TECHNISCHE
UNIVERSITÄT
DARMSTADT

IMPROVING RESEARCH METHODS FOR PROBLEM SOLVING: THE EXAMPLE OF SUDOKU

Dissertation von
Thea Behrens

vom Fachbereich Humanwissenschaften
der Technischen Universität Darmstadt

zur Erlangung des Grades
Doktor rerum naturalium (Dr. rer. nat.)
genehmigte Dissertation

Erstgutachter: Prof. Dr. Frank Jäkel
Zweitgutachter: Prof. Dr. Kai-Uwe Kühnberger

Darmstadt, 2024

Thea Behrens
Improving research methods for problem solving: The example of Sudoku
Darmstadt, Technische Universität Darmstadt
Jahr der Veröffentlichung der Dissertation auf TUPrints: 2024
Tag der mündlichen Prüfung: 18.04.2024

Acknowledgements

First of all, I would like to thank my supervisor, Frank Jäkel, for his continued support and our many fruitful discussions about my research as well as anything related to teaching and cognitive science.

I am very grateful for the support of my colleagues, Max Räuker, Christina Koss, Claire Ott, Susanne Trick, Florian Kadner, Nils Neupärtl, and Philipp Hummel. Together we have attended conferences and exchanged ideas about our research topics as well as about administrative matters and teaching. Special thanks goes to Vildan Salıkutluk and Inga Ibs. We discussed everything related to our research and beyond. They have also read large parts of this thesis and have greatly improved it by giving me helpful feedback.

I would like to thank the students with whom have worked on research projects for courses or their theses. It was fun to work with them and I enjoyed getting their perspectives on our projects: Adrian Kühn, Anna Neubert, Annabell Schmidt, Daksha Katahra, Dennis Duganhodzic, Elifnur Şükran Doğan, Fabiola Schelmbauer, Hanna Behrenbruch, Ingolf Tegtmeier, Isabelle Clev, Janik Schöpper, Karim Abdel Hady, Katrin Scheuermann, Leonie Schüssler, Lilia Stegemann, Marc Saghir, Michelle Kalbfleisch, Myra Zmarsly, Nadine Brass, Sonja Hanek, Timo Rothkegel, Tobias Kühlwein, and Tom Bohlmann. Adrian and Myra deserve a special mention, as they also worked with me as student assistants and were a great help in coding some of the experiments and models.

Last, but not least, I would like to thank my boyfriend, Daniel Mensch, who read all the chapters at several stages, provided valuable feedback and much appreciated support throughout the entire project.

Abstract

Responding flexibly to new rules or constraints and finding tactics on the fly to achieve arbitrary goals are hallmarks of human intelligence. They allow us to adapt to changes in the environment and to thrive under a wide variety of conditions. Studying the solution strategies in puzzles, i.e., the interaction with novel and arbitrary constraints, is a way to study aspects of these core human abilities. This thesis improves on traditional research methods in the area of problem solving by combining qualitative approaches and quantitative modeling, utilizing a broad range of modeling paradigms: production systems, choice models, and hierarchical Bayesian modeling. The "model organisms" on which we test our methods are digit-placement puzzles, most prominently, Sudoku. Since there are several basic tactics for approaching such puzzles, we can study tactic choice and the factors that might influence it.

We present a series of experiments in which participants fill the entire puzzle freely. Concurrent think-aloud protocols enable us to gain a thorough understanding of the tactics used by participants to fill each digit. The studies demonstrate that various digit-placement puzzles are solved using similar methods, and there are two distinct ways in which participants think about the constraints: cell-based and digit-based. Moreover, participants exhibit clear preferences for specific solution tactics, while also utilizing a diverse range of tactics beyond what is required to solve the puzzles. After analyzing data from more than 200 participants, we discover that preferences for tactics change with experience.

We then conduct experiments in which we limit our participants to filling in only one digit per puzzle. This experimental design allows us to control the applicable tactics for each trial. The response times from two experiments indicate that participants can be biased towards a particular tactic by task instructions and task requirements. Based on our experimental findings, we argue that previous research often used biasing task designs and therefore underestimated participants' flexibility and overestimated the importance of a problem's complexity. Furthermore, our experiments demonstrate that participants are able to switch to other tactics if their first attempt does not lead to a solution. We formalize the tactics in a process model and find that the data can only be adequately fitted by including the possibility of switching.

Following up on these experiments, we present a hierarchical Bayesian

model for fitting the response times. We demonstrate how to use process models to analyze response time data and obtain parameter estimates that have a clear psychological interpretation. To estimate the duration of each processing step, we assume that each step has a random duration, modeled as draws from a gamma distribution. Modern probabilistic programming tools enable the fitting of Bayesian hierarchical models to data, allowing for the estimation of the duration of a step for each individual participant. This procedure can also be applied when the step count for each trial is latent, as in our Sudoku model. Our model allows us to estimate tactic choices in the Sudoku task for each participant individually. This approach can be applied to other response time experiments where a process model exists, bridging the gap between classical cognitive modeling and statistical inference.

We also demonstrate how problem solving traces can be analyzed statistically using classical production systems. While research on problem solving traditionally relies on think-aloud protocols in single participants, other research areas usually focus on statistical analyses of overt responses pooled over many participants. To obtain sufficient data for fitting quantitative models on the individual participant level, we introduce a new experimental interface which provides enough data to disambiguate rule selections without relying on labor-intensive methods such as the analysis of think-aloud protocols. To account for the probabilistic nature of rule selection, we use standard choice models, such as the Bradley-Terry-Luce model or the elimination-by-aspects model. The model fits confirm that, as expected, spatial and temporal factors influence rule selection in Sudoku. Through clustering, we find that our participants can be divided into four groups with similar rule preferences.

In summary, we believe that a broad range of methodological approaches is necessary in order to make progress in problem solving research. Therefore, this thesis introduces and uses several experimental designs as well as analysis tools and modeling approaches to contribute to understanding how humans solve digit-placement tasks. We show that there is great potential in the combination of these methods to further improve our understanding of general problem solving.

Contents

List of Figures

List of Tables

List of Abbreviations

AI artificial intelligence

BIC Bayesian information criterion

BTL Bradley-Terry-Luce

cb cell-based

CSP constraint satisfaction problem

db digit-based

EBA elimination by aspects

EIP elementary information processing

IIA independence from irrelevant alternatives

LST Latin square task

NLL negative log-likelihood

NRU number of required units

RL reinforcement learning

ToH towers of Hanoi

Chapter 1

Introduction

Playing is a behavior which can be observed in humans and animals alike. Generally speaking, it is characterized as an activity which has no real consequences, yet for a while the playing individuals are usually quite absorbed in the activity. The goals of the game can be self-set (Davidson et al., 2022) and it is intrinsic motivation, not external reward, that drives individuals to play (Schmidhuber, 2010). Winning or losing in a game usually has no consequences in other aspects of the life of the player, as otherwise the activity would lose its status as "just a game". On the other hand, it is also assumed that playing helps in cognitive and skill development. Often, activities are first practiced in play which are also relevant to adult behavior necessary for survival. Flight animals, for example, can be seen playing chasing, jumping, jinking, and dodging, whereas animals of prey more likely display behaviors like lurking, stalking, and fighting in their games. Self-handicap in play as well as trying new and erratic behaviors might prepare for unusual and unexpected events (Chu and Schulz, 2020).

Human play does not always have the component of physical activity: It can also be a language game or other mental activity. It typically also involves arbitrary rules that can be set quite spontaneously (for example, "walk the whole way without stepping on any gap between the paving stones") or which have been established for generations (for example, the rules for playing chess). If several persons are playing together, they need to agree on the same set of rules (Gray, 2019). Setting abstract and arbitrary rules as well as sharing them with others and following them is an intrinsic part of human playing, reflecting the importance of these abilities for humans (Daston, 2022). We invent abstract rules and follow them, competing against each other in these arbitrary and artificial systems. Whereas many games are played with two or more players either cooperating or competing, there are also single-player games where the player tries to "beat the game" or solve a puzzle.

Structured games are also a popular study domain for research on artificial intelligence (AI). A reason for their popularity is the well-defined and encapsulated nature of such games. In order for a program to play a game, it is not necessary to have a broad understanding of the real world. Explicit rules are easily translated into

formats understandable to a computer. Since the early days of AI, games were a popular challenge and testbed for new AI algorithms and first programs for board games such as chess and checkers were already written in the 1950s (Schaeffer et al., 2007). Programs that could beat world champions in a specific game were milestones in the development of AI. Especially the victory of DeepBlue against Garry Kasparov in chess in 1997 and the victory of AlphaGo against Lee Sedol in Go in 2016 astonished both experts and the wider public. Next milestones have been online vido games such as Dota 2 and StarCraft II, which AI systems can now play as good or even better than humans (Vinyals et al., 2019).

While AI approaches are very good at following explicit rules and optimizing given constraints, humans still outshine them in their ability to flexibly adapt to new objectives and quickly learn from few examples (Johnson et al., 2021; Lake et al., 2019, 2017). Flexibly reacting to new rules or constraints and spontaneously finding tactics to achieve arbitrary objectives are hallmarks of human intelligence. They allow us to adapt to changes in the environment and to thrive under a wide array of conditions. Studying the solution strategies in puzzles, i.e., the interaction with new and arbitrary constraints, is therefore promising to teach us something more fundamental than just the idiosyncrasies of the specific puzzle. Instead, we hope to gain knowledge about core human abilities.

1.1 *The information processing approach*

Newell and Simon (1972) had a lasting effect on the field of problem solving research. They introduced the information processing perspective to the study of human problem solving. Specifically, they described humans as information processing systems, consisting of receptors and effectors to interact with the environment, a central processor, as well as an internal memory storage. The central processor connects the other elements: it receives input from the receptor, can interface with the memory and sends signals to the effectors to act in the world. Such an abstract system can be modeled in programs that can be run on computers. This, in turn, provides the possibility to also translate models of human behavior on a specific task into executable computer programs. "All information processing theories of cognition have this property: they actually perform the tasks whose performance they explain [...] they provide a rigorous test of the sufficiency of the hypothesized processes to perform the tasks of interest." (Simon, 1992, p. 153)

In order to solve a problem, the current state of the world needs to be perceived and a difference to a desired goal state needs to be noted. The problem solving activity can then be described as a successive transformation of a problem state into other states until the desired solution state has been reached. In each state, certain actions can be taken (or operators applied) which transform the current state into a new state. These actions are computed and selected in the cen-

tral processor and then carried out via the effectors.

Solving the problem is equivalent to finding a sequence of actions that transforms the initial state into the goal state. Problem solving can be described as the search for a path in a problem space. If one finds a path from state to state that ends in the goal state, one has solved the problem. Searching for a path is a task that is also very common in the classic literature on AI (Russell and Norvig, 2020, chapter 3). There are numerous strategies of how to proceed with such a search. Random trial and error is of course an option: execute some possible action and observe the outcome. It might just be that one stumbles on the solution. However, many problems require a sequence of actions and often have a relatively large branching factor, therefore the number of paths to explore can grow quickly and it becomes increasingly unlikely to find a solution with a random sequence of actions. First explorative moves are sometimes of this type of strategy when humans approach a new problem domain. However, they usually abandon such a trial-and-error approach quickly.

Two general-purpose heuristics that were found to be frequently used by human participants in problem solving experiments are hill-climbing and means-ends (Simon, 1996; Simon and Reed, 1976). With hill-climbing one tries to get incrementally closer to the solution with each step. It is a very local approach which is prone to get stuck in local optima, instead of finding a path to the desired solution. The means-ends heuristic, on the other hand, starts by evaluating the desired solution state and tries to find important preconditions or subgoals for reaching it. The subgoals can be recursively split into smaller subgoals, until one subgoal can easily be reached from the current state.

Such heuristics have been found by observing participants in the lab solving problems, such as cryptarithmetic puzzles or the towers of Hanoi (ToH) puzzle. This research paradigm started by recording detailed, concurrent think-aloud protocols from the participants during the problem solving episodes. Subsequently, these protocols were transcribed and carefully analyzed. The aim was to infer the mental representation of the problem of a participant (the *problem space*) as well as tracing the traversal of the problem states visited by the participant. The researchers constructed so called problem-behavior graphs where they added each visited state as a node. These graphs thus track the progression of the participant through the problem state in minute detail, also showing returns to the initial state, false starts and dead ends. The next step in the analysis would then be to construct a program, in form of a production system for example, which closely matches the solution trace of the participants.

From such analyses Newell, Simon and colleagues derived general problem solving principles such as the heuristics described above. Even though it was a very successful and influential research program, it was not as successfully continued as other parts of cognitive science (Ohlsson, 2012). It was difficult to aggregate data from several participants and write models that were more general than

capturing the idiosyncrasies of particular participants. One way to arrive at more general models were cognitive architectures, as advocated by Newell (1990). Specifying the architecture on which all of cognition, including problem solving, works, seemed promising. This route, however, quickly abandoned research on problem solving and instead focused on tasks with shorter and more consistent response times such as memory retrieval and reaction to simple stimuli. General questions such as how a problem space is constructed or selected by a problem solver, how learning of new production rules works, and how the selection among productions takes place are still not entirely resolved today.

This work returns to some of the methods employed by Newell and Simon, especially the detailed study of individual problem solving traces. We will expand on it and integrate it with the paradigms of modern cognitive science. With new methods and more powerful computational models, we see great potential for new insights to be generated by studying problem solving (again). Therefore, this thesis presents a collection of studies on problem solving. The overarching theme is the development of methods that can be used for studying problem solving, including modeling approaches as well as behavioral data collection paradigms.

1.2 *Model problems*

There are various paradigms that have been used to study human problem solving and planning. Games and puzzles are probably the two most popular ones. Many games are played by two opponent players, involving elements of theory of mind as well as reasoning and planning with the uncertainty of what the opponent will do. Here, however, we will focus on puzzles which can be solved by a single person. While they are comparably easier to study experimentally, they nonetheless offer a rich testbed for many aspects of human problem solving.

Insight puzzles, like the two-string problem (Maier, 1931), the nine-dot problem (Batchelder and Alexander, 2012) or the mutilated checkerboard (Kaplan and Simon, 1990) often involve a re-conception of some object as another tool or a re-representation of the target domain. In other words, after being stuck on the problem for a prolonged time, a small hint often very suddenly leads people to the solution. Insight problems are hard to study systematically in the lab as they are each unique. While Danek et al. (2014) were able to get some insight on such problems with several trials using magic tricks, it it usually difficult to create multiple experimental trials to see repeated solution attempts of the same participant.

Other puzzles, such as constraint satisfaction problems (CSPs) (e.g., Sudoku, eight queens puzzle, map coloring, crosswords, cryptarithmetic) or move problems (e.g., ToH, Chinese ring puzzle, river crossing, water jug problems), can be solved incrementally. In incremental solution processes, the initial problem state is transformed

into new states by application of operators. Usually, the problem solver has some choice in how to transform the state and different choices lead to different paths through the problem space. The visited states are usually observable and allow the researcher to learn more about how the problem is tackled by the respective player. When attempting to solve such puzzles, it can happen that a legal move or operation leads to a partial solution which is, however, incompatible with the desired end state. In such a case the move needs to be undone in order to be able to reach the full solution. In move problems it is usually possible to reverse the previous move to go back to the previous state. It is also possible to reverse to a previous state in a larger loop, such that the problem solver might not even notice they have already been in the state before and a specific move they did was not leading towards the goal state. In CSPs, such as map coloring or cryptarithmetic, a wrong assignment needs to be canceled with an explicit acknowledgment that it was indeed wrong and cannot be part of the path to the solution going forward.

1.2.1 *Towers of Hanoi*

The ToH is a classical puzzle and became a major research paradigm in the 70s and 80s. Many studies were conducted with this puzzle and many insights on human problem solving were generated from it. In its most classical form, the ToH consists of three pegs and a number of differently-sized discs that can be stacked on the pegs as shown in Figure 1.1. Every disk size is unique. The disks are initially stacked in a pyramid (i.e., sorted from biggest disk at the bottom to smallest at the top) on peg A. The task is to move the entire stack of disks to another peg, say C. There are two rules one needs to obey while doing so: First, only one disk can be moved at a time and, second, no larger disk may be placed on top of a smaller disk. When excluding the move that would directly undo the previous move, there are at most two possible moves at each state of the game. Still, the problem is relatively difficult for people to solve in the first attempt. The minimum number of moves required to solve a ToH puzzle is $2^n - 1$, where n is the number of disks.

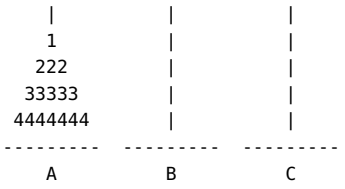


Figure 1.1: Illustration of the initial setup of the ToH puzzle with four disks.

At least four different strategies have been described to solve the ToH puzzle (Simon, 1975). An in-depth analysis of a single participant's think-aloud protocol while solving the same puzzle several times showed the progressive change of planning strategy and problem representation (Anzai and Simon, 1979). A second detailed analysis of the same protocol focused on the rule acquisition events in the protocol and showed that they were taking place in situations where the participant deliberately ignored strategies already known to discover new regularities and rules (VanLehn, 1991). Karat (1982) conducted a study with 192 naive participants and different training regimens, analyzing the data with the help of a production system model. He showed that many participants solve the problem without a full understanding of the underlying recursive structure by

applying local rules to select the next move. The same productions can explain the solution traces of participants of different training groups, and only the duration of the different productions needs to be adapted to fit the response times as well.

Kotovsky et al. (1985) studied several isomorphs of the ToH puzzle, meaning the structure of the state space and the number and restrictiveness of the rules were the same. Nevertheless, different problem statements made a big difference to the time participants needed in order to solve the puzzle. They showed that learning and remembering the rules are major sources of difficulty in puzzles without physical representation and especially when the rules relate less to real-world knowledge. The rule *no bigger disk is allowed to be placed on top of a smaller one* is both easy to verify and to remember, as it has the external representation in the disks and can be directly perceived in the pyramidal shape. Compare this to *If two globes have the same size, only the globe held by the larger monster may be changed in size*. Not only does it require size comparisons over a larger distance, it also involves the sizes of four items in total: the two monsters as well as the two globes. Furthermore, a legal pattern does not correspond to such an easily summarized concept as *increasing in size from top to bottom*. An external representation of the problem which can be manipulated is a big aid, reducing the memory load compared to conditions that have to be solved with a mental representation only.

The ToH puzzle has also been used to study the effects of meta-cognition and self-explanation on strategy learning. In a series of experiments, Berardi-Coletta et al. (1995) showed that answering questions about the solution process (*how are you deciding where to move the next disk?*) led to much better transfer performance in larger versions of the ToH puzzle compared to groups with unconstrained think-aloud instructions or silent controls. Participants initially trained with two- to five-disk versions of the problem and finally had to solve a six-disk problem. Those participants who were prompted by the questions to actively think about how they approached the problem were much better at the final transfer task than the others. Berardi-Coletta et al. (1995) showed that the effect is not due to the verbalization but rather to the meta-cognitive processes: Instructions to "think about the question" (without speaking the thoughts out loud) led to similar performance advantages than instructions where answers had to be given to the experimenter.

1.2.2 *Sudoku*

The ToH puzzle is an impressive example of how much insight can be generated in a relatively simple model domain. There is, however, just one clearly defined optimal path to the solution in this puzzle and once a person has fully understood the recursive structure of the problem it becomes trivial, if tedious, to solve. A slightly more complex model domain is therefore needed to study further aspects of problem solving, such as selection between various competing and

equally valid solution tactics, learning of various and increasingly complex solution tactics, and puzzles that pose difficulty even for experts. One such a "model problem" is the digit-placement puzzle Sudoku and some of its close relatives. These problems are CSPs and have the advantage that many problems of the same type can be generated easily. There are virtually infinitely many cryptarithmetic puzzles or Sudokus that share the same task structure. Transfer of knowledge from one instance to the other is limited to general tactics, no specific value assignment can be copied. It is thus possible to study the problem solving behavior of the same individual for extended periods of time, allowing for data collection in larger magnitudes than with move problems like the ToH puzzle.

The constraints or rules of Sudoku are easily stated, understood and remembered, which is an important prerequisite to successful solution processes (Kotovsky et al., 1985). The fact that Sudoku is a popular puzzle has the further advantage that it is easy to find study participants on different levels of experience with the puzzle. One can thus study the problem solving behavior of experts and beginners alike. Even for easy puzzles there are at least two applicable solution tactics, making it possible to study tactic selection.

There is also a large space of possible alterations of the simple set of rules. One can leave out constraints (e.g., Latin square puzzles omit the box-constraint and only use rows and columns in their specification) or add arbitrary new constraints. One can for example introduce additional connected regions that have to add up to a specific sum, or add other regions in which digits must not repeat, such as in diagonally adjacent cells.

While look-ahead is crucial for games with an opponent (*what opportunities will they have, if I execute this move now?*) it can also be observed in puzzle solutions. The more familiar a person is with a puzzle, the better they usually understand that local consistency now is not enough and may be a bad proxy for global distance to the desired goal state (Simon and Reed, 1976). In digit-placement puzzles look-ahead can sometimes resolve ambiguities and allow the player to find out which digit is the correct choice for a given cell (or which cell is the correct choice for a given digit).

All in all, we argue that Sudoku and related digit-placement puzzles provide a promising testbed for research on problem solving. This thesis builds the foundation of a research paradigm to study such CSPs, using both qualitative and quantitative research methods.

1.3 *Focus of study*

One aim of this thesis is the development and expansion of general methods to study human problem solving, using Sudoku and similar puzzles as a "model problem". We believe that a mix of various methods is required to make progress in problem solving research. Exploration and basic understanding of possible approaches

to a given problem are greatly aided by recording concurrent think-aloud protocols. They are a rich source of data with good temporal resolution and high information content. As participants can freely decide what terms to use to describe their actions, these protocols also provide information about the representations they form of the problem and how these might change over time. A drawback of think-aloud protocols is that they are labor-intensive to analyze and difficult to summarize and aggregate. More restricted data such as response times and answers chosen from a restricted set are easier in this respect and, therefore, have an important part to play. One contribution of this thesis is the development of a new experimental interface, in which participants indicate which information is relevant for deducing a digit for a cell by clicking on it. Such clicking data is much easier to analyze with computer scripts and can, therefore, be collected in larger quantities than think-aloud protocols.

We also employ various modeling approaches in this work. We develop progressively more detailed and fuller models for the task. We start with relatively abstract production rules which describe solution tactics for filling single cells. From there, we move on to more fine-grained process models which make response time predictions for single participants and trials. Finally, we implement a full production system with a choice model to select between different productions. We fit the models to the data of individual participants, acknowledging and describing individual differences in the process.

Regarding the understanding of problem solving, this thesis contributes to the aspect of flexibility in choosing appropriate tactics for solving a given problem. In the puzzles we study, there are at least two different solution tactics to complete the task. While participants exhibit relatively stable preferences for different tactics in all our experiments, they also usually use several different tactics, that is, more variation in approaching the specific task than strictly necessary. In different experiments, we look at the effect of prior experience with such puzzles, task requirements, and instructions on the choice of solution tactics. Generally speaking, tactic choices are adaptive and participants chose effective and efficient solution tactics for the task at hand.

1.4 Contributions and overview

The work presented in this thesis was a collaborative effort. Students who contributed by working under my supervision (for a bachelor's or master's thesis, a class project or as student assistant) are: Adrian Kühn, Daksha Katahra, Fabiola Schelmbauer, Isabelle Clev, Janik Schöpper, Janine Ramolla, Katrin Scheuermann, Lilia Stegemann, Michelle Kalbfleisch, Myra Zmarsly, Nadine Brass, Sonja Hanek, Timo Rothkegel, and Tobias Kühlwein. Some chapters contain previously published text and figures as described below.

Chapter 2 We present a collection of free-filling experiments. The work is mostly exploratory and we employ methods such as think-aloud protocols to get a sense of how people approach various digit-placement puzzles. Overall, we observe many commonalities between the solution tactics applied by the participants in different puzzles. All participants are able to come up with effective and efficient tactics for solving the puzzles, even though we never teach them any procedure explicitly but only state the constraints that must be satisfied in the solution. We observe that there are several tactics to solve such puzzles, and even though there is a big overlap and some tactics are almost universally used by all participants, some are used only by a small set of participants. Four experiments are presented, none of which were previously published.

Experiment 1, a free-filling Sudoku study with 4-by-4 Sudokus, was conducted partly by me and partly by Tobias Kühlwein as part of a practical course in the bachelor of psychology program. Half of the data was transcribed and labeled by him, the other half by me. The analyses are my own work.

Experiment 2, a free-filling Latin square task, was conducted in the context of an experimental practice course for the cognitive science bachelor program under my supervision. Students who collected that data, annotated the think-aloud protocols and implemented the model to categorize the entries are Isabelle Clev, Fabiola Schelmbauer, Katrin Scheuermann, Janik Schöpper and Lilia Stegemann. The analyses presented here are my own work.

Experiment 3, a free-filling Straights study, was conducted as a bachelor's thesis by Myra Zmarsly under my supervision (Zmarsly, 2020). She collected the data and implemented the eye tracking analysis tools. As a research assistant she implemented the model of reasoning tactics. Tobias Kühlwein labeled the protocols of three participants and analyzed them in his bachelor's thesis under my supervision (Kühlwein, 2020). The analyses presented here are partly from their work and partly my own.

Experiment 4 contains the first trial of several online studies. The studies were conducted as part of bachelor's or master's theses, respectively, by Nadine Bras (Bras, 2021), Sonja Hanek (Hanek, 2022), Michelle Kalbfleisch (Kalbfleisch, 2020), Daksha Katahra (Katahra, 2022), and Timo Rothkegel (Rothkegel, 2023). In all these studies we used the same 4-by-4 Sudoku as a warm up task to familiarize participants with the interface and to make sure they understood the rules of Sudokus. The analysis of this data was done by me.

Chapter 3 The third chapter builds upon the insights gained in the free-filling experiments. By design of the puzzle stimuli, we force our participants to switch tactics, as only one leads to a conclusive solution on each trial. A major result is that most participants are able to flexibly switch between the two required tactics and to find the correct solution in all conditions. We observe the influence task instructions can have on the difficulty of the two tactics. We introduce a process model that explains qualitative difference between different

experimental conditions. The chapter is based on the following paper: *Behrens, T., Räuker, M., Kalbfleisch, M., and Jäkel, F. (2023). Flexible use of tactics in Sudoku. Thinking & Reasoning, 29(4):488–530.*

The think-aloud study was conducted as a bachelor's thesis project by Janine Ramolla under the supervision of Max Räuker (Ramolla, 2020). The analyses presented in this chapter were carried out by me. The main experiment was conducted as a bachelor's thesis project by Michelle Kalbfleisch under my supervision (Kalbfleisch, 2020). The process model was developed and fitted by me.

Chapter 4 Here, we expand on the quantitative fitting of the discrete process model predictions to continuous response time data. The statistical framework for doing so is using a hierarchical Bayesian model. This way we can fit parameter values for each participant as well as take all information into account in an optimal way. With two process models which predict different patterns of response times, we can estimate for each participant to what degree they used each of these tactics. This work is currently in preparation for submission.

We use two data sets to illustrate the ideas of the statistical model: The data on children's addition was kindly provided by Sarah Hopkins (Hopkins and Bayliss, 2017). The Sudoku data is the one discussed in chapter 3 already, collected in the bachelor's thesis project of Michelle Kalbfleisch. The model was developed in parts as the bachelor's thesis of Adrian Kühn under my supervision (Kühn, 2021) and in parts by me.

Chapter 5 This chapter closes the loop and returns to an experiment in which 4-by-4 puzzles are filled by participants. With an improved experimental interface we are able to collect data in an online experiment which is rich enough to label the filling events for which tactic was used, without the need for think-aloud data. Participants provide disambiguating information by clicking on all the cells that are relevant for the deduction of the current entry. We are able to label the entries based on this data in an automated fashion, allowing us to analyze significantly more data than in the hand-coded think-aloud experiments. For the experiment in chapter 2 we already implemented the different rules that participants used to make new entries. Here, we additionally model the decision for where and how to continue filling the puzzle between each entry. Using choice models, we can fit preferences for different rules of each participant. Together, the rules and the choice model form a full production system which can fill entire puzzles in similar manners as different participants. This work is currently in preparation for submission.

Chapter 2

Free-filling experiments

2.1 Background and previous work

In this chapter we will use free-filling paradigms, meaning participants have to search and decide where to continue filling the puzzle on each move. Although there is some interest in psychology in digit-placement puzzles, there is very little prior research in free-filling studies. In most studies, the puzzles are used in a very restricted way and participants have to fill in only a single cell per puzzle.

The digit-placement puzzle Latin square has received some attention for its potential as a testing paradigm for fluid intelligence (Birney et al., 2012, 2006; Hartung et al., 2022; Hearne et al., 2020; Perret et al., 2011; Zhang et al., 2009), whereas Sudoku has been used as cognitive training in some cases (Papagno et al., 2013; Nombela et al., 2011). However, most of these studies have little relevance for the present work, as they do not make any detailed analyses of the way people solve these problems. More relevant are the following two papers, as they describe some specific tactics people use in certain puzzle situations.

The study by Qin et al. (2012) provides a lower bound on the realistic speed of carrying out a specific reasoning pattern on 4-by-4 Sudokus. The cell to fill was highlighted, this way participants did not have to search the board for where the rule was applicable. The participants were university students and they had been taught the rule and had trained the task on the day prior to the experiment. The stimuli consisted of puzzles in which the marked cell could be filled by using cell-based tactics. The authors differentiated between puzzles in which a single unit (i.e., row, column, or box) was sufficient to find the correct digit for the cell (*last-in-unit* in our terminology) and those where two or three units together provided the necessary constraints. They also included 2-step conditions, where a second cell was marked which had to be mentally filled first, before finding the value to the cell that was supposed to actually be filled. On the day before the experiment, they taught their participants the seven possible variants of the cell-based tactic (involving either single units, or combinations of units e.g., row and column, row and box, column and box…). During the experiment, participants were expected to select the appropriate tactic and carry it out as quickly and accu-

rately as possible. Elements of search or tactic-discovery were thus explicitly eliminated from the task performance of the participants in this study. All conditions of the experiment (single unit, multiple units and two-step) were solved with very few mistakes (93% to 98% accuracy) and there were no significant differences in accuracy. Response times, however, did show differences between the respective conditions. Whereas the average response time of the one-step one-unit condition was 1.5 seconds, the one-step several-unit condition took 2.9 seconds to answer on average, which is almost twice as long. The two two-step problems took 4 and 6 seconds on average respectively. This experiment by Qin et al. (2012) provides us with a lower bound on expected response times in our own experiments (we do not teach the tactics explicitly, provide no highlights to indicate which cell can be filled and do not administer a training session prior to the experiment). The high accuracy shows that the various cell-based tactics were easily applicable for all participants.

Something like an upper bound for filling duration is provided by a free-filling study by Lee et al. (2008). Their stimuli were 9-by-9 Sudokus with relatively few given digits (28 to 30 of 81), thus only a small proportion of cells could be filled by simple tactics. The participants of this study had never played Sudoku before and they did not receive explicit information about filling tactics in advance. They thus had to discover filling tactics during the experiment and also had to search for cells with enough restrictions to apply these tactics. For 15 minutes, they tried to solve a standard 9-by-9 Sudoku without artificial restrictions. Whenever they filled in a digit, they had to write down a justification, explaining how they knew the chosen digit was the correct value to fill in that specific cell. On average participants managed to fill in 2-3 digits into a puzzle within the allotted 15 minutes, i.e., they needed more than 5 minutes per entry on average. The written justifications provide insights into the kinds of tactics participants used. First of all, participants mostly used logically sound and valid justifications, meaning they understood the rules of the puzzle and correctly reasoned with the given information (i.e., the rules and the given digits). They mostly described one of the basic tactics (as described in subsection 2.1.2) in the justification. Interestingly, they used significantly more *cell-based* tactics than *digit-based* ones.

In this chapter, we will also use free-filling studies, asking participants to fill in entire puzzles without restrictions on the order of the filling. However, our puzzles are smaller which simplifies the search for where to continue filling the puzzle. With concurrent think-aloud protocols we will be able to collect even more detailed information about the solution process than Lee et al. (2008) did with their approach to collect written justifications after each move.

Lee et al. (2008) also describe *advanced tactics*, which are based not only on digits already filled into cells, but also on *possible digits* in cells. When filling a puzzle on paper, one could imagine making little notes in the cells, writing down the digits that are still allowed in a

cell. Usually not all nine digits are options for a cell, as some can be excluded based on the digits already present in the intersecting units. Sometimes two cells within a unit have the same two possible digits as only options. If this is the case, these two digits can be eliminated from all other cells in the same unit. Even though it is not determined yet in what order these two digits will be assigned to the two cells, each of the respective cells will take one of these two digits and no other cell in the same unit can have them. These kinds of advanced tactics are harder to carry out when the interface does not allow for tentative notes as memory aids, but even without notes, beginners are able to discover them (Lee et al., 2008). In an experiment with a specified cell to fill and where advanced tactics were required to find the correct answer, about 40% of the puzzles were correctly solved by a group which was not allowed to take notes. In a second group, which had the possible digits already filled in small font in each relevant cell, the success rate rose to 70%.

In the experiments that will be presented in this chapter, our participants generally preferred to reason based on the basic tactics, as described in subsection 2.1.2. However, in the so called Straights puzzles of section 2.4 the given digits were not sufficient to always infer other definite digits with these basic tactics. In this case the most successful participants spontaneously resorted to using advanced tactics and reasoned with possibilities, even though the experimental interface did not allow to write down tentative notes. Other participants used guessing instead, filling in digits they could not be sure to be correct.

When solving an entire puzzle, the player not only needs to figure out what kind of reasoning rule they can apply, but also *where* on the board they can apply it. Finding intersecting constraints that are strong enough to deduce a new digit that can be filled in is not an easy task. This part of the solution process is especially poorly studied in prior literature. Furthermore, in solving an entire puzzle, people do not carry out each move in isolation. Instead, they often think about several adjacent empty cells at once, knowing a small set of digits that needs to be distributed across them. Filling one of the cells will have immediate consequences for the options in the other cells. Even when not initially planned out in sequence, they often follow up on additional constraints generated by filling in a digit. Are the neighboring cells now determined, too? Alternatively, the placement of the other tokens of the same digit might now be sufficiently constrained.

Recording concurrent think-aloud protocols gave us relatively detailed information on the way participants tackled the task. The protocols do not only cover the successful moves, but provide even more information: They document how the participants searched for how to continue and what features they looked at when trying to find a starting point. Additionally, the protocols contain reasoning chains that did not lead to the filling in of a digit in some empty cell, i.e., also unsuccessful solution attempts.

2.1.1 *Think-aloud protocols as data*

Think-aloud protocols are a great tool in order to understand what people are doing when solving problems. They are a very rich source of data, providing detailed insights into some of the approaches participants use to tackle a problem. Newell and Simon (1972) pioneered research on human problem solving with systematic and very in-depth protocol analysis. They showed that it is possible to construct detailed problem-behavior graphs from said protocols. These graphs reconstruct the knowledge states the participant visited as well as reproduce the operators that were used by the participant to move from state to state.

Although there have been debates about the validity of verbal reports (Nisbett and Wilson, 1977), as of today there is convincing evidence that under the right conditions, think-aloud protocols can give veridical information about the processing tactics of participants (Fox et al., 2011). An important precondition for a reliable think-aloud protocol is to use a task that has a correct answer, instead of asking questions on opinions or preferences for example. It is important to use instructions that encourage participants to verbalize all thoughts that they have during the solution process. When they are speaking the thoughts that they already had (but would have normally thought only silently) this is considered level 1 verbalization (Ericsson and Simon, 1993). Level 2 verbalization consist of thoughts that were not in verbal form, but might have been mental images for example. They therefore need some translation to be put into words but were consciously thought already. In general, level 1 and level 2 verbalization do not alter the "normal" solution process, except for slowing it down as talking takes time (Berardi-Coletta et al., 1995; Fox et al., 2011). When asked to *justify* their actions verbally (instead of just voicing their thoughts), participants are prompted to reflect on the specific procedure and need to verbalize content which would normally not have occurred to them when solving the task. This level 3 verbalization usually changes performance in problem solving tasks, often improving it (Berardi-Coletta et al., 1995). Since we are interested in the unbiased solution process, the studies reported below are aiming at eliciting level 1 and level 2 verbalization, and are not asking for explicit reasons or justifications.

Processes that fall in an intermediate range of automation and novelty are best captured in think-aloud protocols. When participants are discovering new rules or tentatively explore new representations of the problem they often have trouble verbalizing this process and tend to fall silent. On the other hand, when a process is too low-level or automatic, it might not reach consciousness explicitly and thus often fails to show up in the verbal protocol. Duncker (1945) already pointed out that a "protocol is relatively reliable only for what it positively contains, but not for that which it omits. For even the best-intentioned protocol is only a very scanty record of what actually happens." (p. 11). Some thoughts might be too fleeting, others

deemed uninteresting or irrelevant by the participant and, as a consequence, are not being uttered. Nonetheless, concurrent think-aloud protocols are a valuable tool for studying problem solving, providing very detailed information about the involved processes and their temporal order.

2.1.2 *Basic definitions*

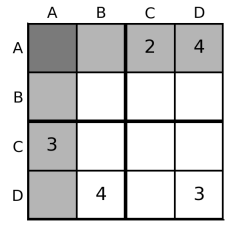
All digit-placement puzzles in this section consist of a square grid of partly filled cells. The grids can be divided in *rows* and *columns*, Sudokus additionally have *boxes*, i.e., 2-by-2 connected regions in the case of 4-by-4 Sudokus. In Sudokus and Latin squares, each digit has to appear exactly once per *unit* (row, column, or box). In Straights, each digit has to appear at most once per unit (the specific rules of Straights will be explained in section 2.4). The *peers* of a cell denotes the set of other cells which share a unit with it: they are either in the same row, column, or box. The digit in one cell cannot appear in any of its peers.

There are three basic tactics which can be applied to all the digit-placement puzzles of this chapter.

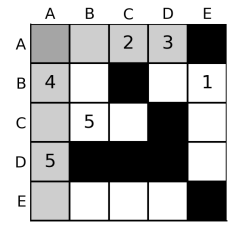
The cell-based tactic You select a cell and look at its *peers*. The digits which are in any of the shared units of this cell have to be excluded as candidates for it. Lee et al. (2008) therefore refer to this tactic as *exclusion tactic*. If only one digit remains as a possible candidate, you can enter it in the cell. Examples for applicable cell-based tactics can be found in Figure 2.1. All digits except for the 1 can be excluded for cell AA in the upper left corner of our example, because they appear already in the peers.

The digit-based tactic Each digit has to appear once in each unit of the Sudoku or Latin square. You can therefore start the search for the next entry with a specific digit in mind and test whether its placement in a specific unit is restricted: For example "Where in this box can I place the 1?". If it is not yet in a given unit, you can try to place it there. Cells which are already filled are, of course, not a possible candidate location. For open cells, you need to check in their peers whether the specific digit is in them already. Each cell that has this digit in its peers can be excluded as candidate location. If only one cell in a unit remains as a candidate location, you can fill it in. Examples for digit-based tactics can be found in Figure 2.2. In the upper row (or box, in the case of Sudoku) the 1 can be placed in one cell only, namely AA. Lee et al. (2008) call this tactic *inclusion* because one tries to include a digit into one specific unit.

Last-in-unit A special case combining both of the two tactics described above arises when a unit is almost full. When $N-1$ cells of a unit of size N are filled, the value of the last empty cell in the unit can easily be determined. In a cell-based mind set, the peers of the

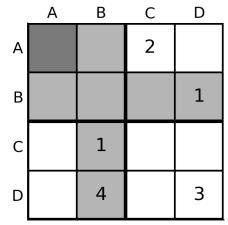


(a) Sudoku

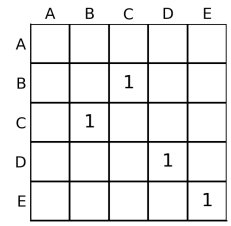


(b) Straights

Figure 2.1: In both puzzles the correct solution for cell AA is 1, which can be found using cell-based tactics. Ask yourself: "Which digit is allowed in cell AA?"



(a) Sudoku



(b) Latin square

Figure 2.2: In both puzzles a 1 can be placed in one cell only, namely cell AA, which can be found using digit-based tactics. Ask yourself: "Where can I put the digit 1?"

empty cell exclude all values but one and only one unit is needed. In a digit-based mindset you can reach the answer easily, too. When looking for where to place the digit in the unit, only one cell can be chosen as all the others are already filled. It is visually quite salient when a unit is almost full and the reasoning becomes rather easy.

2.1.3 *Outlook on chapter*

The first three experiments in this chapter employ think-aloud protocols to get a relatively detailed insight on how people act to solve various related digit-placement puzzles. The experiments are ordered by puzzle complexity. Section 2.2 uses 4-by-4 Sudoku puzzles as stimuli. These are solved comparatively quickly by participants, on average they need less than a minute per puzzle.

More challenging puzzles are used in section 2.3: Latin squares of increasing sizes from 4-by-4 to 7-by-7 are the stimuli. The fact that they have one type of constraint less (no boxes) than Sudokus makes the rules even simpler to state, but it also means there is less leverage to constrain the solution. As this type of puzzle is more difficult to solve the solution times increase and we see more pronounced search phases where participants look for how to continue with the puzzle.

The puzzles in section 2.4 introduce a new kind of constraint which is a bit more difficult to understand than the "each digit exactly once per unit" constraint of Latin squares and Sudokus. Additionally they are the most difficult puzzles, leading participants to reason with *possible digits in cells*, instead of reasoning only based on given or definitely inferred digits. The data of this section not only contains information about tactic application and searching for how to continue, but also gives some insight into learning within the domain of solving these puzzles.

Section 2.5 analyses 253 solutions to one 4-by-4 Sudoku. We do not have think-aloud protocols here, the resolution of the data is thus more coarse, as we only have the filling events to analyze. However, the quantity of the data allows for statistical analyses that were not possible on the other experimental results of this chapter. Especially, the large number of participants allowed us to look at the correlation between prior experience with similar puzzles and tactic choice.

2.2 *Experiment 1: Mini-Sudokus*

In our first experiment we used very simple puzzles, 4-by-4 Sudokus with seven to ten empty cells. The advantage of using simple and therefore rather quickly completed puzzles is that it allowed us to conduct relatively many trials and thus to also see how consistent participants were over time.

2.2.1 *Methods*

There were 20 participants (15 female, 5 male), aged between 18 and 39 (mean: 22.2, SD: 4.74). One participant was excluded from

the analysis due to making too many mistakes and talking too little. Students of psychology and cognitive science participated for partial course credit, other participants received no compensation. Participation was voluntary and participants gave informed consent. The study was approved by the local university's ethics committee. Each participant was recorded individually. They were instructed to think-aloud during the experiment. Instructions were *Please utter all thoughts going through your mind while solving the task without filtering them. Please mention each step you take and also attempts and tryouts that you don't think through till the end.* To get familiar with uttering their thoughts the participants each solved two math problems, one three-digit subtraction and one two-digit multiplication. Subsequently they filled in the Sudoku puzzles on a computer. If they fell silent for longer than 15 seconds, they were reminded to keep talking by the experimenter who was sitting in the same room but out of sight. The think-aloud recordings were transcribed after the experiment. All participants filled the same 20 puzzles but the order was randomized and different for each participant. All puzzles had the same level of difficulty. Of the 16 cells of a 4-by-4 Sudoku, about half of the cells were filled (mean: 8.65), the number of empty cells ranged from seven to ten. In addition to the think-aloud data, we recorded overt responses, mouse movements, and the response time for each entry measured from stimulus onset.

2.2.2 *Response times*

On average, participants needed 48 seconds (SD: 23) per puzzle. Even though most participants were fairly quick in solving the puzzles, the spread was wide. The shortest trial lasted just 16 seconds, the longest 218 seconds. The mean time per move was 5 seconds, with a standard deviation of 5.44 seconds and a median of 3.2 seconds. Overall, participants were quite fast at filling the Sudokus and the puzzles did not pose a big challenge for them.

