Task-independent metrics of computational hardness predict performance of human problem-solving

Juan P. Franco[1], Karlo Doroc[1], Nitin Yadav[1], Peter Bossaerts[1,2], and Carsten Murawski[1]

[1]Brain, Mind & Markets Laboratory, Department of Finance, The University of Melbourne, Parkville, Victoria 3010, Australia
[2]Florey Institute of Neuroscience and Mental Health, Parkville, Victoria 3010, Australia

April 25, 2021

Abstract

The survival of human organisms depends on our ability to solve complex tasks, which is bounded by our limited cognitive capacities. However, little is known about the factors that drive complexity of the tasks humans face and their effect on human decision-making. Here, using insights from computational complexity theory, we quantify computational hardness using a set of task-independent metrics related to the computational requirements of individual instances of a task. We then examine the relation between those metrics and human behavior and find that these metrics predict both performance and effort allocation in three canonical cognitive tasks in a similar way. Our findings demonstrate that the ability to solve complex tasks can be predicted from generic metrics of their inherent computational hardness.

Introduction

The adaptiveness of human organisms is bounded by their limited cognitive capacities, sometimes referred to as bounded rationality [1]. In the words of Herbert Simon, "[h]uman rational behavior [...] is shaped by a scissors whose two blades are the structure of task environments and the computational capabilities of the actor" [1].

In the past several decades, the study of human behavior has focused predominantly on the latter. This work characterizes cognitive capacities and cognitive strategies, algorithms and heuristics people use in different task environments. It includes approaches such as the heuristics and biases program [2], resource and computational rationality [3, 4] as well as ecological rationality [5], among others. However, very little is known about how properties of the task environment affects computational requirements and how they compare to bounds of human cognitive capacities.

Several studies have explored how the task environment affects the performance of specific algorithms or cognitive strategies [6–8]. However, this approach ignores the diversity in strategies implemented not only across humans, but also across situations. Even if the task environment is the same, different people might approach a task using different procedures and might change their procedures

depending on the situation or their level of experience [6, 9–13]. Understanding the interaction between task environment and an agent's computational capabilities is then particularly difficult given the lack of a generic cognitive strategy.

A principled and generic way to characterize the computational requirements of a task environment is to formalize the task as a computational problem and analyze its *problem complexity*. Here, computational requirements are typically analyzed at the level of problems, such as sorting an array of numbers. These requirements are typically expressed in terms of asymptotic worst-case growth of a resource such as compute time or memory. This means that resource requirements are characterized in terms of their growth as a function of the input size of the problem, for example, the length of the array to be sorted. Importantly, this is generally done by considering the growth in requirements in the worst-case as the input size increases. Problems with similar resource requirements, thus defined, are then grouped into complexity classes [14].

As it stands, problem complexity is not amenable to modeling human behavior directly. Critically, although this approach can shed light on the a priori plausibility of models of human behavior [15], it is inadequate for the derivation of empirically testable predictions at finer detail. First, while complexity classes are based on asymptotic growth of resources, in practice many instances (that is, cases of a problem) people face are small in size [16]. Second, complexity classes are typically based on worst-case growth of resources. This means that hardness is defined in terms of resources required to solve the most difficult instance of a problem. However, in most cases, there is substantial variation in resource requirements of instances of the same input size [17, 18], and the worst case is often far away from typical, or average, cases and may not be encountered in the natural environment [19]. Third, the approach classifies problems according to hardness, like a taxonomy, but it does not identify the sources of hardness, for example, which properties of instances make some harder than others. What would be desirable is a set of generic properties of individual instances of a class of problems that are associated with computational hardness in a way that is independent of the problem it belongs to. Similar to how properties like mean, variance and other statistics characterize the level of uncertainty in the class of probabilistic problems. However, there is as yet no analog for characterizing computational hardness.

There is limited research on how properties of instances of problems affect human problem-solving. To date, most studies are based on a problem-specific approach [9, 10, 20, 21]. Hence, their findings may not generalize to other problems. Recent theoretical advances in computer science and statistical physics provide a framework, referred to as typical-case complexity (TCC), that addresses this issue. It allows the characterization of computational hardness of individual instances of a problem. More specifically, it is concerned with the average computational hardness of random instances of a computational problem, linking structural properties of those instances to their computational complexity, independent of a particular computational model [17, 18, 22–24]. This work has identified computational 'phase transitions', which resemble phase transitions in statistical physics and which are related to computational hardness of instances. Such phase transitions have been found in a number of canonical NP-complete problems (i.e., problems that are both in NP and NP-hard) [17, 18, 22–24], including the graph coloring problem [18, 25], the traveling salesperson problem [17] and the K-SAT problems (Boolean satisfiability problems) [18, 25, 26], among others. This program has led to a deeper understanding of computational hardness

by relating it to structural properties of instances. Importantly, it has identified that hardness of an instance is related to a generic instance property, namely *constrainedness* (see Fig 1). The framework has also been useful for understanding patterns in the performance of algorithms [25, 27], and subsequently, generating more efficient algorithms [24].

A recent study applied this framework to study human behavior in the knapsack problem, a (NP-hard) combinatorial optimization problem [28]. The study found that both effort (time-on-task) and ability to solve an instance were related to computational phase transitions, with patterns similar to those exhibited by generic constrained optimization algorithms. An important question is whether these findings generalize to other problems. If they do, then these properties related to computational complexity would be prime candidates for generic measures of computational hardness of human cognition, in the way that statistics like mean, variance and kurtosis serve as generic measures of probabilistic uncertainty in a task [29].

Here, using a behavioral experiment, we study the relation between a set of problem-independent measures of instance complexity and human performance in two canonical NP-complete computational problems, the Boolean satisfiability problem (3SAT) and the traveling salesperson problem (TSP). We then compare those results to results previously obtained for the 0-1 knapsack decision problem (KP) [28], to test their generalizability across NP-complete problems.

Results

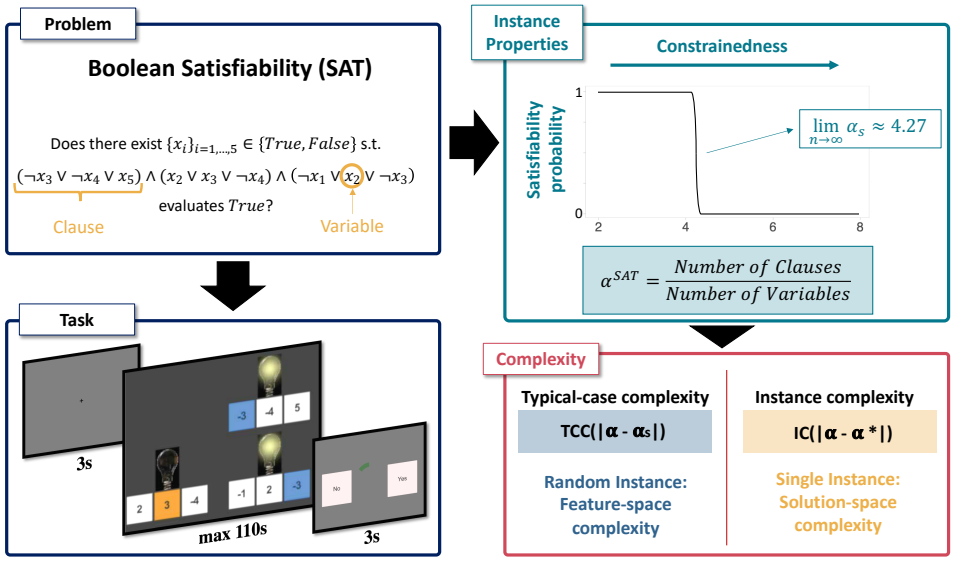
Each participant solved one of the three problems: either 72 instances of the TSP task, 64 instances of the 3SAT or 72 instances of the KP (Figs 1,5). TSP is the problem of determining whether a path of a particular length (or less), connecting a set of cities, exists or not. 3SAT is the problem of determining whether a set of variable configurations (true/false) exist that render a set of clauses true. And KP is the problem of determining whether there exists a subset of items with differing values and weights exceeding a minimum total value while not exceeding a maximum total weight. All three problems are *decision problems*, that is, problems whose answer is either 'yes' or 'no'. If there exists a configuration of variables such that the solution of the instance is 'yes', the instance is called *satisfiable* and *unsatisfiable* otherwise.

