Received: 7 February 2020 | Revised: 3 February 2021 | Accepted: 9 February 2021

DOI: 10.1002/acp.3809

REGISTERED REPORT STAGE 1: STUDY DESIGN

WILEY

The impact of inducing troubleshooting strategies via visual aids on performance in a computerized digital network task

Martin Bordewieck[1,2] | Malte Elson[1,2]

[1]Faculty of Psychology, Ruhr University Bochum, Bochum, Germany

[2]Horst Görtz Institute for IT Security, Ruhr University Bochum, Bochum, Germany

Correspondence
Martin Bordewieck, Psychology of Human Technology Interaction, Faculty of Psychology, Ruhr University Bochum, Universitätsstr. 150, 44801 Bochum, Germany.
Email: martin.bordewieck@rub.de

Funding information
Ministry of Culture and Science of North Rhine-Westphalia, Germany, Digital Society research program, Grant/Award Number: 1706dgn006

Summary

Troubleshooting is a particular problem-solving process comprising error detection, fault diagnosis, and system restoration. Since automation of systems has become increasingly complex and ubiquitous, troubleshooting skills are crucial to maintain safety and security in a variety of contexts. The planned study aims at examining troubleshooting strategies and how to induce them by means of simple visual aids and concise instructions. To this end, a computerized task consisting of network troubleshooting problems will be employed in an experimental study with repeated measures. Indicators of strategy use and performance will be tested for their relation to availability and differential use of visual aids, to cognitive styles that affect how individuals deal with challenges or system information, and to cognitive processes that are involved in metacognition and executive function. The planned research is expected to help gain insights into the cognitive determinants of troubleshooting, reverse engineering, and their links to computational thinking.

KEYWORDS

computational thinking, networks, problem solving, troubleshooting, visual aids

Troubleshooting is a core component of human system interaction that is required when systems do not behave as intended. Troubleshooting is usually referred to as a problem-solving process that comprises (1) detecting aberrant system behavior, (2) diagnosing faults, and (3) repairing or replacing the faulty component (Morris & Rouse, 1985).

The automation of systems in work contexts has become increasingly complex and ubiquitous, and humans are more and more required to act as supervisors or managers rather than simple operators (Romero, Bernus, Noran, Stahre, & Berglund, 2016). Troubleshooting is an important skill to maintain efficiency, but also safety and security in a range of settings (Firesmith, 2003): Preserving a machine's proper and faultless function, for example, is crucial to prevent, detect, and react to accidental harm. In the context of computer systems, its importance is extended to warranting security in terms of preventing malicious harm such as compromised privacy or unwanted computer network access.

1 | TROUBLESHOOTING AS PROBLEM SOLVING

Troubleshooting is considered as one of the most common forms of problem solving (Jonassen & Hung, 2006). In cognitive psychology, problem solving refers to transforming a system from an initial state into a desired goal state, often by non-obvious and nonroutine means (Robertson, 2016). Problems in which both the initial and the goal states are transparent, and the operators available to obtain the goal state are limited and clearly discernible, are called *simple* problems. Frequently, at least two out of these three problem space components are non-transparent, though, which makes a problem ill-defined (or *complex*).

In troubleshooting, a problem is posed by a system's aberration from normal behavior. Usually, the goal state in troubleshooting problems is well-known, which is the expected and regular behavior of a system. Recognizing discrepancies between a system's normal and erroneous states allows for identifying the exact symptoms defining the problem, the fault diagnosis (Jonassen & Hung, 2006).

2 | THE COMPLEXITY OF TROUBLESHOOTING PROBLEMS

Missing information about a system, like the number and location of faulty elements, increases uncertainty. Causal reasoning, including iterations of hypothesis generation and testing, narrows the problem space and reduces uncertainty throughout the processes of fault diagnosis and problem solving (Shreeves, Gugerty, & Moore, 2020). Usually, the operators available for diagnosing or removing the fault are routine procedures that need to be known and practiced, while for new problems they are often indeterminate (Jonassen & Hung, 2006; Morrison & Duncan, 1988).

Such challenges within the problem space render troubleshooting a complex problem (Dörner & Funke, 2017) characterized by (1) a high number of variables involved in the problem space (complexity), (2) interrelated and mutually dependent variables (connectivity), (3) time and event dependent changes within a system (dynamics), (4) uncertainty about states and relations of variables as pointed out above (intransparency), and (5) multiple possibly conflicting goals (polytely) (Funke, 2012).

Dealing with complex systems requires the problem solver to reduce workload and uncertainty by limiting the problem space to a manageable subset of components, or by acquiring more information about the system. To this end, apt troubleshooters apply strategic knowledge by making informative tests and approaching problems in a systematic manner (Morris & Rouse, 1985; Shreeves et al., 2020). Given financial or temporal constraints, adopting efficient strategies is crucial.

2.1 | Troubleshooting strategies

In troubleshooting research, a distinction is made between global and local strategies, with the former being domain-general and helpful in reducing the problem space across tasks and fields, while the latter are domain-specific and help to implement the problem space reduction within a specific system (Jonassen & Hung, 2006).

Global strategies vary in terms of efficiency and system knowledge required.

Trial and error, for example, is common in novice troubleshooters, but hardly ever efficient. Another similarly intuitive, but inefficient approach is creating an exhaustive list of possible faults and testing them successively (Brown, Burton, & Bell, 1975; Johnson, 1991).

In contrast, a topographic strategy is generally more efficient than brute-force strategies (such as trial and error), in that it relies on structural knowledge and involves forward or backward tracing from different points in the system based on their functional states (Brown et al., 1975; Johnson, 1991). Often, the outputs of a system may be the only source of information for the initial step in fault diagnosis, in which case tracing backward is the search method of choice. While tracing backward from negative information (i.e., components that have not failed) is usually the much more efficient approach, human troubleshooters tend to focus on positive information (i.e., bad output indicative of a fault), not only when using a backward topographic strategy (Gugerty, 2007; Morris & Rouse, 1985; Shreeves et al., 2020). This tendency is also predicted by *mental models theory* stating that in faulty logic circuits, individuals typically assume the fault causing a symptom in its immediate vicinity (Goodwin & Johnson-Laird, 2005). Besides this *proximal bias*, according to the *mismatch principle*, individuals focus on components that are expected to transmit a signal, but in fact do not (Goodwin & Johnson-Laird, 2005).

Another simple and efficient global strategy, especially when the system being troubleshot is complex and only little system-specific knowledge is available, is the split-half technique that reduces the problem space by dividing it into halves successively and determining in which half the fault is located (Brown et al., 1975; Johnson, 1991).

3 | TROUBLESHOOTING IN PRACTICE

The rise of computer technology has made human computer interaction the main application field of troubleshooting. A substantial proportion of the work done by information technology (IT) engineers is troubleshooting computer hardware, software, and networks. In IT security, debugging and troubleshooting is of adversarial nature: Both attackers and defenders attempt to detect and troubleshoot anomalies or vulnerabilities, but the latter do so to protect information and system integrity whereas the former aim to intrude, modify, or compromise systems. Troubleshooting knowledge and skills are key to reverse engineering, an integral component of hardware hacking (Fyrbiak et al., 2017), and hackers' thinking with its particularities can serve as a model of highly proficient troubleshooting ability.

4 | INTERINDIVIDUAL DIFFERENCES IN TROUBLESHOOTING

Apart from practical experience and technical skills, successful troubleshooting depends on the abilities to (1) choose and perform informative tests, and (2) employ efficient strategies (Jonassen & Hung, 2006; Morris & Rouse, 1985; Shreeves et al., 2020). An abundance of research on troubleshooting training methods highlights merits of explicit instruction vs. self-discovery, or the type of knowledge provided to foster the transition from conceptual and rule-based approaches to procedural and experiential approaches (Carlson, Lundy, & Schneider, 1992; Ham & Yoon, 2007; Kostopoulou & Duncan, 2001). However, cognitive properties and thinking styles of individuals that might predict the development of global strategic knowledge in troubleshooting are largely understudied.

Some evidence suggests that reflective vs. impulsive thinking affects the extent to which troubleshooters work efficiently and improve with practice (Morris & Rouse, 1985). Strategy use in troubleshooting has also been linked to fluid intelligence and thinking dispositions such as open-mindedness and typical intellectual engagement (Shreeves et al., 2020), as well as metacognitive processing such as executive control, self-regulation, and monitoring processes (Perez, 1991). Furthermore, troubleshooting performance is susceptible to systems' visual characteristics, size, and complexity, and time constraints, all of which variably affect demands put on cognitive processes such as working memory, processing speed, and executive function (Jonassen & Hung, 2006; Morris & Rouse, 1985; Toms & Patrick, 1989).

4.1 | Computational thinking and troubleshooting

Computational Thinking (CT) refers to "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" (Wing, 2017, p. 8). Importantly CT is not limited to computer programming but represents a general, systematic approach to problem solving, and comprises six essential facets: (1) decomposition, (2) abstraction including data collection and analysis, pattern recognition and modeling, (3) algorithms comprising algorithm design, parallelism, efficiency and automation, (4) debugging, (5) iteration, and (6) generalization (Shute, Sun, & Asbell-Clarke, 2017).