2.2.3 *Classification of moves*

As search phases were usually quite short, we decided to only label moves, i.e., reasoning patterns leading to an entry of a digit. There were, of course, also other utterances in the protocols, such as comments on the progress of the experiment, the experimental interface, annoyance about typos or a reasoning chain that did not lead to the entry of a digit. These were rare enough, however, to discard them without losing too much information. After taking a look at the utterances and reasoning patterns of our participants, we implemented the most common ones as a Prolog program. The program was very helpful during subsequent labeling of participants' moves. For a given board configuration and move, the program computed all rules that would allow the entry of the digit. To label a move, one could now select from the list of computed options. Alternatively, it was always possible to enter a new label instead. 79% of all moves

Table 2.1: Labels in the 4-by-4 Sudoku task. Frequency of the label, as well as mean and standard deviation for the response time of the moves with this label are given. There is also a short explanation and an example utterance to show where the label would be applied. Exemplary board situations with mouse movements and think-aloud utterances can be found in Figure 2.3 to Figure 2.6.

		time (sec)			
Label	Freq.	Mean	SD	Description	Example
last-in-unit				3 of the 4 cells in a unit are filled, the participant fills the last one.	"And here we need a 2."
row	16.2	4.25	3.42		
column	13.0	4.11	3.48		
box	14.1	3.52	2.53		
digit-unit				Combination of a unit and a digit to be placed into it. All other empty cells in the unit can be excluded as locations for the digit. When there are three instances of the digit on the board, all units could function as basis unit, no clear basis unit recognizable.	"I'm looking for the 4 in the box. It can't be here, it has to be there."
row	0.7	10.4	4.93		
column	0.7	9.88	5.89		
box	16.9	6.88	5.15		
4th	2.6	8.72	5.40		
2-open				A unit has two empty and two filled cells. Participant notes which digits are missing in it. Some digit in an intersecting unit restricts the placement of one of the two digits. The next move is to fill the other empty cell with the remaining digit.	"Then there can be a 1 and a 3 here. The 3 can't go there, so it goes here. And here goes the 1."
row	1.8	11.01	6.64		
column	0.9	12.75	6.58		
box	5.1	10.59	9.43		
cell-complex	0.9	23.68	11.34	Focus on a single cell. The digits in two intersecting units allow only one digit to be placed into the cell.	"In the column is 2 and 3 and in the row the 4 already. Meaning in this cell only a 1 is allowed"
only-digit-missing	2.5	1.67	1.47	When a participant fills in all missing tokens of each digit in blocks, for the last digit they do not have to check any constraints. They know that in all of the empty cells they need to fill in the one digit that is still missing on the board.	"And in the remaining free cells will be only 2s, because all the rest is already there"
other	24.6	4.83	7.68	correcting mistakes, uncommented move, ...	

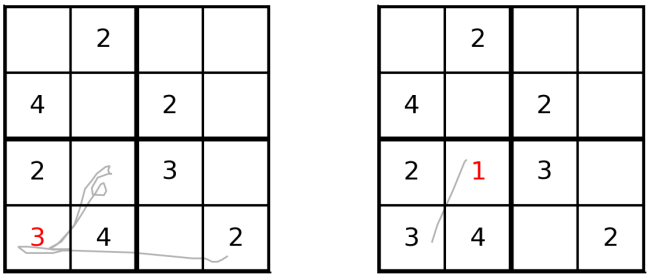
could be assigned one of the proposed labels. Most of the remaining moves had very scarce comments in the think-aloud protocol rendering the disambiguation between two similar labels impossible. Of course, there have also been reasoning errors, typos, corrections and from time to time reasoning patterns that are not adequately captured by the implemented rules. On average, there were about eight rules for each move to choose from in the labeling process (between 0 and 13). The utterances in the think-aloud protocols together with the mouse movements were essential to disambiguate in the case of more than one possible label.

About 20% of each protocol was labeled by two researchers. Inter-rater agreement was 85%. Cohen's kappa is a measure that puts the inter-rater agreement into proportion of "agreement by chance" given the base rates of each rater for each label. Calculated on our data the kappa measure is $\kappa = 0.72$, this value is a sign of "substantial agreement" according to Landis and Koch (1977) (perfect agreement would lead to a value of one, chance-level agreement to a value of zero).

Table 2.1 shows all labels for this data set: how often they occurred, how long a move with this label took on average, a description, and an example utterance. Examples for situations in which the rules were used, together with mouse movements and utterances, can be seen in Figure 2.4 to Figure 2.6. All participants made moves that were labeled *last-in-unit*. These were usually quite quick moves, they took about 4 seconds on average. Most participants were rather indifferent about the unit in which they applied this specific rule. *Digit-unit* is the second most used label. Here, the unit is more important and most participants exclusively used the rule in combination with the basis unit *box*. Only six participants used this move less than 10 times (of about 190 moves in the entire experiment), whereas all others applied it relatively frequently. *Cell-complex* was used regularly by just one participant, three others used it from one to three times. The *only-digit-missing* rule was applied by few participants: One applied it in most of the puzzles, four others used it from one to 15 times.

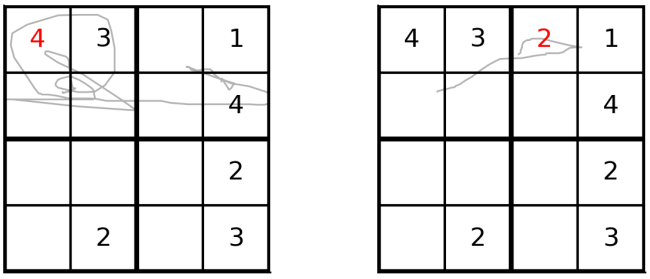
One label is different from the rest, as it does not describe moves in isolation but classifies two moves together: *2-open*. Seven participants applied this rule regularly and reasoned at least once per puzzle about two empty cells and their digits together. Most other participants used this rule at least once, too. Only two participants in our experiment never used the rule at all. This reasoning pattern can be applied when one unit is half filled, i.e., two cells are empty and two are full. The two missing digits have to go into the two empty cells. The participants then looked for an occurrence of one of the missing digits in the peers of the empty cells to resolve the ambiguity of where which digit goes. Figure 2.3 shows an example. This reasoning pattern is characterized by long response times before the first digit is filled in and very short response times for the second digit. All reasoning and explanation as documented in the think-

aloud protocols already happened in the first move, the second digit assignment followed naturally and needed no further reasoning. It should be noted that it would be possible to break such a double move down into two steps and label them independently. The second part would then always be a *last-in-unit* move: putting the last missing digit in the only open cell of the unit. The first step would be classified either as *digit-unit* or *cell-complex*, depending on the exact situation. In Figure 2.3 the reasoning is classified as digit-based because the participant mentioned that the 3 is constrained (by the 3 in the cell in the third row and third column) to go to the lower left cell. If they had started to fill the 1 into the cell where the 3 is not allowed, it would have been classified as *cell-complex*, as the digits of the column and the row together only allow the 1 to be placed there.



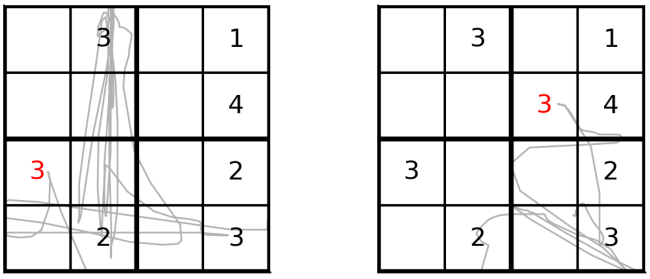
(a) Label: *2-open 1st* (b) Label: *2-open 2nd*

Figure 2.3: Example from participant 14: The digit in red is the new entry, the gray line depicts the trajectory of the mouse. Think-aloud except (a) *Then there can be a 1 and a 3 here. The 3 can't go there, so it goes here.* (b) *And here goes the 1.*



(a) Label: *digit-box* (b) Label: *last-in-row*

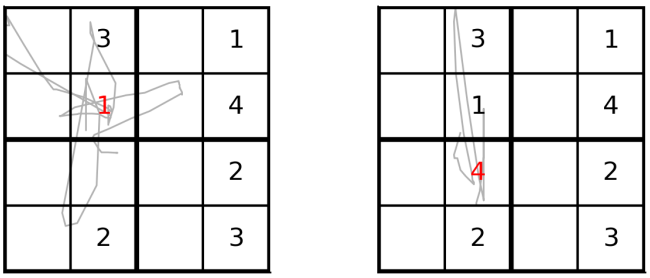
Figure 2.4: Participant 06: (a) *I'm looking for the 4 in the box. It can't be here, it has to be there.* (b) *Now I can fill up the row, a 2 is missing.*



(a) Label: *digit-box* (b) Label: *digit-4th*

Figure 2.5: Participant 03: (a) *There are two instances of 3 and two instances of 2 on the board. There, the, hmm...one 3 blocks this in the lower left and the other that. So a 3 has to be here.* (b) *Now we have three 3s, we know where the last one goes, that is here, because all others are blocked.*

Some specific move examples A noteworthy observation we made on this data set was that participants very often followed up on entries they just made. Even when the reasoning was not explicitly about two cells from the beginning, the next move is often close to the previous one, either by being in the same unit or by using the same



(a) Label: *cell-complex* (b) Label: *last-in-column*

Figure 2.6: Participant 19: (a) *Looking for a cell with just one option... it has to be this cell, because ... in the column is 2 and 3 and in the row the 4 already. Meaning in this cell only a 1 is allowed.* (b) *As a consequence, ... in the second column only the 4 is missing, which we can enter directly.*

digit. Most participants stayed within the same unit and filled the other empty cells in it, which were more constrained after they filled some digits in them already. Others rather followed up on the digit they entered and looked how it constrained the other occurrences of the same digit.

Figure 2.4 to Figure 2.6 show the first two moves of a trial of different participants from the think-aloud experiment. The gray lines show the trace of the mouse. The digit in red font was entered on the move. The accompanying utterances (translated from German) are in the captions of the figures.

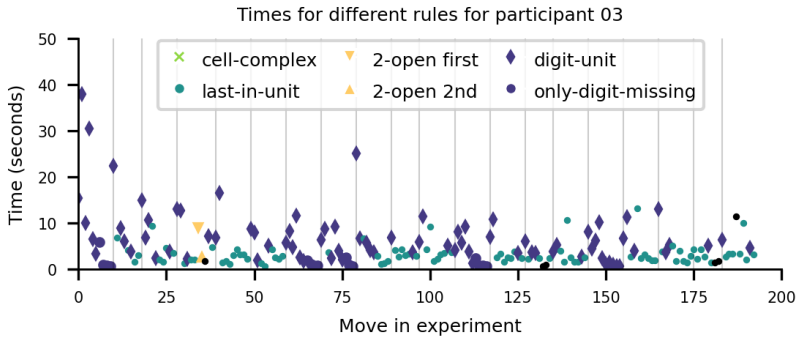


Figure 2.7: Labels for all moves of participant 03: The vertical lines indicate the start of a new puzzle.

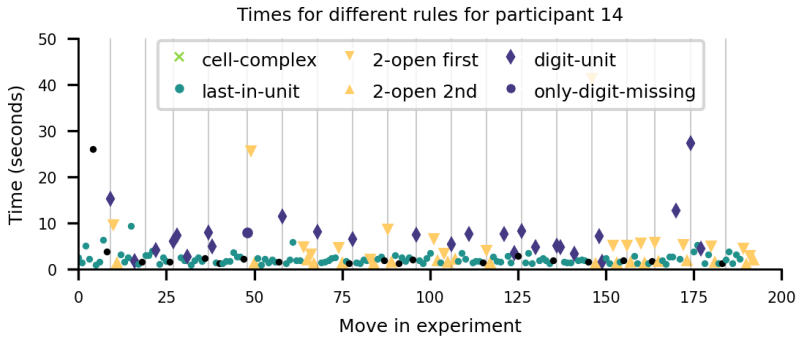


Figure 2.8: Labels for all moves of participant 14: The vertical lines indicate the start of a new puzzle.

Consistency of behavior over time With the relatively many trials per participant it is also possible to look at the consistency of the behavior of each participant. In Figure 2.7 to Figure 2.9, some exemplary participants' labels are plotted over time. The marker's position on

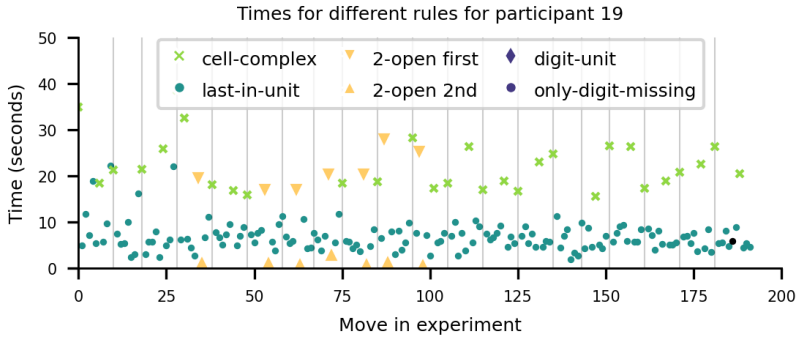


Figure 2.9: Labels for all moves of participant 19: The vertical lines indicate the start of a new puzzle.

the y-axis indicates how long the respective move took, measured in seconds since the previous move. In the first puzzle participants were slightly slower, but otherwise there was not much change over the course of the experiment. Each participant quickly found a set of rules that they then applied throughout the experiment. Most participants seemed not to be aware of the existence of the rules they did not apply. Some explicitly mentioned that the analysis of the data must be boring, because everyone would solve the puzzles in the same way. The figures illustrate how each rule has its own typical duration: *last-in-unit* entries, for example, are typically quite fast and also very consistent within one participant. The first move in a puzzle often required longer time, as participants needed a moment to orient and find out where they can start in the puzzle. Subsequent moves often follow up on the previous move and are thus faster, most obviously so in the case of *2-open*.

2.2.4 *Discussion*

Our main findings are that people know and use a variety of tactics, often only using special cases of more general tactics. This is most strongly seen in the case of the *digit-unit* label, where the majority of participants use the tactic only in combination with the basis unit box. Response times can only be analyzed to a limited degree in this study. When comparing response times between participants, a very influential factor is the amount of talking that they did. A participant who aims at fully explaining why they did a move and how they knew it was legal, takes obviously much longer than a participant who just says "and here a 1", i.e., without any explanation as to why. Such a difference in time does not necessarily reveal anything about the difference in reasoning speed between the two respective participants. What can better be compared, however, are the response times *within* one participant, differentiating between different tactics. Even though some response times can be misleading due to an unsuccessful filling attempt with some other tactic or in some other part of the puzzle, we can see robust effects here. For most participants, the *last-in-unit* tactic was the fastest: it is only topped by *only-digit-missing*, a tactic that is applied by only few participants,

though.

Free-filling experiments entail that the board configurations participants encounter while solving the puzzles are not all identical. Only the initial puzzle situation is seen by all participants. Subsequent configurations of constraints depend on the order in which they fill the puzzle.

These complications in analysis were compensated by the insights we gained in this study. The puzzles could have been solved by using only one of the following tactics in combination with *last-in-unit*: *cell-complex*, *digit-unit*, or *2-open*. It was not necessary to use several of these tactics, but all participants used at least two different reasoning tactics besides *last-in-unit*. The majority of participants had clear preferences for one or two tactics and applied them with about constant frequency throughout the entire experiment. Each tactic had its own relatively consistent execution duration within a participant.

As the puzzles were so small and quickly became very constrained, the majority of all recorded moves were completed with the *last-in-unit* tactic. With this tactic, most participants were very flexible about the unit to which it was applied and used all basis units to a similar degree. When this tactic was not applicable, most participants resorted to using the *digit-based* or *2-open* tactic rather than using the *cell-based* tactic with the union of at least two units.

2.3 Experiment 2: Latin square Task

In the following experiment we used a free-filling, think-aloud paradigm with Latin squares as puzzles. The advantage of Latin squares is that they can be smoothly changed in size. In our experiment we used puzzles with side lengths ranging from 4-by-4 up to 7-by-7, with all intermediate sizes. In Sudoku puzzles only square numbers like 4 and 9 can be used because of the additional box-constraints. The larger puzzle sizes in this study make it possible to pose more challenging problems to the participants and observe more situations in which applying solely the *last-in-unit* tactic is not possible. We can thus learn more about the search process and reasoning in more complex situations than in the small 4-by-4 Sudokus.

2.3.1 Methods

15 participants (8 female, 7 male) took part in the study, and they received no compensation. Their ages ranged from 20 to 58, mean age was 31.6 years (SD: 14.6). Participation was voluntary and participants gave informed consent. The study was approved by the university's ethics committee. The experiment ran for about 30 minutes per person. Depending on their speed, they solved between four and ten Latin square puzzles during this time window. Some motivated participants completed more puzzles, even though the 30 minutes planned for the experiment were exceeded. We increased

the difficulty of the puzzles over the course of the experiment by making them larger and decreasing the number of given digits. The first puzzle was a 4-by-4 Latin square with just six empty cells. The last puzzle was a 7-by-7 Latin square with 30 empty cells. Participants filled the empty cells with a drag and drop interface. Next to the puzzle were stacks of all the digits that could be filled in the puzzle. Participants had to pick up a digit tile with the mouse and drop it into the empty cell they wanted to fill. A trial could only be finished when the puzzle was correctly filled. If, however, there was an error in the completely filled puzzle, the button "delete all wrong digits" could be used. All cells containing a wrong digit would then get cleared and the cells could be filled again by the participant. We ensured the participants did not use the button often during the experiment, but only when they did not see the mistakes at the end of the trial. Participants were instructed to think aloud during the entire experiment and utter their thoughts. We used instructions similar to the ones in section 2.2. Participants practiced thinking aloud with a two-digit multiplication and a three-digit subtraction task. When they fell silent for more than 20 seconds, the experimenter reminded them to keep talking. In addition to the filled in digits, we recorded the mouse movements, the response time for each entry measured from stimulus onset, and the think-aloud protocols.

				Time (min)	
Trial	Puzzle size	Empty cells	Completed by	Mean	SD
1	4	6	15	0.7	0.3
2	4	8	15	1.1	0.9
3	5	14	15	2.6	2.1
4	5	17	15	3.8	2.8
5	5	17	14	7.8	4.0
6	6	22	14	5.3	2.8
7	6	22	12	9.1	2.9
8	6	22	4	5.1	2.0
9	7	30	3	11.4	5.7
10	7	30	1	10.4	—

Table 2.2: Latin square task: List of trials, the size of the puzzle, the number of empty cells in the puzzle, how many participants completed the trial and how long they needed on average for the puzzle.

2.3.2 *Results*

The participants completed between four and ten trials, 14 out of 15 participants completed at least 6 trials, including a puzzle of size 6-by-6. They needed on average between 10 and 35 seconds per empty cell, 12 participants needed less than 20 seconds. The time per empty cell increased over the course of the experiment (from around 6 seconds per empty cell in the first trial to about 20 seconds per empty cell in the last trial), showing that the puzzles were indeed more and more difficult to solve.

2.3.3 *Move classification*

The classification labels for the moves we used in the Latin square puzzles are compatible with the ones used in Experiment 1 for the 4-by-4 Sudoku puzzles, only a bit simplified. With a simple model, we could classify all moves as belonging to one of six categories: filling the last empty cell in a unit (*last-in-unit*), digit-based reasoning (*digit-unit*), cell-based reasoning (*cell*), either digit-based or cell-based reasoning (*both*), no simple rule applicable (*none*, this could either be faulty reasoning, a lucky guess or higher order reasoning based on possibilities in other cells), and wrong inputs (*error*). Wrong inputs might not violate any constraint on the board in that state, i.e., they might be plausible guesses in that state of the board. However, compared to the final solution, they constitute a violation and cannot be logically inferred by any valid rule.

As can be seen in Figure 2.10, the tactic that was used the most is *last-in-unit* (in 47% of all moves). In the early puzzles it was almost sufficient to fill the entire puzzle, but with an increasing number of empty cells in the puzzles it became relatively less applicable. The second most used tactic, with 24%, was to fill in cells based on *digit-based* reasoning. *Cell-based* filling events were recorded rarely (5%), but their fraction is relatively constant across trials. The fraction of *errors* rises sharply on the fifth puzzle (around 20% of all filling events were classified as errors on the fifth trial) and remained on a similar level throughout the rest of the experiment (on average across the entire experiment 17% of the filling events were classified as errors). Moves which could be either cell-based or digit-based (i.e., are labeled as *both*) made up around 5% of the filling events and moves without any label (i.e., labeled *none*, and which most probably were lucky guesses) made up 3%.

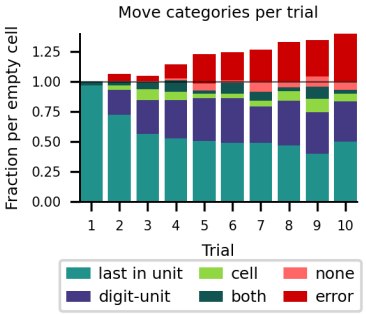


Figure 2.10: Move categories per trial, relative to the number of empty cells in the Latin square task. Without errors or deletions they would sum to one on each trial. The *last-in-unit* was the most easily applicable tactic and was used almost whenever possible. With increasingly complex puzzles the fraction of applicability of this tactic decreases.

2.3.4 *Think-aloud data*

Labels to code the utterances of the protocols were developed based on the data. The think-aloud data match the move classification in most cases, but provide more details about the search process between the filling events. An overview of the labels and what they mean can be found in Table 2.3. The labels for filling events are compatible with the ones from the 4-by-4 Sudoku task (Experiment 1) in section 2.2, only slightly simplified. First of all, we do not distinguish the basis unit because in the Latin square task only rows and columns exist as units, boxes are not present and most participants switch freely between rows and columns. Second, we do not use the 2-open label. With the varying sizes of the Latin squares the situation of having exactly two empty cells in a unit does not occur as often any more. With the additional labels for the search phases we can also label utterances about several missing digits in a unit explicitly. We also do not use the *only-digit-missing* label as we did not observe situations where it would have applied. As the puzzles were more challenging than the 4-by-4 Sudokus of Experiment 1, there were

more and longer phases of searching and reasoning between filling events. We decided to label the search here.

Table 2.3: Latin square task: Labels for the think-aloud utterances. Mean and standard deviation indicate the relative occurrence of labels per participant.

	frequency			
Labels	Mean	SD	Description	Example
Filling				
last-in-unit	0.15	0.07	last cell of a row or column	"And that just leaves the 5."
digit-unit	0.12	0.05	digit-based justification	"There is already a 4 there and there, so the 4 can only go here."
cell	0.04	0.03	cell-based justification	"There is a 4 there, so that means the 3 needs to go here."
try	0.04	0.09	plausible guess, not sure the action is correct	"Hmm, both of those work... Alright, let's just try and see how it goes."
Searching				
focus unit	0.33	0.12	focusing the attention on one row or column	"1, 3, 4, 6, so 2 and 5 are missing. Let's see the 2 can go both here and here."
focus digit	0.13	0.09	focusing the attention on one digit	"Alright, 5, 5, 5, 5...So only one 5 left which has to go here."
search	0.04	0.03	generally looking for where to continue	"Maybe here... no that doesn't work either. Hmm, where should I start?"
hypothesis	0.02	0.02	mentally testing the consequences of filling a certain digit in a certain cell	"If I were to put the 2 here that would mean the 4 can only go there. That would mean..."
Other				
other	0.14	0.06	deletion of wrong inputs, talking to the experimenter, ...	"That was wrong."

As can be seen in Figure 2.11, there are large differences between the participants with respect to their tactic use. Nevertheless, some trends can be reported for the entire group. Overall, it is very clear that when searching for where to continue, our participants liked to focus on one unit at a time. With 33% this is by far the most frequent label in our data. Only a single participant barely used this approach (2%), the participant with the next fewest occurrences of this label is at 20%, for six participants at least 40% of all labels are *focus-unit* (higher than any other label frequency). Focusing on a unit is a classic start for digit-based reasoning. We do not have a label for *focus-cell*, because in the search process this was never mentioned by our participants. Although they do notice cell-based constraints and sometimes fill a cell based on them, often this solution approach stems from starting with a unit-based focus. The other more specified and regularly used search tactic was *focus-digit*, which in case of success would also lead to a digit-based filling event. In cases where the focus on one unit or one digit did not lead to a successful filling event, some participants resorted to guessing, others to playing through the consequences of some guess on the rest of the board.

Testing the consequences of choosing a particular digit only mentally is, of course, more challenging and places a high load on the working memory. This tactic is mainly used by more experienced players, whereas beginners are more likely to fill in a "reasonable guess" without knowing for sure whether this is the correct solution. Both these approaches usually start from a unit with only few empty cells left, so that the potential cell-digit pairings can be reduced to two in the best case (2 empty cells, 2 missing digits in the unit).

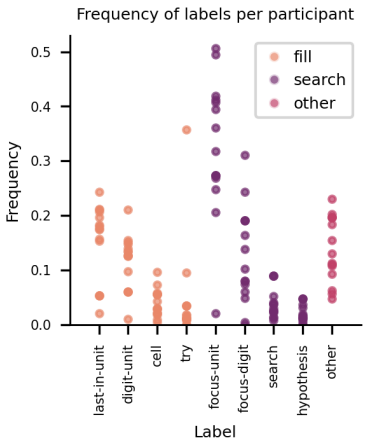


Figure 2.11: Think-aloud labels in the Latin square task. Each dot is the relative frequency of label of one participant.

2.3.5 *Discussion*

In the think-aloud study with 4-by-4 Sudokus, participants needed on average 5 seconds per empty cell, whereas in the Latin square task they needed 16 seconds per empty cell on average, showing that the Latin square puzzles were much more challenging to our participants. The perceived difficulty with the task varied significantly between participants. One participant struggled a lot: they completed four puzzles only and guessed a lot throughout. The other participants had less difficulty with the task and completed at least six puzzles. The puzzles in this experiment were overall more challenging than the small 4-by-4 Sudokus from the previous experiment. One reason for the increased difficulty was the growing size of the puzzles, and there were more empty cells and fewer "easy starting points" for the participants. Whereas in the small Sudokus more than half of the cells can be already filled by some rule at the onset of the puzzle, that fraction is much smaller in the Latin square puzzles we presented in this study. Hence, when checking a unit, cell or digit at random to see whether some conclusions can be drawn there, the chances of success are lower, and more reasoning dead-ends are met.

The resulting longer and more frequent phases of searching for an opportunity to continue filling the puzzle and unsuccessful filling attempts warranted an explicit coding of the corresponding think-aloud passages. We saw two main approaches of searching. One was to focus on specific digits and searching their occurrences on the board, testing whether they constrain the placement of the other instances of the same digit. The other, more frequently used approach was to focus on some row or column, finding out what digits were missing in it and testing whether the placement of the missing digits was constrained by digits in intersecting units. We never saw a participant start their search process by focusing on a single cell and testing which digits were still allowed in it.

2.4 *Experiment 3: Straights*

Another digit-placement puzzle with variable grid size is the game of Straights. This puzzle introduces a new constraint and thus enables and requires additional deduction rules compared to the Latin square task and Sudoku. In the following experiment we use more

difficult puzzles compared to Experiments 1 and 2. Besides think-aloud protocols, here, we also explore eye-tracking as another method to gain insight into the solution process.

2.4.1 *The rules of Straights puzzles*

Straights are puzzles similar to Latin squares: they are quadratic grids of any size (in our experiment we used 4-by-4 and 6-by-6), where some cells are already filled with digits and the task is to fill the empty cells. For an example puzzle see Figure 2.12. As in a Latin square, no digit can exist more than once in a row or column. A new element in Straights is that some of the cells are black instead of empty. These black cells can have a digit in them from the beginning, but can never be filled by the player. An empty black cell in a unit (i.e., a row or a column) also means that not *all* digits will be present in the unit. The black cells can divide a unit into separate parts, like separate words in a cross-word puzzle. Each of these parts has to be filled with digits that can be sorted into an unbroken sequence, a so called *straight*. For example, if there is a 2 and a 3 in a section with three empty cells, the last cell can be filled with either a 1 or a 4, but nothing else. The straights constraint is very powerful, but compared to Sudokus or Latin square puzzles a difficulty in Straights is that one does not know from the beginning which digits will be in a row or column if an empty black cell is present in it. In other words, the constraint is that a digit occurs in a unit *at most once*, whereas in Sudoku and Latin squares it is *exactly once*. Note: not all Straights can be completed to be valid Latin squares: sometimes the black cells would require one digit to complete the row and yet another digit to complete the column.

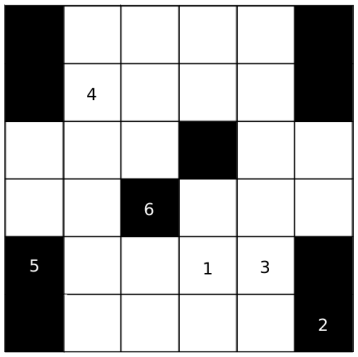


Figure 2.12: Example for a Straights puzzle: All the white empty cells have to be filled with digits between 1 and 6. Each digit can appear *at most once* per row and column. Consecutive white cells within a row or column have to be filled with digits that can form an unbroken sequence. Hence, in the row second to the bottom, only a 2 and a 4 can be filled in for example.

2.4.2 *Methods*

Nine participants (4 male, 5 female) took part in Experiment 3. Their ages ranged from 18 to 53 years, with a mean of 25 years. Students of cognitive science received partial course credit, others did not receive any compensation. Participation was voluntary and participants gave informed consent. The study was approved by the university's ethics committee. None of the participants had ever solved a Straights puzzle before, all of them had at least a little experience with Sudokus. At the beginning of the experiment, the rules of Straights puzzles were explained in six screens. During the experiment participants could re-read the rules at any time. Next, participants were instructed to think-aloud during the entire experiment. To familiarize participants with the think-aloud method, they solved a three-digit addition and a two digit multiplication task and were instructed to describe their solution process speaking out loud. Subsequently, they filled one warm-up 4-by-4 Straights puzzle (recorded as trial 0). Only then did they start with the six main six 6-by-6 puzzles of this experiment. To fill in a digit they had to click on an empty cell and then enter the digit via the keyboard. We recorded their fill-

ing actions, mouse movements, utterances and also eye movements. We used an SMI RED250 eye tracker with a sampling rate of 250 Hz.

2.4.3 *Timing and accuracy*

Filling all puzzles took between 50 and 122 minutes per participant, with a mean duration of 80 minutes (SD: 25). The recorded times and errors per trial are shown in Table 2.4. It can be seen that the smaller warm-up puzzle (trial 0) was solved much faster than the subsequent bigger puzzles. The times and errors improved over the course of the first three puzzles (trials 1, 2 and 3) before increasing again. Trials 2, 4 and 5 were pretty similar performance-wise, but the last puzzle (i.e., trial 6) posed a bigger challenge and required much longer solution times and had a high error rate again.

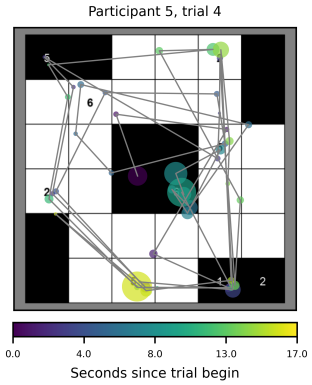
	Time (minutes)			**Errors**			
Trial	Mean	SD	Median	Mean	SD	Median	Per cell
0	3.6	2.4	2.8	5.8	6.9	1	0.72
1	16.1	11.7	13.7	18.9	24.0	5	0.99
2	13.9	10.5	10.2	10.6	16.8	2	0.59
3	7.8	6.5	6.5	4.1	6.0	0	0.18
4	10.9	4.8	10.4	4.2	6.1	1	0.22
5	10.1	5.2	10.4	6.4	10.7	3	0.32
6	17.8	8.9	15.3	16.1	16.3	13	0.85

Table 2.4: Straights task: Time and errors per trial in the Straights experiment

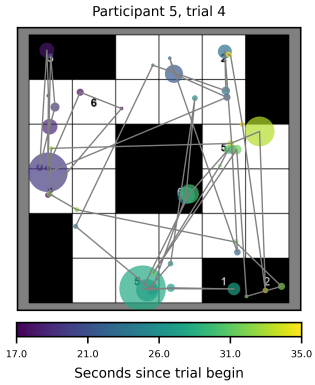
2.4.4 *Eye-tracking*

Thanks to the eye tracker we know at all times at which part of the puzzle the participant looked. Especially the search phases are often not well commented, here the eye tracker gives information about the focus of attention that is otherwise unavailable.

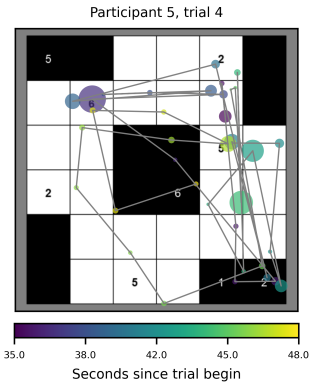
See Figure 2.13 for an example of the search phase at the beginning of a puzzle. The example features participant 5, a person with relatively few errors but quite terse utterances in the think-aloud protocol (11.1 utterances per entry as compared to the average of 29.5, see Table 2.5). This is a participant where the additional information from the eye tracker is most promising, because they probably reason thoroughly but often do not voice the reasoning process in sufficient detail to understand it based on the think-aloud protocol. During the first 17 seconds (Figure 2.13 (a)), the participant looked at the entire board, especially at all the given digits in the puzzle. The following 18 seconds seem to be made up of several short episodes (Figure 2.13 (b)): First, they shortly focused on one three-cell straight with one given digit (column 1, 6 fixations), but seemed to give up on it rather quickly. The other fixations and saccade patterns in this time frame are a bit harder to understand. There are more vertical than horizontal saccades, so the participant seemed to be focusing on columns rather than rows. There were two longer fixations on the 5 in the bottom row and also a back and forth between this five and the



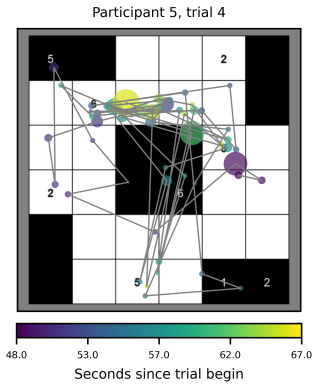
(a) Looking at all given digits



(b) Column 1, digits 5 and 6



(c) Row 2, column 5



(d) Final reasoning: digit 5 in row 2

Figure 2.13: Fixations and saccades of participant 5 at the beginning of trial 4. They only talked about the final reasoning, not the search before. Transcript (translated from German): *"There [row 2] needs to be a 5. Can't be here, here nor here. Therefore it has to be there."*

one in the third row, fifth column. The next figure (Figure 2.13 (c)) shows first a focus on the second row, then a search in columns five and six. The last 20 seconds prior to the entry, the participant looked at all the relevant parts for the reasoning basis for this entry (Figure 2.13 (d)): They looked up and down the second row, in which a 5 needed to be placed, as they mentioned in the think-aloud protocol. Of the four empty cells in the row, three can be excluded as locations for the 5, because of other instances of 5 in intersecting columns. They looked at each of them during this episode. The utterance for the episode is (translated from German): *"There [row 2] needs to be a 5. Can't be here, here nor here. Therefore it has to be there."*. Note that two fives were already looked at intensely at around 25 seconds after stimulus onset, i.e., 20 seconds before the final reasoning episode seems to begin.

It might be possible to do more automated analyses with the eye-tracking data, detecting episodes of general search on the entire board and more focused attention on smaller parts. However, we did not do that. To understand the reasoning processes think-aloud protocols were more helpful for us as they not only inform about what information was used, but also how.

2.4.5 *Think-aloud protocols*

ID	Entries	E./cell	Min.	Sec./e.	Words	Words/e.
1	213	1.69	71	20	4819	22.6
2	348	2.76	68	12	3326	9.6
3	192	1.52	70	22	5339	27.8
4	483	3.83	122	15	8558	17.7
5	178	1.41	104	35	1976	11.1
6	**172**	**1.37**	**50**	**18**	**5474**	**31.8**
7	**154**	**1.22**	**51**	**20**	**6090**	**39.5**
8	**127**	**1.01**	**75**	**36**	**10439**	**82.2**
9	283	2.25	108	23	6671	23.6

Table 2.5: Time, filling events and utterances per participant in the Straights experiment. Entries are abbreviated as *e.*

Table 2.5 compares some high-level data of the problem-solving behavior of all nine participants in this experiment. In total, there were 126 empty cells to fill. Participant 8 only made one typo and thus needed 127 filling events (one cell was filled twice to correct the typo). All other participants made substantially more errors, which included typos, reasoning errors and "just trying out" an assignment. Participant 4, with 483 filling events needed by far the most attempts of all participants (on average this participant filled each cell about 3.8 times).

All think-aloud protocols contain interesting sequences and help us understand how each participant went about solving the puzzles. However, behavior of participants who guess a lot or who frequently make mistakes and then reason based on a faulty puzzle state is hard to analyze formally. Therefore, only the think-aloud protocols of participants 6, 7 and 8, who have both the fewest errors and the most words per filling event, were coded and analyzed in more detail. See Table 2.6 for an overview of the labels and how often they were used by each of these three participants. The coding labels for this experiment do not only include categories for filling and search events, but also reasoning steps that *reduce the set of possibilities*: either the digits a cell could take, digits that are allowed in a unit or straight or locations where a digit is allowed to be placed.

Of those three participants, the relative frequencies of labels do not change much over the course of solving the different puzzles. On average, the analyzed participants used between 3 (participant 6) and 4 (participant 8) labeled utterances per entry. That means, they often verbalize reasoning steps that do not directly lead to an entry. This is also reflected in the relative frequencies of the different labels: Three labels of the category *refine* (*reduce*, *possible digit* and *required digit*) are by far the most common labels. Together they make up more than 65% of all labels for each participant (see Table 2.6). By definition, reasoning steps of this kind do not lead to an entry of a digit, they refine and make explicit the knowledge about *possibilities*. Based on these refined possibilities, the participant is oftentimes able to find the definite assignment of a digit in a next step. If it does not

Table 2.6: Description and examples for each label, with relative occurrence per participant in the Straights task.

	Participant				
Labels	6	7	8	Description	Example
Filling					
last-in-unit	0.09	0.09	0.08	last cell of a unit	"And in the last remaining cell I'll simply put the 3."
straight	0.01	0.02	0.03	justification based on the straight	"Could be 2 or 4...ah no, the 2 can't connect the straight. So 4."
try	0.01	—	—	plausible guess, not sure the action is correct	"okay, let's just try it. I'll just put a 2 here."
fill?	0.05	0.09	0.01	unclear justification	"And here is a 1."
Refine					
required digit	0.30	0.25	0.34	mentioning a digit that *has to be* in some unit or straight	"One of them has to be a 4."
reduce	0.22	0.20	0.21	reducing the digits that are allowed in a cell or straight OR reducing the cells that can take a digit	"The 6 can't go in the cell on the bottom."
possible digit	0.17	0.25	0.12	mentioning a digit that *could* go into a cell or straight	"As a partner for the 2, there has to be a 1 or a 3."
hypothesis	0.04	0.01	0.01	mentally testing the consequences of filling a certain digit in a certain cell	"Alright, if I had a 1 here, I would need a 3 and a 5. 3 would have to go here and 5 there."
Other					
delete	0.06	0.04	—	deletion of wrong inputs	"Let's start over."
other	0.06	0.06	0.19	for example, talking to the experimenter, general comments	"Ah, right, I am supposed to talk."

directly lead to an entry of a digit in one or two steps, and the focus of attention is moved to a different part of the board. It is unclear, however, how much of the generated knowledge is retained over a longer period. Sometimes participants go back and clearly remember their previous inferences: this time over, they only take very few digits into account as possibilities for a cell. On other occasions, however, they have to go through the same reasoning process again and do not recall the result from the last time.

Formal model We built a formal model in the form of a program that can solve all the puzzles of Experiment 3 by applying similar rules as the ones captured by the labels of the think-aloud protocols. The representation we use for the reasoning steps are several possibility-dictionaries. For each empty cell and each straight with open cells, we store what digits could possibly be filled into them. Most rules work on reducing the set of possible digits. For reaching a point where a digit can be entered, usually two or three rules need to applied on the puzzles used in this experiment.

It is difficult to formally asses the similarity of the reasoning chains of a participant and that of the model. Often, participants articulate multiple reasoning steps about one part of the puzzle, abandon that chain of thought and continue somewhere else. Furthermore, they often do not mention some piece of information necessary for the deduction. So the think-aloud protocols neither show a complete list of all reasoning steps, nor do they mention *only* reasoning steps necessary for the next entry.

Hence, when matching a human trace with a model trace, we can neither require all human labels to be present in the model trace, nor all model steps to be mentioned by the human. Nevertheless, there is good correspondence between the labels and the rules we implemented and qualitatively we are able to match the relevant processes quite well.

Some observation regarding participants with many mistakes Participant 4 is the one with the longest time per puzzle (40 minutes for the first "real" one as opposed to the 16 minute average). They tried to find a clearly unambiguous cell-digit combination, but did not look for it very long. Instead, they also went ahead and filled cells with digits which were not the only possible solution. In the first puzzle they did not understand all the rules, so some errors arose because of a lacking understanding of the rules. At the end of the first puzzle they had understood all the rules. Over the course of the first three puzzles they got better in several ways. They kept using guessing or "just trying out" but did so in a more informed way: Most importantly, they knew the rules. They also got better at spotting immediate inconsistencies (e.g., did not fill in the same digit twice in a row or column; already in the beginning they knew the rule but did not always look both into the row and the column). Furthermore, they got better in the search for promising places to continue filling

the puzzle, i.e., either cells where one constraint results in an unambiguous digit assignment or locations where several constraints intersect. And last but not least, they became better at knowing and remembering which entries were based on logical deductions and which involved some guessing. Hence, they also deleted guessed entries faster when they discovered inconsistencies further along the solving process. This way, they could repair the puzzle state locally and reach correct states without having to delete all entries. Til the end of the experiment, they sometimes overlooked constraints and put digits in cells that led to immediate violations but those events became less frequent.

Very similar observations can be made for two other participants as well. From their utterances it is unclear whether they understood from the beginning that only one unique solution exists, they did not necessarily intuit that "no obvious violation now" does not mean "has to be (one of the) correct solution(s)". We can be relatively certain that after a few puzzles they understood that each puzzle has just one solution and that they had to find it.

2.4.6 Discussion

The Straights puzzles were the most difficult of all puzzles in the experiments presented in this chapter. All nine participants completed the same six puzzles, regardless of the time it required to do so. This led to long and exhausting experimental sessions for the participants, but also allowed us to witness some learning taking place.

The difficulty of the puzzles required participants in many cases to reason about sets of possible digits and how these possibilities constrain other possibilities. Another approach chosen was to just fill in one of the options and play it out on the board, instead of reasoning multiple steps ahead mentally. The requirement to reason about *possibilities* instead of just the digits on the board is the foremost difference to all the other puzzles we presented to our participants in this thesis. This does not mean that reasoning based on possible digits cannot occur in Sudokus or Latin square puzzles, we just chose puzzles that were simple enough as to not require such steps. The requirement of reasoning with possibilities (or testing them by filling cells based on "guessing") made labeling the utterances as well as finding a formal model for the processes much harder. Reasoning based on theoretical possibilities is harder to observe, the digits on the board can be seen, but the *possible digits taken into consideration* for cells or regions can only be understood if the participants are very thorough at verbalizing their thought process. If not, there is much more room for misunderstandings between the experimenter and the participant in the labeling process. It is also much harder to find a list of valid reasoning steps in each situation, when not all underlying facts are visible on the board but some only exist in the head of the participant. Even if the participants once talked through some candidate reduction process and it is quite clear what they consider

as possibilities in that moment, it is very unclear to the experimenter how long this derived knowledge remains available to the participant. We know that both cases occur, sometimes they forget and go through the reasoning steps again, sometimes they remember and fill in digits without much of a comment. Reasoning with possibilities instead of just with the specific digits that are already placed on the board clearly poses a challenge to the working memory. It appears quite possible that a better organized working memory is one of the major advantages more experienced players have over beginners.

We explored the use of eye-tracking to gain insight into the problem solving process. We saw that especially for the phases when a participant is searching for where to continue, eye-tracking data has great potential to give additional information besides the think-aloud protocols. Participants are often good at verbalizing reasoning steps they understand well and that they can carry out with relative ease. When they find new deduction rules or when they are unsure how to continue, they usually have a lot more trouble with verbalizing. We did not find a way to more systematically analyze eye-tracking data and generate quantitative data from it. One reason for this is probably also the very explorative nature of the study. With clearer expectations about patterns or different phases it would be easier to use the eye-tracking data. Even though we did not use this type of data much here, we see great potential of this kind of data for future work. For understanding the search phases and how participants decide where on the board to continue, eye-tracking data is very promising. A lot of skill of successful players will come from quickly finding locations where enough constraints hold so that a definite assignment can be made.