Instances varied in their computational hardness (see Methods). Both TSP and 3SAT were self-paced (with time limits per trial), while the KP was not. Results for KP have previously been reported elsewhere and are included here for comparison only [28].

Summary statistics

We first present summary statistics for each of the three tasks. We measured performance as a binary outcome, depending on whether a participant's response was correct or not. Additionally, we studied effort by analyzing time-on-task. This captures another dimension of the agent's problem-solving process that is not entirely determined by performance because, unlike algorithms implemented by electronic computers, humans have the option to stop working independently of the solving strategy. It is worth noting that the effort analysis was not performed for the KP, since this task was not self-paced.

Figure 1: **3SAT problem, complexity metrics and experimental design. The problem.** The aim is to determine whether a Boolean formula is *satisfiable*. **The task.** The Boolean formula is represented with a set of light bulbs (clauses), each of which has three switches underneath (literals) that are characterized by a positive or negative number. The number on each switch represents the variable number, which can be turned on or off (TRUE or FALSE). The aim is to determine whether there exists a way of turning on and off variables such that all the light bulbs are turned on (formula evaluates TRUE). **Instance properties.** The constrainedness of the problem (α) is captured by the ratio of clauses to variables. This parameter characterizes the probability that a random instance of the problem is satisfiable. In the limit this probability undergoes a phase transition around the satisfiability threshold (α_s). **Complexity metrics.** Instances near this threshold are on average harder to solve than instances further away. Average hardness is captured by the typical-case complexity metric (TCC). This metric can be estimated entirely from the features of the problem (feature-space) without the need to solve the problem. Alternatively, instance complexity (IC) can be estimated from features of the solution-space. IC is characterized as the difference between the constrainedness of the instance (α) and α^*, the maximum number of clauses that can be satisfied normalized by the number of clauses.



In the TSP, all instances had 20 cities and a time limit of 40 s. The number of cities and time limit were selected, based on pilot data, to ensure that the task was neither too difficult nor too easy (see Methods). Mean *human performance*, measured as the proportion of trials in which a correct response was made, was 0.85 ($\min = 0.76$, $\max = 0.93$, $SD = 0.05$). Participants' average time spent on an instance was 32.2 s and ranged from 19.9 s to 39.2 s (SD = 5.2). Performance did not vary during the course of the task, but time-on-task decreased as the task progressed (S3 Appendix).

All instances of the 3SAT task had 5 variables and a time limit of 110 s. Similar to TSP, the number of variables and time limit were selected, based on pilot data, to target a specific average performance ($\approx 85\%$; see Methods). Mean *human performance* was 0.87 ($\min = 0.75$, $\max = 0.98$, $SD = 0.06$). The average time spent on an instance varied from a minimum of 15.9 s to a maximum of 104.3 s (mean = 60.2, SD = 18.7). Similar to TSP, performance did not vary during the course of the task, but participants tended to spend less time on a trial as the task progressed (S3 Appendix).

In the KP decision task implemented by Franco et al. [28], all instances had 6 items. This task was not self-paced, that is, participants had exactly 25 seconds to solve each instance and could not skip to the next screen before the time ended. Mean *human performance* was 83.1% ($\min = 0.56$, $\max = 0.9$, $SD = 0.08$). Like in the other two tasks, performance did not vary during the course of the task (S3 Appendix).

Feature-space complexity metrics

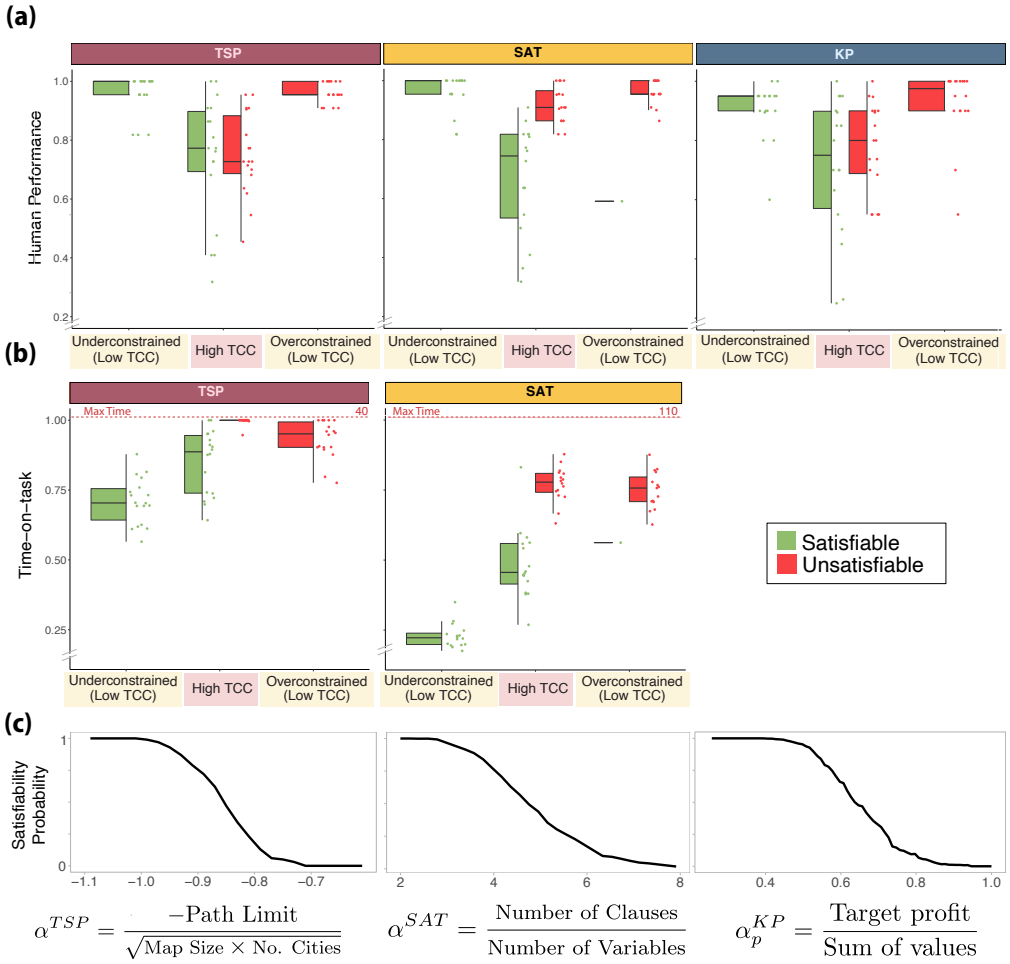
We now examine how generic properties of instances affect the quality of decisions and the computational effort exerted. We study two types of properties: feature-space and solution-space metrics. The main difference between them is that feature-space metrics can be estimated from mathematical properties of the instance without any knowledge of an instance's solution, whereas the calculation of solution-space metrics require knowledge of an instance's solution, that is, require the solution to be computed (Fig 1).

We first examine the effect of typical-case complexity (TCC), a feature-space metric of complexity, on human performance and effort. This measure is based on a framework in computer science developed to study the drivers of computational hardness in computational problems by analyzing the difficulty of randomly generated instances of those problems. The study of random instances has revealed that there is substantial variation in computational resource requirements for instances with the same input length [17, 18, 22, 30]. This variation in computational hardness has recently been related to various structural properties of instances. In particular, it has been shown for several intractable (specifically, NP-complete) problems, including the KP [30], TSP [17] and 3SAT [23, 24], that there exists a set of parameters $\bar{\alpha}$ that captures the constrainedness of an instance. Moreover, it has been shown that there is a threshold α_s such that random instances with $\alpha \ll \alpha_s$ are mostly satisfiable whereas they are mostly unsatisfiable if $\alpha \gg \alpha_s$. Importantly for our study, it has been shown for each of the problems under consideration that instances near α_s are, on average, computationally harder than instances further away from α_s [17, 18, 22, 30]. In our study, we sampled instances with varying values of α and categorize instances with $\alpha \sim \alpha_s$ as instances with a *high TCC* and instances with $\alpha \gg \alpha_s$ or $\alpha \ll \alpha_s$ as *low TCC* (see Fig 2 and Methods).