These general components of CT are mirrored in the applied skillsets of professionals who routinely perform troubleshooting tasks. For example, according to Summers (2015) hackers are characterized by: (1) a high tolerance of ambiguity and appreciation of challenges posed through risk, uncertainty, vagueness, and chaos; (2) superior strategizing and decision making skill; (3) critical or reflective thinking and introspection; (4) use of visualizations (flow-charts, diagrams, etc.) to organize and externalize their reasoning and thinking on a more abstract level; (5) patterning as key technique for detecting regularities and anomalies; (6) ability to decompose a problem into sub-problems that can be worked on in parallel; (7) ability to explore the causal mechanisms that underlie a structure; (8) advanced debugging skill; and (9) high creativity.

Since troubleshooting is associated with the IT domain, we use CT and the particularities in the thinking of highly proficient IT troubleshooters as a framework for determining the cognitive properties and thinking styles that might be crucial to troubleshooting. Thus, we attempt to map the components of hacking (as IT expert troubleshooting), troubleshooting (as conceptualized in this study), and corresponding cognitive variables in a joint framework (see Table 1).

4.2 | Dealing with challenges

4.2.1 | Tolerance of ambiguity

Troubleshooting is characterized by a lack of information that is necessary to understand the system and its malfunction. Tolerance of ambiguity is, broadly speaking, the ability to continue working on a problem without knowing the answer, knowing how or whether it might be obtained, how many possible answers exist, or whether there even is one. Highly proficient IT troubleshooting experts do not only have a high tolerance of ambiguity but even find pleasure in the challenge of reducing ambiguity and uncertainty (Summers, 2015).

TABLE 1 Mapping of computational thinking, hacking and troubleshooting with cognitive styles and processes

Computational Thinking (Shute et al., 2017)	Hacking (Summers, 2015)	Troubleshooting (Jonassen & Hung, 2006; Morris & Rouse, 1985; Perez, 1991)	Cognitive styles & processes
Decomposition	**Decomposing** • Visualizing	**Problem space reduction** • Problem (re)structuring	Attention control Systemizing & Attention to detail
Abstraction • Data collection & analysis • Pattern recognition • Modelling	**Visualizing** • Critical/reflective thinking • Patterning • Exploring causal mechanisms	**Construct conceptual model** • Hypothesis generation and testing ○ Make informative tests ○ Information gathering • Understanding structure and functioning	Numeracy Reflective thinking Attention to detail Systemizing
Algorithms • Algorithm design • Parallelism • Efficiency • Automation	**Strategizing & Decision making** • Decomposing • Patterning • Exploring causal mechanisms	**Strategy use** • local strategies • Strategy switching • Self-monitoring & self-regulation • Understanding structure and functioning	Metacognition Controlling divergent & convergent thinking Inhibitory & Attention control Systemizing
Debugging	**Debugging** • Tolerating ambiguity	**Fault diagnosis & isolation** **Fix & repair**	Need for cognition Ambiguity tolerance
Iteration	**Critical/reflective thinking** • Exploring causal mechanisms	**Hypothesis generation and testing**	Reflective thinking Need for cognition
Generalization	**Patterning** • Visualizing	**Transfer** • global strategies	Executive function (executive control)

Note: This table represents an attempt to map the components essential to troubleshooting and hacking, as well as their assumed corresponding cognitive styles and processes, to the facets of Computational Thinking as a framework of troubleshooting in the IT domain. Please note that this attempt is considered to be neither complete nor necessarily accurate. It is solely a conceptual framework for generating hypotheses within the planned research. The mapping represents the major correspondences in a row-wise arrangement. However, there are substantial overlaps and relationships between facets and components across rows and columns.

4.2.2 | Reflective thinking

Whether troubleshooters employ a brute force approach like randomly replacing possibly faulty components, or a more deliberate strategy might depend on their tendency to engage in reflective thinking, a decision making style based on the assumption of mainly two cognitive processes referred to as system 1 and system 2 (Frederick, 2005). While system 1 is characterized by spontaneous and automatic processing, system 2 encompasses rather slow and elaborative processes, that are computationally costlier, but also more accurate. A higher degree of reflective thinking in terms of relying rather on system 2 enables overcoming a default and intuitive solution approach in favor of further elaboration and reflection (Toplak, West, & Stanovich, 2014). Whether reflective thinking, however, can be considered a stable thinking disposition or a rather changeable characteristic is contested.

4.2.3 | Need for cognition

Intrinsic motivation to engage in and enjoy cognitive activity and effort may be reflected in individuals' need for cognition (Cacioppo & Petty, 1982). Putting effort in reducing ambiguity and employing elaborative strategies in troubleshooting clearly requires this kind of motivation. Expert troubleshooters usually enjoy exploring the causal mechanisms underlying a system (Summers, 2015), and their need for cognition is therefore likely related to their strategic behavior.

4.3 | Dealing with (formal) system information

4.3.1 | Systemizing

Systemizing is a disposition to analyze, understand, predict, control, and construct rule-based systems of any type that exhibit an underlying tripartite structure: input – operation – output (Baron-Cohen, Richler, Bisarya, Gurunathan, & Wheelwright, 2003). This is an integral component of CT and has been linked to interest or working in science and engineering domains that deal with troubleshooting on a regular basis (Billington, Baron-Cohen, & Wheelwright, 2007; Feist, 2012). There is also evidence for a link between systemizing and deliberation, i.e. reflective thinking (Brosnan, Hollinworth, Antoniadou, & Lewton, 2014), as well as code-breaking, an activity that requires identification of vulnerabilities (Harvey, Bolgan, Mosca, McLean, & Rusconi, 2016).

4.3.2 | Attention to detail

In troubleshooting, a fine-grained understanding of a system and its causal mechanisms might require profound attention to detail, which has been linked to systemizing in (high-functioning) autism (Baron-Cohen, Ashwin, Ashwin, Tavassoli, & Chakrabarti, 2009; Grove, Baillie, Allison, BaronCohen, & Hoekstra, 2015), as well as to threat detection skill of hackers and non-hackers (Harvey et al., 2016; Rusconi, Ferri, Viding, & Mitchener-Nissen, 2015; Rusconi, McCrory, & Viding, 2012). Furthermore, in line with the rather anecdotal observation that people with autism are drawn to and particularly talented in the STEM fields (Lorenz, Frischling, Cuadros, & Heinitz, 2016; Wei, Yu, Shattuck, McCracken, & Blackorby, 2013), slightly more pronounced autistic traits (including attention to detail) have been found, for example, in hackers (Schell & Melnychuk, 2010) or mathematicians and computer scientists (Baron-Cohen, Wheelwright, Skinner, Martin, & Clubley, 2001; Ruzich et al., 2015). However, given the limited empirical evidence, we emphasize that assuming these relationships is somewhat speculative.

4.3.3 | Numeracy

Most troubleshooting, particularly when conducted in the IT domain, deals with automated systems that are described in formal language based on mathematics and Boolean algebra. We, therefore, assume that another essential component of advanced troubleshooting skill is mathematical thinking which is reported to have substantial overlap with CT (Sneider, Stephenson, Schafer, & Flick, 2014). Furthermore, decision making in this context requires risk estimation by making probabilistic judgments. The need to make these has been repeatedly found to pose a problem to most troubleshooting individuals, even to more skilled ones (Morris & Rouse, 1985).

4.4 | Metacognition

4.4.1 | Divergent and convergent thinking

While convergent thinking refers to a systematic and analytic mode of reasoning directed to finding one specific correct solution to a problem, divergent thinking represents a rather open and less narrowly focused mode of thinking that widens the problem space representation and helps taking into consideration a broad variety of solution approaches (Wolf & Mieg, 2010). Control over engaging in and switching between these two thinking modes is supposed to be a critical factor in complex problem solving (Dörner, Kreuzig, Reither, & Staudel, 1983). This is, hence, also likely to serve appropriate strategic behavior in troubleshooting.

4.4.2 | Executive function

Suppressing irrelevant visual information through executive functions like inhibitory control and reallocation of cognitive resources might be a crucial metacognitive ability in fault diagnosis. Executive control (monitoring and inhibition), for example, has been linked to reflective thinking and decision making albeit these effects were in part mediated by fluid intelligence and/or numeracy (Del

Missier, Mäntylä, & de Bruin, 2012). Particularly executive control and monitoring processes have been identified as playing important roles in selecting strategies and modifying them as needed in the course of troubleshooting (Perez, 1991).

5 | NETWORK TROUBLESHOOTING PROBLEMS

In an effort to identify cognitive determinants of general troubleshooting skills, Rouse (1978) introduced a troubleshooting task in an abstract format that is representative of various troubleshooting situations. These network troubleshooting problems (NTPs) usually consist of a square number of nodes (often called points or vertices in graph theory) in a rectangular arrangement connected by lines (like wires in an electrical circuit; often links or edges in graph theory) and acting as Boolean AND gates that pass 0's or 1's in left-to-right information flow (see example in Figure 1). Each gate passes on a 1 only if it receives 1's as input exclusively, and passes on a 0 if at least one of the inputs is a 0. Usually, all the leftmost inputs into the network are 1's, so that all the rightmost outputs are supposed to be 1's, too. The task requires troubleshooters to determine the location of one faulty node, that causes one or more 0 outputs on the right end of the network, based on reasoning and information gathering through testing lines and nodes for their signal states.