Another very interesting finding of Experiment 3 is that the less skilled puzzle solvers change their behavior over the course of the experiment. It is difficult to quantify, but a few points are quite clear when looking at the solution traces. An important prerequisite is, of course, for the participant to fully understand all rules, which took some participants one to two puzzles. There is also a difference between *knowing* all constraints and *checking* whether they are all satisfied before entering a digit. Some of the participants with more errors simply did not always check in all relevant units, but rather restricted their reasoning to one unit only. Over time they became somewhat better at checking all constraints before entering a digit. The most important change, however, was their better grasp of and memory for the distinction between "sure entries" and "entries based on guesses." In later puzzles they went back and deleted the content of cells that was not derived by logically sound and compelling reasoning whenever they encountered inconsistencies further along the solution process. They could thus locally repair a faulty state in a more promising way as opposed to randomly changing assignments or deleting all entries, as they resorted to in the beginning.

2.5 *Experiment 4: Mini-Sudoku (again)*

We used a 4-by-4 Sudoku as a warm-up exercise in the beginning of several online experiments, one of which will be discussed in later sections (see chapter 3). The aim of this warm-up puzzle was to familiarize participants with the experimental interface and to make sure they understand the Sudoku rules before the main experiment started. Participants could only start the main experiment once they had filled the 4-by-4 Sudoku correctly.

Here, we will analyze the data from this warm-up trial. Even though we only have the filling events and no disambiguating information such as eye-tracking, mouse-tracking or think-aloud protocols, the data set is interesting to look at, mainly because there are so many participants that we can conduct more quantitative analyses than in any of the previous experiments.

2.5.1 *Methods*

Before the respective experiment began, participants indicated how often they solve Sudokus or similar puzzles on a scale from "never" to "several times per week". Then the experiment began with an explanation of the rules of Sudoku: The three unit types (row, column, and box) were named and highlighted on an image and it was explained that each digit has to occur exactly once per unit. Subsequently a small 4-by-4 Sudoku was displayed and had to be solved by the participants. To do so, participants had to click on an empty cell and then enter a digit via the keyboard. While filling the warm-up puzzle, a button with the words "repeat the rules" was always available to the participants. By clicking on it, participants could go back to the explanation of the Sudoku rules from the beginning. As soon as all empty cells were filled, a button with the words "check solution" appeared. Upon pressing the button, all correctly filled cells turned green, those with a wrong digit in them turned red. If there were some mistakes, they could press a button labeled "delete wrong input", which reset the colors to neutral gray and emptied all cells with wrong digits in them. Correctly filled cells remained filled. The participants could then proceed from this partially filled Sudoku. Only if all entries were correct, participants could proceed to the main part of the experiment.

Overall, we have 253 completed warm-up trials from five separate experiments: One them is Experiment 2 from chapter 3, the others were completed as Bachelor's or Master's thesis and are not further discussed in this work (Bras, 2021; Hanek, 2022; Katahra, 2022; Rothkegel, 2023). We did not record age and sex in all experiments. Students from the Technical University Darmstadt and (in one of the five experiments) Technical University Kaiserslautern received partial course credit for participation. Participation was voluntary and participants gave informed consent. The studies were approved by the university's ethics committee.

Move classification With the Prolog model we developed for the analysis of the think-aloud study with 4-by-4 Sudokus (see subsection 2.2.3), we could also classify the moves in this study. As we do not have any other disambiguating cues such as the mouse movements or utterances, we use a subset of the original labels, leaving out *2-open* and *only-digit-missing*. We also ignore the basis unit, as this is often ambiguous. Whenever a *last-in-unit* label applies, we take this label, as it is the most simple explanation of the move and was a very highly used reasoning pattern in the think-aloud study. In cases where it is not applicable, and either *cell-complex* or *digit-unit* are, we take these as labels. In the few cases where both *cell-complex* and *digit-unit* are applicable, the move gets the label *unclear*. Correct entries that could not be explained by any of the Prolog rules get the label *no label*. The most probable explanation for these cases is faulty reasoning or a lucky guess. Moves that are either entering a wrong digit into a cell or are made in faulty board states are labeled with *error*.

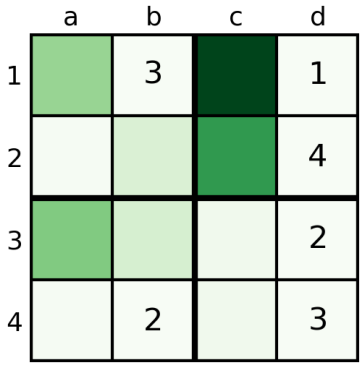


Figure 2.14: The warm-up Sudoku: Heatmap for the most frequently filled first cell (the darker, the more participants filled this cell first).

2.5.2 *Results*

It took participants between 17 and 2471 seconds to fill in the puzzle, with a mean of 86 seconds (SD: 159) and a median of 61 seconds. The puzzle had 10 empty cells that had to be filled, but many participants needed more filling events than 10, meaning that they filled cells several times, presumably to correct mistakes. In total, we recorded 3384 filling events, that is 13.23 on average per person. Of course, there are many participants without a single mistake and some who needed many filling events to get the puzzle right. 120 participants had at least one mistake, 133 had only correct puzzle states. Figure 2.14 shows the puzzle together with a heatmap indicating which cells were most often filled first.

Stage	Poss. states	Vis. states	Coverage	Most frequent
1	10	10	1.0	76
2	45	16	0.36	77
3	120	28	0.23	71
4	210	37	0.18	27
5	252	45	0.18	76
6	210	44	0.21	64
7	120	49	0.41	45
8	45	31	0.69	65
9	10	10	1.0	71
10	1	1	1.0	253

Table 2.7: Warm-up data: Theoretically possible and actually visited states per stage of the filling. When one cell is filled by the participant, there are 10 different possible puzzle states and all of them are realized in our data. "Most frequent" displays the number of participants in the most visited state.

In free-filling experiments, the problem states the participants see quickly diverge from each other. They all start with the same initial puzzle, but depending on where they begin to solve it, they will encounter different intermediate states and will thus also have different solution tactics available at different moments. Each participant has to fill in all 10 empty cells. We use the term *stage* to describe

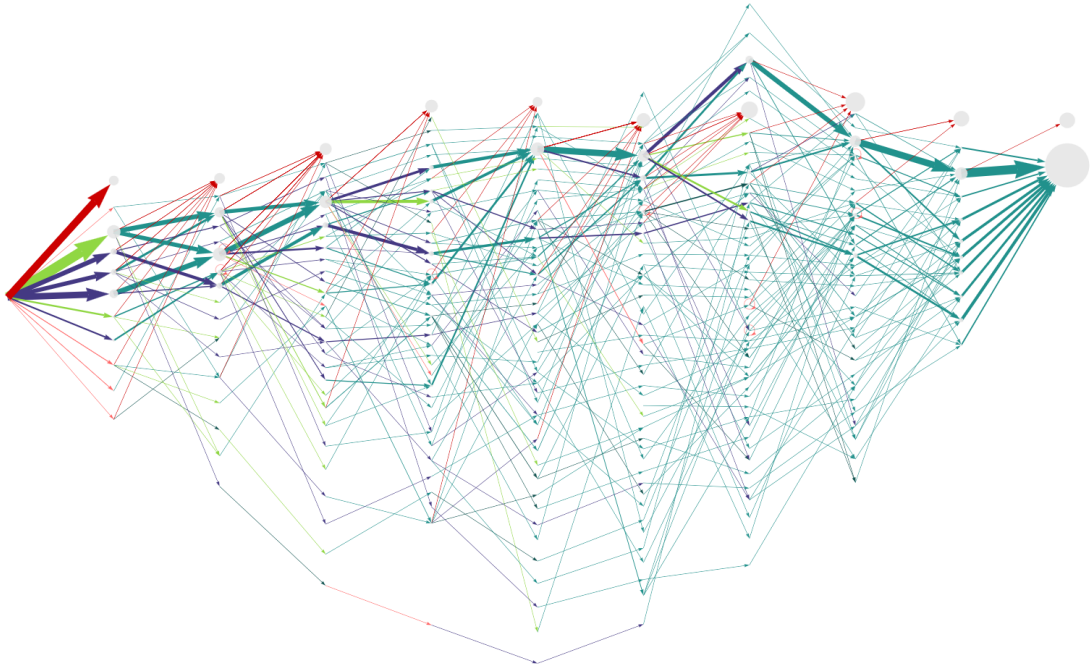


Figure 2.15: A graph displaying all the states that were actually visited by participants. All nodes on the same x-axis belong to the same stage, i.e., have the same number of filled cells. All error states are summarized in a single node per stage (the one with red incoming arrows towards the top). Colors of the edges indicate the rule that would lead from one to the other state. Green is cell-based reasoning, blue digit-based reasoning, teal is a *last-in-unit* rule and red are transitions that cannot be labeled by our Prolog program. The thicker an edge, the more participants used this transition.

all puzzle states with the same number of filled cells. At the first stage, when one cell has been filled, there are 10 possible states of the puzzle (considering only correct filling actions). At the second stage, there are $\binom{10}{2} = 45$ possible correct puzzle states. Possible states increase to $\binom{10}{5} = 252$ in the fifth stage, before they decrease again. See Table 2.7 for a list of the number of possible states per stage. Random filling actions would visit all states with the same probability. The participants in our data are clearly not proceeding randomly but follow similar reasoning rules as the participants in the think-aloud studies did. Not surprisingly, they thus do not cover the entire space of possible correct states, but only a portion of it. At the widest point of the state space graph, coverage is down to 18%, i.e., only 18% of possible correct spaces were visited by at least one participant. In most stages, the most popular state is visited by about a quarter of all participants, which shows a strong concentration on few states. The unequal distribution across states can also be seen in Figure 2.15. More popular paths are generally found at the top of the graph. Each correct state that was visited by at least one participant is represented by a node in the graph. The starting state is at the left, all participants start with the same initial state. Nodes on the same x-coordinate in the graph belong to the same stage: they all have the

same number of filled cells. Edges are colored according to the tactic that most probably explains the move: Green: *cell-based*, blue: *digit-based*, red: no implemented tactic (the nodes without leaving arrows are the collected error states of the stage), teal: *last-in-unit* was possible. Note that we do not know whether the reasoning was indeed *last-in-unit*, some participants might have used digit based reasoning throughout the entire puzzle. As *last-in-unit* is the simplest explanation for a move, however, this is what we use for labeling whenever it is available. At the beginning of the puzzle, no *last-in-unit* move was available, and participants had to either use cell-based or digit-based reasoning (or make a lucky guess). No matter in what order the puzzle is filled, there are always at least three states when a tactic other than *last-in-unit* is required. Most participants are in such states when filling the fourth and the seventh digit. It is clear that there are multitudinous paths that are taken by participants, but there are some nodes where many participants come through.

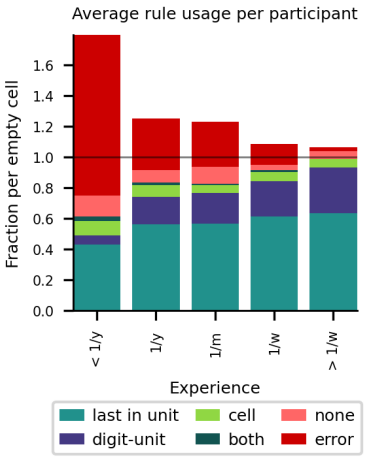


Figure 2.16: Proportion of tactics used for each experience group in the warm-up Sudoku. Beginners make many mistakes and thus have to make almost two filling actions per cell. The most experienced group of participants needs the fewest moves and has the highest proportion of clearly digit-based moves.

Experience Experience was self-reported by the participants on a scale from "never/less than once per year" to "several times per week". 58 participants answered with "never/less than once per year", 73 participants with "once per year", 75 with "once per month", 25 with "once per week" and 22 with "several times per week".

The kind of tactics that were used differ between the experience groups. Figure 2.16 shows a summary of tactic use for each experience group. It is very clear that beginners make more errors than others. More than half of the moves of the beginners were wrong or done in a board state that contained an error.

When looking at the very first move in the puzzle, i.e., at a state where everyone started with the exact same given digits, we see clear trends that change with experience. Table 2.8 shows a summary of the percentage of labels for each experience group. *Last-in-unit* was not applicable in that state and no cell could have been filled with *both* cell-based and digit-based reasoning, so these two labels do not occur. The categories *no label*, *error* and *cell-based* reasoning are most common in the least experienced participant group and decline almost monotonically towards the more experienced groups. Only for the *digit-based* label we see the opposite effect of a steady increase from 16% in the least experienced participants to 77% for the most experienced ones.

Experience	No label	Error	Cell-based	Digit-based
<1/year	0.10	0.31	0.43	0.16
1/year	0.05	0.12	0.38	0.44
1/month	0.04	0.13	0.25	0.57
1/week	0.00	0.08	0.16	0.76
>1/week	0.00	0.00	0.23	0.77

Table 2.8: Warm-up data: Percentage of labels of the first move for each experience group.

2.5.3 *Discussion*

The time per move is pretty similar in the present study and the think-aloud study with similar puzzles (mean think-aloud study 6.8 seconds per move (SD: 7.1), see Experiment 1 in section 2.2, mean this data set 6.0 seconds (SD: 11.5)). When looking at the median, participants in the think-aloud experiment were considerably slower per move than the participant in the current experiment (median think-aloud study 4.37 seconds per move, median this data set 2.7 seconds). Participants in the think-aloud study were required to utter their thoughts, which is known to slow down and can explain the difference (Berardi-Coletta et al., 1995).

Thanks to the relatively large number of participants who solved the same puzzle, we could quantify how much of the possible solution space is actually covered. We see that the coverage of the possible state space is higher in the second half of the stages (i.e., after five digits have been entered) than in the first half. This fact is not really surprising as the number of constraints in the puzzle steadily increases and more and more cells can be filled with simple reasoning tactics.

A new insight this data set provided is the strong correlation of experience with frequency of digit-based tactics. Most clearly this is seen in the very first move of the puzzle, where more than 75% of the two most experienced participant groups used digit-based tactics. In the least experienced group the percentage of participants using digit-based tactics is only a third of that, namely 16%. The two groups in between neatly interpolate between these extremes. Less experienced participants make more errors, but when they reason correctly they are much more likely than more experienced participants to use cell-based reasoning. More than 60% of the correct entries in the first stage by the least experienced participants were cell-based. In the other free-filling studies already discussed in this chapter we usually did not have enough participants to see robust effects of experience on tactic use.

Lee et al. (2008, study 1) found that in a free-filling paradigm their participants (none had prior experience with Sudoku) used more cell-based than digit-based tactics. When looking at the entire group of participants we, to the contrary, find the opposite effect, more digit-based tactic are used. However, when only looking at the inexperienced participants we see a similar pattern in the data. Therefore, it seems that beginners do tend to first use cell-based tactics before discovering and using digit-based tactics.

2.6 *Overall discussion of chapter*

The reasons for us to start out with free-filling and think-aloud experiments were that we wanted to get an understanding of how different participants might approach the task of filling a digit-placement puzzle. Observing several participants doing the same tasks gives

some general insights into possible solution strategies. With all three different digit-placement puzzles in this chapter (i.e., Sudoku, Latin squares and Straights), we have seen participants using similar approaches to solve them. The most common and most trivial tactic is to place the last digit into an almost full unit. All of our participants apply this tactic in all of the different puzzles. Most participants are very flexible about the basis unit in this case. However, it is not always applicable and more complex reasoning tactics are required to fill a puzzle. When looking for how to continue filling the puzzle, a very common approach used for all puzzles is to focus on one unit of the puzzle, determine which digits are missing in it and subsequently check whether some of the missing digits' position can be determined based on the constraints from intersecting units.

It is possible to distinguish cell-based and digit-based reasoning in all three puzzles. Across all of our experiments, both reasoning tactics were used, but digit-based reasoning more often than cell-based reasoning. When looking only at the group of beginners (the group which stated they played Sudoku "less than once per year"), this trend reverses. Beginners tend to use more cell-based reasoning than digit-based reasoning.

In several of the experiments, we have seen that beginners also tend to just fill in plausible guesses if they do not see an easy way of deducing a cell-digit combination for sure. It is unclear whether they fully understand that all the digit-placement puzzles in our experiments have one unique solution. Filling in a plausible guess has a probability of 50% or 33% of being correct (mostly they make entries after reasoning with a result of "either here or there" (digit-based) or "either this or that digit-based" (cell-based), or at most 3 possibilities). It might be that some participants think "if there is no contradiction at this stage of the puzzle, it cannot be false" (and maybe several correct solutions exist for the puzzle). Or they just do not see any other possibility than trying.

More experienced players often rather play the options through in their mind: they test what consequences one of the options would have on other already relatively restricted parts of the puzzle. Playing through such an option mentally places a high demand on working memory, remembering where which digit was assumed to come and then reasoning based on these new constraints that are not visible on the board yet. If they find a contradiction after two to three moves, they know that only the other option can be the correct one and can assign it for sure. Beginners are probably not familiar enough with the reasoning processes to be able to carry out these reasoning chains mentally. When they reach a contradiction after a plausible guess they filled in a few moves earlier, they often do not recall where they made a guessed move and are not able to retrace their reasoning and delete the parts that build upon a loose foundation.

The free-filling paradigms of this section allowed us to observe participants approach different digit-placement puzzles without ad-

ditional experimental restrictions. Observing several participants doing a small set of relatively similar tasks gives some general insights into possible solution strategies. We saw that there is quite a variety of approaches, although most of them can be well described by a small set of rules. Some beginners' behavior is less accurately described by simple deductive rules, as they commit reasoning errors and fill in wrong digits.

There are some drawback to doing experiments with free-filling paradigms. One difficulty is that as soon as participants start filling the puzzle, the problem states they encounter differ, as there are many different orders in which a puzzle can be solved. The diverging puzzle states make it harder to aggregate and compare data across participants. Another disadvantage of the paradigms used in this chapter is that a large portion of the filling events is based on very simple reasoning tactics. In the 4-by-4 Sudokus more than half of the entries were based on *last-in-unit* rules. It is more informative to observe more complex reasoning tactics.

A problem with free-filling experiments in combination with collecting think-aloud data is also that it is difficult to scale up and collect and analyze data from significantly more participants. Data analysis for think-aloud protocols is very labor intensive. With clear expectations about possible reasoning patterns, it was possible in section 2.5 to analyze data from a free-filling experiment without recorded think-aloud traces. By leveraging what we learned in previous experiments, we could infer tactic use in some cases based solely on the entries participants did.

All in all, the work in this chapter is a good foundation to do more controlled experiments. Knowing how participants approach the task of solving a digit-placement puzzle "naturally" without notable constraints from the experimental interface helps to design experiments to answer specific questions. Questions that will be explored in chapter 3 and chapter 4 include: Do all participants discover digit-based and complex cell-based tactics, when that is the only way of solving a task correctly? Do the two tactics differ in difficulty? Under what circumstances is it useful to use which tactic? What might be the reason for a shift in tactic use with increasing experience? And finally, in chapter 5 we look at answering the question whether we can describe the different preferences for rules of individual participants more formally. Building upon the understanding we gained by the free-filling studies, we are well equipped to answer these questions.

Chapter 3

Restricted filling

This chapter has been published as *Behrens, T., Räuker, M., Kalbfleisch, M., and Jäkel, F. (2023). Flexible use of tactics in Sudoku. Thinking & Reasoning, 29(4):488–530. Copyright Taylor & Francis, available online: https://www.tandfonline.com/doi/abs/10.1080/13546783.2022.2091040.*
Some terminology is adapted to fit better with the other chapters and references to the other chapters are inserted where appropriate.

3.1 Introduction

When we solve problems there are usually a number of different strategies that we can use. It is a hallmark of intelligence that we, as humans, are able to flexibly choose between them. Siegler and colleagues have shown that even children already know and use a wide array of strategies. For example, when learning simple addition they usually start with counting on their fingers. Counting all the way from one to the sum is effortful. Hence, they quickly acquire various shortcuts before memory retrieval becomes the dominant strategy (Siegler and Jenkins, 1989). They flexibly interchange strategies and mostly use the most appropriate one, i.e. they use effortful strategies only when simpler strategies fail (Siegler, 1991). Many studies have shown that not just children but also adults often know and use several strategies when engaging with a task. Be it mental rotation (Alderton and Larson, 1994), memory tasks (Alderton and Larson, 1994; Brown, 1995), sentence-picture comparison (Clark and Chase, 1972; MacLeod et al., 1978), deciding between investment options (Lee et al., 2019; Rieskamp and Otto, 2006), fraction magnitude comparison (Fazio et al., 2016), or fault finding in an electrical circuit (Friedrich and Ritter, 2020), there is always more than one strategy. Depending on the situation each strategy might entail different amounts of cognitive effort or might be more or less reliable. Being able to flexibly choose between strategies is thus an essential aspect of problem solving.

Here, we propose that Sudoku provides a suitable test-bed to study and model this flexibility. Previous studies on Sudoku have established that there are two dominant and well-understood strategies (Lee et al., 2008). It is good experimental practice to restrict participants' behavior to one strategy either through task requirements

or instructions. In this way, a strategy can be studied in pure form. Only after all the dominant strategies are understood can we begin to understand their interaction and how people choose between them. However, as we will show in this paper, for reasonably complex and realistic problems, like Sudoku puzzles, it is hard to restrict participants' behavior appropriately and only through modeling several strategies simultaneously can we explain their behavior.

If several strategies can be used in a task, this clearly complicates the design and analysis of problem solving experiments. If participants just use one kind of strategy, researchers will be able to average over all participants and trials, making quantitative analyses easier to carry out. If several strategies are used in the course of the experiment, averaging over them will often lead to erroneous conclusions, as shown in detail by Siegler (1987). Even if a model that assumes only one dominant strategy provides a good overall fit, there is no guarantee that participants only used this strategy. The so-called min strategy for children's addition explained much of the variance in experimental data (Siegler, 1987) and so did the linguistic model for the sentence-picture comparison task (Clark and Chase, 1972). Still, modeling two or more different strategies dramatically improved the fit in both cases (MacLeod et al., 1978; Siegler, 1987) and we present a similar result for Sudoku.

Unfortunately, it is often not easy to tell which strategies participants use in an experiment. Overt responses and response time measures are the bread and butter of experimental psychology but they are seldom enough to classify trials according to the strategy that was used. While cognitive scientists often object to verbal reports, they are hard to avoid in problem solving research and less problematic than usually assumed (Jäkel and Schreiber, 2013). Siegler and Jenkins (1989), for example, asked children after each trial how they obtained the result, and the answers matched well with the overt behavior on the trial as well as with the time they needed. Other possibilities to help distinguish strategies include concurrent verbal protocols and eye- or mouse-tracking. As participants can be very inventive and use strategies that researchers did not expect, researchers can also use these methods to discover strategies in a data-driven way. In general, if researchers are too narrow-minded and expect just one strategy, their experimental setup will often miss the variability that naturally occurs when people solve problems of interesting complexity. This happened, for example, in a study by Ritter and Bibby (2008). They modeled only one strategy for finding faults in electrical circuits. Their SOAR model fit well for 8 out of 10 participants and could explain trial-to-trial variability. However, in a replication of the experiment without a worked example at the beginning, only 4 out of 35 participants were well matched by the model (Friedrich and Ritter, 2020). In this second experiment, Friedrich and Ritter (2020) found and subsequently modeled five more strategies. In our studies on Sudoku we therefore use a mixed-methods approach where response time experiments are combined with modeling and verbal

reports to make sure that we do not overlook any relevant strategies.

As there are usually several strategies for any problem, how do people choose which one to use? There is some research on different factors influencing the choice of strategies: Participants adapt their strategy to the task and learn with experience what the optimal strategy is (Gunzelmann and Anderson, 2003; Lee and Johnson-Laird, 2013; Rieskamp and Otto, 2006). Alderton and Larson (1994), however, found that participants who switched strategies optimally in one task might perform far from optimal in another task (with respect to strategy switching). They concluded that strategy switching is not governed by a central metacognitive process but is task-specific. Personal ability (e.g. verbal ability, spatial orientation) also influences what strategies participants use in a task, such that participants who are generally strong in a domain more likely use strategies relying on operations in this domain (MacLeod et al., 1978; Roberts et al., 1997).

While experience and personal ability are probably also important for strategy selection in Sudoku, here, we study the role of task requirement and task instruction. Sudoku problems can conveniently be constructed in a way that they are only solvable with one of the dominant strategies and not the other. Hence, if participants start with the wrong strategy they will have to switch. Sudokus thus provide us with a suitable test-bed to study strategy choice under changing requirements. Furthermore, through the task instruction we can induce participants to start with one or the other strategy. This is achieved by presenting the problem either as having to fill out a specific cell or as having to find a cell for a specific digit. The way a problem is presented – and therefore represented by the problem solver – is well known to have a huge effect on problem solving (Newell and Simon, 1972; Kaplan and Simon, 1990). It is therefore surprising that previous studies on Sudoku have focused on one particular problem presentation that highlights a cell and requires participants to fill in the correct digit. We show that previously reported pronounced differences in difficulty between different strategies are confounded with this specific mode of presentation.

3.1.1 *Overview of chapter*

In a series of three experiments we try to understand the influence of the two well-known dominant strategies on the difficulty of solving Sudoku puzzles. Experiment 1 is a think-aloud study on how people approach the standard task where one cell is highlighted by the experimenter. In Experiment 2 we directly compare the performance of participants with the standard instruction and a second instruction where the task is to find a cell for a specific digit. As much of the previous literature on Sudoku is focused on the influence of complexity on task difficulty, we conduct Experiment 3 to be able to compare our results to previous studies. Finally, we introduce a simple process model that implements both strategies and

explains how strategy, complexity, task requirement, and instruction together determine task difficulty. But before we can present the experiments and their results, the next section first provides necessary background information on Sudoku.

3.2 *Background on Sudoku*

While Sudoku puzzles are probably known to most readers, a few words about their relation to Latin squares are in order. A Latin square is a square grid of side length N where each field is filled by a digit from the range from 1 to N. Each digit has to appear exactly once per row and once per column. In a Latin square task (LST) the grid is only partially filled and the task of the player is to fill in the missing digits without violating the constraints. Sudoku is similar, it only introduces an additional unit type to the already defined *rows* and *columns*, namely *boxes*. Boxes are of size $\sqrt{N}$-by-$\sqrt{N}$ and the same constraint holds: each digit has to appear exactly once per box. Thus, Sudoku-grids need to be of a side length that is a quadratic number in order to make the boxes possible. Typical Sudokus are of size 9-by-9 and examples are shown in Figure 3.1. The term *peers* describes the set of all cells that are connected to one cell by sharing a unit: the same row, column, or box. Although both puzzles are usually referred to as digit-placement puzzles, numerical properties do not play any role in the problem solution. The digits could be replaced by arbitrary symbols without changing the nature of the task.

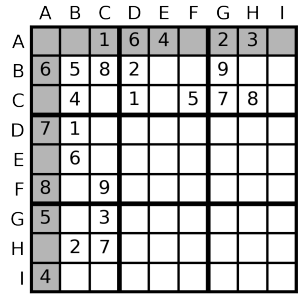
3.2.1 *Basic tactics*

So far, we used the term *strategy* to be consistent with the terminology in the literature on strategy selection. Following Lee et al. (2008) we will, from now on, use the term *tactic* to talk about different reasoning patterns participants use for filling in a single digit in a digit-placement puzzle. Tactics describe more local phenomena and a systematic sequence of tactics would then make up a strategy. According to this definition, some of the phenomena discussed in the introduction would be better described as tactics instead of strategies. Following Lee et al. (2008) we distinguish two basic tactics for filling in single cells: a cell-based tactic and a digit-based tactic. The tactics described in the following are inferring a new digit for some cell, based on the digits already present on the board.

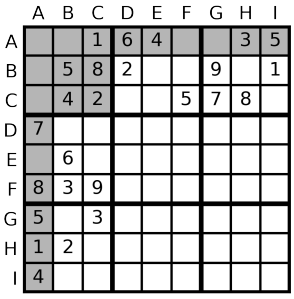
The cell-based tactic Participants select a cell and look at its *peers* (i.e., those cells that are in the same unit as the cell, for example in the same row). All digits that appear in the peers of the cell can be excluded as possible values for it. Lee et al. (2008) therefore refer to this tactic as *exclusion tactic*. Examples for situations in which the cell-based tactic is applicable can be seen in Figure 3.1 (c) and (d). All digits except for the 9 can be excluded for the cell AA in the upper

left corner because they appear already in the same row, column, or box.

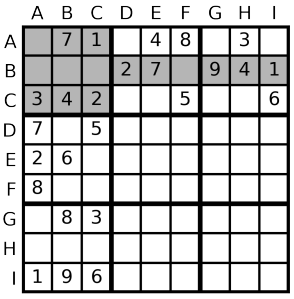
The digit-based tactic Participants start with one specific unit and a specific digit in mind: For example "Where in this box can I place the 9?". Each digit has to appear once in each unit (row, column, box) of the Sudoku. So if it is not yet placed in a unit, one can start looking for candidate cells to place it in. Cells of the unit which are already filled drop out of the candidate set without much further consideration, but some of the empty cells often can be excluded, too. Consider Figure 3.1 (d): the digit 9 is still missing from the upper left box. The 9 in the second row at BG excludes cells BA and BB and the 9 at the bottom of the second column in IB excludes cell CB as possible location. The only cell where a 9 does not lead to a violation of Sudoku rules is the upper left cell AA. Lee et al. (2008) call this tactic *inclusion* because one tries to include a digit into one specific unit.



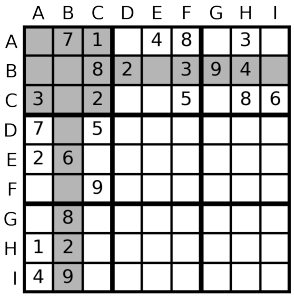
(a) cell-based tactic, 2 units required (cb-2)



(b) cell-based tactic, 3 units required (cb-3)



(c) digit-based tactic, 2 units required (db-2)



(d) digit-based tactic, 3 units required (db-3)

Figure 3.1: The correct solution for the cell AA is 9 in all four puzzles. The units needed to reach this result are marked with shaded background. The best starting question for the cell-based tactic is "Which digit is allowed in cell AA?", all digits except for the 9 can be excluded. The best starting question for the digit-based tactic is "Where in the upper left box (AA-CC) can I put the digit 9?", all cells of the box except for AA can be excluded.

3.2.2 *Tactics and their complexity*

Birney et al. (2006) were the first to introduce the notion of complexity to the study of digit-placement puzzles. They classified puzzles according to their *relational complexity* (Halford et al., 1998), counting how many variables a reasoning tactic requires. However, they

based their analysis purely on the cell-based tactic and ignored the digit-based perspective. Thus, some of the items are not as complex as they seem in their analysis. Later, Lee et al. (2008) introduced the digit-based perspective to studies of Sudoku. They adopted the term relational complexity from Birney et al. (2006) and did not count variables but *units* required for the reasoning process. Here, we use the same complexity measure as Lee et al. (2008) because it is easy to specify for a given puzzle situation. Note, however, that it is unclear how this measure relates to relational complexity as originally conceived. Relational complexity is defined as "the number of interacting variables that must be represented in parallel" (Halford et al., 1998, p. 805). If there is a way to serialize the problem into a sequence of steps that require fewer variables at once, the effective relational complexity is reduced. The relational complexity of a task can therefore only be determined if one has an explicit process model that spells out how participants solve the task and how many independent variables need to be held in mind at any point during this process. Therefore, contrary to previous studies we do not use the term relational complexity and instead only speak of number of required units (NRU).

According to this measure, we can distinguish three levels of NRU within the cell-based tactic. If all cells but one of a unit are already filled, a deduction can be made by focusing solely on this unit. The unit could be either row, column or box, the NRU is always 1. We label these cases as cb-1. However, in many situations a combination of units is required to deduce the next digit. In Figure 3.1 (a) for example, the union of the values from the row and the column (row+column) is needed to exclude all but one digit. Other possible combinations of units are row+box, column+box, row+column+box. We label all cases involving two units as cb-2 and three types of units as cb-3.

The digit-based tactic can similarly be subdivided. The digit-based tactic always operates on a base unit. Additional units are needed to exclude other empty cells from the base unit as candidate locations for the digit. For example, the base unit in Figure 3.1 (d) is box, and two additional units (one column and one row in this case) are needed to rule out the three other empty cells of the box. We count the base unit and all additionally needed units, which in this case results in the categorization *digit-based with 3 required units*, or db-3 for short. In Figure 3.1 (c) the base unit is a box, and only one unit (the second row) is needed to find out the place for the 9, so it would be classified as *digit-based with 2 required units*, or simply db-2.

Several studies have examined the influence of either complexity alone or tactic in interaction with complexity. All these studies highlight one cell in a puzzle (Sudoku or Latin square) and ask participants for the value that has to be assigned to the cell. In the next trial a new puzzle is displayed in the same fashion. Studies that looked at the difference between cell-based and digit-based reasoning found that the cell-based tactic is easier for participants, expressed either

by faster response times or higher accuracy or both (Lee et al., 2008; Perret et al., 2011). All studies find that response times are longer for puzzles with higher complexities, but whether accuracy is affected is not so clear (Birney et al., 2006; Hearne et al., 2020; Lee et al., 2008; Perret et al., 2011; Qin et al., 2012).

Highlighting one cell and asking for the value to be filled in it might bias participants towards using cell-based tactics. For a better understanding of the role of instructions, we devised a new kind of instruction, meant to facilitate digit-based reasoning. We directly compare the performance of participants with the two different kinds of instruction in Experiment 2. With an experimental design that favors digit-based reasoning the bias flips: Sudokus requiring digit-based reasoning are now solved faster and more often correctly than those that require cell-based reasoning. As much of the previous literature is focused on the influence of complexity and some results remain debated, we conduct Experiment 3 to clarify whether NRU alone influences accuracy and response times. We find that a higher NRU leads to longer response times but not to higher error rates in our experiment. Finally we introduce a simple process model that implements both tactics and explains differences between the cell- and the digit-based tactic, and how both instructions and NRU influence task difficulty.

3.3 *Experiment 1: Highlighted cell in a 9-by-9 Sudoku*

In this first study, we followed the usual practice and had participants fill in a predetermined cell in a different puzzle on each trial. To get a better understanding of how participants approach their task we recorded think-aloud protocols. In their first experiment, Lee et al. (2008) asked participants after each digit they filled in how they knew the answer was correct. This way the researchers could classify the tactics their participants employed to fill digits in cells. Our think-aloud data give information on a more fine-grained level, we see what features of the puzzle they pay attention to, can follow their progress and observe how they switch tactics when a reasoning chain does not lead to a solution.

3.3.1 *Participants and methods*

There were 14 participants (9 female, 5 male), aged between 20 and 56 years (mean: 39.1, SD: 15.3). Their self-reported prior experience with Sudoku puzzles is shown in the last column (XP) of Table 3.2. Two participants had to be excluded due to technical issues during the experiment. Each participant was recorded individually. They filled in the Sudoku puzzles on a computer and were instructed to think aloud during the whole time. The puzzles were chosen to require cell-based reasoning, digit-based reasoning, and some more complex reasoning schemata. Puzzle types were chosen adaptively during the experiment to make the problems interesting for the par-

ticipants. The think-aloud recordings were transcribed after the experiment. Participants were instructed to fill in the correct digit in one highlighted cell for the 9-by-9 Sudoku that was presented to them. For each trial, participants had 120 seconds to find the correct digit, after which the Sudoku disappeared from the screen. Response times were measured from stimulus onset. The duration of the experiment was fixed to 60 minutes. The participants solved around 60 Sudoku-cells in this time, ranging from 46 to 77, depending on their speed.

3.3.2 *Results*

For puzzles that could be solved with the digit-based tactic, the most important base unit was box. When comparing the solution times for puzzles where the digit-based tactic was possible with only one of the three units, those that are box-based are solved twice as fast as those that are column- or row-based (32 seconds vs. 71 seconds). Many puzzles could be solved with the digit-based tactic with two or even any of the base units (row, column, or box). Because the applicability of the digit-based tactic with base unit box was so much more decisive for solution times, we distinguish only this feature in Table 3.1 below.[1] A look in the protocol data confirms that only 4 of the 12 participants talked about the digit-based tactic in combination with rows or columns at all. About half of the puzzles were solvable with the digit-based tactic with the base unit *box*. The number of required units (NRU) ranged from 2 to 5 with a mean of 3.2. Another 21% of the puzzles were solvable with the cell-based tactic with 2 or 3 required units (mean: 2.4). Very few (3%) of the puzzles could be solved in both ways and 23% could not be solved directly with one of the two simple tactics. Some of them were solvable with the digit-based tactic with different base units (row or column) or participants needed to mentally infer *another* digit with simple tactics first, in order to figure out the correct digit for the cell in question.

[1] This observation also matches the results in the 4-by-4 Sudoku think-aloud study in section 2.2 in which box was the basis unit more than 20 times as often as either row or column in digit-based reasoning.

	Count	Response Time [sec.]	Accuracy
Digit-based (box)	376	35.61 ± 27.26	0.93
Cell-based	161	31.19 ± 21.13	0.99
Both	28	25.96 ± 25.72	1.00
None of the above	171	79.10 ± 35.10	0.63
All	739	44.35 ± 33.99	0.88

Table 3.1: Mean response time in seconds and accuracy per trial for different puzzle types in Experiment 1.

On average participants needed 44 seconds per puzzle and were correct on 88% of the trials. Table 3.1 shows that trials in which cell-based strategies were applicable were faster and more often correct. When only cell-based strategies were applicable, our participants needed on average 31 seconds, as opposed to 35 seconds when only digit-based (box) strategies were applicable. This difference is not statistically significant. When none of the simple strategies suf-

ficed to solve the puzzle, participants were much slower (79 seconds on average) and when both worked they were faster (26 seconds on average). Looking at the performance of individual participants, the data show a similar pattern overall.

Think-aloud data The transcribed think-aloud protocols were coded sentence by sentence. The coding scheme is more fine-grained but for our purposes it suffices to distinguish between two classes of labels: cell-based or digit-based utterance. Cell-based utterances in the most basic case exclude digits that appear already in the peers of the cell ("it can't be the 1 because it is already in the row"). A basic digit-based statement is something like "I need a 1 in this box", mentioning the digits that are still missing instead of those that can be safely excluded as candidates.

Table 3.2: Some counts for each participant. *Response Time* displays the mean time in seconds the participant needed per trial, *Accuracy* the fraction of correct trials, *Cb-start* how many trials the participant started with a statement that was classified as cell-based, *Db-start* how many trials were started with a digit-based utterance, *Both* how many trials included at least one statement of each of the two classes and *Trials* gives the number of trials each participant completed. Self-reported experience is shown in the last column (XP: 0–less then once per year, 1–once per year, 2–once per month, 3–once per week, 4–more than once per week).

Participant	Response Time	Accuracy	Cb-start	Db-start	Both	Trials	XP
01	45.39 ± 38.05	0.92	0.27	0.69	0.58	62	4
02	62.53 ± 32.68	0.90	0.88	0.04	0.52	48	1
03	42.58 ± 38.52	0.87	0.68	0.29	0.62	63	3
04	52.97 ± 31.41	0.88	0.49	0.47	0.86	51	3
05	37.29 ± 23.75	0.80	0.77	0.17	0.54	71	0
06	44.10 ± 37.62	0.91	0.81	0.17	0.69	64	0
07	54.25 ± 34.61	0.88	0.68	0.28	0.78	50	2
08	48.38 ± 25.42	0.89	0.45	0.53	0.62	55	2
09	34.16 ± 39.52	0.91	0.72	0.27	0.46	78	3
10	34.98 ± 28.82	0.87	0.76	0.20	0.39	75	2
11	44.44 ± 28.78	0.83	0.70	0.28	0.53	60	2
12	44.80 ± 32.14	0.87	0.73	0.26	0.69	62	1
All	44.35 ± 33.97	0.88	0.67	0.30	0.59	739	

In Table 3.2, it can be seen that 9 of 12 participants started at least 68% of their trials by excluding digits that appear in the peers of the highlighted cell, i.e., classical cell-based reasoning. As the participants did not see the same problems in the same order, and depending on their performance saw a different number of problems that required cell-based reasoning, this number is not directly interpretable. However, only about 25% of all trials could be solved by cell-based reasoning. Hence, many participants clearly have a bias to start a trial with cell-based reasoning. When they did not find a solution with their cell-based approach, these participants mostly

switched to digit-based reasoning, trying out the candidate digits they found with the cell-based approach. An example can be seen in Figure 3.2.

Here, the participant 02 (some Sudoku experience, no regular playing) explicitly goes through all digits from 1 to 9 and checks whether they appear in the peers of the highlighted cell, excluding all but 6 and 9. They then check where the 6 and the 9 occur in rows and columns intersecting the box with the highlighted cell and notice that the 9 can be excluded for the three other empty cells of the box. Therefore they can conclude with confidence that the 9 is the correct digit for the highlighted cell.

Participant 05 (never played Sudoku before) uses only the cell-based tactic for the first third of the experiment. If no unique value is found this way, but several candidate digits remain, they just guess one of them. But after a while they notice the digit-based perspective and see that one of the two remaining candidate digits in that puzzle instance can only go in one of the cells of the surrounding box and concludes correctly that this is the only correct solution. After this trial the participant still prefers to use the cell-based tactic, but when it does not lead to a unique answer, tries digit-based reasoning as follow-up. This is especially successful when only two candidate digits remain after the cell-based approach. With more candidates remaining this participant sometimes loses track of which digits were already tried and which were candidates in the first place.

Others solved the puzzles in a different fashion. Some participants regularly start with digit-based reasoning. An example is displayed in Figure 3.3. Participant 01 (plays Sudoku at least once per week) notices many instances of the digit 4 in relevant units: those that intersect the box in which the target cell is located. The 4 has to be filled in the box somewhere, but only for the highlighted cell this does not lead to a contradiction with the existing digits.

Participant 11 (plays about once per month) used only the digit-based tactic at the beginning of the experiment as the main reasoning step. This person often started by remarking some digits that are excluded for the cell, but switched quickly to testing which digits can *only* go into the cell instead of excluding *all* digits in the peers first. Upon encountering two puzzles in close succession which were not solvable with digit-based reasoning this participant really struggled, exceeded the time limit and explicitly remarked (translated from German): "Okay, why is this so difficult? [...] Oh my god, you have to adjust. This is a completely different system than what you usually do." After that this participant still preferred digit-based reasoning but switched effortlessly to the cell-based tactic if necessary.

Participant 02, trial 5

	A	B	C	D	E	F	G	H	I
A	1	2	8		6	9			3
B			5						4
C	4				3		8	9	2
D		5				3			
E				6	4	7			
F			9						
G	3	9		8				6	
H						1			
I	7			5			4		

Figure 3.2: Typical example of starting with cell-based reasoning, excluding digits that appear in the peers and then switching to digit-based reasoning. Transcript: *"Not 1, not 2, not 3. The... not the 4. Not 5. It could be 6. Not 7, not 8... The 6, the 6 there. The 6 and the 9. 9 can't go there, or there. It has to be the 9."*

Participant 01, trial 3

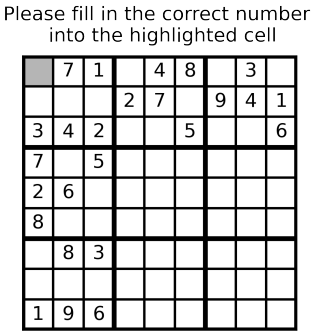
	A	B	C	D	E	F	G	H	I
A				6	4				
B			4	5		8			
C					3		6	4	7
D		4				1		3	
E			2						5
F	3	1							
G			3		6	4	5	7	
H		8	7		2				
I	5	2		1			8		4

Figure 3.3: Typical example for digit-based reasoning. The participant is basing their reasoning on the surrounding box. Transcript: *"... hmm... I'm looking for where to start. There are many 4s already, two come from above [BC, DB]... eh... in the upper row is one, too [GF]. Therefore the 4 has to go here."*

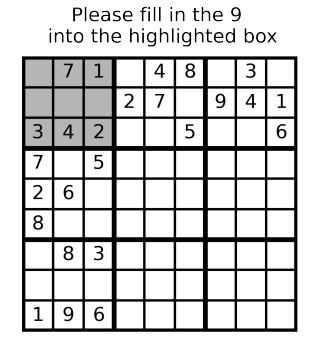
3.3.3 *Discussion*

On the group level this first experiment replicates the results of Lee et al. (2008) and Perret et al. (2011) who found that puzzles requiring cell-based strategies are easier to solve (in terms of accuracy and re-

sponse times). However, our results are not as clear cut and strongly in favor of cell-based reasoning as the previous studies. The accuracy in this experiment was higher for both cell-based and digit-based trials (0.99 and 0.93 respectively) than in previous studies. Lee et al. (2008) for example found accuracies of 0.83 for cell-based and 0.67 for digit-based puzzles (both with 2 required units), but their participants were all beginners who had never solved Sudokus before. As our participants had a wide spread of expertise and most had at least some experience, this difference is not surprising. As we collected think-aloud data we can get a few more fine-grained insights into our participants' reasoning. The participants have preferences for either the cell-based or the digit-based tactic, but they are all able to use the non-preferred tactic as well. Not surprisingly the protocols show that learning takes place. Some participants learn to apply a tactic they did not use at the beginning. Participants often show mixed forms of both tactics. A common pattern is for example to first use cell-based tactics to reduce the candidate set of digits and then follow up with digit-based reasoning to determine the final answer.