We first examine the effect of TCC on human performance across problems. We hypothesized that participants would have lower performance on instances with high TCC compared to those with low TCC. We found that this was indeed the case for both TSP and 3SAT as well as for KP (TSP: $\beta_{0.5} = -2.10$, $HDI_{0.95} = [-2.50, -1.73]$, S1 Table Model 2; 3SAT: $\beta_{0.5} = -1.58$, $HDI_{0.95} = [-1.95, -1.20]$, S2 Table Model 2; KP: $\beta = -1.327$ $P < 0.001$, main effect of TCC on performance, GLMM; Fig 2a).

Instances with an $\alpha \gg \alpha_s$ or $\alpha \ll \alpha_s$ are considered to have a low TCC. However, these instances belong to two structurally different regions, namely an overconstrained and an underconstrained region. We studied whether differences in constrainedness affected performance among low TCC instances. We found that for the TSP and 3SAT, there was no difference in performance between underconstrained and overconstrained regions (TSP: $\beta_{0.5} = 0.14$, $HDI_{0.95} = [-0.58, 0.87]$, S1 Table Model 3; 3SAT: $\beta_{0.5} = -0.43$, $HDI_{0.95} = [-1.12, 0.28]$, S2 Table Model 3; the difference in effect, $overconstrained - underconstrained$, on performance, GLMM). These results are consistent

Figure 2: **Typical-case complexity (TCC). (a) Human performance and satisfiability probability.** Each dot represents an instance of one of the three problems considered. For each instance human performance corresponds to the proportion of participants that solved an instance correctly. The instances are categorized according to their constrainedness region (α) and their TCC. The correct solution (satisfiability) of an instance is represented by its color. **(b) Time-on-task and TCC.** Median time spent solving an instance before submitting an answer. Time is represented as a proportion of the maximum time allotted on each trial (40s in the TSP and 110s in the 3SAT). **(c) Satisfiability probability and constrainedness parameter α.** Probability that a random instance is satisfiable as a function of α (the probability is empirically estimated; see Methods). In the underconstrained region (low TCC) the satisfiability probability is close to one while in the overconstrained region (low TCC) the probability is close to zero. The region with a high TCC corresponds to a region in which the probability is close to 0.5. *The box-plots represent the median, the interquartile range (IQR) and the whiskers extend to a maximum length of 1.5*IQR*



with those obtained previously in relation to KP ($\beta = 0.250$, $P = 0.355$, the difference in effect, $overconstrained - underconstrained$, on performance, GLMM; Fig 2a). Taken together, these findings suggest that the mapping between α and TCC captures the effect of α on performance.

We also expected TCC to have an effect on time-on-task. We hypothesized that participants would spend more time on instances with high TCC. We found this to be the case for 3SAT and TSP

(3SAT: $\beta_{0.5} = 0.149$, $HDI_{0.95} = [0.116, 0.182]$, S5 Table Model 2; TSP: $\beta_{0.5} = 0.118$, $HDI_{0.95} = [0.090, 0.147]$, S6 Table Model 2; effect of TCC on time-on-task as a proportion of the maximum possible time, censored linear mixed-effects models (CLMM), Fig 2b). The effect was mainly driven by the constrainedness level (α). Specifically, participants spent less time-on-task on instances in the underconstrained region (3SAT: $\beta_{0.5} = -0.352$, $HDI_{0.95} = [-0.385, -0.318]$ S5 Table Model 3; TSP: $\beta_{0.5} = -0.199$, $HDI_{0.95} = [-0.233, -0.164]$, S6 Table Model 3; difference in time-on-task between instances in the underconstrained region and those with high TCC ($\alpha \sim \alpha_s$), CLMM). In the TSP, participants spent less time on overconstrained instances compared to those instances with $\alpha \sim \alpha_s$, but this effect was not significant ($\beta_{0.5} = -0.024$, $HDI_{0.95} = [-0.059, 0.011]$, difference in time-on-task between instances in the overconstrained region and $\alpha \sim \alpha_s$, CLMM; S6 Table Model 3). In contrast, in the 3SAT participants spent more time on overconstrained regions compared to those instances with $\alpha \sim \alpha_s$ ($\beta_{0.5} = 0.071$, $HDI_{0.95} = [0.036, 0.106]$, difference in time-on-task between instances in the overconstrained region and with high TCC, CLMM; S5 Table Model 3). It is worth noting that in 3SAT, the more constrained the problem is, the higher the amount of clauses presented, which could have driven this effect.

Our results so far show that participants expend more effort on instances with higher TCC and yet they perform worse on these instances. This suggests a negative correlation between time-spent and performance (TSP: $\beta_{0.5} = -0.1$, $HDI_{0.95} = [-0.13, -0.08]$, S1 Table Model 5; 3SAT: $\beta_{0.5} = -0.02$, $HDI_{0.95} = [-0.02, -0.01]$, S2 Table Model 5); effect of time-spent on performance, GLMM).

Solution-space complexity metrics

In the previous section, we studied the effects of feature-space complexity metrics on human performance and effort. These metrics can be estimated based on a problem's input, that is, without the need to solve the instance. We now turn our attention to complexity metrics based on an instance's *solution space*. We will use the term solution space to refer to the set of *solution witnesses* of an instance, that is, the set of configurations of variables (e.g., possible paths or variable assignments) that satisfy an instance's constraints. Note that in order to estimate solution-space metrics, the instance, or a harder variant, has to be solved. In some cases, all possible solution witnesses must be found.

An important difference in the structure of instances, is their *satisfiability*, that is, whether the instance's solution is 'yes' or 'no'. We found that satisfiability affects performance but that this effect varies between problems. In 3SAT, participants performed worse on satisfiable instances ($\beta_{0.5} = -1.35$, $HDI_{0.95} = [-1.73, -0.99]$, main effect of satisfiability, GLMM, S2 Table Model 8), whereas there was no significant effect of satisfiability on performance in the TSP and the KP (TSP: $\beta_{0.5} = -0.06$, $HDI_{0.95} = [-0.34, 0.22]$, S1 Table Model 6; KP: $\beta_{0.5} = -0.29$, $HDI_{0.95} = [-0.57, 0.01]$, S3 Table Model 1; main effect of satisfiability, GLMM).

Turning our attention to the effect of satisfiability on time-on-task, we find that less time was spent on satisfiable instances in both TSP and 3SAT (TSP: $\beta_{0.5} = -0.17$, $HDI_{0.95} = [-0.20, -0.15]$, S6 Table Model 4; 3SAT: $\beta_{0.5} = -0.32$, $HDI_{0.95} = [-0.35, -0.29]$, S5 Table Model 4; effect of satisfiability on time-on-task, CLMM). We further explored the effect of satisfiability by studying its interaction effect with TCC. We only found an interaction effect between satisfiability and TCC in 3SAT,

in relation to both performance and time-on-task (Fig 2; S1 Appendix).

In summary, we observed that participants spent less time-on-task on satisfiable instances in both TSP and 3SAT, yet the effect of satisfiability on performance varied across problems. Moreover, our results suggest that satisfiability and TCC might interact and affect performance and time-on-task on some problems.

We can analyze the drivers of hardness in satisfiable instances at a more granular level by studying the number of solution witnesses of an instance. This generic feature of decision instances captures the constrainedness of an instance: a higher value of witnesses is related to a lower degree of constrainedness. It is worth noting that this metric is only informative for satisfiable instances (by definition, unsatisfiable instances have zero solution witnesses). Thus, we restrict our analysis to these instances.