Simulating the temporal and monetary constraints usually faced during troubleshooting in real life, troubleshooters resolve NTPs under a time limit and are charged for any line and node tests in virtual currency (whereas reasoning, of course, is free). Further reflecting real structural properties, testing lines is less expensive than testing (or replacing) nodes (Gugerty, 2007; Shreeves et al., 2020). Strategy use can be inferred from the time course and choice of the troubleshooters' test moves.

In NTPs, the most commonly used type of topographic strategies involves focusing on the symptomatic output and tracing backward along the nodes feeding into it (from here on referred to as backtracking). In the Figure 1 example, this brute force approach leaves a subset of 18 possibly faulty nodes (inclusion or backtracking subset, dashed line frame in Figure 1), requiring a fairly large number of tests to identify the faulty node even with split half. This sizeable subset can be reduced significantly by taking into account that nodes feeding into the 0 output cannot possibly be faulty if they feed into a 1 output as well, which is the reasoning that underlies elimination strategy (resulting elimination subset, dark gray nodes and their connecting lines in Figure 1). Exclusive use of only one of these strategies, however, is not common; rather, hybrid strategies are often employed.

6 | VISUAL AIDS FOR SOLVING NTPS

In line with the general finding that apt troubleshooters make fewer tests but take longer for each of them, while poor troubleshooters conduct more tests with shorter time intervals between them (Morris & Rouse, 1985; Rouse & Rouse, 1979; Shreeves et al., 2020), Gugerty (2007) found a strong negative correlation between the

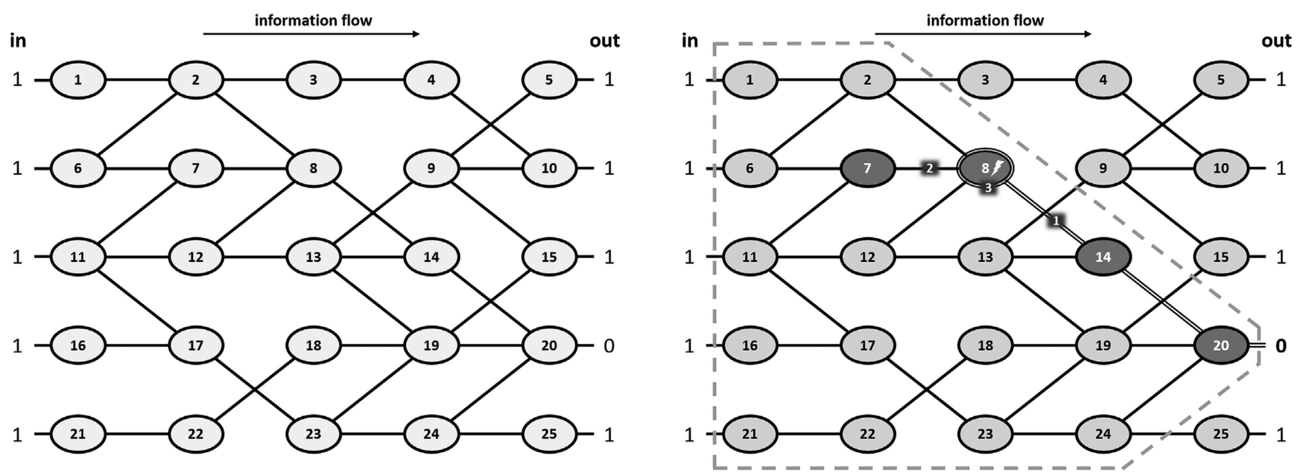

FIGURE 1 Left panel—Example of a network troubleshooting problem (NTP) with 25 nodes and 32 connecting lines similar to the networks used by Gugerty (2007). Information is passed through the network as 0's and 1's in left-to-right direction. Nodes function as AND-gates so that the 0 output of node 20 must be caused by a faulty node (7, 8, 14 or 20 here). Right panel—The same NTP with various information highlighted: The faulty node (white lightning bolt) and the lines carrying a 0 signal are marked with double lines. The nodes that constitute the subset of nodes that cannot possibly be faulty (exclusion subset) are marked in light gray, the nodes that remain after applying the elimination strategy (elimination subset) are marked in dark gray, and the nodes that would be considered for testing according to the backtracking strategy (backtracking subset) are framed by the light gray dashed line. The white numbers in the small dark rectangles represent the minimal number of tests in the required order. Using elimination, troubleshooters start from the 1 outputs on the right and successively eliminate all nodes feeding into them (exclusion subset) obtaining the elimination subset. The combination of elimination with split half represents the optimal algorithm to identify the faulty node (node 8 in the example) largely based on reasoning and a minimal number of tests (here, two line tests and one final node test)

proportion of elimination moves (tests within the elimination subset, i.e., nodes left after using elimination, see Figure 1) and the total number of moves. Thus, the more elimination is used, the fewer tests are needed to identify the faulty node. Further, a positive correlation between the proportion of elimination moves and time per move may be due to elaborate reasoning and reflection prior to conducting tests (Gugerty, 2007).

This points to the conjecture that elimination might be more demanding in terms of working memory resources (Shreeves et al., 2020). Working memory load is assumed to be affected by complexity as well as by visual characteristics of the troubleshooting problem, and accordingly, troubleshooters are known to benefit from visual aids and graphical representations (Morris & Rouse, 1985). Evidence for this effect was reported by Toms and Patrick (1989): In their study using NTPs, participants instructed to employ elimination (vs. backtracking) significantly benefitted from utilizing visual aids in form of node markings. It is of note that this effect was more pronounced for visual compared to verbal memory aids. Again, the use of visual aids is also mirrored in the work of professional troubleshooters, for example, hardware reverse engineers (Rematska & Bourbakis, 2016; Wallat et al., 2019).

Externalizing visually represented information to (re)structure the problem space (1) reduces working memory load and frees up resources for other cognitive processes such as reasoning and reflective thinking, and (2) enables shifting attention to problem elements that are strategically more relevant.

Likewise, visualizing strategy-relevant information in NTPs by the simple means of marking components may facilitate computational or cognitive offloading, for example, the use of physical action to alter the information processing requirements of a task in order to reduce cognitive demand and computational effort, and to overcome capacity limitations (Risko & Gilbert, 2016; Scaife & Rogers, 1996). This main benefit of externalizing visual information is a quite common finding in various domains of problem solving such as mathematical or probability problem solving (Corter & Zahner, 2007; Zahner & Corter, 2010), chemistry problem solving (Dickson, Thompson, & O'Toole, 2016) or hacking (Summers, 2015).

Due to cognitive offloading, external visualization allows for exploring the problem space more thoroughly and helps to form a workable problem space representation as well as to modify and update it. Problem-solving performance is very sensitive to representation formats and visual features of the problem, especially for novices who mostly rely on surface features (Chi & VanLehn, 2012; Simon & Hayes, 1976). Hence, forming an appropriate problem representation is a crucial part of the early problem solving stage of defining and recognizing problems and, therefore, decisive for navigating the problem space and finding a solution (Pretz, Naples, & Sternberg, 2003).

Strategy use on the NTPs is commonly induced by explicit instruction or by specific training. The knowledge taught during the training in Gugerty (2007) was based on the types of knowledge that a production system model required to solve NTPs. It was concordantly conjectured that teaching the same knowledge to human troubleshooters without explicitly explaining the elimination strategy would be similarly beneficial. Gugerty (2007), however, concluded that the model-based training might actually not have taught the cognitive components of the elimination strategy, but rather caused an attentional focus on the kind of knowledge that facilitates learning the elimination strategy and that had already been available in the initial network-task instructions. Priming this prior knowledge through the specific training might have boosted participants' learning processes in favor of elimination.

Shreeves et al. (2020) found evidence for strategy dependent marker use in NTPs indicating that markers are more likely to be used for strategies with higher demands on working memory. Based on the aforementioned, however, we assume that the benefits of visual aiding through marking nodes in the NTPs might go beyond mere working memory aid by additionally directing troubleshooters' attention to information relevant for optimally solving the NTPs. By combining marker use with a concise instruction (and practice) on which components to mark, a guided focus on nodes that can be excluded from testing for being impossibly faulty might, for example, create an attentional focus on the kind of knowledge and reasoning underlying elimination, and favor learning and use of this advantageous strategy.

7 | AIM OF THE PRESENT STUDY

The aim of this study is mainly threefold:

1. Conceptually replicating the findings of Gugerty (2007) and Shreeves et al. (2020) concerning testing behavior and performance on NTPs.
2. Exploring the capability of visual aids such as colored markers combined with concise instructions on how to apply them to induce strategy use.
3. Examining variables related to cognitive properties, processes, and styles possibly involved in solving NTPs as well as their interrelations.

The proposed study is an attempt to investigate how simple (re) structuring of a problem space by means of visual aids impacts and possibly induces strategy use on NTPs instead of having troubleshooters undergo more extensive prior training.

To this end, we adopt the NTPs used by Rouse (1978) and Gugerty (2007) with their abstract, circuit-like appearance (that requires little domain-specific knowledge) and their merits in terms of experimental variability/versatility and control, applying some modifications. Strategy-specific and model-based training prior to solving NTPs will be omitted, and instead visual aids in the form of colored markers will be made available to the troubleshooters, accompanied by some basic instructions for their usage (followed by a guided practice trial): Exclusion of impossibly faulty nodes (Exclusion Condition, reflecting elimination strategy), inclusion of possibly faulty nodes (Inclusion Condition, reflecting backtracking strategy), no particular instruction (No Instruction

Condition). All troubleshooters further work on NTPs in a baseline phase without markers.