(a) cell-based instruction



(b) digit-based instruction

Figure 3.4: The puzzles were the same in both conditions in experiments 2 and 3, only the highlighting and the instruction differed.

3.4 *Experiment 2: The effect of the instruction*

It is possible that the results of our previous experiment are largely influenced by the instructions we used. When only one cell per puzzle has to be filled and this cell is specified by the experimenter and highlighted, it is quite likely that this biases people towards using the cell-based tactic. We therefore conducted an experiment to test the influence of the instructions explicitly. We designed an instruction that was supposed to facilitate digit-based reasoning. We expected that the kind of instruction would have an effect on the initial solution attempt on a puzzle. Our main hypothesis was: (1) There is an interaction between instruction type and required solution tactic. If instruction and required tactic are congruent, performance should be better than in cases where they are not. This hypothesis can be strengthened by: (2) There is no main effect of either instruction type or required tactic. Although Lee et al. (2008), Perret et al. (2011), and Experiment 1 found that the cell-based tactic is easier, we hypothesized that these findings are to a large degree due to the biasing effect of the instructions. As both tactics are very easy to carry out it could be that the effect of the required tactic disappears once instruction type is controlled for. Furthermore, if the difficulty of switching between tactics is symmetric then we also would not expect to see an effect of instruction type. Note that our secondary hypothesis is a null hypothesis that is unlikely to be true exactly (i.e., tactics have equal computation times and switching costs).

3.4.1 *Methods*

We preregistered the experiment and the analysis with the Open Science Foundation on 16 June 2020 (`https://osf.io/2ngc3/`).

Participants 58 participants, (35 female, 23 male), aged between 18 and 62 years, took part. Mean age was 30.35 years (SD: 13.07). We collected their self-reported experience with Sudoku or similar puzzles on a scale from "less than once a year" to "more than once a week." 17 participants selected "less than once per year", 22 "once per year", 10 "once per month", 3 "once per week" and 6 "more than once per week". About two thirds of our participants thus had at most little practice with this kind of puzzles.

Students of psychology and cognitive science participated for partial course credit. All other participants who were recruited from family and friends received no compensation. All participants gave informed consent before participating. Following our preregistration plan, we excluded 12 participants who had an accuracy below 75% from the analysis. Analyses are based on the 46 participants, 23 per condition, who met the inclusion criterion. The number of participants with each experience level can be seen in Table 3.3.

Play frequency	XP level	DB instr.	CB instr.	Sum
less than 1/year	beginner	3	6	9
1/year	intermediate	11	8	19
1/month	regular	5	4	9
1/week	regular	1	2	3
more than 1/week	regular	3	3	6
		23	23	46

Table 3.3: The number of included participants with each level of experience per experimental condition in Experiment 2. In subsequent analyses we group participants with once per month and above under the label *regular*.

Experimental design The experiment was a 2×2 mixed design. The independent variables were *instruction* and *required tactic*. Both can be cell-based or digit-based. If the tactic that is required to solve a puzzle and the tactic that is suggested by the instruction match, we call the condition congruent, if they are different we call them incongruent.

The independent variable *instruction* was measured between participants. Participants were randomly assigned into one of the two instruction groups. In the cell-based instruction the participants had to fill in the correct digit into one highlighted cell of a 9-by-9 Sudoku. This instruction and task is very similar to previous experiments (Birney et al., 2006; Lee et al., 2008; Perret et al., 2011). In the digit-based instruction the participants had to fill in a given digit into one highlighted box of a 9-by-9 Sudoku. The two instruction types are shown in Figure 3.4. In both instruction conditions participants used the mouse to click on the cell they wanted to fill and then entered the digit via the keyboard. The interaction with the experimental interface was thus exactly the same in both groups.

The independent variable *required tactic* was measured within participants. We used four different types of puzzles that differed in the required tactic (digit-based or cell-based) and number of required units (NRU) (2 or 3). Half of the puzzles could only be solved with

the digit-based tactic and the other half of the puzzles could only be solved with the cell-based tactic. The four types of puzzles are shown in Figure 3.1.

Stimuli Nine logically equivalent puzzles were generated from one seed puzzle for each type, using each digit from 1 to 9 once as the correct answer. We exploited the fact that the digits are mere symbols in Sudoku puzzles and can be interchanged. Furthermore, rows and columns belonging to the same set of boxes can be exchanged without changing the relation of constraints. And also bands of three rows or columns belonging to the same set of boxes can be exchanged with each other. By interchanging positions of rows and columns as well as replacing digits with each other we created a set of stimuli that have no directly visible relation with each other. The stimuli were carefully controlled to avoid surface-visible differences between the conditions and exclude as many irrelevant sources of variation as possible. All stimuli have five filled and four empty cells in the target box. Each row and column intersecting the box also had five filled cells, so that they all give the same number of constraints and do not provide an obvious starting point. Each digit occurs on the board two to four times, making none stick out with unusual high or low frequency. When attempting to solve a puzzle with the wrong tactic, three possible answers remain. For puzzles requiring digit-based reasoning, only six values can be excluded for the target cell by looking at the peers, three remain as possible candidates. For puzzles requiring cell-based reasoning, only one of the four empty cells in the target box can be excluded, leaving three cells as possible candidates.

For an example that requires digit-based reasoning look at Figure 3.1 (c) and (d). The correct answer is 9 in cell AA in both cases. When the entire 3-by-3 box is highlighted with the instruction to place the 9 into the correct cell, the answer is easily found, as 3 of the 4 empty cells can be excluded as locations for the 9. When cell AA is highlighted and the instruction is "Please fill in the correct digit in the highlighted cell", the natural starting point is to see what digits can be excluded for the cell. 5, 6 and 9 remain as options in both puzzles. To find the correct answer, you still need to test these candidates using digit-based reasoning. 5 and 6 could still be placed at several locations, but the 9 can only go in cell AA, as placing the 9 in any of the other empty cells of the box would lead to a contradiction with the 9s already there in the rows and columns.

An example that requires cell-based reasoning can be seen in Figure 3.1 (a) and (b). When cell AA is highlighted, one can easily exclude the digits 1 to 8 as possible candidates and fill in the 9 as the only allowed option. But when the entire 3-by-3 box is highlighted and the question is where to place the 9 in it, the answer is not so obvious. Only one of the 4 empty cells can be excluded via a 9 in the same row or column, the other 3 remain plausible options. Only by checking that 9 is the only value that is allowed in cell AA, i.e., using

cell-based reasoning, you can get a definite answer.

Procedure The experiment was implemented with PsychoPy (Peirce et al., 2019) and conducted online on `Pavlovia.org`. The instructions at the beginning of the experiment explained the rules of Sudoku, but gave no hints about possible solution tactics. Participants were instructed to do their best and to work as quickly and as accurately as possible. Before the main experiment started, participants had to solve a 4-by-4 Sudoku completely to familiarize them with the interface and to make sure they understood the rules of Sudoku. The experiment consisted of six blocks and each block consisted of six Sudoku puzzles. After each block the participants received feedback on how many trials they solved correctly in that block. All participants solved the same puzzles and half of the trials were congruent and half of the trials were incongruent with the respective instruction. The order of the trials was randomized for every participant but the first three trials of the first block were chosen to be congruent with the instruction. For each Sudoku participants had 240 seconds to enter their solution. After that the Sudoku vanished from the computer screen and no digit could be entered for this trial anymore.

3.4.2 *Results*

Instruction	Req. tactic	Response time [sec] ± SD	Accuracy
cell	cell	24.69 ± 18.32	0.98
cell	digit	33.65 ± 34.53	0.93
digit	cell	43.33 ± 29.70	0.90
digit	digit	8.67 ± 9.02	1.00
Overall		27.08 ± 27.77	0.95

Table 3.4: Response times and accuracy in the four conditions of Experiment 2.

A summary of the mean response time and accuracy per condition can be found in Table 3.4, mean log-transformed response times are graphically displayed in Figure 3.5. The average accuracy in both instruction groups is approximately equal (95% and 96%) and in both groups more mistakes were made on incongruent trials. Only 3 trials of the entire experiment were not completed in time, they are counted as incorrect trials in the analyses. Accuracy in our experiment was close to ceiling and much higher than in the study by Lee et al. (2008), where beginners only had an accuracy of 83% for cb-2 and 67% for db-2. We therefore did not analyze accuracy further.

In the following analyses of response times only correct trials are included. In accordance with our preregistration plan we log-transformed the response time data because response times are always positive, their distribution is skewed, and because longer average response times naturally have more variance. With cell-based instructions the mean log-response times are almost indistinguishable for both required tactics. With digit-based instructions, however,

participants were much faster for puzzles requiring the digit-based tactic and slower for puzzles requiring the cell-based tactic, see Figure 3.5.

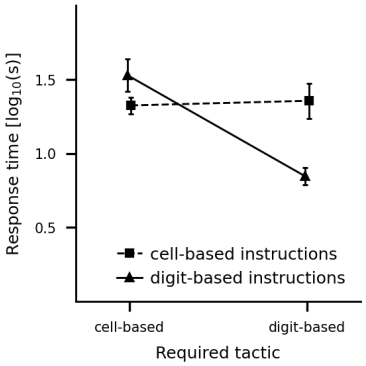


Figure 3.5: Mean values with 95% confidence intervals for $\log_{10}$ response times in the four conditions of Experiment 2. The two required tactics are arranged on the x-axis (factor varied within participants). The two lines display the different instruction groups (factor varied between participants).

An ANOVA with repeated measures on one factor (*required tactic*) revealed main effects of both *required tactic* ($F(1,44) = 101.3$, $p < 0.001$, $\eta^2 = 0.25$) and *instruction* ($F(1,44) = 8.51$, $p = 0.006$, $\eta^2 = 0.06$) as well as a significant interaction ($F(1,44) = 121.7$, $p < 0.001$, $\eta^2 = 0.30$). We repeated the ANOVA with the untransformed response times but with trimmed means, where the highest and lowest response time of each participant was removed. Trimmed means are often used for response time data to make the analysis more robust against outliers. With trimmed means the main effect of required tactic remains significant ($F(1,44) = 15.67$, $p < 0.001$, $\eta^2 = 0.09$) but the main effect of instruction does not ($F(1,44) = 0.89$, $p = 0.35$). There still is an interaction effect ($F(1,44) = 40.57$, $p < 0.001$, $\eta^2 = 0.23$).

Planned post-hoc t-tests (using log-transformed response times again) show that the factor *instruction* had a significant effect on both tactics, congruent conditions being faster than incongruent conditions for both required tactics. Because the variances are not equal, the degrees of freedom are reduced. For the digit-based required tactic the difference is even stronger ($t(32) = 8.17$, $p < 0.001$) than for the cell-based required tactic ($t(33) = 3.45$, $p = 0.002$). Comparing the two required tactics within one instruction group revealed a significant difference only for the digit-based instruction group ($t(34) = 11.55$, $p < 0.001$), but not for the cell-based instruction group ($t(31) = 0.6$, $p = 0.54$).

In the preregistration we specified that we would exclude all participants with an accuracy below 75%. Our rationale was that we did not want to include people in the analysis who guessed often or did not understand the task well. However, unfortunately, the number of people we excluded from the analysis differed between conditions. For the cell-based instructions only two participants were excluded, for the digit-based instructions ten. We repeated the analyses without excluding any participant and got the same results, the effect sizes were slightly smaller but the directions stayed the same and were still significant.

One could be worried that other factors that we did not control in our experiment or did not include in the ANOVA had an influence on the results. For example, we expect that experience has an effect on response times and the ANOVA did not take experience into account. For a within-participant design this would not be a big concern but as the design was a mixed design there might be random fluctuations in experience for the between-participant comparisons. Age also might have a confounding effect on the response times. Additionally, participants might have very different response times for other reasons as well, e.g., conscientious people might double check each step and hence take longer on all trials. Also, we used two levels of NRU (2 and 3) and although all participants worked on the same

Table 3.5: Results of the hierarchical Bayesian linear regression model for Experiments 2 and 3. The column "Estimate" shows the mean posterior values for the parameters, "l-95% CI" and "u-95% CI" are the lower and upper bounds of the 95% credible intervals. The group-level effect indicates how much different participants deviate from the mean intercept. The row "instr-db" indicates how the log response times change when the instruction changes from cell-based to digit-based, similarly for the required tactic in row "req-db". As the minimum number of required units (NRU) was 2, we subtracted 2 from the numerical predictor. The trial index is meant to model learning over the course of the experiment. We dummy-code experience (XP), beginners are the base level, the effect of intermediate or regular experience is shown in the table.

	Experiment 2			Experiment 3		
	l-95% CI	Estimate	u-95% CI	l-95% CI	Estimate	u-95% CI
Population-Level Effects:						
intercept	1.232	1.406	1.575	1.125	1.54	1.954
instr-db	0.137	0.250	0.368	-0.524	-0.424	-0.326
req-db	0.015	0.047	0.079			
instr-db:req-db	-0.818	-0.771	-0.727			
NRU-2	0.030	0.053	0.076	0.055	0.075	0.095
XP-intermediate	-0.206	-0.070	0.069	-0.211	-0.086	0.041
XP-regular	-0.287	-0.143	0.011	-0.275	-0.143	-0.005
trial	-0.006	-0.005	-0.004	-0.019	-0.009	0.000
age	-0.003	0.002	0.006	-0.001	0.002	0.007
Group-Level Effects (participant):						
SD (intercept)	0.138	0.173	0.221	0.136	0.173	0.219
Family Specific Parameters:						
sigma	0.225	0.233	0.242	0.225	0.241	0.258

items, we do expect that NRU has an effect on response times and hence should be included in the analysis. In addition, there might be learning over the course of the experiment and a more careful analysis should take this possibility into account. We therefore fitted a hierarchical Bayesian linear regression model to our data. We included all the mentioned predictors with participants on the group level using the brms package for R (Bürkner, 2017), which is build on the probabilistic programming language Stan (Carpenter et al., 2017). This analysis is the Bayesian analogue of a linear mixed effects model with all the mentioned predictors as fixed effects and participant as an additional random effect. As all puzzles are permutations of the same very small set of carefully designed puzzles we cannot expect the results to generalize to all Sudoku puzzles anyway, hence we did not include the items as a random effect. Experience was entered in the three levels that are indicated in Table 3.3. This analysis was not planned in the preregistration.

Results are shown in Table 3.5. The intercept is found to be at 1.4 and represents the case of cell-based instructions and cell-based required tactic. $10^{1.4} = 25$ seconds fits well with the mean for this condition as reported in Table 3.4. There is considerable variation across participants: The standard deviation around the intercept corresponds to a factor of $10^{-0.173} = 0.67$ or $10^{0.173} = 1.4$. The instruction had a significant effect, participants with digit-based instructions were considerably slower in general than those with cell-based instructions when cell-based tactics were required (instr-db in Table 3.5). The best fitting value 0.25 indicates that participants were $10^{0.25} = 1.78$ times slower in the incongruent condition (44.5 seconds instead of 25 when instruction and required tactic are cell-based, cf. Table 3.4). The effect of required tactic is smaller but also significant. Response times get a little longer when required tactic changes from cell-based to digit-based (req-db in Table 3.5, a factor of $10^{0.047} = 1.1$, i.e., from 25 to 27.5 seconds). Importantly, both main effects (instruction and required tactic) increase response times when switching from cell-based to digit-based reasoning, and hence to incongruent conditions. The main predicted outcome of the experiment, an interaction of required tactic and instruction, is again found as a strong effect (instr-db:req-db in Table 3.5). Compared to the base condition where instruction and required tactic are cell-based, trials where both are digit-based are $10^{0.771-0.25-0.047} = 2.98$ times faster (just 8.4 seconds instead of 25 seconds, cf. Table 3.4). Because of the design of the experiment the predictors of required tactic and instruction together with their interaction fit the raw means in Table 3.4 reasonably well without having to include the remaining predictors of the analysis. All conditions had exactly the same puzzles but if NRU is included in the analysis, it explains some of the variance in response times over trials even though its effect is not very big. Other variables were not balanced as participants were randomly assigned to an instruction group. We found that intermediate experience (XP-intermediate) generally led to faster response times compared to the

beginners, but the Bayesian credible interval includes zero. Regular experience (XP-regular) leads to an even stronger tendency towards shorter response times, but even this credible interval includes zero. The effect of learning looks small (trial in Table 3.5), but the -0.005 means that over the course of the 36 trials participants became on average 1.5 times faster ($10^{0.005*36} = 1.51$). Age has no significant effect on response times and even for the estimated parameter of 0.002 50-year-old participants are only $10^{0.002*(50-20)} = 1.15$ times slower than 20-year-old participants. It is well known that with increasing age cognitive performance, especially in reaction time tasks, deteriorates (Salthouse, 2010). The trend in this direction we see in our data is consistent with the literature. A reason for it not becoming significant might be that we mainly tested younger adults and do not have any participants over the age of 62 in our sample. Overall, the Bayesian analysis confirms the planned ANOVA and the huge interaction effect.

3.4.3 *Discussion*

We found strong evidence for our main hypothesis—an interaction between the required tactic and the instruction. When instruction and required tactic matched, participants were faster than otherwise, especially with the digit-based instructions. Our secondary and much stronger hypothesis—no main effects of instruction or required tactic—was refuted. We found main effects of both, instruction and required tactic.

However, the effect of the required tactic was small in our study. When looking at the two instruction groups individually, the cell-based instructions (that were similar to instructions in other experiments) show a slight trend for faster and more accurate responses for puzzles requiring the cell-based tactic to solve them (see Figure 3.5). However, it is much weaker than effects found in previous studies (Lee et al., 2008; Perret et al., 2011). The other instruction group was digit-based and we know of no previous study with similar instructions. In this group we see a strongly reversed trend: digit-based puzzles were solved much faster and more accurately than cell-based puzzles. Hence, even though the other studies showed a strong effect of the required tactic, it would be a mistake to conclude from these studies that cell-based reasoning is generally faster or easier than digit-based reasoning.

Generally, our participants were quite fast and accurate in all four conditions, showing that they were able to carry out both the digit-based and the cell-based tactic. Even the participants that we excluded due to too many errors were generally very good in the conditions that matched their instructions. Furthermore, our results clearly show that the instructions have a strong influence on the performance in the task, presumably influencing which tactics are tried first. The bias towards the cell-based tactic in previous studies is thus probably at least partially due to the instructions used in those exper-

iments. Our results are, however, not directly comparable to previous experiments, even when restricting the analysis only to the cell-based instruction group. Contrary to other studies where all participants were absolute beginners, most of our participants had at least some prior experience with Sudoku puzzles. We will come back to the influence of experience in the main discussion at the end. In our analysis we excluded participants who had a performance that was worse than 75% correct. This mainly concerned beginners and participants with intermediate experience. And while in the cell-based instruction group only two participants were excluded, ten were excluded for the digit-based instructions. This difference is a little puzzling at first glance. One participant with cell-based instructions made mistakes in about two thirds of the trials and can be excluded because they probably did not understand the task very well. No other participant showed a similarly low accuracy, but some apparently valued fast response times so much that they accepted rather high error rates. They probably entered the first plausible guess, without being sure it was correct. Six participants (five digit-based, one cell-based instructions) fall clearly into this category. They had very short response times throughout the experiment (means between 6 and 14 seconds, whereas the overall mean of all participants was 26 seconds). They were mostly correct on the congruent trials where instruction matched the required tactic, but made many mistakes in the incongruent trials. We think that these participants simply did not bother to think much about the incongruent trials. The remaining five excluded participants had all digit-based instructions. They show an interesting pattern in their response times and accuracies. They made almost no mistakes in congruent trials and were generally below 30 seconds in answering them. On incongruent trials, however, their response times were much longer, well over 60 seconds in most cases. This shows that they did notice that they had to use a more complicated tactic and seriously tried to find the correct answer. They did not succeed very frequently though and answered correctly only about a third of the incongruent trials, which is the rate we expect if people just guess one of the plausible candidates. They struggled with the cell-based trials, the ones that were supposedly easier for beginners according to Lee et al. (2008). With digit-based instructions, this reported ease does not hold.

In summary, the incongruence between instructions and required tactics is the main source of difficulty in our experiment. Some participants were excluded because of the number of mistakes especially in incongruent trials, but also participants that were included in the analysis show longer response times and lower accuracy in incongruent trials. Together these results illustrate, once again, that the instructions of an experiment have a strong influence on the reasoning tactic participants employ.

3.5 *Experiment 3: The effect of NRU*

Due to the slight inconsistencies in the literature regarding the influence of NRU within Sudokus that all require the digit-based solution tactic, we replicated Experiment 2 from Lee et al. (2008), testing the influence of NRU within Sudokus that all require the digit-based solution tactic. We extended their paradigm by having two instruction groups as in Experiment 2. They found an increase in response time with increasing NRU, whereas Perret et al. (2011) found no difference in accuracy between digit-based puzzles of various complexities. Other studies found differences in both accuracy and response times with increasing NRU (Birney et al., 2006; Hearne et al., 2020), but as they did not differentiate between digit- and cell-based tactics, their results are more difficult to interpret.

3.5.1 *Participants, methods, and materials*

We preregistered the experiment and analysis with the Open Science Foundation on 16 June 2020 (`https://osf.io/9z43w/`). The same participants as in Experiment 2 took part in this study. To them it looked like just two more blocks of the same experiment. Instructions stayed the same and the stimuli looked the same. Our cell-based group had instructions that were very similar to the study by Lee et al. (2008). The digit-based group obviously had different instructions than participants in their study (see our previous experiment). Also note that they only studied beginners whereas our sample includes people with more experience. We used only puzzles that required the digit-based tactic and varied the NRU between 2 and 5. For puzzles with 5 required units, there had to be 5 empty cells in the target box, so these puzzles violated some of the properties described above in paragraph 3.4.1. As this study was meant as a replication of Experiment 2 by Lee et al. (2008) and they did not exclude participants based on an a priori criterion, we didn't either and analyzed all 58 participants.

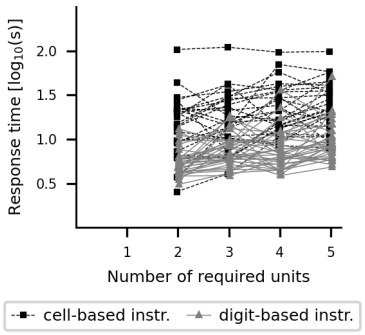


Figure 3.6: The mean log response times in Experiment 4 rise in both instruction conditions with increasing NRU. Each line displays one participant, triangles received digit-based instructions, squares cell-based instructions.

3.5.2 *Results*

Following Lee et al. (2008) we used Page's trend test to test for an effect of NRU on response time. As our experiment had two different kinds of instruction (the cell-based and the digit-based instruction groups), we tested both instruction groups separately. For the cell-based instruction the test statistics were: Page's $L = 660$, $z = 2.4$, $p = 0.02$ and for the digit-based instruction group they were: Page's $L = 918$, $z = 5.6$, $p < 0.001$. The trend of increasing response time with increasing NRU was thus present in both conditions, but stronger in the digit-based instruction group.

In addition, we fit a linear regression to predict the log-transformed response times from the NRU and the instruction type. We found a positive relationship between NRU and response time ($\beta = 0.07$, $p < 0.001$). Hence, if the NRU increases by one, the mean response

time increases by a factor of $10^{.07} = 1.17$. This positive trend can be seen in Figure 3.6. The instruction has a strong influence on the response time, too ($\beta = -0.43$, $p < 0.001$). That is, participants with cell-based instructions are a factor of $10^{0.43} = 2.69$ slower on average than those with digit-based instructions. This regression model explains 38% of the variance of the data (and obviously ignores systematic differences between participants).

We also fit a hierarchical Bayesian linear regression model with all the same predictors as in Experiment 2 (experience, age, order, and a random effect of participant), again using the brms package for R (Bürkner, 2017). This analysis was not planned in the preregistration. As before, we pool participants who indicated they played Sudoku about once per month or more (once per month, once per week, more than once per week) into one group of *regular players*. Results can be seen in Table 3.5 and are very similar to the ones reported for Experiment 2. The size of the influence of instruction looks different from the one found for Experiment 2 but remember that here the required tactic was always digit-based and hence we have to compare the joint effect of the instruction and the interaction in Experiment 2 to the effect of the instruction in Experiment 3: $-0.771 + 0.250 = -0.521$, which is a larger effect of the instruction than observed here with -0.424. Compared to the baseline that were beginners, intermediate experience might lead to moderately shorter response times, but the interval of credible values includes zero (-0.21 to 0.04). Regular experience leads to even shorter response times (credible values -0.28 to -0.02). The best value for the effect of regular experience with Sudoku, -0.143, indicates that those participants were about $10^{0.143} = 1.39$ times faster than beginners. This ratio is roughly consistent with the raw means for response times separated by experience as reported in Table 3.6. Age, again, had no significant effect.

Accuracy does not systematically vary with NRU in our data. In Table 3.6 it can also be seen that accuracy was lower in the incongruent condition, with 86% to 88%, whereas in the congruent condition it was close to ceiling. We expected high accuracies in the preregistration and did not plan to analyze them further.

3.5.3 *Discussion*

We found an increase in response times with NRU but no difference in accuracy. Results are very similar for both instruction groups. Lee et al. (2008) only had cell-based instructions, we added digit-based instructions for a second group of participants. Participants with congruent, digit-based instructions were generally faster but were slowed down by higher NRU. Participants with incongruent, cell-based instructions also showed a marked slow-down with increasing NRU. This experiment confirmed both results from the literature which compared the influence of accuracy within just one tactic. We found slower response times with increasing NRU, just as Lee et al.

NRU		2	3	4	5
cb instructions					
Accuracy					
	All, N=25	.86	.88	.88	.86
	Beginners, N=8	.75	.81	.81	.67
	Intermediate, N=8	.94	.88	.88	1
	Regular, N=9	.89	.94	.94	.89
Time [sec]					
	All, N=25	22.0	25.5	31.0	36.0
	Beginners, N=8	30.0	34.1	36.0	41.6
	Intermediate, N=8	22.9	17.1	25.4	28.1
	Regular, N=9	14.0	25.2	29.6	32.3
db instructions					
Accuracy					
	All, N=33	1	.97	.98	1
	Beginners, N=9	1	1	.94	1
	Intermediate, N=14	1	.96	1	1
	Regular, N=10	1	.95	1	1
Time [sec]					
	All, N=33	6.3	8.7	9.9	12.6
	Beginners, N=9	7.5	7.7	14.6	13.0
	Intermediate, N=14	5.6	10.6	8.2	14.5
	Regular, N=10	6.0	6.8	8.0	9.5

Table 3.6: Mean accuracy and response time in the four conditions of Experiment 3. Data are split according to prior experience with Sudokus. Beginners: "less than once per year", Intermediate: "Once per year", Regular: "Once per month" or more

(2008) did. At the same time we found no difference in accuracy for different levels of NRU within digit-based stimuli, just as Perret et al. (2011) reported before. The two measures—response time and accuracy—are often used as both reflecting the difficulty of the task. Our results suggest, however, that the two measures do not reflect the same aspects of the task.

We mentioned in the introduction that we avoid the term relational complexity in this paper, even though this is the term used in other papers on Sudoku and the LST (Birney et al., 2006; Lee et al., 2008; Perret et al., 2011). The reason being: effective relational complexity, i.e., number of variables to *consider simultaneously*, does not necessarily increase with an increase in the number of units that need to be considered. In Sudoku it is possible to carry out a sequential process that takes more steps when more units have to be considered, but does not get more difficult, i.e., does not require to consider more information *at the same time* and thus does not lead to more errors. We believe that this is the case for the digit-based tactic.

Overall our participants were much faster and more accurate than participants in previous studies. However, when comparing only to participants with no regular Sudoku playing experience and the same instructions as in other experiments, this difference becomes much smaller. The most comparable study is the second study of Lee et al. (2008) that had college students as participants who never be-

fore had filled a Sudoku. Their participants needed two to three minutes on average to respond in this task (cell-based instruction, digit-based with 2 to 5 required units), our beginners of the same condition needed just 30 to 42 seconds on average (see response times for beginners with cb instructions in Table 3.6). The accuracy in the experiment by Lee et al. (2008) was between 50% and 67%. Our beginners with similar instructions were slightly better with 67% to 81% correct (see accuracy for beginners with cb instructions in Table 3.6). One has to keep in mind, however, that our participants had practice with this task from the 36 previous trials of the other experiment (our Experiment 2).

3.6 *A simple process model*

To understand the patterns of response times in the previous experiments better, we designed a simple process model that can carry out both the cell-based and the digit-based tactic and fill a single cell of a given Sudoku. A basic operation required for both tactics is to search a digit in a unit. We call this action a *scan*. A scan is the process of checking a given row, column or box for one specific digit, for example looking whether the 1 occurs in the third row of a Sudoku puzzle. Based on theoretical considerations and insights from the think-aloud protocols of the first experiments, we propose the following algorithms for the two tactics. We do not think that participants follow these procedures to the letter, but they provide a good abstraction of many of the reasoning patterns we saw in the think-aloud protocols of Experiment 1 and explain the patterns of response times we saw in Experiments 2 and 3.

Cell-based instructions and cell-based tactic required With cell-based instructions in a Sudoku that can be solved with cell-based tactics the process of solving a puzzle might take place as follows: The participants need to check each digit, they usually do that in the normal counting order from 1 to 9, as we learned from the think-aloud protocols of Experiment 1. For each digit they have to find out whether it is already placed in the peers of the cell: the same row, same column, or same box. The exact order in which the three units are searched leaves room for individual differences and optimization of the process. Here, we just assume the order is random. As soon as the digit is found in the peers, the search for it can be terminated and the next digit can be processed. If the digit can indeed be excluded through one of the units, on average it will be found after 2 units have been searched (the mean of 1, 2 or 3), i.e., two *scans*. If the digit does not exist in the peers, it is a potential candidate for the cell. One needs to search all 3 units in order to be sure it does not exist. Because the participants cannot be sure that the found candidate is the only candidate digit, they will still have to check all the remaining digits in the same way. If only one digit was *not* found in the peers of the cell, it can safely be filled into the cell. Code 3.1 contains a summary

of these steps in pseudo-code.

Code 3.1: Pseudocode for the cell-based tactic.

```
// Due to the random order in which the 3 units are searched
// 'scans' returns a diff. number each time the code is executed.
// For all 9 digits 1-3 units have to be searched, so 'scans'
// could be any number between 10 and 27.
candidates = set()
scans = 0
for digit in [1..9]:
    digit_found = False
    for unit in shuffle [row, column, box]:
        scans++
        if digit in unit:
            digit_found = True
            break // no need to continue to look at other units
    if not digit_found:
        candidates.add(digit)
if size(candidates) == 1:
    // really only one possible candidate found
    assign_digit_to_cell(candidates.pop(), given_cell)
```

Digit-based instructions and digit-based tactic required When the instructions favor digit-based reasoning, a entire box of the Sudoku is highlighted and the question is "where in this box does the digit X go?". In order to answer, participants need to consider all empty cells in the highlighted box and test whether they can exclude some of them as potential locations for the digit (because the digit appears already in the same row or column). Again, we assume that the order in which they scan row and column is random. If there is only one cell for which the digit in question does not occur in the peers, this is the unambiguous answer. Code 3.2 contains a summary of these steps in pseudo-code.

Code 3.2: Pseudocode for digit-based tactic.

```
// In our experiment all boxes had 4 empty cells.
// Due to the random order in which the 2 units row and column
// are searched, 'scans' returns a different number each time
// this code is executed.
// Empty cells that are in the same row can be eliminated
// in one go, if the digit in question appears in that row
// (same holds for columns).
// For the empty cells 1-2 units have to be searched, so 'scans'
// could be any number between 3 and 8.
scans = 0
candidate_cells = set()
already_eliminated = list()
for cell in empty_cells:
    if not cell in already_eliminated:
        is_candidate = True
        for unit in shuffle [row, column]:
            scans ++
            if digit in unit:
                is_candidate = False
```

```
                // additionally prevent all empty cells
                // in the same unit from further checks
                already_eliminated.append(same_unit(cell, unit))
                break
        if is_candidate:
            candidate_cells.add(cell)
if size(candidate_cells) == 1:
    // only one cell possible
    assign_digit_to_cell(digit, candidate_cells.pop())
```

Mismatch between instructed and required tactic If the instructions do not match the required tactic, the candidate sets found by the algorithms above will not hold a single element but several candidates. By design of our stimuli in Experiment 2, the candidate set will always be of size three, but this is not true for Sudokus in general. If the participants started with the cell-based instructions, they will be left with three possible digits to assign to the cell. In order to find out which one is the correct one, they need to switch to the digit-based tactic and test each of the candidate digits this way. If they are lucky, the first digit they try is the correct one, but it might be necessary to test all three. An example for exactly such a search process can be seen in the protocol example from Experiment 1 in Figure 3.2. This adds one to three times the digit-based search time to the already spent cell-based search time.

Participants with digit-based instructions and a Sudoku that requires cell-based reasoning will be left with 3 candidate cells after they applied the digit-based algorithm. For each of these cells they need to test whether the digit in question is the *only* digit that is allowed there. Hence, in addition to the initially spent search time with the digit-based algorithm the participant has to run the cell-based search 1–3 times.

Simulation results We implemented the algorithms described above as a model for the task and ran it on our experimental design. In Table 3.7 the results of 1000 simulated "participants" are compared to the actual data we recorded in the experiment with human participants from Experiment 2 (means and standard deviations are given). Note that the two measures are not really comparable (steps vs. scans). It is merely a coincidence that the magnitudes are roughly in the same range. The statistics of the model are solely based on the scans carried out, i.e., reflect the number of units that were considered before finding the answer. This model has no parameters to fit, it just executes the assumed processes of the search. Variation in the results comes from the random order in which units are tested and in which the remaining candidates are tested with the other tactic, in cases where instructions did not match the required tactics. Both the mean and the standard deviation of the model simulations are lower than the human data in all conditions. But the relative order of the conditions are well matched.

Instruction	Required tactic	Model [scans]	Human Data [sec.]
digit	digit	4.28 ± 0.96	8.67 ± 9.02
digit	cell	38.40 ± 13.03	43.33 ± 29.70
cell	digit	27.18 ± 5.82	33.65 ± 34.53
cell	cell	15.15 ± 1.69	24.69 ± 18.32

Table 3.7: Model simulations compared to the data from Experiment 2.

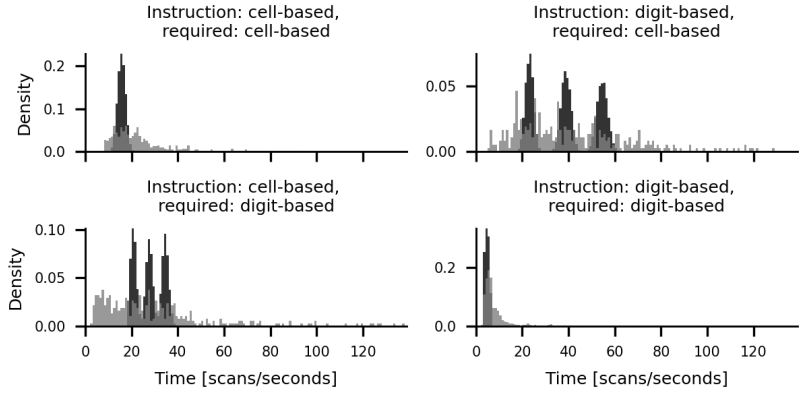


Figure 3.7: Comparison of the model simulations with the data from Experiment 2, showing the effect of required tactic and instruction. The histograms of the model predictions are shown in dark gray, the data from Experiment 2 are in light gray. The 3 peaks in the incongruent conditions of the model reflect the 3 remaining candidates (cells or digits) that can be tested in any order.

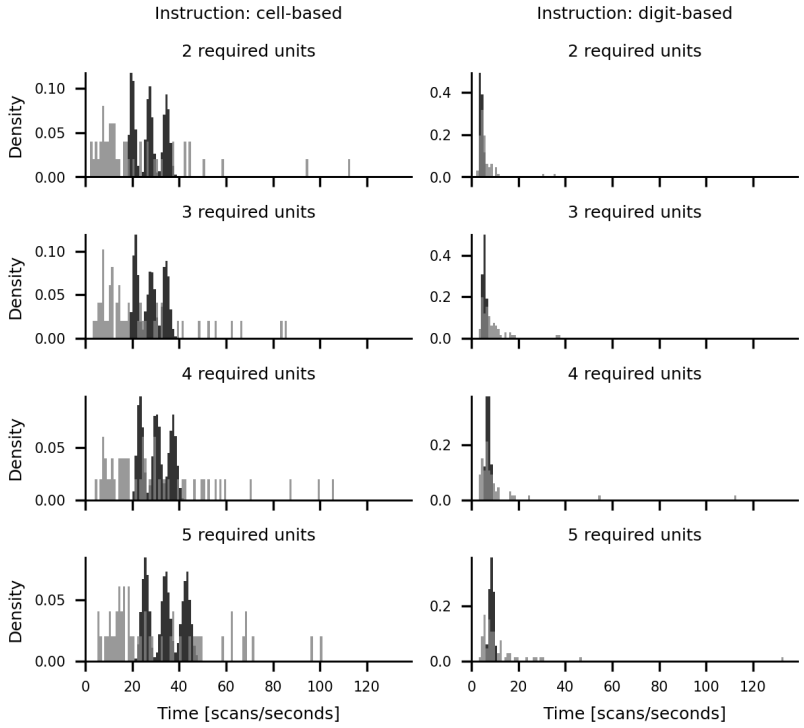


Figure 3.8: Comparison of the model predictions with the data from Experiment 3, showing the effect of number of required units. The histograms of the model predictions are shown in dark gray, the data from Experiment 3 are lighter. The 3 peaks in the incongruent conditions of the model reflect the 3 remaining candidate digits that can be tested in any order.

Figure 3.7 shows the number of scans predicted by the model for each condition of Experiment 2. The fastest and least variable case is the congruent digit-based condition with 3 to 6 scans. The congruent cell-based condition is the next fastest with 11–20 scans. The two incongruent conditions show 3 distinct peaks. The required scans in

these conditions are the sum of first an unsuccessful pass through the instructed tactic, followed by 1–3 attempts with the uninstructed one. For example, if the instructions were digit-based, the digit-based tactic will be tried first (7–8 scans) and then the cell-based tactic is applied to all 3 candidate cells in turn until the answer is found. If, by chance, the first cell that is tried is the correct one, scan numbers are in the first peak, and the last peak represents cases where only the third attempt was successful. For better comparison we have overlaid the human data from Experiment 2. But note that the x-axis for the simulation data and the human data are not comparable (one is number of scans, the other seconds). Figure 3.8 in the appendix shows the same plot for Experiment 3. For both experiments, the model makes clear why the congruent conditions are faster than the incongruent conditions and why the incongruent conditions have a much higher variance. In addition, as we assume that a basic scan takes the same time in the digit-based and cell-based tactics, the model also explains why the digit-based tactic is faster than the cell-based tactic. Note that the model does so purely based on a careful analysis of the puzzles and the two tactics.

3.6.1 *Fitting the model quantitatively*

It is possible to transform the counts of scans into predictions of response times by assuming that each scan takes some variable amount of time. We model the time for a scan as a draw from a Gamma distribution and fit the parameters of this distribution to match the empirical distributions of response times. Additionally we gave the model more flexibility by introducing the possibility to start with the tactic that did not fit the instruction. We saw in the think-aloud data of Experiment 1 that some participants started with digit-based tactics right away, even though instructions were favoring cell-based reasoning. While we used the same parameters for the Gamma distribution of scan duration for all conditions of both Experiment 2 and 3, we allowed a different parameter value for the probability to start with the instructed tactic for each instruction group and experiment. Allowing for this additional variability improved the fits significantly.

There are many potential sources of variation that contribute to the response times in our experiments. Some participants are more proficient or more motivated and thus faster and some trials might take longer because a participant was distracted. In order to fit the model to the response time data we therefore assume that the duration of each scan is random. Allowing the scan duration to vary increases the variability of possible predicted response times and should allow us to fit the model to the histograms from Figure 3.7 and Figure 3.8. It should also give us a reasonable estimate for how long a single scan takes on average. For simplicity, in the following analysis we pool the response time data from all participants (as in the histograms). We model the duration of a scan by a draw from a

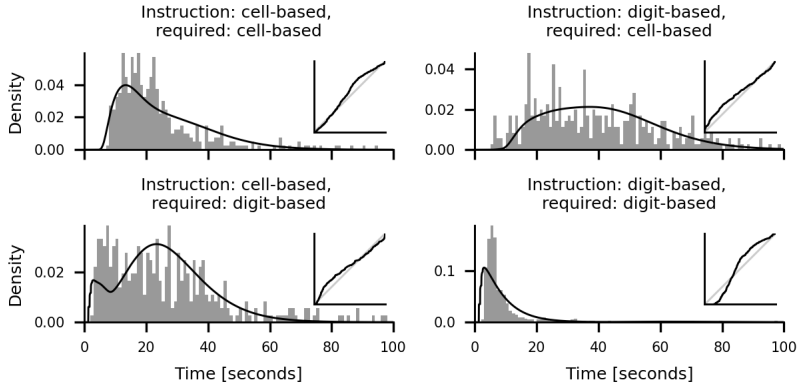


Figure 3.9: Histograms show response times from experiment 2, the lines the fitted model distributions. In the corner of each condition a pp-plot shows the same data. The parameters for the gamma distribution are the same in all four conditions and both experiments. The probability for choosing the instructed tactic is fit individually for each instruction group. They are 0.7 for the cell-based instruction group and 0.96 for the digit-based instruction group.

gamma distribution. The gamma distribution is defined on the positive reals, as is appropriate for scan times. A useful property of the gamma distribution that makes the analysis easier is that the sum of gamma variates with the same scale parameter is again gamma-distributed with the same scale parameter. In addition to each scan step, we assume there is a "zeroth step" before the participants even start with one of the tactics. This initial step models the additional time needed for orientation. We model this orientation time to be gamma-distributed with the same scale parameter.

We expect that participants don't always follow our instructions. We saw in the protocols from Experiment 1 that some participants are not always biased by cell-based instructions but start with digit-based tactics anyway. We therefore assume they only start with the tactic that matches their instructions with a certain probability. For example, if the instructions are cell-based, using digit-based tactics instead leads to a speed up in those cases where digit-based tactics are indeed required. But, of course, if cell-based tactics are required, it will lead to very long searches.

In summary, we used 8 parameters to fit the model to all 12 conditions in the two experiments simultaneously: 3 parameters for the gamma distribution for the duration of each scan (shape, scale, and off-set as the minimum scan time is greater than zero), 1 parameter for the shape of the gamma distribution for the initial orientation time, and 1 parameter for each instruction group of the two experiments that models the probability of really starting with the instructed tactic rather than the other one. The simulated numbers of unit scans are transformed to mixtures of gamma distributions with the parameters described above. Each number of scans forms the basis of one gamma distribution that contributes to the mixture proportional to the fraction of simulations with this number of scans. The simulated relative number of required scans per condition thus give the probability of needing each number of scans. In the congruent digit-based condition of Experiment 3, for example, the number of scans ranges only from 4 to 7, the most probable number being 5. We transform each integer number of scans to a gamma distribution,

i.e. the sum of the gamma distributions for each single scan (remember that the sum of gamma variates of the same scale are gamma variates). This gamma distribution represents the spread of response times we expect for this number of scans and its variance naturally increases with the number of scans because it is the sum of the single scans. The probability for each number of scans is simulated just once and stays fixed in our fitting procedure but the parameters for these gamma distributions are adapted to fit the actual response times from our experiment. To fit the probability of starting with each tactic, we additionally sampled the scans from our model that would result from starting with the tactic that did not fit the instructions. We evaluate the mixture distribution at all data points up to 100 seconds. Response times above this value likely include restarts and recovery from errors which we cannot reproduce well with our model. We minimized the negative log likelihood of the truncated data of the experiments given the model distributions using a standard package for optimization (Virtanen et al., 2020, scipy.optimize with SLSQP).