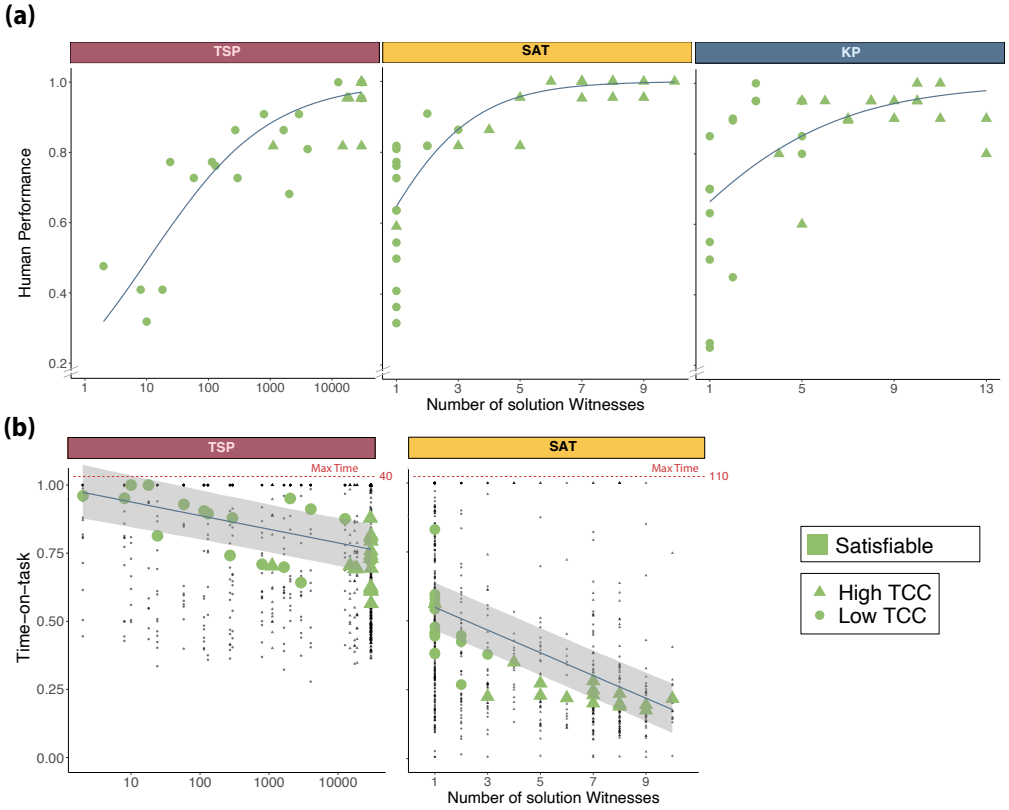
We found, in line with our hypothesis, a positive effect of the number of witnesses on performance in all three problems (3SAT: $\beta_{0.5} = 0.62$, $HDI_{0.95} = [0.49, 0.79]$; TSP: $\beta_{0.5} = 0.45$, $HDI_{0.95} = [0.37, 0.53]$; KP: $\beta_{0.5} = 0.26$, $HDI_{0.95} = [0.19, 0.34]$; main effect of the number of witnesses on performance, GLMM; S4 Table Models 1,4,6; Fig 3a). We further hypothesized that participants would spend less time solving instances with a higher number of witnesses. This was indeed the case (TSP: $\beta_{0.5} = -0.02$, $HDI_{0.95} = [-0.03, -0.02]$, S6 Table Model 5; 3SAT: $\beta_{0.5} = -0.041$, $HDI_{0.95} = [-0.047, -0.036]$, S5 Table Model 5; effect of number of witnesses on time-on-task, CLMM; Fig 3b)). These results suggests that, among satisfiable instances, the more constrained an instance, the harder it is to solve.

It is worth noting that the number of witnesses is a metric conceptually similar to TCC. After all, TCC is a mapping from expected constrainedness (α) to hardness. We studied the link between these metrics and found that the effect of TCC on performance on satisfiable instances is driven, at least partially, by the number of witnesses of an instance (see S4 Appendix).

An alternative solution-space complexity metric that can be used to study the difficulty of all instances (both satisfiable and unsatisfiable) is instance complexity (IC) [28]. It is related to the constrainedness of an instance and the order parameter $\bar{\alpha}$. It is defined based on the distance between the decision threshold of an instance and the maximum attainable value in the optimization variant of the instance. For example, the optimization variant of an instance of the TSP corresponds to finding the minimum path-length connecting all cities. For the KP, it corresponds to finding the maximum value that can fit into the knapsack given the weight constraint. Analogously, for the 3SAT, the optimization version (MAX-SAT) corresponds to finding the maximum number of clauses that can be rendered true simultaneously.

We define IC as the absolute value of the normalized difference between target value of the decision variant and the maximum value attainable of the corresponding optimization variant. In the KP, for example, it is the absolute value of the difference between target profit of the decision instance and the maximum profit attainable of the corresponding optimization instance, divided by the sum of the values of all items, that is,

Figure 3: **Number of solution witnesses.** The number of witnesses is defined as the number of *state-space combinations* (i.e., paths, items or switch-setups) that satisfy the constraints. On satisfiable instances, the problem becomes harder as the number of witnesses approaches 0. Only satisfiable instances are included. **(a) Human performance.** Each green shape represents the mean accuracy per instance. The blue line represents the marginal effect of the number of solution witnesses on human performance (GLMM S4 Table Models 1,4,6). **(b) Time-on-task.** Each green shape represents the median time-on-task per instance. The blue line represents the marginal effect (and 95% credible interval) of the number of solution witnesses on time-on-task (LMM S6 Table Model 5 and S5 Table Model 5). Each black dot corresponds to the time-on-task of one participant while solving a single instance.



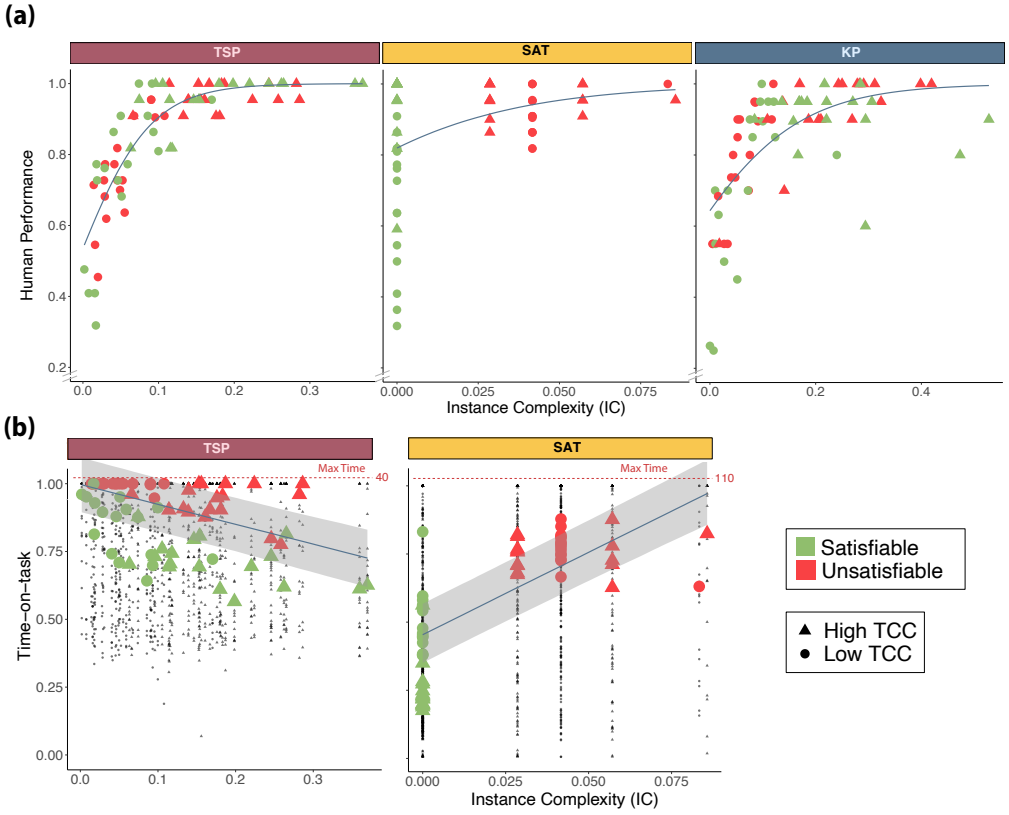
$$IC_{KP} = \left|\alpha_p - \alpha_p^{\star}\right| = \left|\frac{\text{Target profit} - \text{Maximum profit attainable}}{\sum v_i}\right|,$$

where the decision instance and the corresponding optimization instance have the same set of items and the same total weight (capacity) constraint. Intuitively, IC in KP is the normalized value of the distance between the target profit of a decision instance and the maximum profit that can be attained with the same set of items and the same capacity constraint. The corresponding expressions for TSP and 3SAT are provided in the Methods section.

We studied the effect of IC on performance and effort in each of the problems. Note that lower values of IC indicate that the decision threshold is closer to the optimum, which corresponds to a higher level of computational hardness. Therefore, we expected a positive relation between IC and performance.