Our hypotheses are based on the following rationale: In previous studies, use of elimination was induced either by explicit instruction (e.g., Toms & Patrick, 1989) or by teaching the knowledge underlying elimination (Gugerty, 2007). As conjectured by Gugerty (2007), the training effect can be explained in terms of priming this knowledge (see above), which points to more implicit cognitive processing. It is directly stimulating this implicit cognitive processing that the planned study aims for by drawing attention to strategy-relevant knowledge via differentially instructed marker use.

In both elimination and backtracking, participants consider the remaining set of potentially faulty nodes when they select their next troubleshooting test move. However, the key distinction between the two is that in backtracking, participants make diagnostic (effect to cause) inferences mostly from the network components that indicate an error state (transmit 0's). In elimination, on the other hand, participants make diagnostic inferences from components that indicate normal operation (1's) and from components that indicate an error state (0's). Notably, due to the networks' structure, starting point to these inferences can only be the information given through the network outputs (one of which indicates an error state, while four transmit a 1).

Therefore, we believe that, via instructed marker use, drawing troubleshooters' attention to components that they predict to function normally and, thus, not to be faulty (Exclusion Condition) facilitates the reasoning that underlies elimination (components passing 1's can only receive 1's as inputs, i.e., feeding nodes are necessarily functional), and increases the probability of making elimination moves within the remaining subset of possibly faulty components.

In contrast, drawing troubleshooters' attention to components that they consider as possibly faulty and likely to indicate an error state when tested (Inclusion Condition) might promote maintaining attention to components feeding into the symptomatic 0 output while neglecting the type of information that would lead to eliminating some of them (i.e., the components feed into a 1 output at the same time). This leaves troubleshooters with a larger subset of remaining possibly faulty nodes increasing the probability of making test moves other than elimination moves.

8 | HYPOTHESES

Based on the foregoing considerations, we formulate the following hypotheses (primary hypotheses in regular font, *auxiliary hypotheses in italics*):

8.1 | Manipulation check

1) Marker use differs among experimental conditions according to the instruction. Participants in the…

a) … Exclusion Condition mark relatively more nodes within the exclusion subset compared to the other two conditions.

b) … Inclusion Condition mark relatively more nodes within the backtracking subset compared to the other two conditions.

c) … No Instruction Condition mark relatively more nodes within the backtracking subset compared to the Exclusion Condition.

8.2 | Strategy use

2) Strategy use as reflected in the proportion of elimination and backtracking moves is altered by the availability of markers differentially according to marker use instruction in the experimental conditions. Participants in the …

a) … Exclusion Condition show a higher increase in the proportion of elimination moves from the baseline to the experimental trials compared to the other two conditions.

b) … Inclusion Condition show a lower decrease in number and proportion of pure backtracking moves (i.e., *backtracking moves outside the elimination subset) from the baseline to the experimental trials compared to the other two conditions.*

3) The proportion of elimination moves as an indicator of strategy use correlates positively with the cognitive variables tolerance of ambiguity, reflective thinking, need for cognition, systemizing, attention to detail, numeracy and metacognitive processing/executive control across groups in the …

a) … baseline trials.

b) … *experimental trials*.

8.3 | Performance

Performance here, if not stated otherwise, refers to (1) trial success and success rate, i.e. number and proportion of successful identifications of faulty nodes, (2) virtual currency spent on a network and virtual currency saved in total, and (3) time spent on a network both including and excluding unsuccessful trials.

4) Performance differs according to marker use instruction in the experimentalconditions. Participants in the …

a) … Exclusion Condition perform better compared to the other two conditions.

b) … No Instruction Condition perform better compared to the Inclusion Condition.

5) Performance correlates with strategy use as reflected in the proportion of elimination moves across groups and trials. The proportion of elimination moves correlates …

a) … positively with time to initial test move, time per test move, number and proportion of correct node tests, and virtual currency saved across trials.

b) … negatively with time spent on a network, number and proportion of false and missed node tests, number of line tests, and virtual currency spent on a network.

6) Performance correlates positively with the above mentioned cognitive variables across conditions …

a) … in the baseline trials.

b) … in the experimental trials.

7) Performance is altered by the availability of markers. Participants perform better in the experimental trials than in the baseline trials …

a) … across conditions.

b) … within each condition.

8) Judgments of mental, physical and temporal demand, as well as of performance, effort and frustration differentially correlate with performance during the preceding trials.

9 | METHODS

We use a mixed design with a three-level between-subjects factor and a two-level within-subjects factor. All three groups share a baseline condition (no markers available) and undergo different experimental conditions (with markers available) (see Figure 2 for an overview of the experiment's procedure). The three experimental conditions differ with regard to the instruction on how to mark nodes: exclusion of impossibly faulty nodes (Exclusion Condition), inclusion of possibly faulty nodes (Inclusion Condition), no particular instruction (No Instruction Condition).

Dependent variables can be divided into two classes: *strategy use* and *performance*, while a third class of variables comprises the cognitive properties assessed through a test battery and self-report questionnaires (see below).

For a detailed overview of these variables see Appendix 1, Supplementary Information.

9.1 | Sample

9.1.1 | Sample size

The sample size required for detecting small to intermediate effect sizes was determined through a-priori power analyses using the R package *pwr* (Champely et al., 2020) and PANGEA (Westfall, 2015).

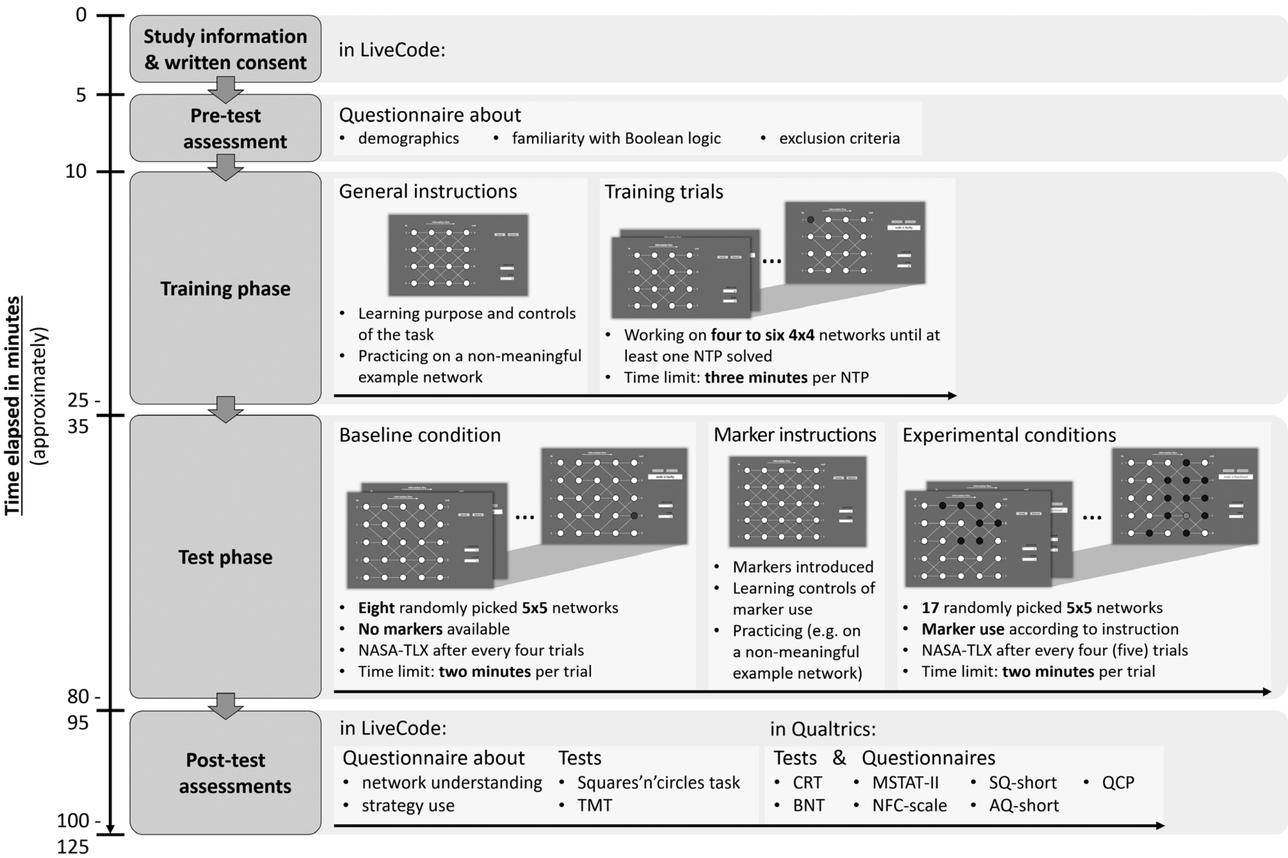

FIGURE 2 Procedure and time course of the planned study's experimental sessions. First, participants receive the study information and give explicit consent. After filling in a questionnaire about their demographics, familiarity with Boolean logic and information related to exclusion criteria, participants get the general task instructions followed by familiarizing themselves with the controls on a non-meaningful example network. Participants who successfully solve at least one of four to six training NTPs proceed with the test phase consisting of eight networks in the baseline condition without markers, followed by marker use instructions with a brief practice phase and 17 NTPs with marker use according to one of the three experimental conditions (exclusion, inclusion and no instruction). Finally, participants provide information about their understanding of the networks and their strategy use, and work on a number of tests and questionnaires that assess cognitive variables relevant to troubleshooting performance and strategy use

See Appendix 2 for the analysis scripts and outputs of *pwr*. For PANGEA, see screenshots from the inputs and outputs in Appendix 3.