3.6.2 *Results of quantitative fit*

The fits of the model to the data can be seen in Figure 3.9 and Figure 3.10. The pp-plots in the corner of each figure plot the two cumulative distributions against each other, model on the x-axis, data on the y-axis. The parameter values that fit the data best are as follows. The offset is at 0.42 seconds, no scan duration is shorter than this. Mean and variance of the scan durations are 0.96 and 3.25 respectively (the shape parameter is smaller than 1). The distribution is very long tailed, skew and kurtosis are 6.53 and 64.14, making some scans last several seconds. What really happens is probably that the participants sometimes forget their progress, mind-wander or make mistakes and restart the entire process. But as we cannot capture any of these processes, the model incorporates the occasional long times with very long-tailed distributions for the scan durations. The best fitting distribution of initial orientation time has a mean of 4.9 seconds. For Experiment 2 the participants in both instruction groups mostly seem to try the tactic congruent with the instructions first. The best fitting probability to start with the tactic suggested by the instruction is 0.96 for the digit-based instruction group and 0.7 for the cell-based instruction group. For Experiment 3 the best fitting probability following the instruction is high for the congruent, digit-based instruction group ($p = 0.99$), but very low for the incongruent, cell-based instruction group ($p = 0.29$). This suggests that not all participants are completely biased by the instructions. Some start directly with the required tactic, despite instructions favoring the other one. In Experiment 3 only puzzles requiring digit-based reasoning were shown and many participants seem to have noticed this and adjusted their solution tactics accordingly. In the first experiment the think-aloud protocols allow us to follow the reasoning steps of the

participants in detail instead of just inferring them based on the response times. There, we already saw that several participants indeed were not always biased to start with cell-based tactics, even though the instructions favored them.

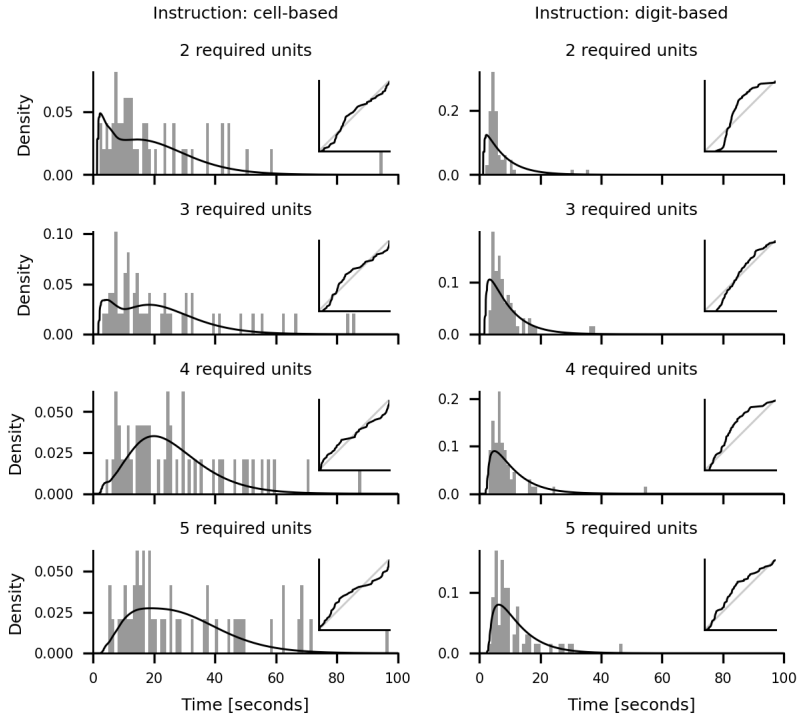


Figure 3.10: Histograms show response times from Experiment 3, the lines the fitted model distributions. In the corner of each condition a pp-plot shows the same data. The parameters for the gamma distribution are the same in all four conditions and both experiments. The probability for choosing the instructed tactic is fit individually for each instruction group. For the digit-based instruction group the parameter is high (0.99), for the cell-based instruction group the best fitting parameter is 0.29, indicating that most of the trials were directly started with the required digit-based tactics.

3.6.3 *Discussion*

Even without any free parameters our model provides a good explanation of the differences in response time in the different conditions of Experiments 2 and 3. It shows that for congruent digit-based conditions far fewer scans need to be carried out than for the congruent cell-based conditions, explaining the main effect of required tactic and why the cell-based instruction group was less influenced by the required-tactic factor than the digit-based instruction group in Experiment 2. Although we designed the model mainly to understand the differences in cell- and digit-based reasoning better, it also trivially explains the longer response times for increasing number of required units in digit-based tactics: higher NRU require more scans and more scans in turn lead to longer response times.

Introducing free parameters to the model—to account for variable scan durations and a certain probability to start with the tactic that was not instructed—allows us to quantitatively fit the model to response time data. The best values for the probability to start with the tactic suggested by the instructions found for Experiment 3 are 0.99 for digit-based instructions (always congruent) and 0.29 for cell-based instructions (always incongruent). This strongly suggests that many participants of the cell-based instruction group realized that

they needed digit-based tactics in this experiment and didn't even bother to first try cell-based tactics. Experiment 2 was constructed such that both tactics were required equally often and there were no obvious surface properties that were informative about the required tactic. Hence in this experiment, it is reasonable that the participants generally followed the instructions. According to our model they did so more in the digit-based instruction group (p=0.96) than in the cell-based group (p=0.7). The first peak in model density in the lower left plot of Figure 3.9 reflects starting with digit-based tactics for the cell-based instruction group. More trials were faster than the model predicts which is also reflected in the early deviation from the diagonal in the pp-plot. The model cannot increase the probability for digit-based tactics further to improve the fit because this would at the same time lead to more very slow responses predicted in the congruent condition of this instruction group. The participants of our experiment were thus cleverer than our model, managing to improve response times in the incongruent condition without hurting performance very much in the congruent conditions.

Our model is obviously still too simplistic to capture all relevant cognitive processes. Traditionally, cognitive architectures, like ACT-R, are used to implement more complete models. In such a model, the time required to give the answer via clicking and typing, for example, would be included for free, i.e., without needing to fit extra parameters, because they are already set in the cognitive architecture. There have been some attempts to model (aspects of) solving Sudokus within ACT-R (Qin et al., 2012; Preuß, 2018). While these models are closer to the actual cognitive processes (with built-in constraints on the speed of certain processes and plausible interactions of various components), these models are also very complex and it is difficult to isolate the key factors influencing the performance. Our model, on the other hand, though too simplified to capture the full details of the process, allows us to use simple and interpretable parameters such as the probability of starting the task according to the instructions. It was created with the aim to understand the faster response times for the digit-based congruent condition as compared to the cell-based congruent condition. This, at first, was a surprising result to us because the literature suggested that cell-based reasoning is easier for participants, and faster (Lee et al., 2008; Perret et al., 2011). We now understand that the reason other studies found that cell-based reasoning is faster than digit-based reasoning is simply an artefact of the cell-based instructions that earlier studies gave and our data and our model show that digit-based reasoning is actually faster given the right starting point. Hence, even though our model describes the tactics at a very abstract level—without recourse to the cognitive architecture—it captures a good portion of what is going on when people solve these simple Sudoku tasks.

3.7 *General discussion*

Overall we observed that our participants were remarkably flexible in applying different tactics to solve Sudoku puzzles. The think-aloud study provided evidence for personal preferences for certain tactics. When the preferred tactic could not be applied, participants were flexible to choose another tactic and approach the problem differently. All of our participants in Experiment 1 were able to use two tactics: cell-based and digit-based. In Experiments 2 and 3 almost all participants successfully solved puzzles that required either kind of tactic. We did exclude some participants who had relatively high error rates on incongruent trials, but the majority had no problems with either tactic. The people that were excluded were mainly in the digit-based instruction group and had trouble carrying out cell-based trials correctly. The cell-based tactic was previously thought of as easier to carry out for beginners. Our experiments showed that this is most likely just an artefact of the way instructions were given.

There are two important variables that have an influence on the difficulty of a trial in a Sudoku or Latin square task (LST): the number of required units (NRU) and the tactic required for solving the task. Most of the research on the LST did not distinguish between cell-based and digit-based tactics, and classified complexity based on the number of variables needed to consider simultaneously for solving the puzzle, which is not the same as our NRU measure. Since they give examples for their stimuli, it is possible to re-classify the stimuli according to our scheme and we can discuss previous results from the literature in the light of our model.

Number of required units and relational complexity Studies that tested the influence of NRU within just one tactic—cell-based or digit-based—generally found an effect on response times but not error rates. Qin et al. (2012) studied only cell-based reasoning in small 4-by-4 Sudokus. They found an effect of NRU on response times, but no significant difference on accuracy (cb-1 vs. cb-2 and cb-3). Within only digit-based trials in 9-by-9 Sudokus Lee et al. (2008) found that increasing the NRU led to significantly longer response times, but the trend for lower accuracy was not statistically significant (db-2 to db-5). Perret et al. (2011) only measured accuracy in a LST and found no difference within digit-based conditions of different NRU (db-2 to db-4). In Experiment 3 we find the same pattern, response times increased with the NRU but accuracy stayed the same. However, based on the pioneering work of Birney et al. (2006) one would expect that response time increases and accuracy decreases with the NRU because it measures *relational complexity* (Halford et al., 1998). A higher *relational complexity* would imply that more variables need to be considered *in parallel*, leading to higher demands of working memory which in turn would result in lower accuracy. The fact that there is no drop in accuracy with increasing NRU can be explained by our model (section 3.6). The model needs more scans if the NRU

increases and more units are involved in a deduction. Obviously, more scans imply longer response times. However, the search for all relevant digits for the deduction is a *serial* process in our model, and therefore requirements for working memory are the same for all levels of NRU. Relational complexity is actually constant in all these cases because the required variables do not need to be considered in *parallel*. Hence, the number of units required for the deduction measures some form of difficulty in Sudoku, but not relational complexity. This distinction is worth keeping in mind for future research. Whether one can really talk about relational complexity in a task depends on the possible strategies for solving it. An explicit process model, like ours, is helpful for making this distinction.

Studies that found not only an increased response time but also lower accuracy for higher complexities generally did not distinguish between cell-based and digit-based puzzles (Birney et al., 2006; Hearne et al., 2020). As their highest-complexity items are all digit-based and their lowest-complexity items all cell-based, it is not clear how much of the reduced accuracy can actually be attributed to the complexity. It might well be that it is rather caused by the switch in required tactics, which would be consistent with the other studies and our model.

Interaction of required tactic and instructions We found that which tactic is more difficult to use depends strongly on the information given in the instructions. When a cell to be filled is highlighted and the instruction is "Please fill in the highlighted cell", cell-based tactics are generally easier for participants, as expressed through higher accuracy and faster response times. Under different conditions, however, digit-based tactics can become much easier and faster to carry out, namely when the digit in question is given, and the location for it needs to be determined. With different instructions and thus different given starting points for reasoning, otherwise identical tasks change massively in difficulty (as expressed by the strong interaction effect we saw in Experiment 2). Trials in which instructions and required tactic matched were solved faster than incongruent trials. In fact, the fastest condition was congruent digit-based.

The model and analysis we presented in section 3.6 gives plausible reasons for the differences between the four conditions of Experiment 2. It showed how the two tactics—digit-based and cell-based—differ (see Figure 3.7). The difference is simply that excluding a digit in the cell-based tactic always requires at least one full scan whereas excluding a cell in the digit-based tactic will only require a scan if a cell is empty. Hence, starting the reasoning process with a cell to fill is very different from starting the process with a digit to place. More concretely, just 3 to 6 scans are required in order to exclude the 3 open cells in our puzzles as possible candidate locations. In contrast, the smallest number of required scans for the cell-based tactic is always 10 because each digit requires at least one scan. The average number of required scans for congruent cell-based stimuli is about

15. In incongruent cases people start out with less information and might need to try several candidates (digits or cells). This makes response times more variable and, on average, longer. But even several runs of the short digit-based search can be shorter than the more tedious cell-based search. This explains why there was only a small difference within the cell-based instruction group between cell-based and digit-based puzzles. For the digit-based instruction group this is reversed, congruent cases are very fast and easily solved, but incongruent cell-based trials require much extra effort and time.

Required tactic with cell-based instructions Previous studies only used cell-based instructions. Not all of them differentiated between digit-based and cell-based required tactics. When they did, they found the cell-based tactic to be easier for participants, as expressed through shorter response time and higher accuracy. Perret et al. (2011) found a big difference between cell-based and digit-based conditions. Cell-based tasks (cb-2) were solved with much higher accuracy (80%) than digit-based ones (≈50% for db-2 to db-4) by children between 8 and 11 years of age. Lee et al. (2008) found much shorter response times and higher accuracies for cb-2 as compared to db-2 trials in a population of university students. Our model predicts this effect, too. As the instructions contain more task relevant information for the cell-based tactic, fewer scans are required to reach the solution, which implies shorter response times. In Experiment 1 we found a trend towards shorter response times in cell-based trials as compared to digit-based trials, but it was very slight and not statistically significant. The same is the case for our second experiment, when looking only at the cell-based instruction group.

The role of experience Some of the differences between our results and what is reported in the literature can be explained by differences in experience. Other studies only looked at the performance of absolute beginners and generally found that cell-based tactics were easier for them. We think that some of this effect can be attributed to the task instructions that favored cell-based reasoning, as explained above. However, also in a free-filling paradigm (where we can't blame the instructions) Lee et al. (2008) found that beginners were more likely to use cell-based tactics. And also in our own free-filling study reported above in section 2.5 we saw the same pattern: Beginners used more cell-based tactics but the proportion of cell-based tactics quickly decreased with prior experience of the participants. Taking all findings together, the advantage for cell-based tactics in experiments with beginners suggests that beginners seem to understand and employ cell-based tactics more easily than digit-based tactics. As we saw in our model, digit-based tactics are, however, much more effective in reaching conclusions and finding specific cell-value assignments. Given that people are quite adept at learning to apply the most efficient strategy (Gunzelmann and Anderson, 2003; Lee and Johnson-Laird, 2013; Rieskamp and Otto, 2006), it seems likely

that with increasing experience Sudoku players prefer to use digit-based tactics more and more.

Overcoming instructions For a good fit of the cell-based instruction group with our model, we needed the additional possibility to start with the uninstructed tactic. The best value for the probability to do this, was around 30% in Experiment 2. In Experiment 3 it becomes even more obvious that many of our participants overcame the instructions. Our model fit suggests that around 70% of the trials of the cell-based instruction group of Experiment 3 were started directly with the uninstructed but required digit-based tactics. Each participant completed 8 trials in this experiment that all required digit-based tactics. Many participants seem to have noticed this task requirement and adapted to it by skipping the uninformative cell-based approach altogether. They quickly learned to ignore the experimenters' misleading instructions, showing once again that researchers should not assume that participants behave as they expect them to.

3.7.1 Conclusion and outlook

Our process model clarifies the role of task requirement, task instruction, number of required units, and how these factors interact to determine the difficulty participants have with a puzzle. Much of the prior research on Sudoku can be understood better in terms of our model. Importantly, we could develop our model only thanks to our mixed-methods approach, which included an exploratory think-aloud study. The think-aloud protocols gave us a very detailed understanding of how participants approach a Sudoku task. In particular, we saw how participants switched to digit-based tactics when they could not find the correct answer via cell-based tactics (for an example see Figure 3.2). This explicit switching of tactics has not previously been reported in the literature on Sudoku or the LST. This insight is at the heart of our model and the two response time tasks that were used to evaluate the model.

More generally, problem solving is a complex human activity that we have not yet fully understood, and there is currently no consensus on what are the best methods to study it (Batchelder and Alexander, 2012; Jäkel and Schreiber, 2013; Ohlsson, 2012). Here, we used think-aloud data to inform a relatively coarse probabilistic process model (coarse compared to the level of detail in, say, ACT-R) and fitted the model to fine-grained response data from experiments that varied relevant task parameters. Similarly, Lee et al. (2019) provide another good example of how to use different sources of data (decisions, mouse clicks, think-aloud protocols) to inform a model and make more accurate inferences about the strategies participants used in each trial of an experiment. We believe that research on problem solving would benefit from using this kind of combined qualitative and quantitative modeling more systematically. Sudoku provides us

with a convenient test-bed for developing this approach further.

Our results, once more, showed that we should appreciate that there are generally many different strategies for solving a given problem. We should therefore embrace and study participants' flexibility to make progress on the big open questions in problem solving research. The two instruction types we employed in our studies both restrict the participants in their responses and most likely lead to strategies that are quite different from "natural" Sudoku filling. The advantage of such restrictions is better experimental control. However, if we are genuinely interested in participants' flexibility of choosing between different tactics, constraining participants less will probably lead to more interesting observations. In a paradigm where participants freely fill a Sudoku puzzle, there will be many more opportunities for strategy and tactic selection than in the restricted tasks that we used here. While, in theory, it should be possible to obtain a big data set from people solving complete Sudokus online (see e.g., Pelánek (2011)), these data will usually not be enough to infer the tactic that was used in each step. There is only one careful lab study we know of that employed a free-filling paradigm and also studied participants' tactics: Lee et al. (2008) asked their participants after each cell they filled to justify how they knew the entry was correct. However, this method does not provide any information as to how the participants decided on what tactic to use, which cell to attack, or which digit to try and fill into a unit. In future work, we will address the important question of how tactics are chosen more directly by studying how people solve complete Sudokus (see chapter 5).

Chapter 4

Hierarchical Bayesian model: EIP regression

This chapter has been accepted as *Behrens, T., Kühn, A., and Jäkel, F. (2024). Connecting process models to response times through Bayesian hierarchical regression analysis. Behavior Research Methods*. Some terminology is adapted to fit better with the other chapters and references to the other chapters are inserted where appropriate.

Models of mental processes have a long tradition in cognitive science and psychology. They specify a series of (mental) operations necessary to complete a task and can make predictions about the difficulty and successes of different strategies under varying conditions. The amount of mental processing required to complete a task is assumed to be reflected in the time it takes to do so. Response times are easy to measure in psychological experiments and, if analyzed with the right tools, can thus be very informative about the underlying mental processes. For a psychological model it is desirable to have the possibility of fitting it to the data of individual participants. It can then make more precise statements about the fit and the variability between participants. Here, we propose a Bayesian hierarchical regression analysis to link classical process models to response time data of individual participants.

In human computer interaction detailed process models of specific tasks have a long history (John and Newell, 1989). For example, GOMS models (goals, operators, methods, selection rules) are used to analyze the complexity of a task for a specific user interface. They can make predictions about the time required to complete a task as well as the working memory load for the user (Estes, 2021). They do so by using an inventory of basic building blocks, simple atomic actions, for which duration and memory load have been measured carefully. When building a model for a new task, these building blocks can be combined to get an estimate of the overall duration. Although in this approach process models predict response times directly, they are not adequate as psychological models as they make predictions about a standard expert user only and cannot be used to fit data and find out about a specific person's abilities, for example. Similar caveats hold for cognitive architectures (like Soar (Laird, 2012), ACT-R (Anderson, 2007) or Clarion (Sun, 2016)).

The modeling approach we will introduce here, however, builds upon simpler and more abstract process models. Process models that can count elementary information processing (EIP) steps for single experimental trials. Such process models have, for example, been used extensively in the decision-making literature (Bettman et al., 1990; Payne and Bettman, 2004; Payne et al., 1988). The duration of a single elementary information process could be estimated by simple analysis tools, such as linear regression. The analysis we advocate for builds on this idea but improves it in several important ways. First, we make the duration of a single EIP step inherently probabilistic, turning the model into a more plausible cognitive model. For simplicity, however, we will focus solely on serial processing models where each EIP is identically distributed for each participant and there is across-stage independence (Townsend and Ashby, 1983). Second, we use gamma distributions instead of normal distributions to model the duration of an EIP step. Gamma distributions are more adequate in many psychological models, as already advocated by Maris (1993). Third, we use a hierarchical Bayesian framework, allowing to jointly fit the model to the data of individual participants. And lastly, we add some extensions to the model such that it can also deal with situations where the exact EIP step count is latent or there are several strategies with associated process models and EIP step counts. We call the resulting analysis (hierarchical) EIP regression.

Examples where this analysis can be applied include children's addition, which can often be well predicted by the smaller of the two addends (Groen and Parkman, 1972), mental rotation, where response times increase linearly with angular difference between the two stimuli (Shepard and Metzler, 1971), and visual search in a feature-conjunction display (Treisman, 1982). In these cases, the regression models are not just mere statistical tools to analyze the data, they can be interpreted as simple cognitive models. The slope of the regression tells us something about the processing speed of some cognitive component. In the case of simple addition, the elementary information process is counting up a number. A regression model then tells us something about the speed of mental counting. When our model predicts half a second increase in response time for each increase in the smaller addend, that means that children need about half a second to carry out one EIP step, i.e. counting up one number. In the experiment where two depictions of 3D objects had to be compared, the response time increased linearly with the angular difference between the two depictions. The fact that the data could be fit by linear regression, with the angular difference as a predictor of response time, tells us that the internal process seems to involve mentally rotating one of the objects to match the image of the other. We also learn that this mental rotation seems to be done at approximately constant speed which we can read off of the regression slope.

In this paper we demonstrate how this big class of process models can be fit to response time data. In all cases where the cognitive

process can be expressed in counts of EIP steps, they can be fitted with the model we present here. The serial processing model that we assume may, of course, be wrong. EIP steps might not follow a gamma distribution. Even if they do, they might not follow the same distribution within each participant. This could be because there are really several different elementary processes instead of just one. Or the speed of processing at one stage might depend on the speed of processing at other (earlier) stages and, therefore, across-stage independence does not hold. In general, the processing might also happen (partially) in parallel. As mentioned before, it is surprisingly hard to identify the true serial or parallel architecture from response times alone (Townsend and Ashby, 1983). However, for many practical applications – e.g., in human-computer interaction or as in the above example, where children count on their fingers – these assumptions are plausible, and the true architecture is of less interest than a model of the expected increase in response time with relevant task variables.

4.1 *EIP regression*

We assume that each EIP step takes some time, so that trials that require more EIP steps will, on average, have proportionally longer response times than trials with fewer EIP steps. Additionally, there is a constant offset component in the model, accounting for processes that are the same in all trials (e.g. orienting oneself or initiating the motor response of pressing a key). Hence, the expected response time of participant i on trial j can be described with the following linear equation:

$$\mu_{ij} = a_i x_{ij} + b_i. \tag{4.1}$$

This mean response time μ_{ij} is given by the constant time b_i (accounting for the steps that are the same across trials) and the parameter a_i that is multiplied by the number of EIP steps x_{ij} (that vary between trials). We want to estimate the two parameters a_i and b_i to find out the processing speeds for each participant.

In linear regression, a noise term—usually normally distributed with constant variance—is added to the mean to account for the variability of the data. Here, instead, we assume that every single EIP step is of random duration. EIP steps will not take the exact same amount of time in each execution. We model each EIP step as a draw from a gamma distribution with mean a_i and the constant part as a single draw from a different gamma distribution with mean b_i. The fact that gamma distributions are skewed and only defined on positive values make them very well suited for modeling the duration of an EIP step (Maris, 1993). A natural effect of summing several random EIP steps is that the expected spread of the overall response time increases with a higher EIP step count. See Figure 4.1 for an example of how the regression line with the gamma density around it might look like. Hence, using gamma distributions for EIP steps

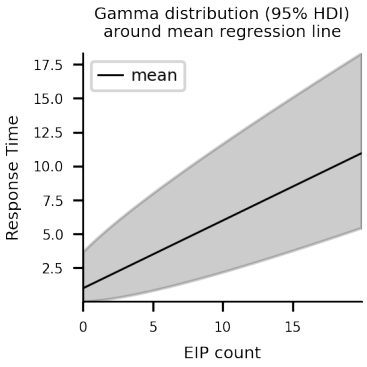


Figure 4.1: Regression line with asymmetrical gamma density around it. Here the parameters are $a = 0.5$, $b = 1$, $\theta = 1$. With an increasing count of EIP steps the spread increases.

fixes two conceptual flaws compared to simply assuming normal errors and homoscedacity for Eq. 4.1, as it is usually done. First, response times are constrained to be positive and, second, the variance increases with the step count, as it should.

Let $z_n \sim \text{Gamma}(k_n, \theta_i)$ be the random processing time of the n^{th} EIP step. Usually, gamma distributions are specified using parameters for shape, k, and scale, θ. Mean and standard deviation of the distribution are then $m = k\theta$ and $s = \sqrt{m\theta}$. We prefer, however, to parameterize the gamma distribution directly by mean and standard deviation because they are easier to interpret. We denote this parameterization of the gamma distribution Gamma′. With $m_n = k_n\theta_i$ the random processing time z_n for the n^{th} EIP step can then equivalently be written as

$$z_n \sim \text{Gamma}'(m_n, \sqrt{m_n\theta_i}). \tag{4.2}$$

We assume that the z_n are identically and independently distributed. Psychologically, this means that for each participant there is just one elementary information process and that there is across-stage independence (Townsend and Ashby, 1983). We also assume that the EIP steps are processed serially, hence, we now want to know the total response time distribution of $N+1$ random steps, $y = \sum_{n=0}^{N} z_n$. We assume all gamma distributions of participant i share the same scale parameter θ_i. In this way, the overall response time distribution for y is easy to compute, because the sum of several independent gamma variates z_n from distributions with the same scale θ_i and shape parameters k_n is again gamma distributed: $y \sim \text{Gamma}(\sum_{n=0}^{N} k_n, \theta_i)$. The mean of this gamma distribution is $M = \theta_i \sum_{n=0}^{N} k_n = \sum_{n=0}^{N} m_n$ and its standard deviation is $S = \sqrt{M\theta_i}$. And, hence, the overall response time $y \sim \text{Gamma}'(M, S)$ in our alternative parameterization of the gamma distribution.

Remember that each participant i is modeled by three parameters that describe the gamma-distributed response time of each trial: The mean time each EIP step takes, a_i, the mean offset, b_i, and the scale of the gamma distributions θ_i. Hence, if participant i needs $N = x_{ij}$ EIP steps on trial j, $m_0 = b_i$ for the initial step and $m_n = a_i$ for the other identical EIP steps (where n ranges from 1 to x_{ij}), then the overall mean of the summed EIP-step-times is $M = a_i x_{ij} + b_i = \mu_{ij}$ (Equation 4.1) with standard deviation $S = \sqrt{\mu_{ij}\theta_i} = \sigma_{ij}$. The random response time y_{ij} of participant i on trial j is therefore

$$y_{ij} = \sum_{n=1}^{x_{ij}} z_n \sim \text{Gamma}'\left(\mu_{ij}, \sigma_{ij}\right). \tag{4.3}$$

The basic EIP regression model is, thus, a linear gamma regression with the intuitive interpretation of a latent random duration for each processing step.

Importantly, because we know the distribution of the sum of the latent gamma variates we do not have to model the duration of the single EIP steps z_n explicitly. This is a great computational advantage in the hierarchical Bayesian model that we propose in the next section. Other commonly used distributions for response times (e.g., the

Weibull or log-normal distributions) do not have this property and would, therefore, be a lot more cumbersome to work with. In later sections we will extend the use of the EIP regression model to cases where not just the duration of the intermediate steps is unknown but the overall number of EIP steps is also latent. If we assume a distribution for the number of steps, the expected response time of participant i on trial j will be

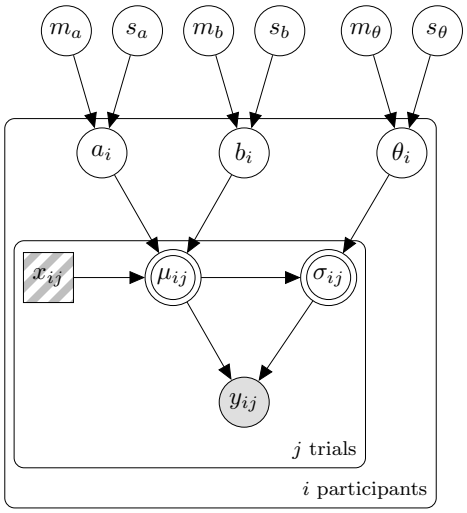
$$\begin{aligned}\mathbb{E}(y_{ij} \mid a_i, b_i, \theta_i) &= \mathbb{E}(\mu_{ij} \mid a_i, b_i, \theta_i) \\ &= \mathbb{E}(a_i x_{ij} + b_i) \\ &= a_i \mathbb{E}(x_{ij}) + b_i \end{aligned} \tag{4.4}$$

We know the distribution of x_{ij} and can thus easily calculate its expected value. To find the unique combination of values for a_i and b_i one needs at least two experimental conditions with different expectations for $\mathbb{E}(x_{ij})$. While it is possible to estimate the parameters from the means, it is statistically much better to try and model the full response time distributions y_{ij} to infer the parameters. In order to do so, we need to take into account the full distribution of the number of EIP steps x_{ij}. This complicates the statistical modeling considerably because, in this case, the response time distribution is a mixture of gamma distributions. Before we introduce these complications with an example model for response times in a Sudoku task, we will first present a hierarchical extension to model individual differences and apply this model to response time data of children adding numbers.

4.1.1 *Bayesian hierarchical EIP regression*

By adding one layer of priors we can easily extend the basic gamma regression model to a hierarchical model and estimate the parameters of several participants in parallel. Hierarchical Bayesian models have big advantages over estimating each participant on its own: parameter estimation becomes more robust and one can borrow strength across participants (Gelman and Hill, 2007; Rouder et al., 2003).

As the parameters for each participant (a_i, b_i, and θ_i) all need to be positive, we sample them from log-normal priors. Usually, log-normal distributions are characterized using parameters μ and σ, which are the mean and standard deviation of the logarithm of the distribution (which is a normal distribution). Similar to the gamma distribution, we define a different parameterization denoted by LogNormal′, using the mean and variance of the distribution itself. This parameterization leads the parameters to be measured in seconds and, hence, makes them easier to relate directly to the data. Both of these hyperparameters need to be positive, too. For the hyperpriors on the means, we use a half-normal distribution with $\sigma = 5$. For the hyperpriors on the variances, we use half-Cauchy distributions with $\beta = 5$. These values proved to be reasonable in our applications of the model. They do favor small values but are wide enough to allow for a wide range of values.



$$
\begin{aligned}
m_a, m_b, m_\theta &\sim \text{HalfNormal}(\sigma = 5) \\
s_a, s_b, s_\theta &\sim \text{HalfCauchy}(\beta = 5) \\
a_i &\sim \text{LogNormal}'(m_a, s_a) \\
b_i &\sim \text{LogNormal}'(m_b, s_b) \\
\theta_i &\sim \text{LogNormal}'(m_\theta, s_\theta) \\
x_{ij} &\leftarrow \text{EIP step count} \\
\mu_{ij} &\leftarrow a_i x_{ij} + b_i \\
\sigma_{ij} &\leftarrow \sqrt{\mu_{ij}\theta_i} \\
y_{ij} &\sim \text{Gamma}'(\mu_{ij}, \sigma_{ij})
\end{aligned}
$$

Figure 4.2: Hierarchical EIP regression model. The outer box is the core regression model for each participant i. The observed response time y_{ij} depends on a known (or latent, see later sections) number of EIP steps x_{ij}. The expected response time μ_{ij} is a linear function of the number of steps with slope a_i and offset b_i. The standard deviation σ_{ij} scales with θ_i.

The graphical model of our hierarchical EIP regression can be found in Figure 4.2. In the display of the graphical models we follow the conventions used in (Lee and Wagenmakers, 2014): Round nodes are continuous, squared nodes represent discrete values; open nodes are latent variables, shaded ones are observed; a node with double border is deterministic. In addition, plates enclose parts of a graph to denote independent replications.

This basic model can be extended to more complex cases when the number of EIP steps for a trial are not observed or when there are several tactics to solve a task which would produce a different number of EIP steps. We will deal with such extensions later in the paper. In general, we implemented all graphical models discussed in this paper with PyMC (Salvatier et al., 2016) and used the No-U-Turn Sampler (NUTS, (Hoffman and Gelman, 2014)) to find good parameter values. We let four chains run in parallel for 2000 iterations, after tuning the hyperparameters of NUTS with 1000 samples which were discarded. Convergence was checked via visual inspection of the traces as well as the diagnostic parameters $\hat{R}$ and effective sample size (ESS) (Vehtari et al., 2021). The code for all models can be found here: https://osf.io/rgh3j/.

4.2 *A first example: Addition*

The simplest possible use-case for the model is when there is just one cognitive tactic that we want to model and the EIP step count for each trial is known. We will show the application of the model in one such simple case: Adding two numbers.

When children learn to add, they usually start by putting up their fingers for each addend and then simply count the fingers (Siegler and Jenkins, 1989). Before they reach the proficiency level of adults and can retrieve the answer to small addition problems from memory, they usually discover several shortcuts to counting explicitly

through all the numbers from one to the sum. A quite sophisticated tactic, called min-counting, is to start counting at the larger of the two addends (e.g. $3 + 5$: start at 5 and count three numbers up to get the answer) (Siegler and Jenkins, 1989). During the learning process, children usually use several tactics concurrently. With experience, the most efficient tactics come to dominate, which are the retrieval tactic followed by the min-counting tactic (Siegler, 1987).

Hopkins and Bayliss (2017) examined what tactics children in 7th grade use to solve simple addition problems where both addends are single digits. Here, we use their data to illustrate how our model can be used to infer the temporal properties of EIP steps in the min-counting tactic. 200 children from 13 schools with a mean age of 12.38 years took part in the study. The addition part of the study consisted of 36 trials, all single digit additions with addends greater than 1. After each answer a child gave, the experimenter asked how they arrived at the answer. Answers were classified as min-counting, retrieval, decomposition and other (e.g. "don't know"). For details of the experiment see Hopkins and Bayliss (2017). Here we are only concerned with the min-counting trials which make up about a quarter of the data (of the 6855 correct trials, 1786 were min-counting trials).

The process model for min-counting trials comprises the following parts: read the question, find the bigger number, count from that number as many steps up as the other number indicates, state the answer. Each counting step in this model thus constitutes an EIP step. We can therefore map the counting speed of child i onto the parameter a_i and the duration of the other processes onto the parameter b_i.

4.2.1 *Results of EIP regression*

The estimates for the group parameters of the Bayesian hierarchical EIP regression model can be found in Table 4.1. These group parameters tell us the mean parameter values and how the parameters vary over participants. For example, looking at the mean values for a and b we see that on average the offset parameter is roughly four times as big as the slope parameter. Hence, on average the constant processes (orienting, response initiation, etc.) take about four times as long as each counting step.

For each participant, we also get the individual posterior distributions of all three parameters. The distribution of parameter densities for all participants can be seen in Figure 4.4. The regression line for a single, exemplary participant can be seen in Figure 4.3. The plot on the left of Figure 4.3 shows the conditional fit for the location of the mean (given the data) with its confidence intervals (95% HDI). These are calculated as the respective quantiles of the matrix $a_i x + b_i$, where x is the vector $[2, ..9]$ and a_i and b_i are all the posterior samples of the respective values for participant i.

The x-axis depicts the EIP step count, the y-axis the response time

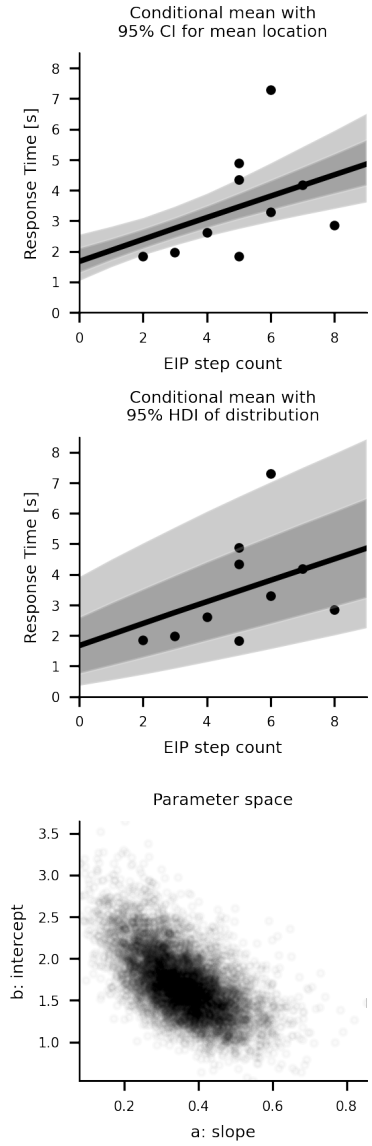


Figure 4.3: Results for one exemplary participant for the EIP model of min-counting. The regression line and the data points are the same in the first two plots. In the bottom most plot the individual samples of the two parameters a and b are depicted.

	mean	SD	HDI 2.5%	HDI 97.5%
$m_a[s]$	0.434	0.029	0.379	0.493
$m_b[s]$	1.724	0.084	1.561	1.889
$m_\theta[s]$	0.509	0.047	0.421	0.602
$s_a[s]$	0.293	0.040	0.221	0.373
$s_b[s]$	0.447	0.100	0.253	0.637
$s_\theta[s]$	0.485	0.089	0.324	0.653

Table 4.1: The parameters found for the EIP regression for the counting trials. The m parameters are the mean and the s parameters the standard deviations of the prior distributions of the respective participant parameters. All values are in seconds.

in seconds. The dots mark individual response times of this participant (only those trials where the participant reported they had used "counting" to find the answer). There were three trials with the minimum value 5 (with response times varying between 1.8 seconds to 4.8 seconds), for all other minimum values at most two trials were solved by counting by this participant. The solid line is the mean of all posterior regression lines of this participant. Despite the hierarchical model, the 95% HDI around the mean is still on the order of one second. However, note that the regression of this student is based on only a few trials. We again see that the constant offset is relatively large, around 2 seconds, compared to the average time this participant needs to do one counting step. In the middle of Figure 4.3, the same mean line is depicted, the shaded area around it illustrates the density of the gamma distribution around it. Except for one data point all the data are well within the expected variance. To help the reader relate standard linear regression analysis to the EIP regression and appreciate the difference as well as the similarity, we included a linear regression model in the appendix (section A.2). The main results are not much different in the two models, the difference is mainly in the shape of the expected distribution of the data around the regression line.

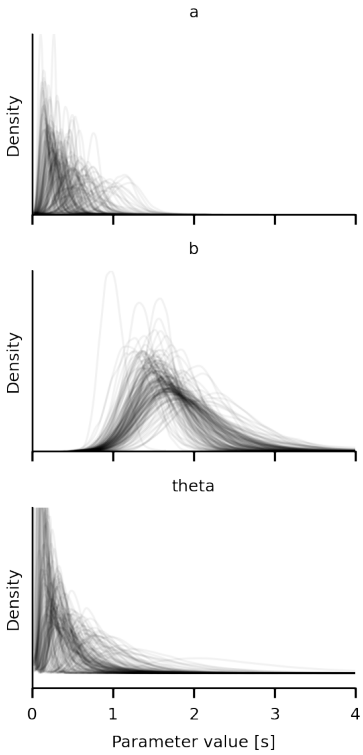


Figure 4.4: The distributions of the participant parameters for all participants for the EIP model of min-counting. Each line shows the posterior density of values for the respective variable for one participant.

4.3 *A more complex example with latent steps: Sudoku*

In the previous example, the number of processing steps that each participant went through was known. Hence, we could simply plot the number of EIP steps against the response times and the linear fit gave us estimates of the offset and processing time of each EIP. However, for more complex tasks, usually, we will not know the number of processing steps exactly. As an example for such a task we will look at Sudoku. In general, there are often different ways to solve a problem. In the previous addition example, there was min-counting and retrieval from memory. For Sudoku, too, there are different tactics to fill a cell in a puzzle. Contrary to the data for the addition example, where we knew how a student solved the addition problem, here, we will look at data where we do not necessarily know which tactic was tried. Furthermore, also contrary to the counting example, tactics are rarely so simple that each participant executes the tactic in the same way in each situation. Participants do not

follow a tactic deterministically but make probabilistic choices. For example, in Sudoku a participant might sometimes choose to first look at the columns and then at the rows and sometimes do it the other way round. Thus, the number of processing steps needed to solve a problem is not fixed but random and, usually, unobserved. Fortunately, the basic EIP regression model can be extended to deal with these complications.

In this section, we will first cover a previously published study on Sudoku (Behrens et al., 2023) to illustrate how these additional complications commonly arise in problem-solving data. In this previous paper we also published a process model for how people solve Sudokus but only applied it to group data. Here, we will showcase how Bayesian hierarchical EIP regression can be combined with the process model to analyze these data at the single-participant level. These analyses will be very similar for other problem-solving tasks where the results of some reasoning processes can be observed in behavior, but many cognitive steps are latent.

4.3.1 *Sudoku tactics*

Let us first look at the two most simple tactics in Sudoku. They work with different elements as focus: "*What* digit can I place *in this cell*?" vs. "*Where* can I place *this digit*?" The cell-based tactic (CB for short) tests for a single empty cell of the puzzle which digits can be excluded from it by looking at the surrounding cells. When a digit appears already in the same row, column, or box, it cannot be placed in the cell under consideration. If all but one digit can be excluded, the one remaining digit is the solution for the cell. The digit-based tactic (DB) focuses on a specific digit which occurs already several times on the board. When it does not yet occur in some unit, for example a 3-by-3 box, one can see whether the other occurrences of the digit restrict where in the given box this digit can be placed. If all empty cells of the box but one can be excluded as locations for the digit, the one remaining cell has to be filled with the digit. See Figure 4.5 for examples of the tactics.

cell-based tactic

	A	B	C	D	E	F	G	H	I
A				7		8	5	1	4
B		5	4		6	3		2	
C	1	8	2			9	3		
D	6	2							
E	4								
F			5						
G		9	7						
H	3		8						
I	5	1							

number-based tactic

	A	B	C	D	E	F	G	H	I
A			3	7	2			1	4
B	7				6		8	2	9
C	1	8	2	5		9			
D		2							
E	4								
F	8	3	5						
G		9							
H	3	4	8						
I			6						

Figure 4.5: Examples for the two tactics. The correct answer is 9 for the cell AA in both puzzles. For the cell-based tactic, the easiest way to find the answer is by asking "what digit can I place in cell AA?" The easiest way for finding the answer in the digit-based puzzle is by asking "where in the upper left box (A:C–A:C) can I place the 9?"

4.3.2 *Experiment and instruction groups*

In the experiment, we had two different instruction groups that were supposed to bias participants to use one or the other tactic. The task for participants of both instruction groups was to fill in one correct digit in a given Sudoku per trial. In order to do so, they had to click on an empty cell with the mouse and then enter a digit via the keyboard. In the cell-based instruction group one cell was highlighted, and the instructions were "Please fill in a digit in the highlighted cell". For the digit-based instruction group, a 3-by-3 box was highlighted and the instructions were "Please fill in the X into the highlighted box" (where X was replaced with a specific digit in each trial). Both instruction groups saw the same puzzles in a random order, half of which could only be solved with CB tactics,

the other half only with DB tactics. Hence, if a participant always started with the same tactic, in half of the cases it led to the correct answer, in the other half they would have to follow up on the first attempt with the second tactic in order to successfully solve the puzzle. The experiment was thus a 2×2 design, with the independent variables instruction and required tactic. The data set, the experimental materials, and the analysis code are available at the OSF project site https://osf.io/rgh3j/. The data set consists of 46 participants, 23 per condition, who met the inclusion criterion (at least 75% correct responses). Every participant completed 36 trials, all saw the same puzzles in randomized order. For both required tactics we had two seed puzzles and created 9 isomorphic stimuli from them. The isomorphs were created by exchanging rows and columns and interchanging the values for different digits. Although the isomorphs look different on the surface, they afford the same logical inferences and the process model predicts the same number of EIP steps for them.[1] We, therefore, do not distinguish between the individual puzzles in the following analyses, but just use one distribution of EIP step counts for each combination of instruction and required tactic.

[1] Some differences in EIP step counts arise due to the order in which digits are tried. If a smaller digit is tried first and it is the required target digit then this puzzle has a lower number of EIP steps. But we do not want to make the strong assumption that our participants always follow this exact order. So we lump all target digit scan distributions together.