We found a positive non-linear relation in all problems (KP: $\beta_{0.5} = 9.05$, $HDI_{0.95} = [7.20, 11.02]$, S3 Table Model 2; TSP: $\beta_{0.5} = 21.13$, $HDI_{0.95} = [17.63, 24.91]$, S1 Table Model 7; 3SAT: $\beta_{0.5} = 30.30$, $HDI_{0.95} = [21.95, 39.24]$, S2 Table Model 6; the effect of IC on performance, GLMM; Fig 4a).

Figure 4: **Instance complexity.** Instances become harder as $IC = |\alpha_p - \alpha_p^\star|$ approaches 0. **(a) Human performance.** Green and orange shapes represent the mean accuracy for each instance. The blue lines represents the marginal effect of IC on human performance (GLMM S3 Table Model 2, S1 Table Model 7, S2 Table Model 6). **(b) Time-on-task.** Green and orange shapes represent the median time-on-task for each instance of the TSP and 3SAT problems. The blue lines represents the marginal effect (and 95% credible interval) of IC on time-on-task (LMM S6 Table Model 6, S5 Table Model 6). Each black dot corresponds to the time spent by a single participant on a particular instance.



IC is a metric at the level of individual instances and thus we expected that it captures a substantial amount of the variability in performance between instances. Indeed, IC was able to explain a high proportion of the variance in average instance performance in the TSP and the KP (KP: $R^2 = 0.65$; TSP: $R^2 = 0.75$) but was lower in 3SAT (3SAT: $R^2 = 0.16$). We explore this further in the S5 Appendix.

Next, we explored how well IC predicted time-on-task. We expected a negative relation between IC and the average time spent on an instance. This was the case for TSP ($\beta_{0.5} = -0.735$, $HDI_{0.95} = [-0.901, -0.581]$, main effect of IC on time-on-task, CLMM; S6 Table Model 6; Fig 4b), but for the 3SAT we found a significant positive effect ($\beta_{0.5} = 6.04$, $HDI_{0.95} = [5.41, 6.70]$, main effect of IC on time-on-task, CLMM; S5 Table Model 6; Fig 4b). Based on this result, we hypothesized that the positive effect of IC on time-on-task in 3SAT could have been driven by the effect of satisfia-

bility, but we are unable to test this hypothesis directly. Therefore, we investigated the effect of IC on time-on-task in unsatisfiable instances only and found a non-significant negative effect ($\beta_{0.5} = -0.557$, $HDI_{0.95} = [-1.912, 0.782]$, main effect of IC on time-on-task for unsatisfiable instances, CLMM; S5 Table Model 7). These results indicate a negative relation between IC and time-on-task in the TSP, whereas in 3SAT the results are inconclusive.

We have shown that generic instance-level complexity metrics are able to explain differences in performance and time-on-task across instances and problems. However, it remains an open question whether these generic properties can shed light on how humans solve those problems. To explore this question, we investigated whether our metrics could explain differences in the number of clicks across instances. The number of clicks is a useful metric in studying the algorithms implemented by humans. Specifically, the number of clicks is related to the way that the problem's state space is explored. In the 3SAT, the state space consists of all possible on-off switch setups (2^5 possible combinations) while in the TSP the state space consists of all possible ordered path selections ($2^{\binom{20}{2}} = 2^{190}$ possible combinations). Arguably, participants search the state space by clicking on different state combinations in order to decide whether an instance is satisfiable or not. Differences in the quantity of clicks used to solve an instance can shed light into how the state space is explored (under the assumption that the state space is explored by clicking on elements in the task). We investigated whether generic properties of the instance captured differences in the number of clicks.

We found that the length of search in the state space, that is, the set of paths or variable configurations, is related to two properties of the instance, namely satisfiability and complexity. Search was longer in general in the case of unsatisfiable instances and there was a positive effect of TCC on search length. Moreover, longer search was also related to lower values of IC and lower number of witnesses (see S2 Appendix). These patterns suggest that the length of search in the state space can be explained, at least partially, by the properties of an instance. Interestingly, the effect of our metrics on performance can shed light on the possible strategies used by participants (see Discussion).

Discussion

Human behavior arises as an interaction between the agent, subject to limited cognitive capacities, and its environment [1]. Much research on this interaction to date has focused in characterizing cognitive strategies employed by agents in a given environment [2, 5, 31]. Comparatively little work has investigated how properties of the task environment relate to cognitive demands and how these interact with cognitive capacities. In the present study, we propose a generic framework for quantifying computational hardness of cognitive tasks based on structural properties of individual instances of the the underlying computational problem. We find that a set of metrics based on these properties predict both task performance and effort exerted across three cognitive tasks related to different NP-complete computational problems.

More specifically, using a controlled experiment, we show that three generic properties of NP-complete problems, typical-case complexity (TCC), the number of solution witnesses, and instance complexity (IC), affect human performance and effort exerted when performing a task. While the extent of effort increased with higher complexity of instances, efficacy, and thus performance in those instances,

decreased. We show that the relation between the complexity metrics presented on the one hand and task performance and effort exerted on the other, are similar across three different NP-complete problems.

Our results complement findings from computer science and suggest that hardness stems partially from intrinsic difficulty of the problem and the instance, regardless of the algorithm and the computing device used. In particular, our findings suggest that the same intrinsic hardness metrics describe the performance of algorithms executed by both electronic computers [17, 18, 22–24, 30] and humans. This is particularly interesting because the theory in which our analysis is based is derived without taking into account limits on human computation. For instance, no memory constraints are imposed on the solving algorithms. Interestingly, our results also show that computational hardness affects how much time an agent decides to spend on an instance. This is far from obvious because, unlike the standard algorithms executed by electronic computers, humans have the option to stop working independently of the solving strategy.

Critically, our results provide support for the premise that a comprehensive and accurate characterization of human behavior requires the study of both 'blades' of the scissors: an agent's cognitive capacities as well as the task environment. The proposed approach can shed light on how to operationalize bounded rationality [1] by shaping the canvas to which cognition must be confined in order to model a computationally feasible agent.

Computational complexity in cognition

The role of computational complexity in cognition has been studied before. Problem complexity has been used to study the limits of what is potentially human computable [15, 32, 33]. According to this work, many tasks we face in our lives—and corresponding computational models of human behavior—are computationally intractable (NP-hard) [15], including planning, learning and many forms of reasoning (for example, analogy, abduction and Bayesian inference) [15]. This means that the computational requirements quickly grow to levels that make solving those tasks infeasible within a reasonable amount of time and memory.

This analysis is, however, too coarse to explain differences in performance and behavior across the class of human-computable problems. Such differences have generally been ascribed to the solver or the agent [6–8, 21, 34–36]. This approach, however, is problematic given the diversity of algorithms implemented and their specificity to a particular problem [11].

We propose that a new level of analysis be included in the study of cognition: instance-level complexity. This additional level of analysis describes the generic or intrinsic complexity of problems at a more granular level. In the present study, we show that our conceptual approach captures differences in behavior across different NP-complete problems without reference to an algorithm or particular computational device. More specifically, we explored the effect of three generic complexity metrics on human performance. Each of them can be used to unearth generalities in human behavior. Typical-case complexity (TCC) captures the average hardness of a random ensemble of instances of a problem based on its constrainedness. Critically, TCC can be computed ex-ante—without knowledge of an instance's solution. Instance complexity (IC) maps constrainedness to complexity, but does this at the level of a single instance rather than an average across instances. Finally, the number of solution witnesses captures a

structural property of an instance that is related to the hardness of search for satisfiable instances.

Our three metrics capture generalities in behavior using generic metrics of computational hardness on NP-complete problems, just like metrics of uncertainty, such as mean, variance and other statistics, capture generalities in behavior in probabilistic problems. Importantly, our framework can be applied to other decision problems in classes P or NP [18, 22, 23], and has also been shown that it can be extended to optimization problems [28].

The generality of TCC is limited by its dependence on a particular sampling distribution. We sampled instances for each of the problems from a specific procedure in which the components of the instances were randomly sampled from uniform distributions. We leave it to future research to study whether TCC can be extended to other probability distributions, and particularly, to those found in real life [19].