Calculations were based on effect sizes reported in Gugerty (2007) and theoretical considerations for smallest effect sizes of interest (SESOIs). Gugerty (2007) reports high correlations between indicators of performance and of strategy use: $r = -0.90$ for number of tests needed to identify the faulty node and the percentage of elimination moves. Similarly, there was a positive correlation of $r = 0.76$ between elimination use and time per move as another performance indicator. To replicate these observations, and further examine the relationships between strategy use and various cognitive variables, we plan to detect small to intermediate effect sizes of $r > 0.3$ for one-tailed correlation tests at 90% power, yielding a minimal sample size of 92 (which would allow us to detect r's > 0.26 at 80% power and r's > 0.33 at 95% power).

Gugerty (2007) reports effect sizes of Cohen's $d \geq 0.8$ for the between-subjects effect of strategy training condition on the percentage of elimination moves as the main indicator of strategy use, which is also the most relevant dependent variable in the present study. However, differences in methods of inducing strategy use in the present study and Gugerty (2007), as well as other studies employing similar network tasks (Carlson et al., 1992; Ham & Yoon, 2007; Kostopoulou & Duncan, 2001; Toms & Patrick, 1989), need to be accounted for. Thus, we benchmark on the smallest between-subjects effect of strategy training condition on any dependent variable reported in Gugerty (2007), which was Cohen's $d = 0.47$ for the effect on number of tests (Cohen's d among group comparisons ranging from 0.47 to 1.15).

Using PANGEA (Westfall, 2015), a one-way between-subjects ANOVA with three levels each consisting of 31 participants (93 in total) and 17 replicates (17 trials in experimental conditions, see below) yielded a uniform sensitivity to effect sizes of $d = 0.51$ for between-subjects contrasts at 90% power ($d > 0.44$ at 80%, $d > 0.57$ at 95% power).

The overall advantage of marker availability will be tested with a within-subjects one-sided ANOVA with two levels (baseline, experimental condition). Across all 93 subjects, we would be able to detect mean differences of $d > 0.30$ at 90% power ($d > 0.26$ at 80% power, $d > 0.34$ at 95% power). Please note that for this a-priori power analysis, only eight of 17 replicates in the experimental conditions were used in order to balance the number of observations in both of the factor levels, since the baseline condition only comprises eight trials. For the actual analysis of the data, all replicates will be used.

Finally, in order to account for an interaction of marker availability and marker use instruction, we computed sensitivity analyses for a 2 (no markers vs. markers) $\times$ 3 (exclusion vs. inclusion vs. no instruction) mixed-design repeated ANOVA, resulting in $d > 0.37$ at 90% power ($d > 0.32$ at 80% power, $d > 0.41$ at 95% power) for an interaction effect: We expect that conditions will differ concerning strategy use or performance in the experimental trials, but not the baseline trials.

Note that the sampling plan with a target $N = 93$ is optimized for the primary hypotheses. Any other hypothesis tests might have less power, so that their results will be considered and interpreted only most cautiously. For a table summarizing the a-priori power analyses see Appendix 4.

To account for data loss due to exclusion criteria as specified below or in Appendix 5, we will oversample our target of $N = 93$ and collect data from an initial sample size of 110 participants. However, with regard to randomized assignment of conditions data collection will continue and further participants will be recruited, if necessary, to reach our target sample of at least 31 valid and complete cases in the smallest group.

9.1.2 | Sample characteristics

We aim at obtaining a high variance with respect to participants' strategy use, task performance, and cognitive abilities. However, a few restrictions will apply to the sample: Participants will be recruited at Ruhr University Bochum, surrounding institutions of higher education and relevant online platforms (documented in the Appendix), such that most will be students of varying age. Students from STEM fields (including computer science) will be excluded, as we expect ceiling effects in their performance on the NTPs of the complexity level used in this study. For the same reason, we exclude students with advanced programming experience (as defined in Appendix 5). General inclusion criteria are normal or corrected-to-normal sight and being fluent in German language. Participants should report being free from neurological or cognitive impairment. Written consent will be obtained from every participant before starting the experiment.

Participants will either receive 10€/hr or course credit (where applicable and desired). In case they cannot complete (e.g., due to exclusion criteria) or quit the study prematurely, participants will be recompensed on a pro-rata basis.

The study is approved by the ethics committee at the department of psychology, Ruhr University Bochum (application no. 493).

9.2 | Materials

All the codes used in the experiment, along with this manuscript, appendix, data, and analysis scripts will be made available in an open repository (see Appendix under https://osf.io/xrk2j/).

9.2.1 | Task and stimuli

We use NTPs adopted from Gugerty (2007), originating from Rouse (1978), as introduced above. Each of the 25 five-by-five networks is designed to contain exactly one faulty node causing exactly one 0 output.

Network complexity is controlled for by setting the mean number of connecting lines in the networks to 32 (varying from 31 to 33) with 7 to 9 lines per column providing slight variation without affecting complexity too much, since this is a major source of variation in problem difficulty (Morris & Rouse, 1985; Rouse & Rouse, 1979). Further, the locations of the faulty nodes as well as the 0 outputs are distributed uniformly across all 25 networks as listed in Appendix 6.

Since the elimination subset is always a (sub-)subset of the backtracking subset, with the only difference across networks being the ratio of their sizes, we ensure that elimination is consistently superior to backtracking by making the elimination subset contain at the utmost 45% of the number of nodes and lines within the backtracking subset but always at least three nodes and two lines in any network. Finally, the networks and the virtual currency budget are designed and set in such a way that using elimination always results in savings on each trial.

The networks' structures and properties are specified in a script (Appendix 7) and finally visualized during the experiment using the software LiveCode (2020). See Figure 3 for an example of how networks are presented in the actual experiment.

9.2.2 | Tests and questionnaires

Interindividual differences in several cognitive dimensions described in the introduction will be assessed by the use of well-established psychological test paradigms and self-report instruments which are listed along with their properties in Table 2. All tests and questionnaires will be presented in German. Where missing, translations made by the authors' research group are provided. Short descriptions of the instruments as well as english versions of the tests and questionnaires are attached in Appendix 8.

9.3 | Procedure

An overview of the sessions' procedure and time course is depicted in Figure 2.

The study will be conducted entirely online using applications developed in LiveCode (2020) for the NTP experiment followed by two executive function tests, and the online survey software Qualtrics (https://www.qualtrics.com/) for the remaining tests and questionnaires. After receiving the required weblink participants can complete the study at any time and place, while a fully double-blind procedure is warranted, that is, neither participant nor experimenter knows which condition a participant is assigned to.

9.3.1 | Pretest and training phase

After filling in a pre-test questionnaire about demographics and knowledge relevant to the study (see Appendix 9) the training phase is initialized providing the general instructions for the task (see Appendix 10) and opportunity to get familiarized with the testing controls and the virtual currency (see Appendix 11 for a more detailed description of the procedure and the functioning of the computerized task). Finally, four to six training networks with 4x4 nodes and a time limit of 3 min each are presented in random order. Only during the training phase participants receive feedback about an appropriate number of test moves and the position of the faulty node, if not identified. Participants failing to solve any of these up to six NTPs will be excluded from further participation.

9.3.2 | Test phase

Participants who have completed the training phase successfully proceed with the test phase and are informed that the test networks differ from the training networks only in consisting of 5×5 nodes and in

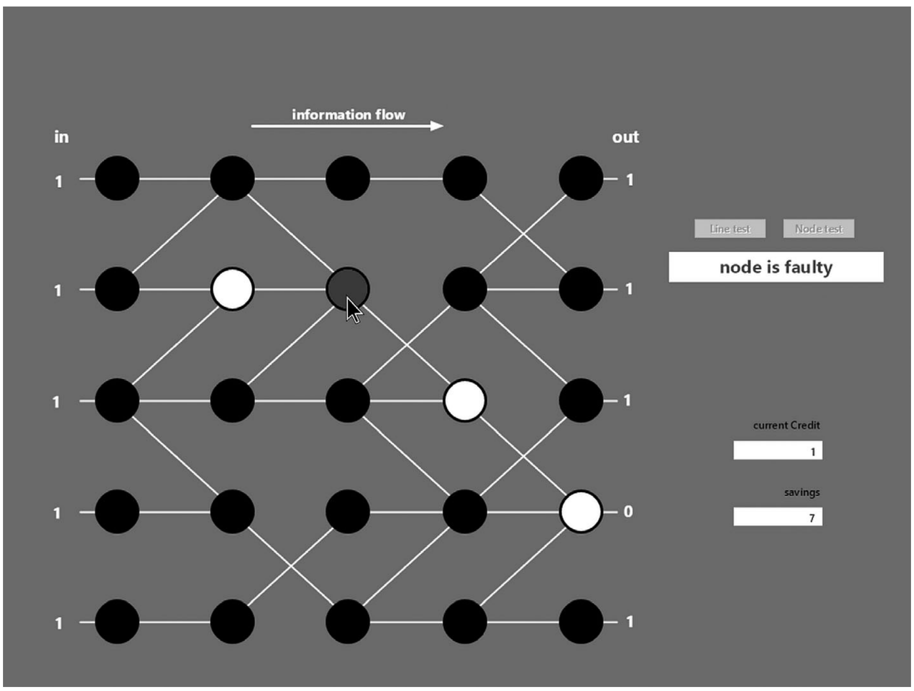

FIGURE 3 Screenshot from the experiment in LiveCode. The network is the same as in Figure 1. One can see the black markers (blue in the experiment) and the current balance as well as the savings across trials. The screenshot was taken in the moment of identifying the faulty node during node test mode, as can be told from the mouse cursor position on the node in row 2, column 3