4.3.3 *Process model (First strategy)*

The information given in the two instruction groups is different (fixed cell vs. fixed digit) and the process model we developed accordingly also covers the two different cases. The basic processing step that we use in the process model is to search for a digit in a given unit (e.g., looking for the 3 in the first row). We call such a step a scan. Scans are the EIP steps for all following models. Note that each scan consists of several sub-steps, i.e., looking at each cell in a unit, that are not modeled explicitly. The model makes predictions about the required number of scans for a given puzzle and instruction. Importantly, the process model makes some random choices and therefore the number of EIP steps needed in a trial varies randomly. For example, in Figure 4.5 in the CB example, participants will at some point have to scan the A-row and the A-column for a 3 to check whether it can be filled into cell AA. We assume that participants will randomly either first check the A-row or the A-column. Participants who first check the A-column get lucky and only need one scan to exclude the 3. Participants who first look at the A-row will also have to search the A-column and therefore need two scans. In the case where the first executed tactic does not lead to a unique answer, the other tactic has to be applied as a follow-up. The strategy we implemented as a process model here is to always start with the tactic that best fits the instruction they see. We thus expect more scans in incongruent trials, i.e., where the required tactic does not match the one favored by the instruction type. For a detailed description of the proposed algorithms behind the process model see (Behrens et al., 2023). We let the model run on on the puzzles for 1000 trials to get a distribution of

the EIP step count in each of the four conditions.[2] Hence, instead of knowing the precise number of EIP steps for each trial as in the min-counting example, here, we only know the probability distribution over the number of EIP steps that will be needed by our stochastic process model. In Figure 4.6 the solid lines show the distribution of EIP step counts in the different conditions.

[2] For some models it will be easy to compute this distribution by hand but in general it will be easier to estimate this distribution by sampling.

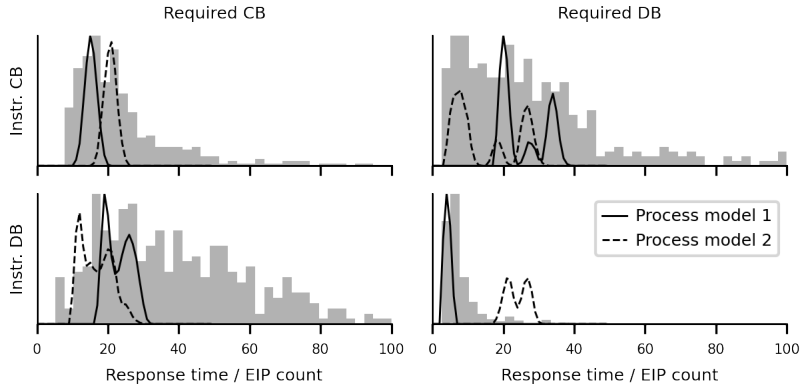


Figure 4.6: The gray bars represent the response times of the participants in the four conditions of the Sudoku experiment, the lines depict the EIP step counts for two different strategies. For the first Sudoku model, only the solid lines are relevant. They show the EIP step counts for the process model of the first strategy we implemented. The dashed lines (strategy 2) show the alternative process model that is additionally used in the EIP model with strategy selection. Note that the x-axis depicts seconds for the response times but 'number of EIP steps' for the scan distributions.

4.3.4 *EIP regression with latent steps*

The core of the model is identical to the model we used for the counting data. The added complexity stems from the fact that we do not have a definite EIP step count per trial, but a distribution over EIP steps instead. We still want to estimate how long each participant needs on average to carry out an EIP step, i.e., a scan. Additionally, we estimate an intercept term to take care of the processing time not captured in the process model (reading the task, typing the answer). Of course, this is also a validation attempt of the process model. Only if the model makes reasonable predictions about the required number of EIP steps in the different conditions (or at least their relative proportions) can a good fit be found that explains the response times mainly on the basis of the EIP step count.

As mentioned above, before we fit the EIP regression model, we first let the process model run on the stimuli to get a discrete distribution of the EIP step counts per condition. As we now know this distribution, we can treat the number of EIP steps x_{ij} as a latent variable in our Bayesian model. The only change that is required to the original Bayesian hierarchical EIP regression from before is that we need to provide the discrete prior distribution for each x_{ij} (that depends on the stimulus that was shown in trial j and the instruction participant i received). Hence, each y_{ij} is now not gamma distributed anymore but a mixture of gamma distributions over the latent number of EIP steps. When fitting the EIP regression model, we could just draw from the latent distribution to get one specific EIP step count for each draw. Instead, we marginalize over this distribution and work directly with the mixture of gamma distributions for y_{ij}.

Marginalization has the advantage that the samples converge more quickly to a stable distribution. Without marginalization, we need to take more samples in each chain to reach convergence. Once the chains have reached convergence, the results are the same in the two approaches, as one would expect given their equivalence. We report the results from the model with marginalization here.

4.3.5 *Results of latent EIP regression*

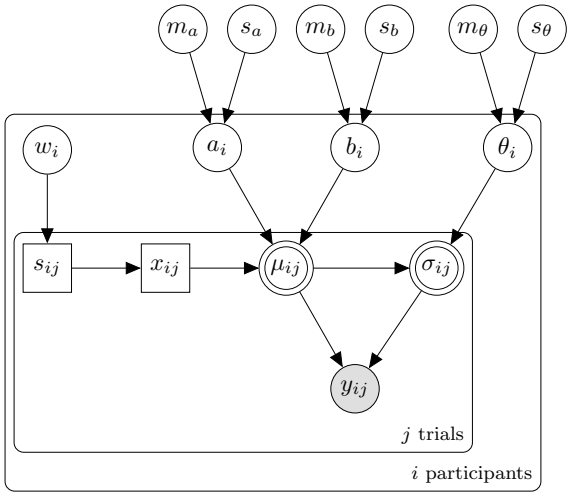
	mean	SD	HDI 2.5%	HDI 97.5%
$m_a[s]$	1.526	0.386	0.945	2.249
$m_b[s]$	7.270	1.694	4.315	10.696
$m_\theta[s]$	5.243	0.705	3.960	6.639
$s_a[s]$	2.458	1.736	0.746	5.346
$s_b[s]$	14.424	7.528	4.266	28.945
$s_\theta[s]$	4.542	1.249	2.523	6.972

Table 4.2: The parameters found for the EIP regression with latent EIP step count for the Sudoku data. The m parameters are the mean and the s parameters the standard deviations of the prior distributions of the respective participant parameters.

The values of the group parameters can be seen in Table 4.2. For each participant, i, individual values are sampled for the parameters a_i, b_i and θ_i. Their mean is depicted on the left of Figure 4.10. According to these fits, participants take between 0.1 and 3 seconds for one EIP step. The intercept term accounts for all the other processes in the response time. It is below 5 seconds for about half of the participants. The other half has very variable intercept terms, including values of up to 25 seconds. Such high intercept terms are a sign of a rather poor process model because in our case the response times ranged from about 10 to about 60 seconds. If more than half of the time needs to be accounted for by the intercept term instead of the sum of the EIP steps, this means that the distributions of EIP steps do not match participant behavior very well. Luckily, not all is lost, and we can extend the model further to better account for the data.

4.3.6 *EIP regression with strategy selection*

The strategy described above assumes that first the instructed tactic is carried out fully, and only if it did not lead to a unique solution, subsequently the other tactic is carried out. However, it might also be that our participants do not always apply the tactics in this order, instead, they might be starting with the tactic that does not fit the instruction. If they do this, they need more EIP steps in the congruent trials but fewer in incongruent trials. We defined "starting with the other tactic" as a second strategy, implemented a corresponding process model and again sampled the expected numbers of EIP steps per condition to get distributions of scan numbers. The dashed line in Figure 4.6 shows the corresponding distributions. We extend the EIP regression model to incorporate strategy selection for each participant. In order to estimate to what extent each of the two strategies explains their response time patterns, for each trial, we draw from a



$$
\begin{aligned}
m_a, m_b, m_\theta &\sim \text{HalfNormal}(\sigma = 5) \\
s_a, s_b, s_\theta &\sim \text{HalfCauchy}(\beta = 5) \\
a_i &\sim \text{LogNormal}'(m_a, s_a) \\
b_i &\sim \text{LogNormal}'(m_b, s_b) \\
\theta_i &\sim \text{LogNormal}'(m_\theta, s_\theta) \\
w_i &\sim \text{Beta}(1, 1) \\
s_{ij} &\sim \text{Bernoulli}(w_{i_k}) \\
x_{ij} &\sim \text{EIP step count}(s_{ij}) \\
\mu_{ij} &\leftarrow a_i x_{ij} + b_i \\
\sigma_{ij} &\leftarrow \sqrt{\mu_{ij}\theta_i} \\
y_{ij} &\sim \text{Gamma}'(\mu_{ij}, \sigma_{ij})
\end{aligned}
$$

Figure 4.7: EIP regression model with latent EIP step count and strategy selection for the Sudoku experiment.

Bernoulli distribution with probability w_i and decide which strategy s_{ij} is used to explain the response time on trial j. We sample these weights w_i from a uniform beta distribution (Beta(1,1)). From the EIP step distribution corresponding to the selected strategy, one specific EIP step count, x_{ij}, is sampled. From here on the rest of the model is the same as in the basic EIP regression model. See Figure 4.7 for the corresponding graphical model. Again, we can use marginalization to reach convergence in fewer sampling iterations. In this case, we use the w_i parameter as a mixing weight of the two strategy distributions. Note, however, that the distributions of EIP step counts for the two strategies have considerable overlap (see Figure 4.6), so it is probably impossible to say very precisely how much of each of these strategies contributed to the performance of a participant.

4.3.7 *Results of strategy selection analysis*

The values of the group parameters can be seen in Table 4.3. Compared to the values found by the simpler model, the intercept term m_b decreased significantly, while the variance related term, m_θ, decreased and slope, m_a, increased a bit (the confidence intervals for m_a and m_θ are overlapping in the two models). The standard deviations of the prior distributions for the three parameters (s_a, s_b and s_θ) decreased all at least a bit, indicating that the individual participants now have more similar parameter values compared to the model with just one strategy. The widths of the highest density intervals of all population parameters (except for s_b decreased, showing that the model is more precise in its results. The densities of the participant parameters can be seen in Figure 4.9, split into the two instruction groups. The mean of the slope and intercept terms for both models can be seen in Figure 4.10. The DB-instruction group gets consistently very low weight parameters for the new strategy (bottom right of Figure 4.9), meaning that they overwhelmingly do

as the first strategy proposed, i.e. start with the DB-tactic. They were well fit by that model and the additional strategy did not change the fit much. Accordingly, the blue triangles did not move very much between the two plots in Figure 4.10. The most striking difference between the two models in Figure 4.10 is that the intercept terms, b_i, decreased dramatically for many participants in the CB-instruction group. In this group, some participants get a high weight for the newly added strategy. Together, this indicates that around half of the participants in the CB-instruction group did not follow the first strategy's assumptions, but instead often opted for using the DB-tactic early, even though it was not favored by the instructions. With the additional tactic in the model, a much bigger portion of the overall response time can be explained by the EIP step count assumed in the model instead of needing to be covered by the "catch all" intercept term. This shift alone makes the new model with strategy selection a much better model in our eyes. When the response times can be explained by differences in required EIP steps, it means that there is the possibility of some correspondence between the process models and the processes in the head of the participants.

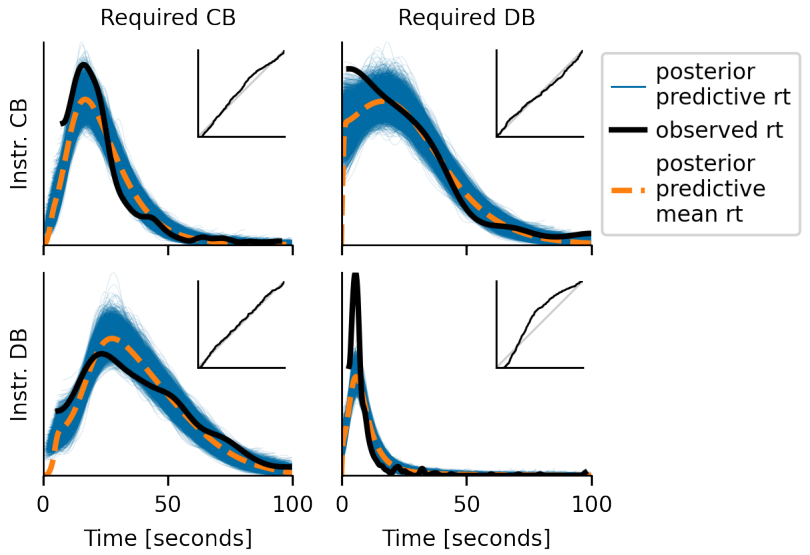


Figure 4.8: The posterior predictive distributions for the EIP regression model with latent EIP step count and strategy selection for the Sudoku experiment. The inset figures show the PP-plots for the respective condition.

in Figure 4.8 the posterior predictive distributions of the model for the four conditions of the experiment can be seen. The lower right part of the figure shows the congruent digit-based condition. Here, the observed response times contain many very short answers, which the model cannot match perfectly. The other three conditions are fit very well, as can be seen in the almost perfect diagonal PP-plots in the insets.

	mean	SD	HDI 2.5%	HDI 97.5%
$m_a[s]$	1.490	0.141	1.232	1.782
$m_b[s]$	2.589	0.950	1.053	4.433
$m_\theta[s]$	3.525	0.532	2.557	4.555
$s_a[s]$	0.893	0.192	0.577	1.265
$s_b[s]$	6.724	10.001	0.548	17.713
$s_\theta[s]$	3.296	1.032	1.759	5.344

Table 4.3: The parameters found for the EIP regression with latent EIP step count and additional strategy selection for the Sudoku data.

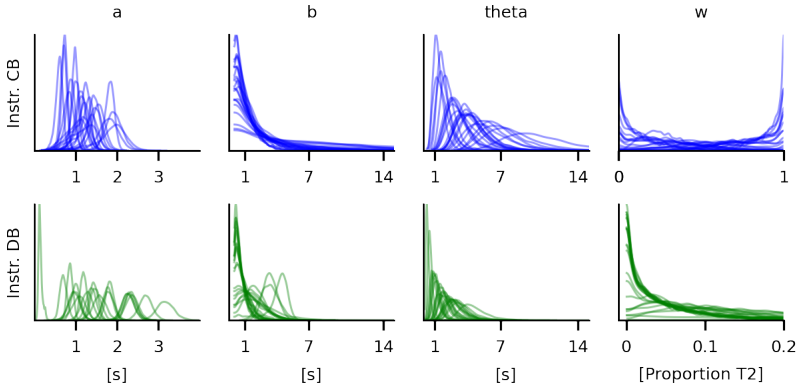


Figure 4.9: The posterior density of all the participant parameters in the Sudoku model with strategy selection. The two rows of the plot show different instruction groups.

4.3.8 *Model comparison*

Besides the observation that the parameter values are in more plausible ranges in the model with strategy selection as compared to the one without, we can also look at the fit of the data more formally. We use 10-fold cross validation to do so. We split the data such that all participants are equally represented in all folds. We train the model on nine tenths of the data and test the performance on the last tenth, which was not used for this training round. This is done ten times, where each fold of the data is the held-out part once. We thus get a measure for how well the model can predict data it has not seen during training. As performance measure we use expected log pointwise predictive density (ELPD) as described by Vehtari et al. (2017), which is the log likelihood of the data given the entire distribution of parameter estimates instead of point estimates of parameters. The ELPD for the EIP regression with latent EIP step count is -5552.32 (with a standard error of 49) and for the model with additional strategy selection it is -5489.08 (with a standard error of 49), so there is a difference of 63.24 (with a standard error of 18.19 on this difference), showing that the data are much more likely under the more complex model. The ArviZ project (Kumar et al., 2019) also provides a tool to compute an estimated leave-one-out ELPD on the traces. The data contain outlier points resulting in warnings about highly influential observations. Nevertheless, the results by the ArviZ toolbox confirm the results of the cross validation.

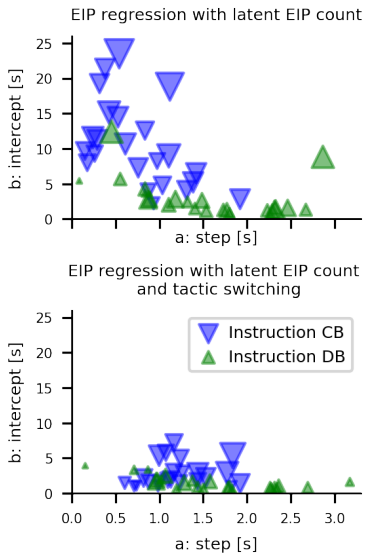


Figure 4.10: The mean of the a_i and b_i parameters for each participant as fit by the two Sudoku models. The color and orientation of the triangles mark the two instruction groups. The size of the triangles reflects the value of θ, so smaller means better predictable.

4.4 Discussion

We have shown how to connect process models to response time data using gamma regression. The key idea is to treat each EIP step as having a gamma-distributed random processing time. We assume the same scale parameter to ensure that the overall response time is also gamma distributed. We call this core model EIP regression. This model can be extended to a Bayesian hierarchical regression model. Such a model allows us to fit the predictor variable of a process model, i.e., the number of EIP steps, to the response time data of individual participants. We illustrated how the model works by applying it to children's addition (and you can find a comparison to a standard regression analysis in the appendix, see section A.2).

In a second example, we developed a process model of a Sudoku task which made predictions about the required number of processing steps in different conditions. As the process model is probabilistic in itself, the predicted number of processing steps is not a single number but a distribution of possible values instead. Even though the number of EIP steps is not observed, the Bayesian hierarchical EIP regression can still be fit to those data. Connecting the discrete predictions of the process model with the quantitative probabilistic model allows us to do two important things: First, we can get an estimate for how long a processing step should take given our process model and the data we collected. Hence, we can make statistical inferences about processes that are never observed in isolation. Second, it also allowed us to statistically assess the process model itself. We saw that our first attempt did not match the data of the cell-based instruction group very well. The average response time assigned to the processing steps was very short and most of the overall response time had to be explained by the intercept term. We implemented a second process model that allows for starting out with the tactic that does not fit the instruction. In the EIP model, we added a weight parameter for each participant to fit the degree to which each of the two strategies explains the response time patterns. An second version of the process model that allows for starting with different solution tactics improved the fit immensely. The improvement was clearly shown in a model comparison (much better log likelihood of the data) as well as in the values for the individual parameters.

Using a probabilistic programming framework, like PyMC (Salvatier et al., 2016; Wiecki et al., 2022), allows us to define models that would not be expressible in traditional statistical analyses. For example, the generalization from an observed to a latent number of processing steps is straightforward in PyMC but would be very hard using standard regression tools. In general, one motivation for this paper was to showcase how probabilistic programming can be used to bridge the gap between classical cognitive modeling and statistical inference. Traditionally, process models are hypothetical algorithms of how participants solve a problem and make qualitative predictions about processing times, but they are seldom scrutinized

statistically. Lee and Wagenmakers (2014) have long advocated the use of Bayesian tools for cognitive modeling and their book provides an excellent collection of basic and advanced models. Our contribution here is to provide one model, EIP regression, that can easily be adapted to many use-cases. If you have a probabilistic process model that you can implement in a computer program, and the program has clearly identifiable elementary information processes, then you can use EIP regression to infer the model's parameters. Importantly, these parameters have a clear psychological interpretation in terms of the average processing time of an EIP. In addition, as the core model is a simple regression model, the data analysis is very similar to standard analyses in psychology. We thus believe that EIP regression can easily supplement many existing process models.

In fact, decision-making tactics have already been analyzed in depth using EIP models and standard regression tools (Bettman et al., 1990; Payne and Bettman, 2004; Payne et al., 1988). A host of different decision-making tactics exist, some relying on single features, some considering several or all features of the choice options, but they all share the same elementary information processes. Some models have even differentiated between different processes (e.g., counting, multiplication, comparisons, reading) within the same experiment and estimated the duration of each of these individually. This was, however, only possible for simple models and through very clever experimental design. In contrast, estimating the duration of several interacting elementary information processes even for complex models is a straightforward extension of the EIP regression we presented in this article. Ideally, in the future EIP regression should be integrated with GOMS-like modeling frameworks, e.g., Cogulator (Estes, 2021). This would immediately make modern statistical estimation and model comparison tools available for a large class of classical cognitive models that are also widely applied in human-computer interaction.

Chapter 5

Statistical modeling of rule selection

This chapter is in preparation for submission.

The ability to solve problems is at the core of human intelligence. Newell and Simon (1972) revolutionized our understanding of problem solving by developing computer programs that can mimic human problem-solving behavior. They first recorded detailed think-aloud protocols while participants solved various problems. From these protocols they could then infer the problem space as well as the operations that participants used to search the space. They also carefully described in which context each operator could be applied. Importantly, as in some contexts several operators could be applied, they could also infer a preference order from a participant's think-aloud protocol. Using all these ingredients they developed computer programs with a repertoire of a participant's operators in the form of if-then rules. In these early programs, operators were always applied in a fixed preference order. Only when a rule was not applicable or did not lead to a change in the problem state the next rule was tried. Amazingly, with the right preference order such a production system could provide reasonably good explanations for a participant's trace through the problem space (Newell and Simon, 1972, Chapter 6). However, it was clear from the start that participants do not always follow a fixed preference order. Rule selection is probably probabilistic and highly context-dependent.

It has often been commented that Newell's and Simon's approach to problem solving has not yielded many deep theoretical insights beyond its early successes (Batchelder and Alexander, 2012; Jäkel and Schreiber, 2013; Ohlsson, 2012). One obvious problem of their qualitative approach is its reliance on think-aloud protocols. There is no simple way to systematically aggregate such data and analyze them statistically – as it's done in most other areas of cognitive science (Ohlsson, 2012). Here, we therefore develop a statistical model for probabilistic and context-dependent rule selection that can be fit to coded protocols or directly observable behavior. In this way, we can quantify the relative importance of different production rules and their statistical dependencies. Furthermore, we can cluster participants into groups that show similar behavior. We illustrate this

statistical approach to analyzing problem-solving behavior using Sudoku puzzles.

There are good reasons Newell and Simon relied so much on think-aloud protocols in their early exploratory studies. Think-aloud protocols are well-suited to trace the problem-solving process in a very open experimental setting. Participants can freely choose what to say and are not restricted to some predefined set of operators to chose from. The downside of think-aloud protocols is the very labor-intensive coding process, which relies on human coders. Despite all the recent technological advances in natural language processing, transcribing and labeling a protocol still typically takes several times longer than it took to record the protocol in the first place. Furthermore, there will always remain some degree of subjectivity in the labels even if multiple coders label the same data.

Because of these drawbacks of think-aloud protocols, experimenters usually try to design tasks in a way that externalizes as much of a participant's internal problem-solving process as possible. Participants should not plan and simulate everything only in their head but should draw it out or make moves in the environment (virtual or physical). This then leaves an easily recordable and interpretable trace. The *Towers of Hanoi* are one such example. The path to the solution is easily enacted in a physical or virtual representation of the problem and participants are unlikely to do lot of purely mental planning. This is one of the reasons the *Towers of Hanoi* are such a popular puzzle to study the application and learning of problem solving strategies (Anderson et al., 2005; Anzai and Simon, 1979; Kotovsky et al., 1985; VanLehn, 1991).

Once we have a trace of the problem-solving process (from a coded think-aloud protocol or other data sources), the most interesting part of data analysis still lies ahead. What where the rules and problem solving strategies that gave rise to this trace? Even a coded think-aloud protocol is a rich source of data that is not easily reduced to simple statistical tests. Even though it is not a very common technique, think-aloud protocols have a small but constant appearance over the years (Blech et al., 2019; Brandstätter and Gussmack, 2013; Chi, 1997; Ericsson and Simon, 1993; Fox et al., 2011; Walsh and Gluck, 2015). However, the process data are rarely analyzed at the same level of detail as in the work of Newell and Simon (1972). They carefully constructed so-called problem-behavior-graphs from the annotated protocols. These graphs trace the mental states of the participants, the operators that are presumably applied, and the resulting changes in problem state representations. They then built a production system with if-then rules to describe and simulate the participant's problem-solving behavior. To determine the execution priority of the different rules, they used a fixed preference order, estimated from the data. A fixed order, however, is not the best description of participants' behavior. Instead, participants usually display variability which would be better described with a probabilistic mechanism. Hence, our analysis focuses on statistically

estimating choice probabilities.

The easiest way to estimate rule selection probabilities would be to simply count how often each rule was chosen in a protocol to estimate a rule's probability. This approach works well in stable environments where all rules are applicable at any moment. However, when the applicability of rules changes from move to move, this analysis does not yield the full picture. Some rules might be used rarely, but the reason could simply be that they were applicable only on very few occasions. Also, it matters which other rules were applicable at the same time. We thus need to take into account the context in which a rule was chosen. In the analyses by Newell and Simon (1972) this was taken care of by the preconditions of the rules. However, if choices are probabilistic and context-dependent, you can only get reasonable count-estimates for the choice probabilities if you collected enough data for each relevant context. Unfortunately, in many problem solving tasks, contexts vary a lot over moves and you might only observe each context very few times, sometimes even just once. This makes estimating the probability for a rule to be selected a challenging statistical problem. Luckily, this estimation problem is not unique to rule selection. In fact, it is the core problem of choice modeling (Luce, 1959). Choice models allow us to estimate choice probabilities in varying contexts. We therefore propose to use such models to statistically analyze rule selection in production systems. After we estimated its parameters, we can use a choice model to compute the probability that a participant will select a production rule for any problem state. Together with the production rules that can change the problem state, we can then stochastically simulate complete traces for solving a problem. In this way, we can generate behavior similar to the behavior of the participants and can assign a probability to any given trace. A production system together with a choice model is thus a compact description of the distribution over possible traces.

In the present study we collect problem solving data on 4-by-4 Sudokus. These 4-by-4 Sudokus are an ideal problem domain for us because they are quick to fill and we can therefore present many puzzles in a single experimental session. Furthermore, there are several distinct rules for filling them and at any point in time several of them can be applied. Participants thus have a choice between different rules at all times. The many choice opportunities make 4-by-4 Sudokus an excellent problem domain for studying rule selection. Different knowledge of or preferences for rules will lead to different solution paths and these differences are easily observable in the overt behavior. We model these preferences with choice models that can be fit to the data of individual participants.

For our previous think-aloud study (see section 2.2), we already identified several inference rules that were used in similar puzzles to fill the empty cells. Labeling the moves of the think-aloud study was successful for a big proportion of the data, but very tedious and sometimes subjective. As most cells can be filled by several rules,

even the think-aloud data were frequently too sparse to uniquely determine the rule that was used. For choice modeling, the amount of data from the think-aloud study was unsatisfactory. We therefore designed an interface that allows us to infer the rules that were used on each move without having to rely on think-aloud data. In this new interface, participants used the mouse to click on and highlight all the cells in a Sudoku that were relevant for filling in a digit. We have thus externalized the most important aspect of the inference process to be able to automatically identify the rules that were used by the participants. This allowed us to efficiently collect (mostly) unambiguous problem-solving traces. These traces are then the basis for fitting choice models to describe each participant's preference for different rules.

5.1 *Production system*

To help us label the data of the think-aloud study in section 2.2, we developed a Prolog program that can solve Sudokus using the same rules our participants used in that study. It can be used to generate a list of all possible rules for each empty cell of a given Sudoku board.

These rules can also be seen as productions in production systems. The typical way of defining productions is by if-then rules. If certain conditions are met, then apply the change stated in the second part of the rule. In cases when the if part of several productions is met, a choice must me made which of them to "fire". Selection could be random, according to a fixed preference order, or probabilistic according to some utility weight attached to each rule. In this chapter we show that the utility weights for probabilistic selection can be learned from data to match each participant's behavior.

The following productions are written with the abstract variable UNIT. When fitting the choice probabilities, we actually fit a weight for each unit type (row, column and box) separately for those rules that have UNIT in their name. When a rule is applied, all the variables in it are instantiated with specific values such as *cell-2* for CELL and *row-1* for UNIT. The same rule might be applicable in several location, i.e., with several different values inserted for the variables.

Code 5.1: The following three productions summarize the principles of the actual code. The all-caps word are variables and need to be bound to specific values. All conditions in the "if" part need to hold simultaneously.

```
all-digits = set{1, 2, ..., N}

if
  CELL is empty
  CELL is part of ROW
  DIGIT is all-digits \ digits-in-ROW
  | DIGIT | = 1
then
  fill DIGIT in CELL
  label: last-in-row
```

```
// analogous definitions for
// last-in-column
// last-in-box

if
  CELL is empty
  CELL is part of UNIT.1
  CELL is part of UNIT.2
  DIGIT is all-digits \ (digits-in-UNIT.1 ⋃ digits-in-UNIT.2)
  | DIGIT | = 1
then
  fill  DIGIT in CELL
  label: cell-complex

if
  DIGIT is not in BOX
  CELL is empty-cells-in-BOX \ cells-where-DIGIT-in-peers
  | CELL | = 1
then
  fill DIGIT in CELL
  label: digit-box

// analogous definitions for
// digit-row
// digit-column
```

5.2 *Empirical study*

We designed an observational study in which participants solved 4-by-4 Sudokus. In order to be able to distinguish between inference rules, we required participants to highlight all cells that were relevant to their deduction. Figure 5.1 shows the interface of the study. We used a subset of the labels described in section 2.2 that will be shortly described here again. First, in cell-based reasoning, a participant is trying to fill a *specific cell* and, given the digits in the other highlighted cells, only one digit is still possible. An example can be the seen in Figure 5.1 (a). The participant is trying to fill the cell with the green outline. As there is a 4 already in the same row and 1 and 3 are already in the same column, only the 2 is still possible for the green cell. Second, in digit-based reasoning, a participant tries to put a *specific digit* they have in mind into a unit and the cell that is filled is the only cell in this unit, where the digit is still possible. An example can be the seen in the right panel of Figure 5.1. The participant is trying to place a 1 into the upper left box. The 1 cannot go into the cell where there's already a 2. But it also cannot go into the second row because it already has a 1. This leaves only the green cell for the 1. Lastly, there is the *last-in-unit* move, where a digit is filled into the last empty cell of a unit. Reasoning can be either cell- or digit-based in this case.

5.2.1 Methods

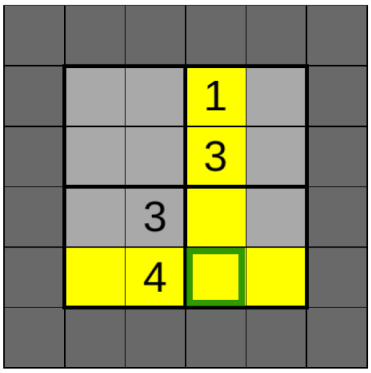
Participants The study was conducted online. Students of psychology and cognitive science received course credit for participation. 32 participants (21 female, 11 male) completed the experiment. Their age ranged from 18 to 31 years (mean: 21.8, SD: 2.69). They indicated their prior experience with Sudoku on a discrete 5-point scale. Two participants chose "more than once a week", three "once per week", ten "once per month", nine "once per year" and eight "less than once per year".

Procedure At the beginning of the study the rules of Sudoku and the interface were explained. The instructions for each trial were "We would like to understand how you proceed while solving the puzzle. Please mark the relevant digits and cells for each step." Highlighting was done by clicking on a cell. Additionally it was possible to highlight a entire unit (4 cells in the same row, column or 2x2 box) at once by clicking on buttons surrounding the board. See Figure 5.1 for a view of the interface. On every move when the participant tried to fill a digit into a cell, the program checked that the inference was licensed by the highlighted cells alone, i.e., no other constraints are needed to uniquely determine the digit that belongs into the cell. If a participant filled a cell and the highlighted cells did not license that move, a pop-up window appeared on the screen and informed the participant that the highlights were not sufficient and the entered digit was not accepted. This check prevented participants from ever entering a wrong digit into a cell and therefore only correct digits (with appropriate highlights) were accepted. We never explained the different strategies to our participants and left the instructions intentionally rather vague to avoid biasing the actions of the participants.

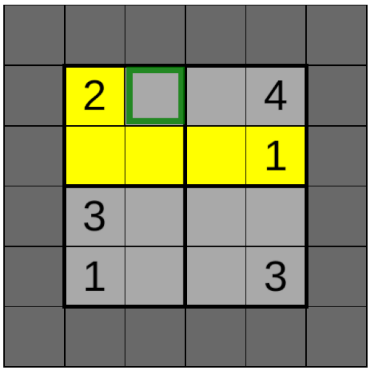
The study comprised 60 4-by-4 Sudokus presented in a different random order to each participant. We recorded overt responses, mouse movements, and the response time for each entry measured from stimulus onset.

Stimuli We used 10 different 4-by-4 Sudokus as bases and generated 6 isomorphs for each of them. Isomorphs were created by switching rows within a box (i.e., 1 with 2 or 3 with 4) or blocks of rows (1,2) with (3,4), same for columns. The digits in Sudoku are just symbols, their numerical value is irrelevant. Our base Sudokus were defined with letters as symbols. Each isomorph had a random translation from letters to digits (while using only the digits from 1 to 4). The Sudokus had between 4 and 8 given digits, with an average of 5.8. It is possible to solve all Sudokus of our study by exclusively using digit- or cell-based rules.

Labeling the data Even for the small 4-by-4 Sudokus, several different inference rules can be used to fill in digits. Based on an earlier think-aloud study and other literature (Behrens et al., 2023; Lee et al., 2008;



(a) Cell-based reasoning



(b) Digit-based reasoning

Figure 5.1: Examples of the interface. Only the 16 central cells belong to the puzzle. The darker gray ones surrounding them provide convenient highlighting of entire Sudoku units (the adjacent row, column or box). The active cell with a green frame will be filled when a digit is entered.

Qin et al., 2012) we isolated seven rules that are commonly used in this setting (these rules are called tactics in previous work). Three of those are digit-based and only differ in whether a box, column or row is the base unit (corresponding labels are *digit-box, digit-col, digit-row*). Cell-based reasoning gets the label *cell-complex* (we do not distinguish which combination of units was used exactly). The remaining three labels could be either cell- or digit-based and are used when the last value of a unit is filled (corresponding labels are *last-in-box, last-in-col, last-in-row*). We implemented these rules in a Prolog program that takes a partially filled Sudoku puzzle as input. It returns a list of applicable rules for each empty cell.

Whether the justification of an entry was digit-based or cell-based (or potentially both) was already determined during the experiment, when checking whether the provided highlights of cells were indeed sufficient to deduce the entry. The exact label for each move was determined off-line after the experiment. The highlights of the participants were used for disambiguation when several rules could have led to the entry. The heuristic we use for determining the base unit is based on the number of highlights per unit. The unit with the most highlighted cells is assumed to be the base unit for reasoning.

When, for example, several digit-based rules are applicable, the unit with the most highlighted cells was chosen as the base unit. An example can be seen in Figure 5.1 (a). When the 1 is filled into the cell with the green frame, it has to be digit-based reasoning. Looking at the neighbors of the cell, two possible digits remain (1 and 3), therefore cell-based reasoning does not lead to a unique digit and cannot be applied. But when looking for where to place the 1, the cell remains the only option for all three containing units (box, row, column). The placement of the highlights in this example strongly suggest that the base unit used was box because three cells of the box are highlighted, whereas only one cell of the row and one cell of the column. The label for this move would thus be *digit-box*.

Some participants highlight only a few filled cells individually, without highlighting the entire unit. In these cases it might be impossible to say what base unit was used, only the digit-based nature of the move can be deduced for certain. In these cases we label the move with *digit-?* without a specific unit. This is still valuable information that can be used in fitting the choice-models as it excludes more than half of the rules.

The *last-in-unit* labels were given to a move when all highlights are within one unit. The basis unit is easily determined in this case. As we do not distinguish between the different combinations of units that could be used for *cell-complex* reasoning, we do not need highlights for disambiguation here.

5.2.2 *Results*

The mean solution time per puzzle was 44 seconds (SD: 26). The shortest trial took 10 seconds, the longest 384 seconds. The first

few trials took a little longer in this experiment, suggesting that the participants needed a while to get familiar with the interface. After about 5 trials the response times stay mostly constant.

With the defined disambiguation rules, 89% of the data could be labeled precisely. 4% of the moves could not be labeled at all as the highlights were sufficient for the move, but not clearly matched by any pattern. It is for example possible to highlight all cells on the board. In this case the relevant information will be highlighted, but in such an unspecific way that it does not tell us anything about the reasoning process of the participant. Another 7% of moves were clearly digit-based, but the base unit was ambiguous. For a detailed breakdown of occurrences of each label please refer to Table 5.1.

Table 5.1: How often each label occurred in the click-reasoning experiment.

Label	relative	absolute
last-in-box	0.230	4508
last-in-col	0.210	4111
last-in-row	0.186	3651
cell-compl	0.044	862
digit-box	0.164	3216
digit-col	0.030	593
digit-row	0.030	584
digit-?	0.068	1335
unclear	0.037	724
total		19584

5.3 Statistical modeling of rule selection data

5.3.1 Choice models

In order to statistically estimate the probability that one inference rule is chosen over another, we use choice models. Research on how people choose from a set of available options has a long tradition, both in psychology and in economics (Luce, 1959; McFadden, 1973). While people will not always choose the same option in recurring situations, they still show preferences in their probabilistic choices. Hence, if repeated choices can be observed, one can statistically infer the underlying preferences. In theory, one could group the data by context and then estimate the choice probabilities for each context separately. However, in our data a context recurs only 3.5 times per participant on average, rendering this idea impossible. Using choice models, we can still estimate these probabilities. This is possible because such models make assumptions about how the choices in different contexts relate to each other (Train, 2003).

Bradley-Terry-Luce model The simplest choice model is the Bradley-Terry-Luce (BTL) model (Bradley and Terry, 1952; Luce, 1959). Each potential option x is assumed to have a positive weight $u(x)$. These weights formalize a participant's preferences for the different options. The probability of selecting option x from a choice set A is equal to the weight attached to this option divided by the weight of all other items in the choice set:

$$P(x, A) = \frac{u(x)}{\sum_{y \in A} u(y)}. \tag{5.1}$$

Obviously, the larger the weight of option x, the higher the probability that it will be picked from the set of available options A. Importantly, the probability of choosing x also depends on the other options available in context A. The bigger A and the more attractive the other options in A, the lower the probability of choosing x. Note that the unit of u is arbitrary and cancels.

Elimination-by-aspects model In circumstances where some of the choice options are similar, a fundamental assumption of the BTL model is likely to be violated: Independence from irrelevant alternatives (IIA). In the BTL model the ratio r_{xy} of how often option x is chosen over option y is the same for all choice contexts (divide the two choice probabilities in Equation 5.1 and the denominator cancels). In particular, if you are given the choice between x and y, the introduction of a third option z should not change the ratio r_{xy}. However, if z is similar to x or y, it usually does (Debreu, 1960).

The elimination by aspects (EBA) model can deal with situations where the choice options are similar and the IIA assumption is likely to be violated (Tversky, 1972). In the EBA model, each option is represented by a set of aspects. The decision is a multi-step process, focusing on a single aspect in each step. When an aspect is selected, only options that have this aspect are kept for further consideration. All other options are eliminated from the choice set. This process continues until only one option remains. Or if several identical options remain, one is picked at random with equal probabilities. In the EBA model the aspects of the options have associated weights, not the options themselves (as in the BTL model). The probability to select a certain aspect for consideration is proportional to the weight associated with this aspect. A highly weighted aspect is thus much more likely to be decisive than a minor one. The choice probabilities for the EBA model can be computed recursively by

$$P(x, A) = \frac{\sum_{\alpha \in x' \setminus A^0} u(\alpha) P(x, A_\alpha)}{\sum_{\beta \in A' \setminus A^0} u(\beta)} \tag{5.2}$$

where x' are the aspects of the chosen item, A^0 is the set of aspects shared by *all* items in the context, $u(\alpha)$ is the estimated weight of aspect α, A_α is the subset of A consisting of all items that have aspect α and A' are the aspects of all items in the context. If no aspects are shared between any of the options, the EBA model reduces to the BTL model.

5.3.2 *Three choice models for rule selection*

We can now adapt the above choice models to the case of rule selection in Sudoku: The options available for choice are cell-rule combinations. In a puzzle with four open cells, there might be five rules per cell available to fill them. In this case the choice set would consist of 20 options, only one of which the participant chooses to fill out a cell. We do not expect participants to be consciously aware of all available options. They probably choose the first cell-rule combination they notice as possible. Still, choice models are a good way to capture the preferences of the participants. The models can make context-dependent predictions of the next action of the participant. Given a configuration of the board (plus the previous move for our full model), which cell-rule combination is most likely to be picked by a specific participant? Using choice models, it is possible to compute

the probabilities for all available options in the context. Of course, it is also possible to compute the marginals: the probability for a certain rule or for each of the empty cells.

Model 1: Rules only (M1) The basic model is defined as a BTL model and only knows about the applicable cell-rule combinations in a specific situation and tries to predict the probability of choosing each of them. In our experimental data the choice context, i.e., all applicable rules in a situation, can be determined automatically, with the help of our Prolog program. For the cell that was filled, 4.39 rules were consistent with a move on average. However, for the entire board (i.e., all empty cells before the move was made), 15.52 rules were applicable on average. As we only identified seven rules, this means that most rules are applicable in several places for many board configuration. We assume that if a rule is applicable in several places, each rule application is a different option. We further assume that the weight of the option is only determined by the rule and not by where on the board it can be applied.[1] The actual choice is given by the labels that were identified in the protocol analysis. All moves are included in the fitting and testing of the models. This includes those moves with no or only incomplete labels. When no label for a move is given, we marginalized over all available options for that cell in that context. Incomplete labels specify a subset of the rules that might have led to the filling of the digit. They reduce the number of possible labels from seven to some smaller number and are still valuable information. In this case marginalization has to be done only between the potential labels, all other rules can be ignored.

[1] Of course, we expect there to be spatial biases but here we are not interested in them and trying to estimate them would increase the number of free parameters massively.

Instead of estimating u directly, we estimate the log weights $v(x) = \log(u(x))$ to ensure positivity of u. Furthermore, for the base model, we assume that the choices on all M moves a participant makes are independent. With x_i being the rule that was chosen on move i, and A_i being the set of applicable rules the overall negative log likelihood for a participant is

$$NLL_{BTL} = -\log\left(\prod_{i=1}^{M}\frac{e^{v(x_i)}}{\sum_{y\in A_i} e^{v(y)}}\right) \tag{5.3}$$

$$= -\sum_{i=1}^{M}\left(v(x_i) - \log\left(\sum_{y\in A_i} e^{v(y)}\right)\right). \tag{5.4}$$

As this function is convex we can easily find the unique optimum for v by numerical optimization. It can, however, happen that a participant used a rule every time it was applicable or never at all. In these cases the v that optimize the negative log likelihood tend to plus or minus infinity. To regularize the solution we added a zero-mean Gaussian prior for v and minimized the negative log posterior using a standard package for optimization (Virtanen et al., 2020, scipy.optimize with BFGS). The prior also has the benefit of implicitly choosing the arbitrary unit for u. We chose the variance of the Gaussian prior such that it is reasonably uninformative given the precision of the data.

Model 2: Rules and cell aspects (M2) The IIA assumption of BTL probably does not hold for our Sudoku contexts. It is likely that participants systematically scan the board for filling opportunities, for example row by row or box by box. Thereby they focus their attention on one part of the board at a time. Imagine a situation in which three cells can be filled with the same rule: one cell in the first row that is otherwise full and two cells in the third row with the two other cells of the row being filled, too. Assume that a participant has decided on using this rule. The rules-only model (M1) would then assign all three cells the same probability of $1/3$, as they all involve the same rule and therefore each cell-rule combination has the same weight. This, however, is not plausible (Debreu, 1960, cf.). If the participant first randomly picks a row with equal probability and then a cell within the row, the cell in the first row will have a probability of $1/2$ and the other two cells have a probability of $1/4$ each. These choice probabilities are much more plausible, especially as cells in more constrained units with fewer open cells are chosen more frequently than cells in relatively empty units.

We can capture these statistical regularities by adding locational aspects to the rule options and using the EBA model instead of BTL. As in M1, each cell-rule combination has one aspect that represents one of the seven rules. In addition, we represent each cell by the row, column and box they belong to (row and column together already uniquely identify each cell, but we want the very salient unit of boxes to be represented, too). To keep the number of free parameters reasonably small we restrict the model to using just one weight per unit type. There is one weight parameter for the four rows, one for the four columns, one for the four boxes. Hence, we do not model any spatial biases (e.g., a preferences for row 1 over row 2), but only a bias to restrict the selection to one type of unit (boxes rather than columns for example).