Importantly, we provided two alternatives to TCC (IC and number of witnesses), which do not depend on a sampling procedure. These metrics quantify the hardness of specific instances of problems. However, they do come at a cost: these metrics are computationally intensive. That is, in order to compute them, the decision problem, or a harder variant, needs to be solved first. For IC to be estimated, the optimization variant of the instance needs to be solved, whereas to compute the number of witnesses, all of the possible witnesses of an instance need to be counted.

We argue that the computational requirements of calculating these metrics is not prohibitive in the context of the study of human problem-solving and cognition in general. These metrics can be used to predict generalities in human behavior with the aid of any of the resources at hand, including electronic computers. Therefore, since the practical instances of problems solvable by humans are relatively small compared to those solvable by electronic computers, cognitive scientists effectively have access to an oracle machine to estimate computationally intensive metrics.

Future directions

This paper focuses on generalities across problems within a well-defined class (i.e., NP-complete). A related question is whether intrinsic characteristics specific to a problem can complement the generic metrics presented here. Intrinsic metrics of complexity, specific to a problem, have been previously shown to affect performance. Specifically, for all three problems considered in this study, measures derived from the features of the problem have been shown to affect computational time of algorithms executed on electronic computers [37–40]. Additionally, problem-specific complexity metrics have been show to be related to human performance in the optimization variants of the TSP [9, 10]. Future work should be undertaken to study how instance-complexity generic metrics and problem-specific measures jointly affect human performance.

Importantly, our results suggest that the metrics put forward in this study are generic as they provide both ex-ante and ex-post predictability across different problems. However, our work also highlights that certain structural properties of the problem might have problem-specific effects that could interact with the effect of generic metrics of hardness. This is particularly evident in the 3SAT. In this task we find that IC explains less of the variance in performance than in on the other two tasks and that the effect of IC on time-on-task is inconclusive. This might be related to the intertwinement of satisfiability and IC

in this problem. Specifically, the structure of the 3SAT problem generates an unavoidable confounding between these two metrics given that all satisfiable instances have $IC = 0$, thus rendering IC incapable of explaining variance across satisfiable instances. This is further relevant because in this task, unlike in the other two, satisfiability has a significant effect on performance. Taken together, this suggests that the effect of IC on performance in the 3SAT might be incongruously driven by satisfiability, in a way that cannot be differentiated in our experimental design. In future studies, it should be attempted to disentangle these effects, for example, by studying the related maximum satisfiability problem (MAX-SAT). More importantly, these results warrant further investigation of the effect of satisfiability, and other structural properties, on human behavior. Moreover, future work could explore the differences in these effects across classes of problems. For instance, NP-complete problems could be categorized into finer classes based on the effect of particular properties on human problem-solving. Our findings would suggest that more abstract logical problems might be solved differently to other more life-pertinent problems, such as KP and TSP.

We investigated the effect of different metrics of instance-level complexity keeping the size of instances fixed. An additional dimension in this framework that has been shown to affect human behavior is an instance's size [7, 9] and the size of the state space (that is, the number of possible combinations or paths) [6, 41]. Additionally, the instance complexity metrics we presented are based on the satisfiability threshold and the number of witnesses. Recently, it has been shown that the performance of algorithms, designed for electronic computers, as α approaches α_s, is not only related to the decrease in the number of witnesses, but also to the shattering of witnesses into distinct clusters [25, 42]. Further research is needed to integrate these different dimensions of complexity and determine their combined effect on human problem-solving.

We have argued that the framework presented here can be used to characterize the effect of the task environment on human performance. However, instance-level complexity metrics can also shed light on the type of strategies employed by agents. Note, for example, that our results suggest that participants did not predominantly perform random search. In a random algorithm, random combinations from the *state space* (i.e., paths or variable configurations) are tried and an answer (yes/no) is selected depending on whether a solution witness is found. If participants implemented such an algorithm, we would expect the length of search to be similar on all unsatisfiable instances given that completing an exhaustive search of the state space is unlikely because of the limits on time and number of clicks. This, however, is not what we found. In unsatisfiable instances the length of search, time-on-task (in TSP) and performance were affected by IC. In fact, by applying the same argument, we can rule out other more directed search heuristics such as greedy algorithms [43], which have been proposed to be linked to human behavior [6]. Overall, our results suggest that when people solve decision problems, they implement procedures to exclude alternatives from the witness solution set. If this was not the case, we would not find any effect of our complexity metrics among unsatisfiable instances. Further research is needed in order to disentangle between prospective algorithms and explore how instance-level complexity measures can be used to inform the study of algorithm selection in humans.

* * *

We provide empirical evidence that studying the intrinsic computational hardness of the task environment predicts human cognitive effort and performance on a task. This has important practical implications, which could help improve human decision-making. The approach presented here could be used to quantify the computational hardness of problems people face in everyday life, such as making financial investments or health insurance decisions. Our generic approach would provide a rigorous method to estimate average quality of such decisions. Both designers of products as well as regulators could use a framework like ours to identify upper limits in the complexity of the tasks that consumers face when dealing with those products and services.

Methods

Ethics statement

The experimental protocol was approved by the University of Melbourne Human Research Ethics Committee (Ethics ID 1749594.2). Written informed consent was obtained from all participants prior to commencement of the experimental sessions. Experiments were performed in accordance with all relevant guidelines and regulations, including the Declaration of Helsinki.

Participants

A total of 47 participants were recruited in two separate groups from the general population (group 1: 24 participants; 12 female, 12 male; age range = 19-35 years, mean age = 24.1 years; group 2: 23 participants; 13 female, 10 male; age range = 18-32 years, mean age = 23.3 years). Inclusion criteria were based on age (minimum=18 years, maximum=35 years) and normal or corrected-to-normal vision.

Each group of participants were asked to solve a set of random instances of a computational problem. Group 1 participants were presented with 64 instances of the Boolean satisfiability problem (3SAT). Group 2 participants were presented with 72 instances of the decision variant of the traveling salesperson problem (TSP).

Experimental tasks

Boolean satisfiability task

This task is based on the 3-satisfiability problem. In this problem, the aim is to determine whether a boolean formula is *satisfiable*. In other words, given a propositional formula, the aim is to determine whether there exists at least one configuration of the variables (which can take values TRUE or FALSE) such that the formula evaluates to TRUE. The propositional formula in 3SAT has a specific structure. Specifically, the formula is composed of a conjunction of clauses that must all evaluate TRUE for the whole formula to evaluate TRUE. Each of these clauses, takes the form of an OR logical operator of three literals (variables and their negations). An example of a 3SAT problem is:

Does there exist $x_i \in \{TRUE, FALSE\}$ s.t.
$(\neg x_3 \vee \neg x_4 \vee x_5) \wedge (x_2 \vee x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$
evaluates $TRUE$?

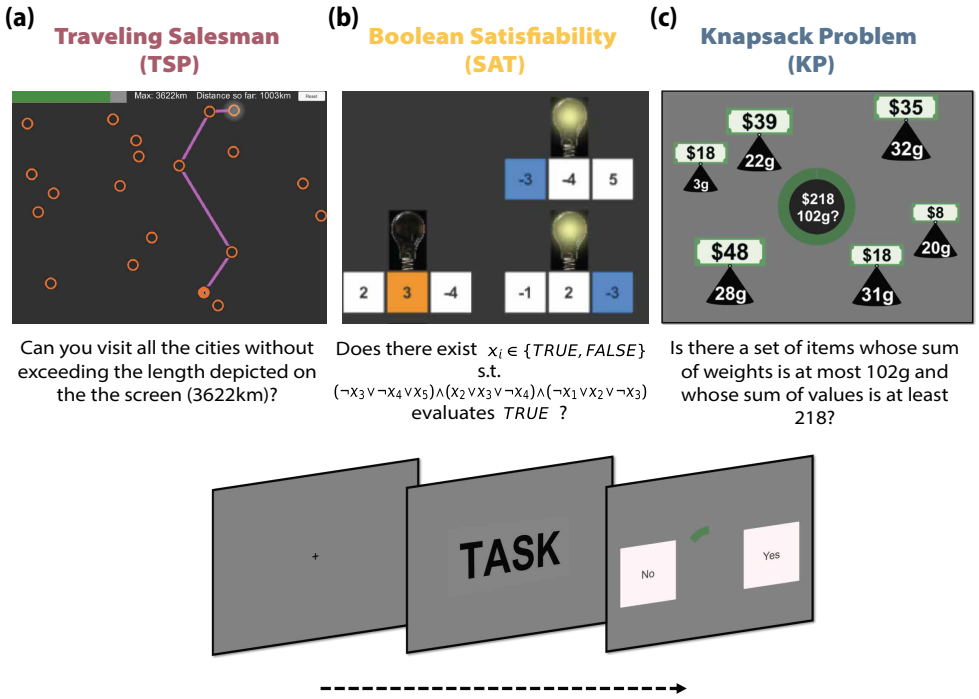
In order to represent this in an accessible way to participants we developed a task composed of switches and light bulbs (Fig 5b). Participants were presented with a set of light bulbs (clauses), each of which had three switches underneath (literals) that were represented by a positive or negative number. The number on each switch represented the variable number, which could be turned on or off (TRUE or FALSE). The aim of the task is to determine whether there exists a way of turning on and off variables such that all the light bulbs are turned on (that is, the formula evaluates TRUE).