TABLE 2 List of instruments used for assessing the cognitive variables examined in the study

Cognitive variable	Instrument	Details (use in the present study)	Number of items	Response scale
Tolerance of ambiguity	MSTAT-II (McLain, 2009)	Multiple Stimulus Types Ambiguity Tolerance Scale-II - derived from the MSTAT-I (22 items), reduced to 13 items	13	Likert scale from 1 (strongly agree) to 7 (strongly disagree)
Reflective thinking	CRT (Frederick, 2005)	Cognitive Reflection Test - CRT version extended to seven items by Toplak et al. (2014)	7 (4 + 3)	One correct answer
Need for cognition	NFC scale (Cacioppo & Petty, 1982)	Need for Cognition Scale - German translation (Bless, Wänke, Bohner, & Fellhauer, 1994) of a validated English 6-item version (de Holanda Coelho, Hanel, & Wolf, 2018)	6	Likert scale from 1 (extremely uncharacteristic of me) to 5 (extremely characteristic of me)
Systemizing	SQ (Baron-Cohen et al., 2003)	Systemizing Quotient - short version (Samson & Huber, 2010)	13	Likert scale from 1 (strongly agree) to 4 (strongly disagree)
Attention to detail	AQ (Baron-Cohen et al., 2001)	Autism-Spectrum-Quotient - attention to detail subscale (10 items) from the AQ plus the other four subscales à two items from short version (Allison, Auyeung, & Baron-Cohen, 2012)	10 (+ 8)	Likert scale from 1 (definitely agree) to 4 (definitely disagree)
Numeracy	BNT (Cokely, Galesic, Schulz, Ghazal, & Garcia-Retamero, 2012)	Berlin Numeracy Test - commonly used adaptive testing mode	2–3 (of 4)	One correct answer
Divergent and convergent thinking	QCP (Dörner et al., 1983)	Questionnaire for cognitive process variables - two subscales "controllability of the activation of divergent and convergent thinking" and "controlled divergent thinking"	18 (10 + 8)	Likert scale from 1 (does not apply to me) to 7 (applies to me)
Executive control	Squares 'n' circles task (custom made)	Cancelation task: Processing speed/ accuracy and inhibitory control - task: manually mark as many target stimuli as possible within a time limit in 12 arrays of 48 stimuli each	12 sets à 48 stimuli	Completion time and errors
	TMT (Reitan, 1955, 1958)	Trail Making Test - Part A (processing speed) and Part B (executive control)	2	Completion time
Metacognitive judgments	NASA-TLX (Hart, 2006; Hart & Staveland, 1988)	National Aeronautics and Space Administration—Task Load Index - judgments of perceived workload demands as well as perceived performance, effort and frustration related to a task.	6	20 point scales anchored with "low" and "high," or "good" and "poor"

Note: Main characteristics of the used instruments. Find the descriptions of these instruments along with their English versions in Appendix 8.

a time limit of now 2 min. They are asked to work as quickly and as accurately as possible, and to work most efficiently in terms of making as few tests as possible and watching their virtual currency balance on each trial (see Appendix 10).

Participants first work on eight baseline trials without markers. The availability of markers in the subsequent 17 trials of the experimental conditions is introduced through another short instruction, followed by an opportunity to get used to the controls of the marking function (see Appendix 11 for a more detailed description). Before proceeding with the final 17 trials participants receive the individual marker use instructions (see Appendix 10) depending on the experimental condition: (a) to mark only nodes they can exclude from the subset of possibly faulty nodes (Exclusion Condition), (b) to mark only nodes they can include in the subset of possibly faulty nodes (Inclusion Condition), or (c) to use markers as visual aids in whatever way seems helpful to them (No Instruction Condition). To make sure that participants pay attention to and understand the instructions, they are complemented by a short presentation on different possible approaches of using markers, a negative example of non-compliant marker use, an animated example of correct marker use, and finally a dedicated practice trial with feedback. Explicating the strategies, however, remains avoided. All participants are strongly encouraged to make extensive use of markers as visual aids before conducting any test, in order to increase their efficiency on the task.

Throughout the test phase NTPs are interspersed with the NASA-TLX after every block of four trials (five trials in the last block).

9.3.3 | Post-test phase

After finishing the experiment, a short questionnaire asks participants about their network understanding as well as about their strategy and marker use (see Appendix 12). Then, still in the LiveCode (2020) application, they complete two executive function tests in random order, followed by the remaining tests and questionnaires in Qualtrics (https://www.qualtrics.com/) assessing cognitive properties in random order.

Finally, following a short debriefing, participants receive final instructions on how to obtain their compensation and are acknowledged for their participation.

Find a more detailed description of the procedure and of features of the experimental setup such as randomization, participant code generation and handling of non-compliant participants in Appendix 11.

9.4 | Measuring strategy use

Participants' testing behavior in the planned study is measured through the proportion of elimination moves, which is the number of tests within the elimination subset compared to the number of tests within the whole backtracking subset (that contains the elimination subset). Accordingly, this ratio can result in values from 0 to 1, where a 0 indicates that exclusively tests of network components outside the elimination subset have been conducted, while testing only network components within the elimination subset yields a 1. Hence, the closer to 1 values of this variable are the more likely a participant is using the elimination strategy.

However, since the elimination subset necessarily lies within the backtracking subset, use of the elimination strategy cannot be unambiguously determined through the proportion of elimination moves, because users of backtracking are likely to test a considerable portion of components within the elimination subset by chance, too.

Obtaining a more reliable indicator of elimination use, therefore, requires correcting the proportion of elimination moves for the probability of making elimination moves by chance given use of a backtracking strategy, short: $P(E|B)$. Gugerty (2007) and Shreeves et al. (2020) neatly determined this probability by means of a computer model that solves NTPs using backtracking in thousands of iterations and thereby provides a solid estimation of $P(E|B)$. This approach also takes into account the dynamic nature of solving NTPs that entails iteratively updating the subset of potentially faulty nodes based on the knowledge gained from each test move (Shreeves et al., 2020). Therefore, the method allows for reliably categorizing participants as backtracking or elimination users, which qualifies it as gold standard for this purpose.

For the statistical analyses of the planned study the proportion of elimination moves will be complemented with the same variable likewise corrected for $P(E|B)$. Given the more exploratory aim of the planned study, however, this probability will be approximated through a more basic approach: An a-priori probability is derived from each network's structure, i.e. from the ratio of elements in the elimination subset and in the backtracking subset, that reflects the probability of making an elimination move by chance, if tests were conducted randomly without any updating. To prevent underestimating the targeted probability too much, it is calculated for a reduced backtracking subset which extends only from the rightmost to the leftmost elimination subset element (see Appendix 13 for a much more detailed description of the rationale).

Please note, however, that this basic approach to measuring testing behavior without taking updating into account might limit the conclusions that can be drawn from the study's results with regard to participants' actual strategy use, since we do not categorize into backtracking and elimination users, but examine testing behavior that might be indicative of one strategy or the other. The results will, hence, be limited to estimating how likely participants use certain strategies.

9.5 | Statistical analyses

A complete overview of the variables assessed, computed and analyzed in the study along with their precise definitions can be found in Appendix 13.

9.5.1 | Descriptive statistics

For all the dependent variables, we will report range, mean, median, standard deviation, 95% confidence intervals, box-plots, and a zero-order correlation table. Every measure will be provided for the whole sample, as well as for baseline and experimental conditions separately.

For questionnaires and test instruments item and scale statistics will be provided.

9.5.2 | Inferential statistics

In line with the hypotheses and the a-priori power analyses accordingly, there are principally three types of statistical tests that apply to the data obtained in the planned study: correlation tests, one-way between-subjects General Linear Models (GLMs) with three levels, one-way within-subjects GLMs with two levels. The GLMs will be integrated into a 2 (no markers vs. markers) × 3 (Exclusion vs. Inclusion vs. No Instruction) mixed-design GLM, in order to additionally account for an interaction effect.

Additionally, the hypothesis tests in the study will be realized in R performing linear, logistic, and negative-binomial mixed-effects regressions depending on the respective expected data distributions according to the level of measurement of the different dependent variables. After data collection, however, the empirical distributions might require making adjustments to the used models, which will then be reported in addition to those proposed here.