While participants who prefer the cell-based rules first need to select a cell before they can apply a rule, participants who prefer the digit-based rules have to select a digit first. Hence, they will often scan the board for cells to fill in a specific digit. In this way they will introduce statistical regularities into the cell-rule choices that depend on the digit rather than the unit. We therefore also added an aspect for the digit to be entered into a cell. In order to keep the number of free parameters small we assume that each digit has the same weight and there are no biases in picking one digit over another. The total number of weights to be estimated is thus 11: seven for the rules plus four for the aspects row, column, box and digit.

In summary, each option is represented as a set of aspects that describes a cell-rule combination. Consider the following example:

```
{'last-in-col', 'col-1', 'row-2', 'box-1', 'dig-4'}
```

The first aspect is the name of the rule, the others specify the digit that is the correct solution for the cell and the cell location via row, column and box. The complete context to choose from is a list of

sets. The same cell can often be filled through several rules, in which case there will be several options in the choice context with the same cell-aspects that differ only in the rule aspect.

As before, we optimize the log-weights $v(x) = \log(u(x))$—but this time for the aspects in an EBA model—to ensure positivity.

$$NLL_{EBA} = -\log\left(\prod_{i=1}^{M} \frac{\sum_{\alpha\in x'\setminus A^0} e^{v(\alpha)} P(x, A_\alpha)}{\sum_{\beta\in A'\setminus A^0} e^{v(\beta)}}\right) \tag{5.5}$$

$$= -\sum_{i=1}^{M} \log\left(\frac{\sum_{\alpha\in x'\setminus A^0} e^{v(\alpha)} P(x, A_\alpha)}{\sum_{\beta\in A'\setminus A^0} e^{v(\beta)}}\right) \tag{5.6}$$

This formula does not simplify as nicely as in the BTL case. We could not prove that the function is convex. Nevertheless optimization worked well and led to stable results, which we ensured by optimizing the values repeatedly with different initial values. We use the same zero-mean Gaussian as above to regularize the solution and minimize the NLL in the same way as for the rules-only model.

Model 3: Full model with serial dependencies (M3) So far we assumed that the choices in each move are independent of each other. However, there are obvious serial dependencies between moves. In the think-aloud study described in section 2.2 we saw that participants often follow up on information they just generated. For example, they often fill in cells in the same unit as the one they just filled before. Participants who prefer digit-based rules on the other hand, often fill in the same digit as they did in the previous move. We can model these statistical regularities across trials by temporarily increasing the weights of the aspects corresponding to the previous entry (row, column, box, digit). The factor by how much these aspects should be boosted is a free parameter of the model. For example consider the situation in Figure 5.1 (a). When a 1 is entered in the cell with the green frame (row 1, column 2), for the next move the aspects *row-1*, *column-2*, *box-1* and *digit-1* will be boosted by multiplying their weights by a factor.

The total number of free parameters to be estimated is now 12: seven for the rules, four for the aspects row, column, box and digit, plus one for the boost factor. Note that the three models are nested. The rules-only model, M1, has the seven free parameters for the rules. For M2 we then add the distinguishing aspects for the cells and four free parameters. Lastly, for M3 we add the boost parameter in order to capture serial dependencies.

5.3.3 *Model fits*

We fit the three nested choice models to each participant. All three models are well calibrated. The probability assigned to a potential move by a fitted model reflects very well the actual percentage of times such a move was selected by the participant during the course of the experiment. The fits of the models for different participants

might differ (see Table 5.2), but they all accurately reflect the probabilities for the selection of different options. Figure 5.2 shows a summary plot of the calibration of all three models over all participants.

All three models have weights for the rules, some add additional weights for the cell aspects and for momentarily increasing the focus on aspects that correspond to the previous entry. We fit the log weights (v), but report the back-transformed weights (u). As mentioned above, the absolute value of the weights is not important, only the ratio between them matters. As the absolute values are not meaningful, we normalize them such that the weights for the rules sum (without the cell aspects and the boost) to one. In this way the parameters of all three models can be compared to each other. For most participants, the resulting normalized weights for the rules are almost indistinguishable for the three models. They are not identical but the differences are smaller than 0.05 for 27 out of 32 participants. Only five participants have a deviation larger than 0.05 between the normalized weights of the three models. All of these have in common that they have at least one of the weights for the cell aspects unusually high. The similarity in the weights for the rules in the three models shows that the additional aspects in the M2 and M3 model are relatively orthogonal to the rules.

The weights found for the rules by the choice models differ in important ways from the pure counts of labels. They are shown together for some participants in Figure 5.3. The frequency with which each rule was used by the participant is depicted by the gray bars. The estimated weight of each rule is shown by the black bars. There is some correspondence between the bars but one can also see notable differences. Whenever the black bar exceeds the gray bar this means that the participant highly valued this rule and chose it very frequently compared to the availability. A gray bar considerably higher than a black bar shows that this rule was used to a considerable degree, but has been available even more than that: Compared to the availability it has not been chosen that much. This can be clearly illustrated with participants 06 and 32 (top right panels of Figure 5.3). These participants applied the *cell-complex* rule in almost a quarter of all their moves. However, this rule is applicable very often, and taking the availability into account shows that it was by far not the most preferred rule for the participant. The three *last-in-unit* rules are used with a similar frequency as *cell-complex* but they were less often available. These participants seem to have a clear preference for the *last-in-unit* rules over *cell-complex*, a fact that would not be visible from raw label-counts alone.

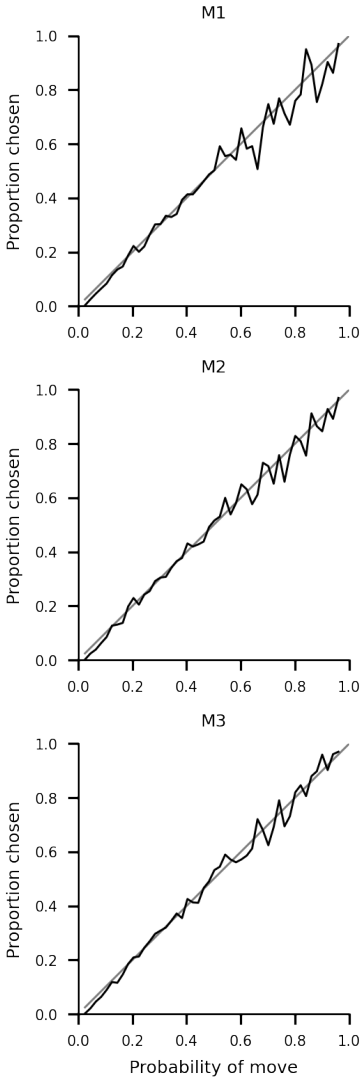


Figure 5.2: Calibration plots for all three models. Due to the smaller number of predictions of high probability the calibration gets less precise.

Model 1: rules only (M1) There are seven rules and corresponding weights to estimate for each participant. All participants have one to four favorite rules with large weights (the average is 3), all other rules get small weights with less than 4% of the overall mass. Most participants get large weights for the three *last-in-unit* rules.

Model 2: Rules and cell aspects (M2) The additional aspects (3 location, 1 digit) generally get weights that are much smaller than the most favored rules, but bigger than the least favored rules. Only three participants (a subset of the five who have very different rule-weights in different models as mentioned above) have single cell aspects (box or digit) that are bigger than the biggest rule aspects. The little overall mass of weight put on these aspects for the other 29 participants suggests that the applicable rules are much more important for the selection of the next move than the cell aspects. Still, adding these aspects generally improved the fit (see model comparison below).

Model 3: Full model with serial dependencies (M3) The cell aspects vary more between the M2 model and the M3 model than the rule aspects. Even when multiplying the cell aspects by the boost factor, all but the three participants mentioned above have larger weights for the rules than for the cell aspects.

For M3 the boost parameter has a value between 2 and 10 for all participants, the mean is 5.1 (SD=2.4). This confirms that strong serial dependencies are present in all our participants. When a specific row is five times more likely to be picked, this significantly influences the predictions of the model. Even for a participant with low row-focus in general, the most recent row will more likely be picked than any other row, given that the same rules are applicable in them. Specifically, a boost value of 5 means that when two options are equal in all respects except for whether they are in the same row that was filled by the previous move or in another row, the one in the same row will be picked in 5 out of 6 cases (83%). This is because in EBA only the aspects that differ between two options matter for the choice probability. And when the weight u of one of the row-aspects is multiplied by a boost factor of $b = 5$, that means the probability that the corresponding item will be picked is

$$P(\text{item in boosted row}) = \frac{bu}{bu + u} = \frac{b}{b + 1}. \tag{5.7}$$

5.3.4 *Model comparisons*

We performed five-fold cross-validation to test which model performs best and to ensure they do not overfit the data. The data of each participant was randomly split into five parts. Each part acted as the test set once while the model was trained on the other 4/5 of the data. The models are evaluated based on the mean log-likelihood of the test sets. In order to be able to judge the absolute values of the log-likelihood we also implemented a baseline model. It assigns the probability of $1/|A|$ to each option in context A for each choice situation. Compared to this baseline, all three models improve the fit immensely for all participants. M2 is better than the simple M1 model for 24 of the 32 participants. This proportion is bigger than what we would expect if there was no systematic difference in fit.

According to a sign-test this difference is unlikely to occur by chance ($p = 0.0035$). For all participants the full M3 model fits best. This result is very unlikely to occur by chance ($p = 0.5^{32} \approx 2.3 \times 10^{-10}$)

We report the cross-validated average negative $\log_2$ likelihood for each participant and model in Table 5.2. We use base 2 instead of the natural logarithm because it is easier to interpret. It can be thought of as bits of information for each decision and conveys the number of options per trial the model guessed between. The baseline of about 3.8 corresponds nicely to the roughly 15 options per trial on average ($2^{3.83} = 14.2$). If a model reduces this number by 1 it means that the effective choice set from which to guess was reduced to half the original size.

When averaging over all participants the difference in the $\log_2$ likelihood per move between the simplest M1 and the M2 model is 0.033, which sums to 20.196 for the entire experiment (0.033×612 moves $= 20.196$), which in turn means, that the data are $2^{20} = 1,048,576$ times more likely under the more complex model. The difference between the M2 and the M3 model is even bigger: 0.071 per trial or 43.452 for the entire experiment, meaning the data are 2^{43} times more likely under M3 than under M2.

The three models described above are nested: when setting some parameters to zero the more complex ones could describe and predict the exact same patterns of data as the simpler ones. Hence, likelihood-ratio tests can also be used for model comparisons. Likelihood-ratio tests and the Bayesian information criterion (BIC) largely confirmed the results we got via cross-validation. Likelihood ratio tests had the exact same numbers as the cross validation (for 24 out of 32 participants, M2 is better than M1, for all 32 participants the M3 model is best). The BIC is a bit more critical in the comparison of the M2 model to M1. Only for 18 participants the more complex M2 model has a better BIC value than the simpler M1 model. This is a ratio that could easily be due to chance ($p = 0.2983$ according to a sign test). In the comparison of the M3 model to the others, BIC agrees with the other tests, the M3 model has the best BIC value for all participants.

Even the best fit of the most predictable participant is only at 1.55 bits per trial on average, which corresponds to guessing from a bit under 3 options. On average the M3 model has 2.55 bits per trial, which corresponds to guessing from about 5.8 options. Of course there are many situations when the model assigns a high probability of 80% or more to the actually selected move. On the other hand, at the beginning of a puzzle for example, there is bigger uncertainty in the model. Participants with the worst overall fits often placed the highlights in a way that makes exact labeling impossible. When they have many *digit-?* labels for example, the fits cannot be good because the model cannot learn which basis unit should be used. See also subsection 5.3.6 for a discussion of the (in-)consistency of behavior of different participants.

	M1	M2	M3	baseline
01	2.785	2.738	**2.554**	3.832
02	2.932	2.906	**2.775**	3.885
03	2.416	2.361	**2.285**	3.805
04	2.960	2.968	**2.886**	3.808
05	2.780	2.763	**2.734**	3.876
06	2.113	2.108	**2.062**	3.868
07	2.605	2.576	**2.525**	3.859
08	2.705	2.641	**2.603**	3.811
09	2.586	2.563	**2.477**	3.833
10	2.620	2.593	**2.584**	3.826
11	2.622	2.608	**2.552**	3.840
12	2.750	2.708	**2.698**	3.867
13	1.618	1.601	**1.554**	3.801
14	3.032	3.031	**2.952**	3.795
15	3.688	3.582	**3.449**	3.896
16	2.425	2.437	**2.369**	3.794
17	2.807	2.797	**2.728**	3.825
18	3.410	3.331	**3.204**	3.917
19	2.250	2.224	**2.089**	3.795
20	2.907	2.917	**2.862**	3.763
21	2.482	2.214	**2.068**	3.941
22	2.540	2.555	**2.528**	3.808
23	2.618	2.556	**2.540**	3.856
24	3.479	3.484	**3.445**	3.816
25	2.235	2.222	**2.138**	3.823
26	2.411	2.339	**2.308**	3.856
27	2.819	2.841	**2.806**	3.836
28	2.408	2.388	**2.305**	3.779
29	2.634	2.592	**2.486**	3.821
30	2.850	2.851	**2.827**	3.836
31	2.454	2.404	**2.305**	3.835
32	2.049	2.052	**1.966**	3.865
mean	2.656	2.623	**2.552**	3.836

Table 5.2: Average negative $\log_2$ likelihood per move for the three different models as found by cross-validation together with the baseline model. The best model for each participant is printed in bold. For the entire experiment the log-likelihood is 612 times bigger, as there were that many filling events per participant.

5.3.5 *Clustering of participants*

Even though each participant has their unique profile of best fitting weights, some seem to have quite similar preferences for rules and units. For a more formal assessment of similarity, we clustered the participants with k-means (we used the algorithm provided by scikit-learn 1.3.0 (Pedregosa et al., 2011)). We can base the clustering on the weights of each of the three models. We decided to use the best-fitting M3 model as the basis for clustering. A good number of clusters is four for our group of participants. With more clusters, some clusters contain only a single participant. We repeatedly started the clustering with different random seeds. The best-fitting partition for four clusters has the following four clusters:

- Cluster 1, 13 participants
- Cluster 2, 11 participants
- Cluster 3, 5 participants
- Cluster 4, 3 participants

In Figure 5.3, two exemplary members of each cluster are shown. About 40% of our participants are in cluster 1. These participants have a strong preference for the three last-in-unit rules. They use other rules when they have to, but none of them gets a strong weight. Participants from the next biggest cluster 2 also have a preference for the three last-in-unit rules, but additionally they clearly prefer *digit-box* over the other alternatives. Members of cluster 3 prefer to use the three last-in-unit rules, too. When these cannot be applied, they resort to *cell-complex* rules, instead of any of the digit-based ones. The last and smallest cluster (only 3 participants) is most unique: these participants are very digit- and box-based in their approach. The two rules they apply the most are *last-in-box* and *digit-box*. They do not use any other rule at all.

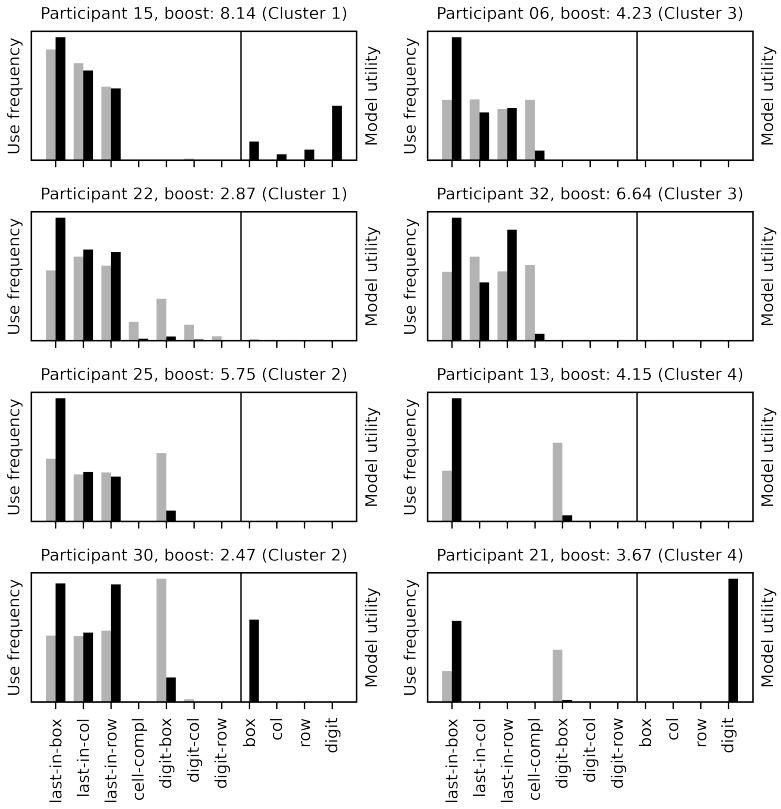


Figure 5.3: All participants were clustered based on the weights of the M3 model. Representative participants of each cluster are depicted. Model weights (black bars) and rule frequencies (light gray bars) are plotted side by side for comparison. The bars to the right of the dividing line represent additional aspects that do not correspond to rules.

5.3.6 *Consistency of behavior in similar situations*

As mentioned in the methods section, we presented six isomorphs of ten different seed puzzles to each participant. They thus saw six

logically equivalent puzzles of each base puzzle during the experiment. This enables us to assess the degree of consistency of behavior in similar situations for each participant. Some participants with a few very specific rule preferences will nearly always choose the the same action in a similar situation. Others have much more variable or flexible behavior and behave differently each time they encounter a certain situation. We can compare a participant's behavior to the traces generated by the model. When we limit the analysis to the first few moves per puzzle, it is even possible to exhaustively generate all possible start sequences and calculate their likelihoods according to the model. Generally speaking, the participants with the best fits in Table 5.2 are the more deterministic participants.

A participant with one of the worse fits is participant 02. They are variable in how they start this particular puzzle, two times they use a clearly cell-based start, the other four times a digit-based start (see Figure 5.4). Follow-up moves most often stay in the same unit, but whether row or box are preferred to stay in is unclear since both are used. In one puzzle participant 02 follows up on the digit instead and fills in all instances of 1 first. An additional difficulty for the model is that many of the digit-based labels are under-specified, it remains unclear which unit was the base for the deduction and thus the model has to work with more unspecific data. The very flexible behavior is reflected in the model, too. When scoring all possible first three moves in the puzzle, the 40 most probable moves add up to only 70% of probability mass. Many starting sequences are possible according to the model, but none are very salient or likely.

A much more predictable participant is participant 32. They always start with a complex cell-based reasoning step and generally seem to prefer to stay in the same unit and follow up with only-value rules (see Figure 5.5). According to the model the 40 most probable puzzle starts can account for more than 90% of the probability mass.

5.4 *Discussion*

In this work we used 4-by-4 Sudokus to study probabilistic rule selection in humans. At each moment in the puzzle the participants have on average 15.52 different cell-rule combinations to choose from. While we do not think that they are explicitly aware of all these options, choice models are still a very good tool to describe the statistical preferences for specific options each participant has. We showed that choice models provide a compact and quantitative description of the statistical regularities in each participant's behavior. We expect this form of analysis to be helpful for other discretely labeled data, too. It contains much more nuanced information as compared to simple counts of labels (as used in standard protocol analysis). The high-level quantitative description of preferences of each participant also allowed us to further investigate another influence on the selection of a move at each moment: the dependence on the previous move. We could quantify how strongly participants preferred to stay

within the same unit for their next move.

5.4.1 *The benefits of externalizing thinking*

One important aspect that enabled the statistical modeling done in this article is improved data collection. With the interface we developed we could collect data in an online experiment and label each move automatically. Similar to research on decision making (Johnson et al., 1989; Rieskamp and Otto, 2006) or planning (Callaway et al., 2021), we relied on externalization of the reasoning process to understand the actions of our participants better and gain detailed insights without requiring think-aloud protocols. We developed an interface in which participants not only filled the 4-by-4 Sudokus but also indicated which cells were relevant for each inference. They did so by highlighting these relevant cells by clicking on them. The highlights allowed us to clearly assign a rule label to 89% of all moves. Knowing only which cell was filled at each moment would not have been enough. On average 4.39 rules are applicable for each filled cell. Only about 4% of all filling events took place in cells where only a single rule was applicable and could have been labeled without the highlights. We developed the interface with the highlighting of relevant cells after our experience from a previous study (see section 2.2) that relied on think-aloud protocols for labeling. It would have been a lot of work to transcribe and hand-label this amount of data from think-aloud protocols. While think-aloud protocols are immensely useful in the early stages of a study on problem solving, it is important that research on problem solving moves beyond the traditional analysis of the idiosyncratic behavior of single participants. With the new paradigm we recorded more than 4 times as much data as in the think-aloud study and still reached results much faster. Our adaptation of the Sudoku task might serve as an example for other problem solving tasks.

5.4.2 *Choice models for problem solving traces*

As we expected, the Sudoku data clearly showed that the raw frequency of rule application can be misleading with respect to the relative importance of a rule for a participant. This mismatch is caused by the uneven distribution of rule applicability. Some rules can be applied in several places for most moves, others only in one place every now and then. We are interested in the conditional choice probabilities of rules given the context. But most contexts repeat not nearly often enough in the course of the experiment to count the choices in each context separately (on average every context comes up 3.5 times in our data). Here, we showed that choice models are excellent high-level models to describe how participants select rules in given contexts. These models take into account the context in which a rule was applied but do not need repetitions of the exact same choice situation in order to estimate the rule weights. This makes it possible to find a ranking among the rules and important aspects to describe

each participants' statistical preferences and predict choices in novel contexts. Importantly, a fixed order for rule preference—as Newell and Simon (1972) used in their analyses—cannot match participants' behavior fully, because they usually do not behave deterministically. A probabilistic ranking of the rules is therefore more appropriate. While production systems can easily incorporate probabilistic rule selection (and often do, e.g., in ACT-R), the rule weights are usually not directly estimated from data. Choice modeling provides a sound framework for doing so. We showed that simple BTL-choice models are well suited to model the preferences for the different rules of our participants.

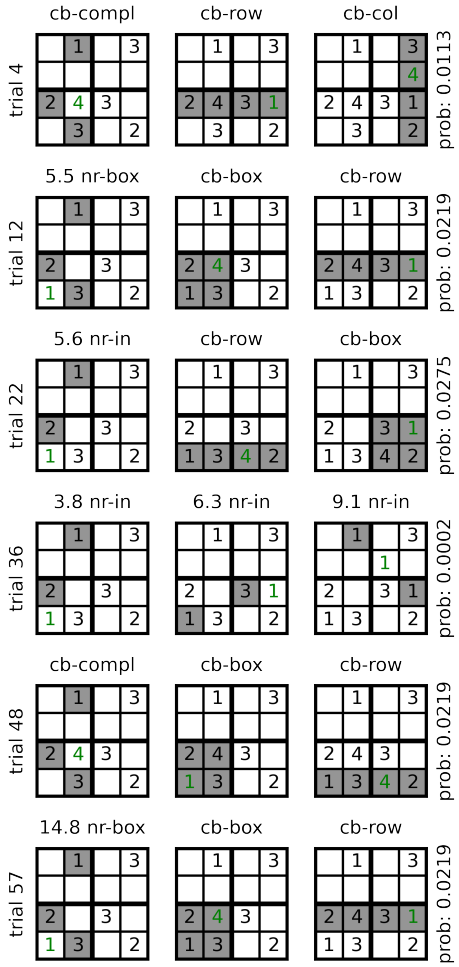
The rather high level nature of our model made it possible to model another factor that influences the choice of the next move besides the preferences for the different rules, namely the dependence on the previous move. We used an EBA model to integrate the influence of several aspects on the choice of the participant. Choice options were cell-rule combinations, where each cell was represented by the aspects row, column, box, and digit. These cell aspects could be boosted, in case they were identical to the previous move. When fitting this model for each participant, we found that the rules are much more important than the cell aspects for the majority of our participants. As rules were often applicable in several places, the cell aspects were still influential: they provide disambiguation between several instances of the same rule. The boost factor, which can be interpreted as an attentional focus, makes sure that an option that shares a unit with the previous move (say it is in the same row) is more likely to be picked than another cell where the same rule can be applied. We found that this attentional focus increased the fit of the model significantly for each of our participants and that its strength differs between participants.

5.4.3 Clustering of participants

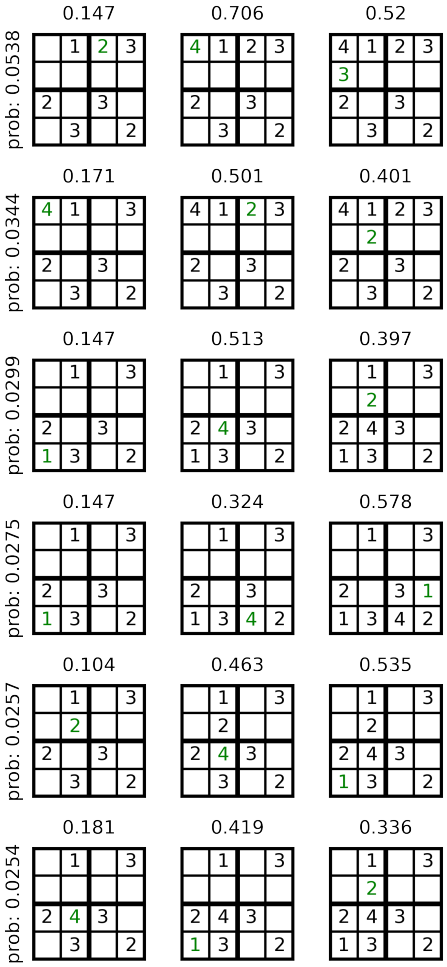
It is true that different participants approach problem solving tasks in different manners. Nonetheless, it is desirable to also find common elements to describe their actions. The compact representation of their preferences via weights for a limited number of aspects makes it possible to compare the profiles of participants and find marked similarities as well as differences. It became clear that some participants behaved similarly. In a cluster analysis we found four clusters in which the participants can be grouped. Finding not only the inter-individual differences but also similarities among groups of participants was possible only because we had a sufficient number of participants. The fact that we were able to process the data with scripts with not much additional processing time per participant made it possible to go beyond the usually small sample size in problem solving research.

5.4.4 *Limitations and outlook*

Choice models can capture statistical regularities in the rules participants apply in a problem solving task. They are, however, not a cognitively plausible model for how participants actually go about selecting cells and rules for a move. The options in the choice models consist of all the cells that can currently be filled. However, the participants cannot know which cell-rule combinations will or will not work before actually trying them. They do not explicitly represent the complete set of options and then choose from it. Instead they have to search for cell-rule combinations that leads to a unique answer. It is likely that participants use heuristics, like looking at units with more filled cells first. In doing so, they will sometimes also try cell-rule combinations that do not lead to an entry. These failures, unfortunately, leave no trace in the observed choices (they will prolong response times though). The statistical regularities that we observe in the data are a result of this partially unobservable heuristic search process. It is remarkable that even without specifying these details of the cognitive process, we can still model the statistical regularities in the behavior reasonably well. In fact, we believe that the statistical analyses we presented here will help us to better understand the underlying cognitive processes. For example, we have seen that the statistical preference for a rule has to be disentangled from the raw frequency that it was used. Also, we could see that participants' preferences for rules come in clusters and these clusters are likely to represent different stable strategies for solving 4-by-4 Sudokus. Future work will try and the extend the EBA model to include search heuristics and learning while keeping the model simple enough to be fit quantitatively to empirical data.

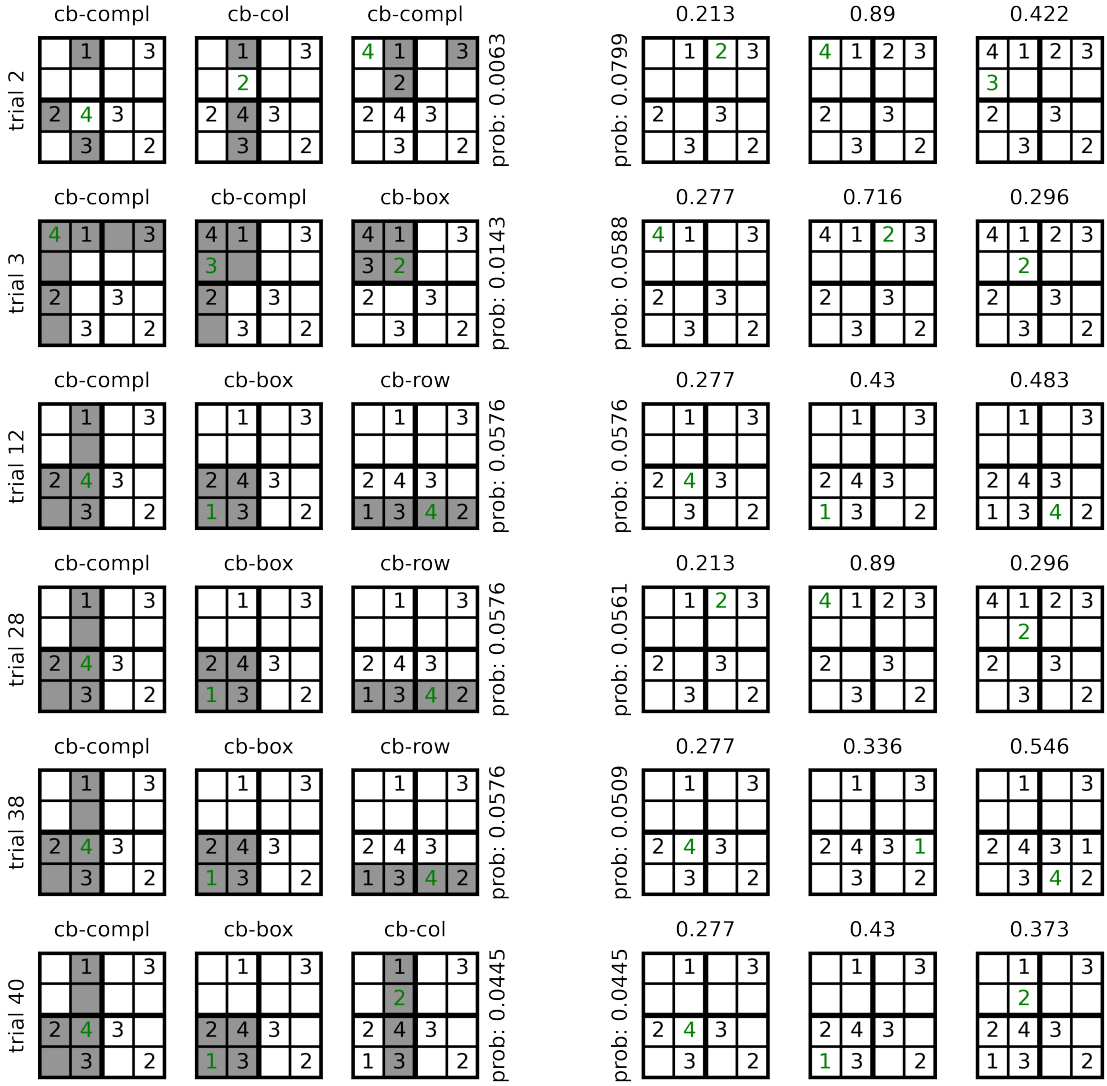


(a) First moves of participant 02



(b) Model of participant 02

Figure 5.4: A participant with very variable behavior: participant 02. (a) The first three moves in all isomorphs of puzzles 'h' of participant 02. The isomorphs are transformed such that they all look the same here. Highlights as clicked by the participant in gray. The newly filled digit is printed in green. The titles of the subplots give the label of the move. (b) The six most probable starts of the puzzle according to the M3 model of participant 02. The probability of the entire sequence is on the left, the probability of each individual move in the title of the subplots.



(a) First moves of participant 32

(b) Model of participant 32

Figure 5.5: A participant with more predictable behavior: participant 32. (a) The first three moves in all isomorphs of puzzles 'h' of participant 32. The isomorphs are transformed such that they all look the same here. Highlights as clicked by the participant in gray. The newly filled digit is printed in green. The titles of the subplots give the label of the move. (b) The six most probable starts of the puzzle according to the M3 model of participant 32. The probability of the entire sequence is on the left, the probability of each individual move in the title of the subplots.

Chapter 6

Discussion and outlook

Throughout this thesis we have used a wide array of methods to study the behavior of people while they solve digit-placement puzzles. We believe that in order to make progress in research within the domain of higher cognition in general, it is necessary to adopt a versatile set of tools and a mix of qualitative and quantitative data. To get a general understanding of the domain, we started with informal introspection (Jäkel and Schreiber, 2013), moving on to think-aloud studies, and then to response-time experiments, we developed more and more detailed models of solution tactics for the task. From the first models, which were relatively abstract production rules for different filling tactics in chapter 2, we progressed to more detailed process models of some of the tactics in chapter 3, which we fit to individual response times in chapter 4. In chapter 5 we increased the time frame our models could describe: By using choice models to select productions, we moved from modeling single filling events to filling the entire puzzle.

In chapter 2 we reported the results of several think-aloud studies. Even though it is a labor-intensive research method and furthermore does not lead to results which are readily usable as the basis for quantitative model fitting, it has an important part to play in research on higher cognition. We found several filling tactics that were relatively similar across different puzzles. All our participants were able to find simple tactics for filling the puzzles without any explicit instructions on how to do so. They displayed relatively stable preferences for specific filling tactics, with more experienced players using different tactics than inexperienced ones. Without think-aloud protocols, we would probably have overlooked the *only-digit-missing* tactic, as we did not anticipate it. Behaviorally it could have been *last-in-unit* as well, but the utterances accompanying the filling events clearly showed a different reasoning tactic. Problem solving approaches, such as an exploration of different problem spaces, as participant S8 in Newell and Simon (1972, chapter 7) did, can only be observed in relatively unconstrained experiments with an open and rich data source such as think-aloud protocols. Whenever we are interested in processes that have a deliberate and conscious component, think-aloud studies have the potential to provide very rich information about them. They allow for relatively open tasks, as the think-aloud

protocols enables us to understand the unique interpretation of the task a participant came up with.

A measure with even higher temporal resolution than think-aloud protocols is eye-tracking. We employed it in addition to the think-aloud and mouse-tracking recording in section 2.4. For us, however, the data from the eye-tracker was not as helpful for the explorative analyses than the other two in combination. The mouse-tracking was well aligned with the utterances in the think-aloud protocol and helped to disambiguate the referents of the utterances. We think that eye-tracking would be more informative with a better and more detailed model of what participants probably do in the task. With the process model we developed for the cell-based and the digit-based trials in Sudoku puzzles, it would now be interesting to check whether the saccades of people doing the task fit the patterns predicted by the model. All in all, we think eye-tracking can be a valuable source of information, but for a task such as ours, it is better suited with good models of the solution tactics, instead of during the first exploration phase.

When one has acquired a good overview of possible tactics participants might employ in a given task, it is reasonable to move on to more controlled experiments in order to find out specific details. It is important to acknowledge the influence of the scaffolding provided by a more controlled experiment. If there is only one specific way in which a participant can answer, they will try to do so to the best of their ability. However, this might not be the first answer that came to their mind or the most natural way to solve the task for them. They were solving the specific task posed by the experiment, which is not necessarily the same as the "natural" one that was the inspiration for the experiment. See for example Straub and Rothkopf (2022) for an argument on why it makes sense to infer the costs participants have and optimize, which might differ from the ones the experimenter sets and expects. A good understanding of relatively "natural" behavior of participants in a relatively unconstrained task environment, helps the researcher to see the changes and limitations imposed by more rigorous experimental control. It is important to keep these in mind when generalizing the results of an experiment.

In the more controlled experiments in chapter 3, we disentangled the effects of the number of required units (NRU) as well as required tactic and the interaction with the task instruction. Already with simple statistical tests on the response times we found significant interaction effects. In combination with a process model, however, the results could be interpreted in much more detail. We showed that an increase in the NRU lead to longer response times, but no significant difference in accuracy. The process model explains this one-sided influence by showing that more processing steps are required with higher numbers of required units, but the burden on working memory does not increase, as the processes can take place in serial order. The difference in response time between congruent cell-based and digit-based trials is also explained by the process model: The

digit-based condition needs many fewer scans and can therefore be answered more quickly.

The statistical model we developed to fit the process model's steps to the individual response time patterns of the participants allowed us to deepen our understanding of the data even further (see chapter 4). We generated parameter estimates for scan durations and intercepts for each participant and could also estimate to which degree they used each tactic. Estimating the tactic use for each participant based solely on the response times was only possible with the help of the process models which we developed from the insights we gained from the analyses of the think-aloud protocols. As a possible next step it would be informative to conduct an eye-tracking study to test whether the eye movements of participants confirm the assumptions of the processing model. The elementary information processing (EIP) regression model we developed in this chapter can estimate the duration of processing steps even if they number of processing steps on each trial is latent. The model is general and can be applied to analyze response time data in light of a processing model and return parameter estimates with a clear psychological interpretation.

Another way in which we expanded our initial modeling was by combining choice models with the production rules for filling single cells (see chapter 5). In this way, we built a production system which can fill a entire puzzle in similar ways as different participants do. In filling Sudoku puzzles, the exact same context rarely repeats but choice models allowed us to generalize over different contexts and estimate the preference weight of each participant for each production. The probabilistic description of most likely choices in each context are an adequate model for the variable behavior of our participants. In order to collect enough data to fit such models, we developed a novel experimental interface, in which participants indicated which parts of the puzzle were relevant for finding the value for the cell they filled in on each move. The filling events alone are mostly ambiguous and do not reveal which reasoning tactic was used to deduce the value. With the additional information, 89% of the moves could be labeled automatically, reducing the time and effort of data preparation by orders of magnitude compared to hand-labeling think-aloud protocols. In future work it could be interesting to connect the two approaches (a detailed process model and a production system) to build a more detailed production system. Such a model would make more detailed predictions about response times of individual participants for specific puzzles. However, further investigation would be needed to understand how these productions are learned in the first place and how preferences for them arise over time. Ideas about how to approach such questions are presented in section 6.1. We observed in our experiments that performance is not always perfect and errors can occur. More detailed models, therefore, should also include the possibility of errors and not only describe perfect execution of solution tactics. Introducing a limited

working memory, for example, might be one way of accounting for errors.

To sum up, we believe that a broad range of methodological approaches is necessary in order to make progress in problem solving research. The work in this thesis is an exploration of possible experimental designs as well as analysis tools and modeling approaches. Their combination has great potential to further improve our understanding of general problem solving tactics.

6.1 *Outlook: Aspects of learning*

We propose digit-placement puzzles as an ideal domain to study learning in problem solving on different levels. When telling novices the rules of such a puzzle, they are quickly able to come up with simple solution tactics. In our own experiments with 4-by-4 Sudokus, all participants solved their first puzzle within just a few minutes. In a study by Lee et al. (2008), beginners managed to place on average two to three digits in a complex 9-by-9 Sudoku within 15 minutes. That means, when starting to work on a digit-placement puzzle, people are translating the constraints into some form of executable rule to find digits they can enter in the puzzle. Over the course of the experiment a speedup could be observed in most of our experiments, indicating a form of learning. The routine participants developed in applying rules could be observed in the utterances in the think-aloud protocols, too. The utterances got shorter and often explicitly indicated routine actions by referring to repetition. Practice leading to routine and faster execution is probably the form of learning that is easiest to study in experiments.

It is also possible to continuously increase the difficulty of the puzzles during an experimental session. We did so in the Straights experiment in section 2.4, for example. In this case, learning is not restricted to developing a routine, but participants are challenged to find new and possibly more complex solution tactics in order to solve the puzzles. Digit-placement puzzles offer a wide range of difficulty: It is possible to devise puzzles that still pose a challenge to very experienced players. It is also possible to introduce new constraints in a puzzle like Sudoku, allowing to place fewer digits at the start. New constraints require participants to find new tactics, integrating already learned approaches with new knowledge to form new and more complex tactics.

Sudoku in particular offers to study the performance of players of different levels of experience. As it is such a popular puzzle, it is not difficult to find participants who regularly solve Sudokus in their spare time. It is thus possible to compare the performance and tactics of different learning stages without having to train the experts in the lab necessarily. Another option would be to give participants practice tasks to do at home.

It is even possible to analyze data of people solving Sudokus without conducting experiments: There are both large data sets (Pelánek,

2011) and smaller, high quality ones by puzzle champions with concurrent commentary by the solvers, explaining their thought processes (youtube.com/@CrackingTheCryptic).

6.1.1 *Preference learning*

Modeling learning in the domain of Sudoku and similar puzzles requires the inclusion of several aspects. One such aspect is the development of preferences for some tactics over others. Preferences might arise through mechanisms of reinforcement learning (RL). In RL an agent learns which actions to take in what state through rewards. It is formalized as a Markov decision process with states and actions the agent can take. Depending on the action the agent takes, the state changes to a new state. The transition to a new state can be either deterministic or probabilistic. With trial and error the agent learns in which states which actions likely lead to a high reward and thus transitions from exploring the state space to exploiting known paths to a reward. In Q-learning the agent learns a table of expected rewards, one entry for each state-action pair. Replacing an explicitly enumerated table of rewards with a deep neural network that approximates the value function has enabled RL to be applied to much larger problems, for example learning to play Atari games from pixel input and the score alone (Mnih et al., 2015).

In ACT-R a learning mechanism called utility learning (Anderson, 2007, chapter 4) is implemented which can give rise to preferences for specific solution tactics. It is very similar to RL. A simplification in ACT-R's utility learning as compared to Q-learning is that utilities are learned for productions (i.e., actions) instead of state-action pairs. This assumes that actions (if they are applicable at all) always have the same utility which is independent of the state of the problem (Brasoveanu and Dotlacil, 2021). With utility learning, an agent will be more likely to repeat actions that lead to a favorable outcome in the future.

If simple tactics are implemented in the language of ACT-R, each of them would need several productions to fire in sequence before a digit could be entered. In this case, the normal ACT-R utility learning mechanism could be applied out of the box. Paths leading to an entry with fewer productions would be more strongly reinforced than ones requiring more productions to fire. Here, we used more macroscopic productions (see Code 5.1) which all lead directly to an entry. In this sense, all productions are equally good and no basis for preferring one over the other is given. It would be possible, however, to combine the EIP steps of chapter 4 with these productions. The productions would then return a count of EIP steps each time they were applied to a specific puzzle situation. The number of EIP steps could then be used as basis for the temporal discounting factor, leading to a preference of productions which need fewer EIP steps over time. Unsuccessful attempts to apply a production to a specific puzzle situation would need to lead to a negative utility update.

Such RL mechanisms should give rise to preferences for groups of productions that optimally play together. It would be interesting to compare these preferences with the preference clusters we found in chapter 5. RL might be a good explanation for how such preferences form and could provide additional explanations as to the likely preconditions for different preference-patterns.

6.1.2 *Chunking*

Another aspect of learning concerns the sped-up and in some cases automated execution of known procedures. Procedures which for a beginner require conscious effort can often be carried out much faster and without much thought by highly trained individuals. At least some of it can be explained by chunking: growing building blocks of perception, cognition, and action. Over time, events that often occur together in the environment will be grouped and processed as single unit instead of in individual parts. In the Towers of Hanoi puzzle, for example, a participant developed the concept of *moving an entire pyramid of disks* (Anzai and Simon, 1979). This chunk of actions made planning of long-term goals for her much easier, as it lifted the burden of enumerating the entire sequence of moves. And in chess, masters are able to remember more pieces from meaningful positions (ones that actually occurred during a game) than from random arrangements of chess pieces on a board, presumably also because they can perceive some groups of pieces as unitary chunk (Chase and Simon, 1973). Some common patterns in chess even have their own names, like *castled-King position* or *pawn chain*. Such concepts help in the game when the player recognizes a pattern: They might immediately see some of the strengths and weaknesses of the position, without having to consider each piece and its options in isolation. Some chunks form in conscious effort, others arise automatically. When solving a task together in collaboration, humans quickly invent names to refer to recurring situations (Angerer and Schreiber, 2019). A name helps to refer to a complex concept and might also help solidify and consolidate a chunk in the mind (Gentner, 2003). It has already been shown for infants and young kids that labels help to find commonalities between objects and understand more complex concepts (Althaus and Plunkett, 2016; Gentner et al., 2021).

Procedural chunking is implemented in ACT-R as production compilation (Anderson, 2007; Taatgen et al., 2006). When several productions are activated in direct succession, a new production, combining their effects can be created. An example for this might be the addition of two numbers, say 2 and 5. There is a general production which can take two variable digits as inputs and look up their sum in long term memory. A second production can harvest the returned answer from memory and output the answer. If these two productions fire, a new, very specific production can be created, which answers 7 immediately upon seeing the digits 2 and 5 in a summing context. The utility of such a new production is initially

very weak. Only if it is encountered very often will it become the dominant production to use. As the compiled production does not require a memory retrieval, it gives the answer much quicker than the two productions. With a lot of practice the answer to the specific combination of digits thus becomes automated. Of course, only sums that have been encountered very often will be answered in such automated fashion, others have to be retrieved from memory and will be answered more slowly.