At the beginning of each trial, participants were presented with a different instance of the 3SAT problem. A bar in the top-right corner of the screen indicated the time remaining in the trial. Each participant completed 64 trials (4 blocks of 16 trials with a rest period of 60 seconds between blocks). Trials were self-paced with a time limit of 110 seconds. Participants could use the mouse to click on any of the variables to select their value ($\{blue = TRUE, orange = FALSE\}$). A light bulb above each clause indicated whether a clause evaluated to TRUE (light on) given the selected values of the variables underneath it. The number of clicks in each trial was limited to 20. The purpose of this limit was to discourage participants from using a trial-and-error strategy to solve the instances. When participants were ready to submit their solution, they pressed a button to advance from the screen displaying the instance to the response screen where they responded YES or NO. The time limit to respond was 3 seconds, and the inter-trial interval was 3 seconds as well. The order of instances and the side of the YES/NO button on the response screen were randomized for each participant.

Instance sampling A random instance is a selection of clauses and literals in which M clauses of three literals are chosen randomly. Each of the literals is associated with one of N variables. Both numerical [23] and analytical [24] evidence suggests that in the limit $N \to \infty$, there exists a value of the clause to variables ratio $\alpha = M/N$, α_s^{SAT}, such that typical instances are satisfiable for $\alpha < \alpha_s^{SAT}$, while typical instances are unsatisfiable for $\alpha > \alpha_s^{SAT}$. The current best estimate for the satisfiability threshold, α_s^{SAT}, as $N \to \infty$ is 4.267 [24] (note that the value of α_s is a function of the number of literals per clause, which was fixed at 3 in this study). As $N \to \infty$, instances near the threshold are on average harder to solve [18, 22]. We exploit both the threshold phenomenon in satisfiability and its link to computational hardness.

We generated random instances with different degrees of complexity by varying α. We picked a value of α, starting at the lower bound of its range and incrementing in steps of 0.1 until the upper bound was reached. For each value of α, we computed the number of clauses M by multiplying α and a fixed value of N and rounding to the nearest integer. N (and the time limit for the task) was determined before hand using pilot data to ensure that the task was not too easy nor too hard for participants (i.e. to ensure sufficient variation in performance). Importantly, N was also restricted to values in which the corresponding number of clauses could fit in the screen of the task. Specifically, we restricted the number of clauses to be at most 36.

Figure 5: **Experimental Tasks. (a) Traveling salesperson task.** Participants are given a list of cities displayed on a rectangular map on the screen and a limit L on path length. The problem is to determine whether there *exists* a path connecting all N cities with a distance at most L. The task was interactive. Participants could click from city to city and the corresponding path and distance traveled would display and update automatically. This stage lasted a maximum of 40 seconds. Afterwards, participants had 3 seconds to make their response. **(b) Boolean satisfiability task.** In this task, the aim is to determine whether a Boolean formula is *satisfiable*. The Boolean formula is represented with a set of light bulbs (clauses), each of which has three switches underneath (literals) that are characterized by a positive or negative number. The number on each switch represents the variable number, which can be turned on or off (TRUE or FALSE). The aim of the task is to determine whether there exists a way of turning on and off variables such that all the light bulbs are turned on (the corresponding Boolean formula evaluates to TRUE). The task was interactive. Participants could click on switches to turn them on and the corresponding literals and light bulbs would change color automatically. This stage had a time limit of 110 seconds. Afterwards, participants had 3 seconds to make their response (either a 'YES' or 'NO'). **(c) Knapsack decision task.** Participants are presented with a set of items with different values and weights. Additionally, a capacity constraint and target profit are shown at the center of the screen. The aim is to ascertain whether there exists a subset of items for which (1) the sum of weights is lower or equal to the capacity constraint and (2) the sum of values yields at least the target profit. The task was not interactive. This stage lasted for 25 seconds. Finally, participants had 2 seconds to make their response.



Once N was fixed, we generated 1000 random instances for each value of M. Each random instance was generated by first selecting the literals for each clause. Each literal is represented by a positive or negative sign (negation of a variable) and is sampled from the set $\{-1, +1\}$ with equal probability. Afterwards, three variables were selected for each clause by sampling without replacement from the set of N variables [18, 23].

From the randomly generated instances we first determined the satisfiability threshold of our finite

instances ($N = 5$). That is, we calculated the value of α at which half of the randomly generated instances were satisfiable and half were unsatisfiable. This was the case for $\alpha = 4.8$. Based on this we selected a subset of random instances to use in the task.

We asked participants to solve a set of instances randomly sampled from three different regions: an underconstrained region ($\alpha \ll \alpha_s^{SAT}$), a region around the satisfiability threshold ($\alpha \sim \alpha_s^{SAT}$) and an overconstrained region ($\alpha \gg \alpha_s^{SAT}$). Instances near the satisfiability threshold are defined to have a *high TCC*, whereas instances further away from the satisfiability threshold (in the under-constrained or over-constrained regions) are defined to have a *low TCC*. We selected 16 instances from the underconstrained region ($\alpha = 2$) and 16 instances from the overconstrained region ($\alpha = 7$). We then sampled 32 instances near the satisfiability threshold ($\alpha = 4.8$), such that 16 of the selected instances were satisfiable and 16 were not satisfiable.

In order to also ensure a sufficient degree of variability between instances near the satisfiability threshold, we added an additional constraint in the sampling. For each set of instances (satisfiable and not satisfiable) we forced half to have algorithmic complexity less than the median algorithmic complexity at this value of α, and the other half to be harder than the median. The algorithmic complexity was estimated using an algorithm-specific ex-post complexity measure of a widely-used algorithm (*Gecode* propagations parameter). Gecode is a generic solver for constraint satisfaction problems that uses a constraint propagation technique with different search methods, such as branch-and-bound. We chose an output variable, the number of propagations, that indicates the difficulty for the algorithm of finding a solution and whose value is highly correlated with computational time. We did not use compute time directly as a measure of complexity because for instances of small size, like the ones used in this study, compute time is highly confounded with overhead time. Thus, our set of instances in the region $\alpha \sim \alpha_s$ comprised 8 instances in each of the following categories $\{\text{satisfiable, unsatisfiable}\} \times \{\text{low/high algorithmic difficulty}\}$.

Traveling salesperson task

This task is based on the traveling salesperson problem. Given a set of N cities displayed on a rectangular map on the screen and a limit L on path length, the decision problem is to answer whether there *exists* a path connecting all N cities with a distance of at most L (Fig 5a).

In the TSP task, each participant completed 72 trials (3 blocks of 24 trials with a rest period of 30 seconds between blocks). Each trial presented a different instance of TSP. Trials were self-paced with a time limit of 40 seconds. Participants could use the mouse to trace routes by clicking on the dots indicating the different cities. The length of the chosen route was indicated at the top of the screen (together with the maximum route length of the instance). When participants were ready to submit their answer, they pressed a button to advance from the screen displaying the cities to the response screen where they responded YES or NO. The time limit to respond was 3 seconds, and the inter-trial interval was 3 seconds as well. The order of instances and the sides of the YES/NO button on the response screen were randomized for each participant.