Variables measured on absolute scale for being count data are analyzed on trial level by means of a generalized linear mixed-effects model for the negative binomial family with repeated measurements. The influence of person variables on strategy use and performance are examined through linear regressions on both trial and person level. Finally, variables with binary outcomes such as trial success (faulty node identification) are accounted for by employing logistic mixed-effects regressions with repeated measures. Where applicable, both trial and participant are treated as random factors.

Previous studies using NTPs have observed elimination use with a default rate of 10–20% in their samples (Gugerty, 2007; Shreeves et al., 2020). Such a default rate might reduce the effect of marker use instruction and, thus, limit the explanatory power of the study's results. Although one could assume that the experimental conditions will be equal with regards to this default rate due to random assignment, we will further account for this issue by reporting all the analyses using the whole sample as well as for a subsample which excludes participants who started using elimination consistently during baseline.

Additional and exploratory analyses might be performed after data collection and will then be reported separately.

For the inferential statistics, the R script containing the planned statistical analyses is provided in Appendix 14.

9.6 | Timeline for conduction and completion of the study

After in-principle acceptance and final preregistration, recruitment of participants will be initiated followed by testing. Reaching the targeted sample size is expected to take 12–14 weeks.

Data processing and analyses will be performed within three more weeks. Finally, another 3 weeks will be dedicated to preparing the final manuscript, so that we expect 20 weeks to elapse from publishing the final version of the preregistration to stage two submission (see Appendix 15 for a visualization of the timeline).

ACKNOWLEDGEMENTS

This research is supported by the Digital Society research program funded by the Ministry of Culture and Science of North Rhine-Westphalia, Germany (grant number 1706dgn006). We thank Leo Gugerty for his personal correspondence that was helpful for the design of this research.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data and materials of the study are publicly available at the Open Science Framework: https://osf.io/xrk2j/

ORCID

Martin Bordewieck https://orcid.org/0000-0001-8999-1127
Malte Elson https://orcid.org/0000-0001-7806-9583

REFERENCES

Allison, C., Auyeung, B., & Baron-Cohen, S. (2012). Toward brief "red flags" for autism screening: The short autism Spectrum quotient and the short quantitative checklist in 1,000 cases and 3,000 controls. *Journal of the American Academy of Child & Adolescent Psychiatry*, *51*(2), 202-212.e7. https://doi.org/10.1016/j.jaac.2011.11.003

Baron-Cohen, S., Ashwin, E., Ashwin, C., Tavassoli, T., & Chakrabarti, B. (2009). Talent in autism: Hyper-systemizing, hyper-attention to detail and sensory hypersensitivity. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*(1522), 1377–1383. https://doi.org/10.1098/rstb.2008.0337

Baron-Cohen, S., Richler, J., Bisarya, D., Gurunathan, N., & Wheelwright, S. (2003). The systemizing quotient: An investigation of adults with Asperger syndrome or high–functioning autism, and normal sex differences. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, *358*(1430), 361–374. https://doi.org/10.1098/rstb.2002.1206

Baron-Cohen, S., Wheelwright, S., Skinner, R., Martin, J., & Clubley, E. (2001). The autism-Spectrum quotient (AQ): Evidence from Asperger syndrome/high-functioning autism, males and females, scientists and mathematicians. *Journal of Autism and Developmental Disorders*, *31*(1), 5–17. https://doi.org/10.1023/A:1005653411471

Billington, J., Baron-Cohen, S., & Wheelwright, S. (2007). Cognitive style predicts entry into physical sciences and humanities: Questionnaire and performance tests of empathy and systemizing. *Learning and Individual Differences*, *17*(3), 260–268. https://doi.org/10.1016/j.lindif.2007.02.004

Bless, H., Wänke, M., Bohner, G., & Fellhauer, R. F. (1994). Need for Cognition: Eine Skala zur Erfassung von Engagement und Freude bei Denkaufgaben [Need for cognition: A scale measuring engagement and happiness in cognitive tasks]. *Zeitschrift Für Sozialpsychologie*, *25*(2), 147–154. https://pub.uni-bielefeld.de/record/1779110

Brosnan, M., Hollinworth, M., Antoniadou, K., & Lewton, M. (2014). Is empathizing intuitive and systemizing deliberative? *Personality and Individual Differences*, *66*, 39–43. https://doi.org/10.1016/j.paid.2014.03.006

Brown, J. S., Burton, R. R., & Bell, A. G. (1975). SOPHIE: A step toward creating a reactive learning environment. *International Journal of Man-Machine Studies*, *7*(5), 675–696. https://doi.org/10.1016/S0020-7373(75)80026-5

Cacioppo, J. T., & Petty, R. E. (1982). The need for cognition. *Journal of Personality and Social Psychology*, *42*(1), 116–131. https://doi.org/10.1037/0022-3514.42.1.116

Carlson, R. A., Lundy, D. H., & Schneider, W. (1992). Strategy guidance and memory aiding in learning a problem-solving skill. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *34*(2), 129–145. https://doi.org/10.1177/001872089203400201

Champely, S., Ekstrom, C., Dalgaard, P., Gill, J., Weibelzahl, S., Anandkumar, A., … De Rosario, M. H. (2020). Package 'pwr.' *R Package Version* 1.3-0. Retrieved from https://cran.r-project.org/web/packages/pwr/pwr.pdf

Chi, M. T. H., & VanLehn, K. A. (2012). Seeing deep structure from the interactions of surface features. *Educational Psychologist*, *47*(3), 177–188. https://doi.org/10.1080/00461520.2012.695709

Cokely, E. T., Galesic, M., Schulz, E., Ghazal, S., & Garcia-Retamero, R. (2012). Measuring risk literacy: The berlin numeracy test. *Judgment and Decision making*, *7*(1), 25–47.

Corter, J. E., & Zahner, D. C. (2007). Use of external visual representations in probability problem solving. *Statistics Education Research Journal*, *6* (1), 22–50 Retrieved from https://www.stat.auckland.ac.nz/~iase/serj/SERJ6(1)_Corter_Zahner.pdf

de Holanda Coelho, G. L., Hanel, P. H. P., & Wolf, L. J. (2018). The very efficient assessment of need for cognition: Developing a six-item version. *Assessment*, *27*, 1–16. https://doi.org/10.1177/1073191118793208

Del Missier, F., Mäntylä, T., & de Bruin, W. B. (2012). Decision-making competence, executive functioning, and general cognitive abilities. *Journal of Behavioral Decision Making*, *25*(4), 331–351. https://doi.org/10.1002/bdm.731

Dickson, H., Thompson, C. D., & O'Toole, P. (2016). A picture is worth a thousand words: Investigating first year chemistry students' ability to visually express their understanding of chemistry concepts. *International Journal of Innovation in Science and Mathematics Education*, *24*(1), 12–23.

Dörner, D., & Funke, J. (2017). Complex problem solving: What it is and what it is not. *Frontiers in Psychology*, *8*(7), 1–11. https://doi.org/10.3389/fpsyg.2017.01153

Dörner, D., Kreuzig, H. W., Reither, F., & Staudel, T. (1983). *Lohhausen. Vom Umgang mil Unbestimmtheit und Komplexität [Lohhausen. Dealing with uncertainty and complexity]*. Bern, Switzerland: Huber Retrieved from https://fis.uni-bamberg.de/handle/uniba/27941

Feist, G. J. (2012). Predicting interest in and attitudes toward science from personality and need for cognition. *Personality and Individual Differences*, *52*(7), 771–775. https://doi.org/10.1016/j.paid.2012.01.005

Firesmith, D. G. (2003). Common concepts underlying safety, security, and survivability engineering. *Software Engineering Institute*. Pittsburgh, Pennsylvania: Carnegie Mellon University. Retrieved from https://doi.org/10.1184/R1/6572621.v1

Frederick, S. (2005). Cognitive reflection and decision making. *Journal of Economic Perspectives*, *19*(4), 25–42. https://doi.org/10.1257/089533005775196732

Funke, J. (2012). Complex problem solving. In *Encyclopedia of the sciences of learning* (Vol. 3, pp. 682–685). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4419-1428-6_685

Fyrbiak, M., Strauss, S., Kison, C., Wallat, S., Elson, M., Rummel, N., & Paar, C. (2017). Hardware reverse engineering: Overview and open challenges. In *2017 IEEE 2nd International Verification and Security Workshop (IVSW)* (pp. 88–94). IEEE. https://doi.org/10.1109/IVSW.2017.8031550

Goodwin, G. P., & Johnson-Laird, P. N. (2005). Diagnosis of ambiguous faults in simple networks. In Bara, B. G., Barsalou, L. W., Bucciarelli, M. (Eds.), *Proceedings of the Twenty-Seventh Annual Conference of Cognitive Science Society* (791–796). Mahwah, NJ: Erlbaum.

Grove, R., Baillie, A., Allison, C., BaronCohen, S., & Hoekstra, R. A. (2015). Exploring the quantitative nature of empathy, systemising and autistic traits using factor mixture modelling. *British Journal of Psychiatry*, *207* (5), 400–406. https://doi.org/10.1192/bjp.bp.114.155101

Gugerty, L. (2007). Cognitive components of troubleshooting strategies. *Thinking & Reasoning*, *13*(2), 134–163. https://doi.org/10.1080/13546780600750641

Ham, D.-H., & Yoon, W. C. (2007). The training effects of principle knowledge on fault diagnosis performance. *Human Factors and Ergonomics in Manufacturing*, *17*(3), 263–282. https://doi.org/10.1002/hfm.20074

Hart, S. G. (2006). NASA-task load index (NASA-TLX); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *50*(9), 904–908. https://doi.org/10.1177/154193120605000909

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In P. A. Hancock & N. Meshkati (Eds.), *Human mental workload* (pp. 139–183). Amsterdam: North-Holland.