Similar production compilation mechanisms could be used in Sudoku models to create productions that are very specific to some pattern on the board. See for example Figure 6.1 for a possible pattern in a Mini-Sudoku. Every time the same digit appears in the two opposing corners of the puzzle, the other two instances of the digits have to appear on the other diagonal, but in the two inner most cells. Another pattern might be called *windmill*, see Figure 6.2: if two outer cells, one directly above and one directly below the middle bar contain the same digit, the other two instances of the digit have to appear on the outer periphery of the puzzle as well, directly right and left of the vertical middle bar. Also in 9-by-9 Sudokus, there are named patterns that regular players typically know (*hidden pair, naked pair, X-wing, swordfish...*). The deductions licensed by such patterns could be made with a more general rule, but with a lot of practice such patterns might become salient and lead to almost automated responses, lifting the burden of thinking through the reasoning behind it again. Rules for such specific patterns can well be created by a mechanism like ACT-R's production compilation. In language modeling, the trade-off between re-computation and storage of new rules has been modeled with the help of a fragment grammar, a probabilistic formalism that stores frequently co-occurring strings such as idioms in their own building blocks (O'Donnell et al., 2009).

Chunking might be a necessary precondition for participants to find new and more complex solution tactics. By automating some simple steps, it frees the participant from reasoning through simple processes such that they have the free mental capacity to think of new tactics. In a two-player game, more experienced players can plan more steps in the future than novices do, which improves their performance (van Opheusden et al., 2023). It was already remarked by Roberts and Erdos (1993) that participants who struggle the most with a task are the ones who do not have the resources to develop new and maybe simpler solution tactics. Simple addition is a case in point: When children learn to add single digits, they usually start with counting tactics to do so (Siegler and Jenkins, 1989). Starting with counting all numbers in the sum they progress to use more efficient tactics and eventually have enough practice that they can simply retrieve the answers to the most common questions. Retrieval is much faster than counting and is very helpful when performing more complex mathematical operations such as adding multi-digit numbers with paper and pencil. However, children who count very slowly and make many mistakes there, also are the ones who are

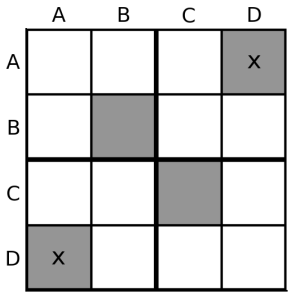


Figure 6.1: A possible pattern in Mini-Sudokus: *Diagonals*. Two digits of the same kind are on the outer ends of one diagonal, the other two need to be on the inner ends of the other diagonal.

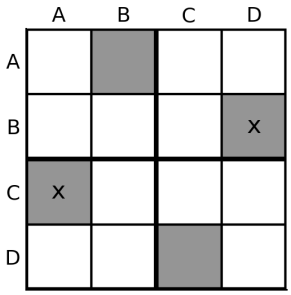


Figure 6.2: A possible pattern in Mini-Sudokus: *Windmill*. Two digits of the same kind are on just above and below the horizontal middle bar at the outer periphery of the puzzle, the other two then need to go right and left of the vertical middle bar on the outer periphery as well.

least likely to develop retrieval tactics for simple addition problems, presumably *because* of their slow counting (Hopkins and Lawson, 2006).

The causal link between the slow counting and problems to learn new tactics for addition tasks is hard to establish in studies with children because it interacts with many other factors. With a task such as Sudoku it might be possible to study it in the lab: Enumerating all digits is a frequent sub-routine in order to find a digit which is still allowed in a specific cell, for example. One could replace the digits with arbitrary other symbols without changing the fundamental properties of the puzzle, as no numerical properties of the digits play a role in a normal Sudoku puzzle. However, other symbols would be more difficult to enumerate exhaustively for the participants, slowing down this process. One could thus devise an experiment in which the discovery and use of tactics is compared between two groups: one with the conventional digits (easy and fast enumeration) and one with arbitrary symbols (slow, effortful, and error-prone enumeration). This would be a good test of the effect of a slowed-down subroutine.

6.1.3 *Learning new rules*

The previous sections have dealt with preference learning on existing rules or efficiency gains through chunking existing productions. But could it be possible to model how people learn fundamentally new rules that go beyond already implemented procedures? Some form of inductive learning is required here (Schmid and Kitzelmann, 2011).

One way in which humans learn new tactics or an entirely new task is by demonstration: An expert solves the task while the novice watches and then attempts to copy the behavior of the expert. Understanding the intentions of the expert and reasons for specific actions helps the learner to solve the task successfully. Reasonable imitation usually involves some form of abstraction and analogous reasoning, 14 months old infants can already translate a movement to a different extremity instead of blindly copying the movement of the adult (Gergely et al., 2002).

Program induction generalizes from a small set of demonstrations to a general function that can work on new inputs (Gulwani, 2011; Kitzelmann, 2008). There is also work that interleaves program induction with building a more expressive language through abstracting common induced structures and storing them as new building blocks (Ellis et al., 2020). For learning new rules from examples we probably have already collected an ideal data set: The data we collected in chapter 5 does not only contain filling events, but all cells that were relevant for the deduction are highlighted, too. We have already labeled each entry in that data set with a rule from the set found in previous think-aloud studies. It might be possible to further cluster the moves with the same label to get even finer classes

of rules to learn. In future work, this data set should be suited to inductively learn the rules that describe the inference made in the examples.

Humans, however, cannot only learn from demonstration, they can also find new tactics on their own (Ericsson et al., 1980; Gray and Lindstedt, 2017). Such strategy discovery is often a conscious effort, marked by some form of meta-cognition, where the problem solver observes their own tactic and tries to improve it (Jäkel and Schreiber, 2013). Analogy and abstraction might play a very important part in finding new tactics in such cases and they are often considered a key element of our intelligence (Gentner, 2003; Mitchell, 2021). In the domain of Sudoku, the following are observations most beginners likely make: The most simple analogy in the domain of Sudoku is "rows are like columns" and any reasoning tactic that is applicable to rows is similarly applicable to columns. For many cases boxes are also equivalent to rows and columns and a possible abstraction over these three elements is *unit*. A further insight could be that sometimes the *union of two or three units* can take the place of a single unit. A more advanced abstraction could be that in some cases it is possible to reason on the basis of *candidate digits* in a similar way as with *known digits*. Analogies and abstractions like these make it possible to reason about situations which would have seemed unsolvable before.

Within-domain analogies and abstractions will probably go a long way in explaining tactic learning in Sudoku. To explain problem solving more generally, however, it will also be necessary to include analogies between different domains (Gick and Holyoak, 1980; Mitchell, 2021). Whether analogical reasoning will suffice to explain the learning of new rules in Sudoku is an empirical question that remains to be tested.

6.2 *Conclusion*

The flexibility to choose suitable tactics for tackling a task is a core human ability that can be observed in game-like settings. Flexibly adapting to new objectives or taking new constraints into account is an area in which humans still outperform artificial intelligence (AI) approaches (Johnson et al., 2021; Lake et al., 2019). The paradigms proposed in this work are well suited to study this ability. The qualitative insights gained from the think-aloud protocols were an important prerequisite for our quantitative modeling of participants' behavior in follow-up experiments. We found that people are adept at switching tactics on the fly when the task demands it. Methodologically, this work has been inspired by Newell and Simon (1972), specifically by also studying individual problem solving traces and the use of production systems. We have built on these methods and combined them with tools of modern cognitive science, namely choice models and hierarchical Bayesian models. Digit-placement puzzles proved to be a rich experimental paradigm allowing us to

study and model different aspects of tactic preferences and tactic selection. There is great potential to advance our knowledge of human problem solving, flexibility in tactic choice and learning of new tactics with these methods and tasks.

Appendix A

Appendix for: Hierarchical Bayesian model

A.1 Posthoc parameter recovery study

As a sanity check we can test how well the model fits the parameters for artificially generated data with known values. Of course, when generating data from the model, Bayesian inference should be able to re-discover the values from the generated data. Such a parameter recovery study does not tell us about the model's adequacy for real-world data. For that aspect it is more useful to compare different models, similar to our comparison of the two Sudoku models (Lee et al., 2019, Appendix B).

When we generate data from our model we know the true values of each parameter. We can then check where the true value is in the posterior distribution. Intuitively, the best case would be if the true value is very close to the mean of the distribution. Of course, also the width of the posterior distribution should be well calibrated. More specifically, the quantiles of where the true values fall within the posterior distribution should follow a uniform distribution. Hence, when one has many true values, one can calculate for each of them the quantile within the corresponding posterior distribution of samples and check whether they are correctly distributed.

In order to quantify how many participants and trials per participant are necessary in order to reach a specific size of highest density interval (HDI), one can simulate data of different sizes and fit the model on them. Naturally, with more data the size of the HDI shrinks. The numbers required to reach a desired size of course depend on the model specification. It thus makes sense to run such simulations with the specific model one wants to use for data analysis. To give an impression of possible results of such a study, we show different combinations of participant number and trials per participant for the experiment and model reported above ("EIP regression with strategy selection") in Table A.1. To generate the data we set the six population parameters to approximately the means found in the model for the actual experiment. We used $m_a = 1.5$, $m_b = 2.5$, $m_\theta = 3.5$,$s_a = 1.3$, $s_b = 3$, and $s_\theta = 3$.

As expected, the size of the HDI shrinks with more data. Participant parameters can be estimated more precisely with more trials per participant, whereas population level parameters get more pre-

cise estimates with an increasing number of participants. Depending on which of these values one needs to a certain precision, one can adapt the experimental design to reach the goal. One thing that becomes apparent in this simulation study is that the model has most problems in finding precise values for the b parameter and its priors. The sizes of the confidence intervals are much larger than for any of the other parameters.

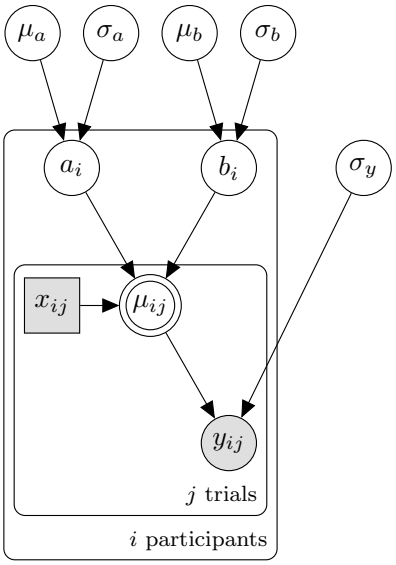
We only fit one simulated experiment per combination of participant and trial number, due to computation time. While the smallest combination was fit in just 10 minutes, the largest ones needed several days on our machine, the latent steps make computations very slow. All the models are well calibrated. The true values fall evenly within the quantiles of the posterior distribution. PP-plots showing how closely the fitted distributions match the theoretical expectations can be found at https://osf.io/rgh3j/.

Part.	**10**			**50**			**150**
Trials	30	150	450	30	150	450	30
m_a	1.92	1.94	2.42	0.68	0.47	0.49	0.32
m_b	5.57	5.61	3.66	5.97	2.90	3.44	2.52
m_θ	4.04	3.29	3.83	1.82	1.93	2.27	1.17
s_a	3.13	2.98	4.56	0.90	0.56	0.64	0.41
s_b	56.90	101.33	75.37	53.22	16.77	17.14	17.53
s_θ	8.83	5.54	6.76	2.67	3.12	3.99	1.90
a	0.59	0.36	0.16	0.86	0.34	0.23	0.67
b	5.58	4.29	1.32	10.27	3.95	3.42	6.58
θ	3.57	1.92	1.23	4.76	2.65	1.59	4.76
w	0.40	0.25	0.17	0.54	0.30	0.19	0.54

Table A.1: Sizes of highest density intervals for different numbers of participants and trials for the EIP regression with latent steps and strategy selection. Experimental design of the simulations was exactly like in the reported experiment above, only the number of participants and trials per participant were varied.

A.2 *Linear regression as comparison*

We can also implement the traditional linear regression as a hierarchical Bayesian model. We do so to compare the results to the EIP regression model on the data of children's addition. We estimate parameters for the mean (a_i) and slope (b_i) for each participant, the variance is a population parameter in this model. We model the participant parameters as draws from normal distributions. The graphical model can be found in Figure A.1.



$$
\begin{aligned}
\mu_a, \mu_b &\sim \text{Normal}(0, 10) \\
\sigma_a, \sigma_b &\sim \text{HalfCauchy}(\beta = 5) \\
a_i &\sim \text{Normal}(\mu_a, \sigma_a) \\
b_i &\sim \text{Normal}(\mu_b, \sigma_b) \\
\sigma_y &\sim \text{HalfCauchy}(\beta = 5) \\
x_{ij} &\leftarrow \text{EIP step count} \\
\mu_{ij} &\leftarrow a_i x_{ij} + b_i \\
y_{ij} &\sim \text{Normal}(\mu_{ij}, \sigma_y)
\end{aligned}
$$

Figure A.1: Hierarchical linear regression model for the counting trials. The predicted mean is determined by the parameters a_i and b_i together with the value of the min addend x_{ij}.

A.2.1 *Results linear regression*

		mean	sd	HDI 2.5%	HDI 97.5%
μ_a	[s]	0.433	0.030	0.374	0.494
μ_b	[s]	1.726	0.106	1.515	1.929
σ_a	[s]	0.267	0.019	0.229	0.304
σ_b	[s]	0.231	0.149	0.000	0.494
σ_y	[s]	1.584	0.028	1.533	1.643

Table A.2: The parameters found for the hierarchical linear regression. All values are in seconds.

Again, we get parameters for each participant and hyper-parameter distributions describing the group. Group parameters can be found in Table A.2, the distribution of participant parameters is plotted in Figure A.2. The regression line for a single, exemplary participant can be seen in Figure A.3.

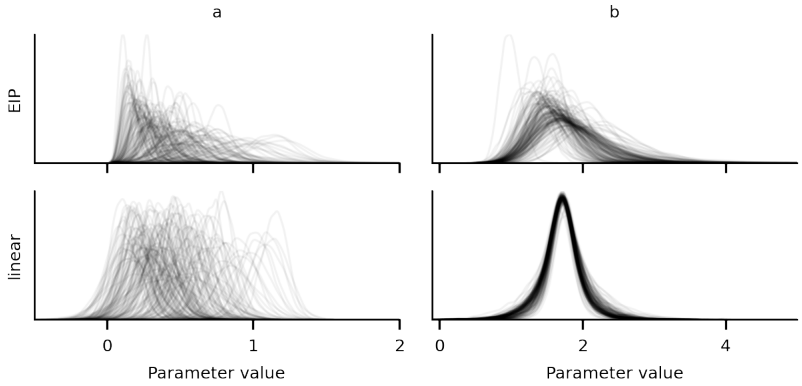


Figure A.2: The distributions of the participant parameters for all participants for the two models of min-counting. Each line shows the posterior density of values for the respective variable for one participant. Top: EIP regression model, bottom: linear regression model. Note how some distributions of the variable a get credible values of below zero in the linear regression model.

A.2.2 Comparing the results of the two models

The two models, EIP regression and standard linear regression, use different distributions for the priors as well as the spread around the predicted mean, so one would expect slightly different results. An important difference is also that the EIP regression model fits a variance parameter for each individual participant, whereas in the linear regression model the variance parameter is shared by all participants. However, generally the two models agree very well with each other. In the group parameters, the means and standard deviations of a and b are almost identical, only the standard deviation for the b parameter is bigger in the EIP regression model than the linear regression. The parameters per participant can be compared easily. a_i is the value of the slope, b_i the value of the intercept for each participant. Both models try to fit the mean of the participant data. in Figure A.2 the densities of all participants for both parameters are depicted. The rough ranges of plausible parameters is the same in both models, in the linear regression model some values below zero are included, which is excluded by the specification of the EIP regression model.

It is also possible to translate the parameters into regression lines for single participants in both models. An example can be seen in Figure A.3. The mean prediction is very similar in both models. The biggest difference between the two models is how the variance of the data points around the mean is handled. In the linear regression model, steps of the same duration are added and variance is only added via a normal distribution around the so calculated mean in the end. In the EIP regression model on the other hand, we assume that each step that contributes to the overall time is a random variable that can have different duration. So a larger spread for bigger values of the min addend (x_{ij}) is expected by the EIP regression model but not by the linear regression model. In the linear regression model, the standard deviation of the normal distribution is just one parameter that is shared by all participants, in the EIP regression model the precision parameter θ_i is fit for each participant i and can thus be different across participants. An obvious advantage of the Gamma

density is, that it will never place probability mass below zero. Statistically, using the estimated leave-one-out ELPD or cross validation ELPD, the EIP regression model has much better log-likelihood (-2783.81 with standard error 49.85 for the EIP model and -3473.51 with standard error 115.58, the difference is thus 689.70 with a standard error of 91.34) Hence, while the EIP regression is theoretically clearly superior, in this example, it does lead to conclusions that are extremely similar to those of a standard linear regression analysis – as performed in the study of Hopkins and Bayliss (2017), whose data we analyzed here.

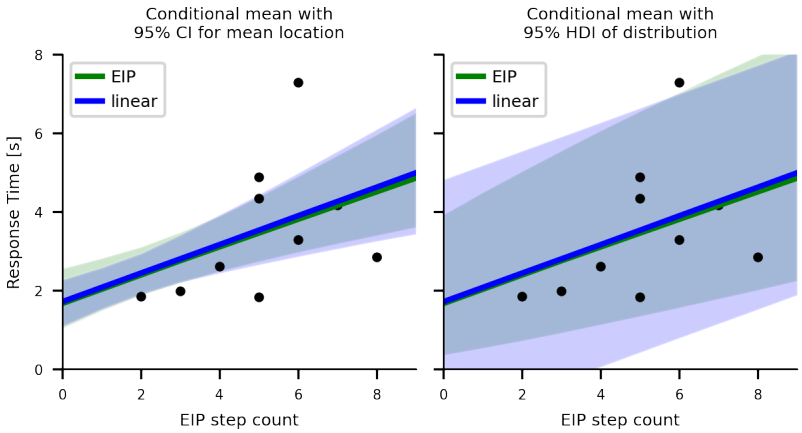


Figure A.3: Direct comparison of the two model results. The point estimates for the regression line is very similar in both models. The biggest difference is in the lower plot. In EIP regression, the expected spread around the mean is adjusted for each participant individually. It is not symmetric around the regression line, but has a bigger spread above than below. Additionally, it increases with the value of the min addend. In the linear regression, we fit just a single spread parameter for the entire population, it is symmetric around the regression line and thus can include negative response times in the expected region.

Bibliography

Alderton, D. L. and Larson, G. E. (1994). Cross-task consistency in strategy use and the relationship with intelligence. *Intelligence*, 18:47–76. doi: 10.1016/0160-2896(94)90020-5.

Althaus, N. and Plunkett, K. (2016). Categorization in infancy: Labeling induces a persisting focus on commonalities. *Developmental Science*, 19(5):770–780. doi: 10.1111/desc.12358.

Anderson, J. R. (2007). *How can the human mind occur in the pysical universe?* Oxford series on cognitive models and architectures. Oxford university press. doi: 10.1093/acprof:oso/9780195324259.001.0001.

Anderson, J. R., Albert, M. V., and Fincham, J. M. (2005). Tracing problem solving in real time: fMRI analysis of the subject-paced Tower of Hanoi. *Journal of cognitive neuroscience*, 17(8):1261–1274. doi: 10.1162/0898929055002427.

Angerer, B. and Schreiber, C. (2019). Representational dynamics in the domain of iterated mental paper folding. *Cognitive Systems Research*, 54:217–231. doi: 10.1016/j.cogsys.2018.11.011.

Anzai, Y. and Simon, H. A. (1979). The theory of learning by doing. *Psychological Review*, 86(2):124–140. doi: 10.1037/0033-295X.86.2.124.

Batchelder, W. H. and Alexander, G. E. (2012). Insight problem solving: A critical examination of the possibility of formal theory. *The Journal of Problem Solving*, 5(1):56–100. doi: 10.7771/1932-6246.1143.

Behrens, T., Räuker, M., Kalbfleisch, M., and Jäkel, F. (2023). Flexible use of tactics in Sudoku. *Thinking & Reasoning*, 29(4):488–530. doi: 10.1080/13546783.2022.2091040.

Berardi-Coletta, B., Dominowski, R. L., Buyer, L. S., and Rellinger, E. R. (1995). Metacognition and problem solving: A process-oriented approach. *Journal of Experimental Psychology: Learning, Memory and Cognition*, pages 205–223. doi: 10.1037/0278-7393.21.1.205.

Bettman, J., Johnson, E. J., and Payne, J. W. (1990). A componential analysis of cognitive effort in choice. *Organizational behavior and human decision processes*, 45:111–139. doi: 10.1016/0749-5978(90)90007-V.

Birney, D. P., Bowman, D. B., Beckmann, J. F., and Seah, Y. Z. (2012). Assesment of processing capacity: Reasoning in Latin Square tasks in a population of managers. *European Journal of Psychological Assessment*, 28(3):216–226. doi: 10.1027/1015-5759/a000146.

Birney, D. P., Halford, G. S., and Andrews, G. (2006). Measuring the influence of complexity on relational reasoning: The development of the Latin Square Task. *Educational and Psychological Measurement*, 66(1):146–171. doi: 10.1177/0013164405278570.

Blech, C., Gaschler, R., and Bilalić, M. (2019). Why do people fail to see simple solutions? Using think-aloud protocols to uncover the mechanism behind the Einstellung (mental set) effect. *Thinking and Reasoning*, 26(4):552–580. doi: 10.1080/13546783.2019.1685001.

Bradley, R. A. and Terry, M. E. (1952). Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39:324–345.

Brandstätter, E. and Gussmack, M. (2013). The cognitive processes underlying risky choice. *Journal of behavioral decision making*, 12:185–197. doi: 10.1002/bdm.1752.

Bras, N. (2021). User interfaces for solving Sudoku. Bachelor's thesis, Technical University of Darmstadt.

Brasoveanu, A. and Dotlacil, J. (2021). Reinforcement learning for production-based cognitive models. *Topics in Cognitive Science*, 13. doi: 10.1111/tops.12546.

Brown, N. R. (1995). Estimation strategies in the judgement of event frequency. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 21(6):1539–1553. doi: 10.1037/0278-7393.21.6.1539.

Bürkner, P.-C. (2017). brms: an R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1). doi: 10.18637/jss.v080.i01.

Callaway, F., van Opheusden, B., Gul, S., Das, P., Krueger, P., Griffiths, T. L., and Lieder, F. (2021). Rational use of cognitive resources in human planning. *psyarxiv Preprints*. doi: 10.31234/osf.io/byaqd.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*, 76(1). doi: 10.18637/jss.v076.i01.

Chase, W. G. and Simon, H. A. (1973). Perception in chess. *Cognitive psychology*, 4(1):55–81. doi: 10.1016/0010-0285(73)90004-2.

Chi, M. T. H. (1997). Quantifying qualitative analyses of verbal data: A practical guide. *The Journal of the Learning Sciences*, 6(3):271–315. doi: 10.1207/s15327809jls0603_1.

Chu, J. and Schulz, L. E. (2020). Play, curiosity, and cognition. *Annual Review of Developmental Psychology*, 2:317–343. doi: 10.1146/annurev-devpsych-070120-014806.

Clark, H. H. and Chase, W. G. (1972). On the process of comparing sentences against pictures. *Cognitive psychology*, 3(3):472–517. doi: 10.1016/0010-0285(72)90019-9.

Danek, A. H., Fraps, T., von Müller, A., Grothe, B., and Öllinger, M. (2014). Working wonders? Investigating insight with magic tricks. *Cognition*, 130:174–185. doi: 10.1016/j.cognition.2013.11.003.

Daston, L. (2022). *Rules: A short history of what we live by*. The Lawrence Stone Lectures. Princeton University Press. doi: 10.1515/9780691239187.

Davidson, G., Gureckis, T. M., and Lake, B. M. (2022). Creativity, compositionality, and common sense in human goal generation. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Debreu, G. (1960). Review of RD Luce, individual choice behavior. *American Economic Review*, 50(1):186–188.

Duncker, K. (1945). On problem-solving. *Psychological monographs*, 58(5). doi: 10.1037/h0093599.

Ellis, K., Wong, C., Nye, M., Meyer, M. S., Cary, L., Morales, L., Hewitt, L., Solar-Lezama, A., and Tenenbaum, J. B. (2020). DreamCoder: Growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning. *arXiv*. doi: 10.48550/arXiv.2006.08381.

Ericsson, K. A., Chase, W. G., and Faloon, S. (1980). Acquisition of a memory skill. *Science*, 208:1181–1182. doi: 10.1126/science.7375930.

Ericsson, K. A. and Simon, H. A. (1993). *Protocol analysis: Verbal reports as data (revised edition)*. the MIT Press. doi: 10.7551/mitpress/5657.001.0001.

Estes, S. (2021). Cogulator: A primer [White paper]. *The MITRE Corporation*.

Fazio, L. K., DeWolf, M., and Siegler, R. S. (2016). Strategy use and strategy choice in fraction magnitude comparisons. *Journal of Experimental Psychology*, 42(1):1–16. doi: 10.1037/xlm0000153.

Fox, M. C., Ericsson, K. A., and Best, R. (2011). Do procedures for verbal reporting of thinking have to be reactive? A meta-analysis and recommendations for best reporting methods. *Psychological Bulletin*, 137(2):316–344. doi: 10.1037/a0021663.

Friedrich, M. B. and Ritter, F. E. (2020). Understanding strategy differences in a fault-finding task. *Cognitive Systems Research*, 59:133–150. doi: 10.1016/j.cogsys.2019.09.017.

Gelman, A. and Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press. doi: 10.1017/CBO9780511790942.

Gentner, D. (2003). Why we're so smart. In Gentner, D. and Goldin-Meadow, S., editors, *Language in mind: Advances in the study of language and thought*, pages 195–235. Cambridge MA: MIT Press.

Gentner, D., Shao, R., Simms, N., and Hespos, S. (2021). Learning same and different relations: cross-species comparisons. *Current Opinion in Behavioral Sciences*, 37:84–89. doi: 10.1016/j.cobeha.2020.11.013.

Gergely, G., Bekkering, H., and Király, I. (2002). Rational imitation in preverbal infants. *Nature*, 415(755). doi: 10.1038/415755a.

Gick, M. L. and Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology*, 12:306–355. doi: 10.1016/0010-0285(80)90013-4.

Gray, P. (2019). Evolutionary functions of play: Practice, resilience, innovation and cooperation. In Smith, P. K. and Roopnarine, J., editors, *The Cambridge Handbook of Play: Developmental and Disciplinary Perspectives*. Cambridge University Press.

Gray, W. D. and Lindstedt, J. K. (2017). Plateaus, dips, and leaps: Where to look for inventions and discoveries during skilled performance. *Cognitive Science*, 41:1838–1870. doi: 10.1111/cogs.12412.

Groen, G. J. and Parkman, J. M. (1972). A chronometric analysis of simple addition. *Psychological review*, 79(4):329. doi: 10.1037/h0032950.

Gulwani, S. (2011). Automating string processing in spreadsheets using input-output examples. In *Proceedings of the 38th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, volume 46, pages 317–330. doi: 10.1145/1926385.1926423.

Gunzelmann, G. and Anderson, J. R. (2003). Problem solving: Increased planning with practice. *Cognitive Systems Research*, 4(1):57–76. doi: 10.1016/S1389-0417(02)00073-6.

Halford, G. S., Wilson, W. H., and Phillips, S. (1998). Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology. *Behavioral and brain sciences*, 21:803–865. doi: 10.1017/S0140525X98001769.

Hanek, S. I. (2022). The effect of puzzle composition on tactic selection in Sudoku. Bachelor's thesis, Technical University of Darmstadt.

Hartung, J., Goecke, B., Schroeders, U., Schmitz, F., and Wilhelm, O. (2022). Latin Square Tasks: A multi-study evaluation. *Intelligence*, 94. doi: 10.1016/j.intell.2022.101683.

Hearne, L. J., Birney, D. P., Cocchi, L., and Mattingley, J. B. (2020). The Latin Square Task as a measure of relational reasoning: A replication and assessment of reliability. *European Journal of Psychological Assessment*, 36(2):296–302. doi: 10.1027/1015-5759/a000520.

Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623. doi: 10.5555/2627435.2638586.

Hopkins, S. and Bayliss, D. (2017). The prevalence and disadvantage of min-counting in seventh grade: Problems with confidence as well as accuracy? *Mathematical Thinking and Learning*, 19(1):19–32. doi: 10.1080/10986065.2017.1258613.

Hopkins, S. L. and Lawson, M. J. (2006). The effect counting speed has on developing a reliance on retrieval in basic addition. *Contemporary Educational Psychology*, 31:208–227. doi: 10.1016/j.cedpsych.2005.06.001.

John, B. E. and Newell, A. (1989). Cumulating the science of HCI: From S-R compatibility to transcription writing. In *CHI '89 Proceedings*, pages 109–114.

Johnson, A., Vong, W. K., Lake, B. M., and Gureckis, T. M. (2021). Fast and flexible: Human program induction in abstract reasoning tasks. In *Proceedings of the 43rd Annual Meeting of the Cognitive Science Society*.

Johnson, E. J., Payne, J. W., Bettman, J. R., and Schkade, D. A. (1989). Monitoring information processing and decisions: The mouselab system. Technical report, Duke University.

Jäkel, F. and Schreiber, C. (2013). Introspection in problem solving. *The Journal of Problem Solving*, 6(1):20–33. doi: 10.7771/1932-6246.1131.

Kalbfleisch, M. (2020). Der Effekt der Instruktion und der relatinalen Komplexität beim Problemlösen am Beispiel von Sudokus. Bachelor's thesis, Technical University of Darmstadt.

Kaplan, C. A. and Simon, H. A. (1990). In search of insight. *Cognitive Psychology*, 22(3):374–419. doi: 10.1016/0010-0285(90)90008-R.

Karat, J. (1982). A model of problem solving with incomplete constraint knowledge. *Cognitive Psychology*, 14(4):538–559. doi: 10.1016/0010-0285(82)90018-4.

Katahra, D. (2022). Rows, columns, and boxes in Sudoku: How flexibly do people switch between different units? Master's thesis, Technical University Kaiserslautern.

Kitzelmann, E. (2008). Analytical inductive functional programming. In Hanus, M., editor, *International symposium on logic-based program synthesis and transformation*, pages 166–180. Springer.

Kotovsky, K., Hayes, J. R., and Simon, H. A. (1985). Why are some problems hard? Evidence from the Tower of Hanoi. *Cognitive Psychology*, 17:248–294. doi: 10.1016/0010-0285(85)90009-X.

Kumar, R., Carroll, C., Hartikainen, A., and Martin, O. (2019). ArviZ a unified library for exploratory analysis of Bayesian models in Python. *Journal of Open Source Software*, 4(33):1–5. doi: 10.21105/joss.01143.

Kühlwein, T. (2020). Denkprotokolle beim Zahlenrätsel Straights. Bachelor's thesis, Technical University of Darmstadt.

Kühn, A. (2021). Probabilistic modeling of response times in Sudoku. Bachelor's thesis, Technical University of Darmstadt.

Laird, J. E. (2012). *The Soar cognitive architecture*. MIT press. doi: 10.7551/mitpress/7688.001.0001.

Lake, B. M., Linzen, T., and Baroni, M. (2019). Human few-shot learning of compositional instructions. In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society*.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:1–72. doi: 10.1017/S0140525X16001837.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174. doi: 10.2307/2529310.

Lee, M. D., Gluck, K. A., and Walsh, M. M. (2019). Understanding the complexity of simple decisions: Modeling multiple behaviors and switching strategies. *Decision*, 6(4):335–368. doi: 10.1037/dec0000105.

Lee, M. D. and Wagenmakers, E.-J. (2014). *Bayesian cognitive modeling: A practical course*. Cambridge university press. doi: 10.1017/CBO9781139087759.

Lee, N. Y. L., Goodwin, G. P., and Johnson-Laird, P. N. (2008). The psychological puzzle of Sudoku. *Thinking & Reasoning*, 14(4):342–364. doi: 10.1080/13546780802236308.

Lee, N. Y. L. and Johnson-Laird, P. N. (2013). Strategic changes in problem solving. *Journal of cognitive psychology*, 25:165–173. doi: 10.1080/20445911.2012.719021.

Luce, R. D. (1959). *Individual choice behavior: A theoretical analysis*. Wiley, New York. doi: 10.1037/14396-000.

MacLeod, C. M., Hunt, E. B., and Mathews, N. N. (1978). Individual differences in the verification of sentence-picture relationships. *Journal of verbal learning and verbal behavior*, 17:493–507. doi: 10.1016/S0022-5371(78)90293-1.

Maier, N. R. F. (1931). Reasoning in humans. II. the solution of a problem and its appearance in consciousness. *Journal of Comparative Psychology*, 12(2):181–194. doi: 10.1037/h0071361.

Maris, E. (1993). Additive and multiplicative models for gamma distributed random variables, and their application as psychometric models for response times. *Psychometrika*, 58(3):445–469. doi: 10.1007/BF02294651.

McFadden, D. (1973). Conditional logit analysis of qualitative choice behavior. In Zarembka, P., editor, *Econometrics*, pages 105–142. New York: Academic Press.

Mitchell, M. (2021). Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101. doi: 10.1111/nyas.14619.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518:529–533. doi: 10.1038/nature14236.

Newell, A. (1990). *Unified theories of cognition*. Harvard University Press, Cambridge, MA.

Newell, A. and Simon, H. A. (1972). *Human problem solving*, volume 104. Prentice-Hall Englewood Cliffs, NJ.

Nisbett, R. E. and Wilson, T. D. (1977). Telling more than we can know: Verbal reports on mental processes. *Psychological review*, 84(3):231–259. doi: 10.1037/0033-295X.84.3.231.

Nombela, C., Bustillo, P. J., Castell, P. F., Sanchez, L., Medina, V., and Herrero, M. T. (2011). Cognitive rehabilitation in Parkinson's disease: Evidence from neuroimaging. *Frontiers in Neurology*, 2:233–245. doi: 10.3389/fneur.2011.00082.

O'Donnell, T. J., Tenenbaum, J. B., and Goodman, N. D. (2009). Fragment grammars: Exploring computation and reuse in language. techreport MIT-CSAIL-TR-2009-013, Massachusetts Institute of Technology.

Ohlsson, S. (2012). The problems with problem solving: Reflections on the rise, current status, and possible future of a cognitive research paradigm. *The Journal of Problem Solving*, 5(1):101–128. doi: 10.7771/1932-6246.1144.

Papagno, C., Semenza, C., and Girelli, L. (2013). Meeting an "impossible challenge" in semantic dementia: Outstanding performance in numerical Sudoku and quantitative number knowledge. *Neuropsychology*, 27(6):680–690. doi: 10.1037/a0034457.

Payne, J. W. and Bettman, J. R. (2004). Walking with the scarecrow: The information-processing approach to decision research. In Koehler, D. J. and Harvey, N., editors, *Blackwell handbook of judgment and decision making*, pages 110–132. Blackwell Publishing Ltd. doi: 10.1002/9780470752937.ch6.

Payne, J. W., Bettman, J. R., and Johnson, E. J. (1988). Adaptive strategy selection in decision making. *Journal of experimental psychology: Learning, Memory, and Cognition*, 14(3):534–552. doi: 10.1037/0278-7393.14.3.534.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peirce, J., Gray, J. R., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., Kastman, E., and Lindeløv, J. K. (2019). PsychoPy2: Experiments in behavior made easy. *Behavior research methods*, 51(1):195–203. doi: 10.3758/s13428-018-01193-y.

Pelánek, R. (2011). Human problem solving: Sudoku case study. Technical report, Masaryk University, Faculty of Informatics.

Perret, P., Bailleux, C., and Dauvier, B. (2011). The influence of relational complexity and strategy selection on children's reasoning in the Latin Square Task. *Cognitive Development*, 26:127–141. doi: 10.1016/j.cogdev.2010.12.003.

Preuß, K. (2018). An ACT-R model of skill development in the case of sudoku. Master's thesis, Albert-Ludwigs-Universität Freiburg i. Br.

Qin, Y., Xiang, J., Wang, R., Zhou, H., Li, K., and Zhong, N. (2012). Neural bases for basic processes in heuristic problem solving: Take solving Sudoku puzzles as an example. *PsyCh Journal*, 1:101–117. doi: 10.1002/pchj.15.

Ramolla, J. (2020). Probabilistische Performanzeinschätzung bei Sudokus. Bachelor's thesis, Technical University of Darmstadt.

Rieskamp, J. and Otto, P. E. (2006). SSL: A theory of how people learn to select strategies. *Journal of Experimental Psychology: General*, 135(2):207–236. doi: 10.1037/0096-3445.135.2.207.

Ritter, F. E. and Bibby, P. A. (2008). Modeling how, when, and what is learned in a simple fault-finding task. *Cognitive science*, 32(5):862–892. doi: 10.1080/03640210802221999.

Roberts, M. J. and Erdos, G. (1993). Strategy selection and metacognition. *Educational Psychology*, 13(3-4):259–266. doi: 10.1080/0144341930130304.

Roberts, M. J., Gilmore, D. J., and Wood, D. J. (1997). Individual differences and strategy selection in reasoning. *British Journal of Psychology*, 88:473–492. doi: 10.1111/j.2044-8295.1997.tb02652.x.

Rothkegel, T. (2023). Mental set effect in Sudoku. Bachelor's thesis, Technical University of Darmstadt.

Rouder, J. N., Sun, D., Speckman, P. L., Lu, J., and Zhou, D. (2003). A hierarchical Bayesian statistical framework for response time distributions. *Psychometrica*, 68(4):589–606. doi: 10.1007/BF02295614.

Russell, S. and Norvig, P. (2020). *Artificial intelligence: A modern approach (4th edition)*. Pearson.

Salthouse, T. A. (2010). Selective review of cognitive aging. *International Neropsychological Society*, 16(5):754–760. doi: 10.1017/S1355617710000706.

Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. (2016). Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2. doi: 10.7717/peerj-cs.55.

Schaeffer, J., Burch, N., Björnsson, Y., Kishimoto, A., Müller, M., Lake, R., Lu, P., and Sutphen, S. (2007). Checkers is solved. *Science*, 317:1518–1522. doi: 10.1126/science.1144079.

Schmid, U. and Kitzelmann, E. (2011). Inductive rule learning on the knowledge level. *Cognitive Systems Research*, 12(3–4):237–248. doi: 10.1016/j.cogsys.2010.12.002.

Schmidhuber, J. (2010). Formal thory of creativity, fun, and intrinsic motivation (1990-2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247. doi: 10.1109/TAMD.2010.2056368.

Shepard, R. N. and Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703. doi: 10.1126/science.171.3972.701.

Siegler, R. S. (1987). The perils of averaging data over strategies: An example from children's addition. *Journal of Experimental Psychology: General*, 116(3):250–264. doi: 10.1037/0096-3445.116.3.250.

Siegler, R. S. (1991). Strategy choice and strategy discovery. *Learning and Instruction*, 1:89–102. doi: 10.1016/0959-4752(91)90020-9.

Siegler, R. S. and Jenkins, E. (1989). *How children discover new strategies*. John M. MacEachran memorial lecture series. Lawrence Erlbaum Associates. doi: 10.4324/9781315807744.

Simon, H. A. (1975). The functional equivalence of problem solving skills. *Cognitive Psychology*, 7(2):268–288. doi: 10.1016/0010-0285(75)90012-2.

Simon, H. A. (1992). What is an "explanation" of behavior? *Psychological Science*, 150(3):150–161. doi: 10.1111/j.1467-9280.1992.tb00017.x.

Simon, H. A. (1996). *The sciences of the artificial (3rd edition)*. The MIT Press. doi: 10.7551/mitpress/12107.001.0001.

Simon, H. A. and Reed, S. K. (1976). Modeling strategy shifts in a problem-solving task. *Cognitive Psychology*, 8:86–97. doi: 10.1016/0010-0285(76)90005-0.

Straub, D. and Rothkopf, C. A. (2022). Putting perception into action with inverse optimal control for continuous psychophysics. *eLife*, 11:e76635. doi: 10.7554/eLife.76635.

Sun, R. (2016). *Anatomy of the mind: Exploring psychological mechanisms and processes with the Clarion cognitive architecture*. Oxford University Press. doi: 10.1093/acprof:oso/9780199794553.001.0001.

Taatgen, N., Lebiere, C., and Anderson, J. (2006). Modeling paradigms in ACT-R. In Sun, R., editor, *Cognition and multi-agent interaction: From cognitive modeling to social simulations*, pages 29–52. Cambridge University Press.

Townsend, J. T. and Ashby, F. G. (1983). *The stochastic modeling of elementary psychological processes*. Cambridge University Press, Cambridge.

Train, K. (2003). *Discrete choice methods with simulation*. Cambridge University Press. doi: 10.1017/CBO9780511805271.

Treisman, A. (1982). Perceptual grouping and attention in visual search for features and for objects. *Journal of Experimental Psychology*, 8(2):94–214. doi: 10.1037/0096-1523.8.2.194.

Tversky, A. (1972). Elimination by aspects: A theory of choice. *Psychological review*, 79(4):281–299. doi: 10.1037/h0032955.

van Opheusden, B., Kuperwajs, I., Galbiati, G., Bnaya, Z., Li, Y., and Ma, W. J. (2023). Expertise increases planning depth in human gameplay. *Nature*. doi: 10.1038/s41586-023-06124-2.

VanLehn, K. (1991). Rule acquisition events in the discovery of problem-solving strategies. *Cognitive Science*, 15:1–47. doi: 10.1016/0364-0213(91)80012-T.

Vehtari, A., Gelman, A., and Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-outcross-validation and WAIC. *Stat Comput*, 27:1413–1432. doi: 10.1007/s11222-016-9696-4.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. (2021). Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of MCMC. *Bayesian Analysis*, 16(2):667–718. doi: 10.1214/20-BA1221.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silve, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575:350–354. doi: 10.1038/s41586-019-1724-z.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, I., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). Scipy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*.

Walsh, M. M. and Gluck, K. A. (2015). Verbalization of decision strategies in multiple-cue probabilistic inference. *Journal of Behavioral Decision Making*, 29(1):78–91. doi: 10.1002/bdm.1878.

Wiecki, T., Salvatier, J., Vieira, R., Patil, A., Kochurov, M., Engels, B., Lao, J., Carroll, C., Osthege, M., Martin, O. A., Willard, B. T., Seyboldt, A., Rochford, A., rpgoldman, Paz, L., Meyer, K., Coyle, P., Gorelli, M. E., Kumar, R., Abril-Pla, O., Yoshioka, T., Ho, G., Kluyver, T., Beauchamp, K., Andorra, A., Pananos, D., Spaak, E., Edwards, B., Ma, E., and Domenzain, L. M. (2022). PyMC: v4.1.3. Zenodo. doi: 10.5281/zenodo.6838902.

Zhang, L., Xin, Z. Q., Lin, C., and Li, H. (2009). The complexity of the latin square task and its influence on children's performance. *Chinese Science Bulletin*, 54:766–775. doi: 10.1007/s11434-009-0079-5.

Zmarsly, M. (2020). Menschliches Problemlösen bei dem Zahlenrätsel Straights. Bachelor's thesis, Technical University of Darmstadt.

Erklärung zur Abschlussarbeit

Hiermit erkläre ich, Thea Behrens, dass ich die vorliegende Arbeit selbstständig, ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt habe. Ich habe mit Ausnahme der zitierten Literatur und anderer in der Arbeit genannter Quellen keine fremden Hilfsmittel benutzt. Die von mir bei der Anfertigung dieser wissenschaftlichen Arbeit wörtlich oder inhaltlich benutzte Literatur und alle anderen Quellen habe ich im Text deutlich gekennzeichnet und gesondert aufgeführt. Dies gilt auch für Quellen oder Hilfsmittel aus dem Internet. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Thesis Statement

I herewith formally declare that I, Thea Behrens, have written the submitted thesis independently without any outside support and using only the quoted literature and other sources. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I have clearly marked and separately listed in the text the literature used literally or in terms of content and all other sources I used for the preparation of this academic work. This also applies to sources or aids from the Internet. This thesis has not been handed in or published before in the same or similar form.

Datum/Date	Unterschrift/Signature