Harvey, I., Bolgan, S., Mosca, D., McLean, C., & Rusconi, E. (2016). Systemizers are better code-breakers: Self-reported systemizing predicts code-breaking performance in expert hackers and naïve participants. *Frontiers in Human Neuroscience*, *10*(229), 1–14. https://doi.org/10.3389/fnhum.2016.00229

Johnson, S. D. (1991). Training technical troubleshooters. *Technical and Skills Training*, *27*(7), 9–16.

Jonassen, D. H., & Hung, W. (2006). Learning to troubleshoot: A new theory-based design architecture. *Educational Psychology Review*, *18* (1), 77–114. https://doi.org/10.1007/s10648-006-9001-8

Kostopoulou, O., & Duncan, K. D. (2001). Abstract and reduced-context representations in fault-finding training. *Ergonomics*, *44*(2), 175–201. https://doi.org/10.1080/00140130121312

LiveCode 9.6.1 Community Edition [Computer software]. (2020). Retrieved from https://downloads.livecode.com/livecode/

Lorenz, T., Frischling, C., Cuadros, R., & Heinitz, K. (2016). Autism and overcoming job barriers: Comparing job-related barriers and possible solutions in and outside of autism-specific employment. *PLoS One*, *11* (1), e0147040. https://doi.org/10.1371/journal.pone.0147040

McLain, D. L. (2009). Evidence of the properties of an ambiguity tolerance measure: The multiple stimulus types ambiguity tolerance scale–II (MSTAT–II). *Psychological Reports*, *105*(3), 975–988. https://doi.org/10.2466/PR0.105.3.975-988

Morris, N. M., & Rouse, W. B. (1985). Review and evaluation of empirical research in troubleshooting. *Human Factors*, *27*(5), 503–530. https://doi.org/10.1177/001872088502700502

Morrison, D. L., & Duncan, K. D. (1988). Strategies and tactics in fault diagnosis. *Ergonomics*, *31*(5), 761–784. https://doi.org/10.1080/00140138808966720

Perez, R. S. (1991). A view from troubleshooting. In M. U. Smith (Ed.), *Toward a unified theory of problem solving: Views from the content domains* (pp. 127–166). New York, NY: Routledge. https://doi.org/10.4324/9780203052334

Pretz, J. E., Naples, A. J., & Sternberg, R. J. (2003). Recognizing, defining, and representing problems. In J. E. Davidson & R. J. Sternberg (Eds.), *The psychology of problem solving* (pp. 3–30). Cambridge, England: Cambridge University Press. https://doi.org/10.1017/CBO9780511615771.002

Reitan, R. M. (1955). The relation of the trail making test to organic brain damage. *Journal of Consulting Psychology*, *19*(5), 393–394. https://doi.org/10.1037/h0044509

Reitan, R. M. (1958). Validity of the trail making test as an indicator of organic brain damage. *Perceptual and Motor Skills*, *8*(3), 271–276. https://doi.org/10.2466/pms.1958.8.3.271

Rematska, G., & Bourbakis, N. G. (2016). A survey on reverse engineering of technical diagrams. In *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)* (pp. 1–8). IEEE. https://doi.org/10.1109/IISA.2016.7785372

Risko, E. F., & Gilbert, S. J. (2016). Cognitive offloading. *Trends in Cognitive Sciences*, *20*(9), 676–688. https://doi.org/10.1016/j.tics.2016.07.002

Robertson, S. I. (2016). *Problem solving: Perspectives from cognition and neuroscience* (2nd ed.). New York, NY: Routledge. https://doi.org/10.4324/9781315712796

Romero, D., Bernus, P., Noran, O., Stahre, J., & Berglund, Å. F. (2016). The operator 4.0: Human cyber-physical systems & adaptive automation towards human-automation symbiosis work systems. *IFIP Advances in Information and Communication Technology*, *488*, 677–686. https://doi.org/10.1007/978-3-319-51133-7_80

Rouse, W. B. (1978). Human problem solving performance in a fault diagnosis task. *IEEE Transactions on Systems, Man, and Cybernetics*, *8*(4), 258–271. https://doi.org/10.1109/TSMC.1978.4309946

Rouse, W. B., & Rouse, S. H. (1979). Measures of complexity of fault diagnosis tasks. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(11), 720–727. https://doi.org/10.1109/TSMC.1979.4310112

Rusconi, E., Ferri, F., Viding, E., & Mitchener-Nissen, T. (2015). XRIndex: A brief screening tool for individual differences in security threat detection in x-ray images. *Frontiers in Human Neuroscience*, *9*, 1–18. https://doi.org/10.3389/fnhum.2015.00439

Rusconi, E., McCrory, E., & Viding, E. (2012). Self-rated attention to detail predicts threat detection performance in security X-ray images. *Security Journal*, *25*(4), 356–371. https://doi.org/10.1057/sj.2011.27

Ruzich, E., Allison, C., Chakrabarti, B., Smith, P., Musto, H., Ring, H., & Baron-Cohen, S. (2015). Sex and STEM occupation predict autism-Spectrum quotient (AQ) scores in half a million people. *PLoS One*, *10*(10), e0141229. https://doi.org/10.1371/journal.pone.0141229

Samson, A. C., & Huber, O. W. (2010). Short German versions of empathizing and systemizing self-assessment scales. *Swiss Journal of Psychology*, *69*(4), 239–244. https://doi.org/10.1024/1421-0185/a000028

Scaife, M., & Rogers, Y. (1996). External cognition: How do graphical representations work? *International Journal of Human-Computer Studies*, *45*(2), 185–213. https://doi.org/10.1006/ijhc.1996.0048

Schell, B. H., & Melnychuk, J. (2010). Female and male hacker conferences attendees: Their autism-spectrum quotient (AQ) scores and self-reported adulthood experiences. In *Corporate hacking and technology-driven crime: Social dynamics and implications* (pp. 144–168). IGI Global. https://doi.org/10.4018/978-1-61692-805-6.ch008

Shreeves, M., Gugerty, L., & Moore, D. (2020). Individual differences in strategy use and performance during fault diagnosis. *Cognitive Research: Principles and Implications*, *5*(1), 49. https://doi.org/10.1186/s41235-020-00250-5

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Simon, H. A., & Hayes, J. R. (1976). The understanding process: Problem isomorphs. *Cognitive Psychology*, *8*(2), 165–190. https://doi.org/10.1016/0010-0285(76)90022-0

Sneider, C., Stephenson, C., Schafer, B., & Flick, L. (2014). Computational thinking in high school science classrooms. *The Science Teacher*, *81*(5), 53–59. https://doi.org/10.2505/4/tst14_081_05_53

Summers, T. C. (2015). *How hackers think: A mixed method study of mental models and cognitive patterns of high-tech wizards*. Case Western Reserve University Retrieved from https://etd.ohiolink.edu/!etd.send_file?accession=case1427809862

Toms, M., & Patrick, J. (1989). Components of fault-finding: Symptom interpretation. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *31*(4), 465–483. https://doi.org/10.1177/001872088903100409

Toplak, M. E., West, R. F., & Stanovich, K. E. (2014). Assessing miserly information processing: An expansion of the cognitive reflection test. *Thinking and Reasoning*, *20*(2), 147–168. https://doi.org/10.1080/13546783.2013.844729

Wallat, S., Albartus, N., Becker, S., Hoffmann, M., Ender, M., Fyrbiak, M., … Paar, C. (2019). Highway to HAL: Open-sourcing the first extendable gate-level Netlist reverse engineering framework. In *ACM international conference on computing Frontiers 2019, CF 2019 - proceedings* (pp. 392–397). ACM. https://doi.org/10.1145/3310273.3323419

Wei, X., Yu, J. W., Shattuck, P., McCracken, M., & Blackorby, J. (2013). Science, technology, engineering, and mathematics (STEM) participation among college students with an autism spectrum disorder. *Journal of Autism and Developmental Disorders*, *43*(7), 1539–1546. https://doi.org/10.1007/s10803-012-1700-z

Westfall, J. (2015). PANGEA: Power analysis for general ANOVA designs. http://jakewestfall.org/publications/pangea.pdf

Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, *25*(2), 7–14. https://doi.org/10.17471/2499-4324/922

Wolf, K. M., & Mieg, H. A. (2010). Cognitive determinants of the success of inventors: Complex problem solving and deliberate use of divergent and convergent thinking. *European Journal of Cognitive Psychology*, *22*(3), 443–462. https://doi.org/10.1080/09541440902916845

Zahner, D. C., & Corter, J. E. (2010). The process of probability problem solving: Use of external visual representations. *Mathematical Thinking and Learning*, *12*(2), 177–204. https://doi.org/10.1080/10986061003654240

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Bordewieck M, Elson M. The impact of inducing troubleshooting strategies via visual aids on performance in a computerized digital network task. *Appl Cognit Psychol*. 2021;35:833–847. https://doi.org/10.1002/acp.3809