Instance sampling A TSP instance is a collection of N cities, a matrix of distances $\boldsymbol{d}$ between each pair of cities, and a limit L on path length. Here, we restrict the problem to the euclidean TSP; that is,

we constraint our distance matrices $\boldsymbol{d}$ to those that can be represented in a two-dimensional map of area M^2.

Just like for 3SAT, it has been proposed that there exists a parameter α^{TSP} that captures the constrainedness of the problem, specifically $\alpha^{TSP} = -L/(M\sqrt{N})$ [17]. Evidence suggests that in the limit $N \to \infty$, there exists a value of α, α_s^{TSP}, such that typical instances are satisfiable for $\alpha \ll \alpha_s^{TSP}$, while typical instances are unsatisfiable for $\alpha \gg \alpha_s^{TSP}$. α_s^{TSP} for the euclidean TSP is estimated at -0.7124 ± 0.0002 in the limit $N \to \infty$ [17, 44]. As $N \to \infty$ instances near α_s^{TSP} have been shown to be, on average, harder to solve [17]. We use the this insight to vary typical-case complexity of finite instances.

Instances of the TSP had $N = 20$ cities. This value, and the time limit for the task, were determined using pilot data to ensure that the task was not too easy nor too hard for participants (i.e. to ensure sufficient variation in performance). Random instances of the euclidean TSP were then generated by choosing (x,y) coordinates for each of the $N = 20$ cities, uniformly at random from a square with side length $M = 1000$ [17]. We generated 100 sets of coordinates; that is, 100 distance matrices $\boldsymbol{d}$. For each distance matrix, we generated instances with different values of L. We did this by varying the value of α, which was incremented in the range $[-0.25, -1.25]$ with step size 0.02.

To determine the location of the satisfiability threshold in our sample of random instances (with $N = 20$), we determined the value of α at which half of the randomly generated instances were satisfiable and half were unsatisfiable. The satisfiability threshold was located at $\alpha^{TSP} = -0.85$. We randomly sampled instances at this value of α such that half of the selected instances were satisfiable and half were not satisfiable. We also ensured that half of the instances had a number of propagations above the median and half of them had a number of propagations below the median (see description of 3SAT above for details). Thus, our set of of instances in the region $\alpha \sim \alpha_s$ comprised 9 instances in each of the four following categories: $\{\text{satisfiable}, \text{unsatisfiable}\} \times \{\text{low/high algorithmic difficulty}\}$.

For the underconstrained region, $\alpha \ll \alpha_s$, we randomly chose 18 instances from the set of 100 randomly generated instances with $\alpha^{TSP} = -0.99$. For the overconstrained region, $\alpha \gg \alpha_s$, we randomly chose 18 instances from the set of 100 randomly generated instances with $\alpha^{TSP} = -0.71$. We made sure that no two instances in our set of selected instances had the same set of city coordinates.

Knapsack task

In this paper we report on the experimental data collected on the knapsack decision task by Franco et al. [28]. Statistical results from [28] were used when available.

The knapsack task is based on the 0-1 knapsack problem (KP). An instance of this problem consists of a set of items $I = \{1, \ldots, N\}$ with weights $\langle w_1, \ldots, w_N \rangle$ and values $\langle v_1, \ldots, v_N \rangle$, and two positive numbers c and p denoting the capacity and profit constraint (of the knapsack). The problem is to decide whether there exists a set $S \subseteq I$ such that $\sum_{i \in S} w_i \leq c$, that is, the weight of the knapsack is less than or equal to the capacity constraint; and $\sum_{i \in S} v_i \geq p$, that is, the value of the knapsack is greater than or equal to the profit constraint.

In their study they implemented the knapsack decision problem in the form of the task presented in

Fig 5c. In their task all instances had 6 items ($N = 6$) and w_i, v_i, c and p were integers. In the task each participant completed 72 trials (3 blocks of 24 trials with a rest period of 60s between blocks). Each trial presented a different instance of the KP. Trials had a time limit of 25 seconds and were *not* self-paced. A green circle at the center of the screen indicated the time remaining in each stage of the trial. During the first 3 seconds participants were presented with a set of items of different values and weights. Then, both capacity constraint and target profit were shown at the center of the screen for the remainder of the trial (22 seconds). No interactivity was incorporated into the task; that is, participants could not click on items. When the time limit was reached, participants were presented with the response screen where they responded YES or NO. The time limit to respond was 2 seconds, and the inter-trial interval was 5 seconds. The order of instances and the sides of the YES/NO button on the response screen were randomized for each participant.

Instance sampling It has been proposed that there exists a set of parameters $\bar{\alpha}^{KP} = (\alpha_c^{KP}, \alpha_p^{KP})$ that captures the constrainedness of the problem, specifically $\alpha_p^{KP} = \frac{p}{\sum_{i=1}^{N} v_i}$ and $\alpha_c^{KP} = \frac{c}{\sum_{i=1}^{N} w_i}$ [30]. These parameters characterize where typical instances are generally satisfiable (under-constrained region), where they are unsatisfiable (over-constrained region) and where the probability of satisfiability is close to 50% (satisfiability threshold). Instance near the satisfiability threshold have been shown to be, on average, harder to solve [30].

Instances in Franco et al. [28] were selected such that α_c^{KP} was fixed ($\alpha_c^{KP} \in [0.40, 0.45]$) and the instance constrainedness varied according to α_p^{KP}. 18 instances were selected from the under-constrained region ($\alpha_p \in [0.35, 0.4]$; *low TCC*) and 18 from the over-constrained region ($\alpha_p \in [0.85, 0.9]$; *low TCC*). Additionally, 18 satisfiable instances and 18 *unsatisfiable* instances were sampled near the satisfiability threshold ($\alpha_p \in [0.6, 0.65]$; *high TCC*).

Like for 3SAT and TSP, high TCC instances were selected such that they varied according to the number of propagations (see description of 3SAT sampling for details).

Procedure

After reading the plain language statement and providing informed consent, participants were instructed in the task and completed a practice session. Each experimental session lasted around 110 minutes. The tasks were programmed in Unity3D [45] and administered on a laptop.

Participants received a show-up fee of AUD 10 and additional monetary compensation based on performance. In the 3SAT, they additionally received AUD 0.6 for each correct instance submitted plus a bonus of AUD 0.31 per instance if all instances in the task were solved correctly. In the TSP, participants received 0.3 per correct instance submitted plus 0.14 per instance if all instances were solved correctly. In the KP task [28], participants received a show-up fee of A$10 and earned A$0.7 for each correct answer.

Note that the 3SAT and TSP tasks were self-paced (with time limits per trial), whereas the KP was not.

Derivation of metrics

We estimated a collection of metrics based on the features of each instance and its solution space. We estimated one feature-space metric and several solution-space metrics. We first defined Typical-case complexity (TCC) according to the problem-parameter α for each task. Estimation of this metric is tightly related to the instance sampling procedure and its derivation is described in the previous section. Instances were sampled such that there was an equal number of instances with low and high TCC on each of the problems.

Once instances for the tasks were sampled, we estimated their solution-space metrics. We estimated the number of solution witnesses for 3SAT instances using exhaustive search and used the Gecode algorithm [46] for TSP instances. For the TSP, we allowed the algorithm to stop after finding 30,000 solution witnesses. This was done to reduce the computational requirements of solving an instance. 15 TSP instances reached the 30,000 maximum imposed. Given the variability in the number of witnesses in the TSP, the results on number witnesses are reported in logarithmic scale (natural logarithm).

We define the instance complexity metric (IC) as the absolute value of the normalized difference between target value of the decision variant and the maximum value attainable of the corresponding optimization variant. In the KP, the optimization variant's problem is to find the maximum value attainable given the weights, values and capacity. In the TSP, the optimization variant is to minimize the path traveled given a distance matrix. In the 3SAT, the optimization variant (MAXSAT) is to find the maximum number of satisfiable clauses given the Boolean formula presented. Explicitly, IC is defined as follows:

$$IC_{KP} = \left|\alpha_p - \alpha_p^\star\right| = \left|\frac{\text{Target profit} - \text{Maximum profit attainable}}{\sum v_i}\right|$$

$$IC_{TSP} = \left|\alpha - \alpha^\star\right| = \left|\frac{\text{Path limit - Minimum path}}{\sqrt{\text{Map area}} \times \text{Number of cities}}\right|$$

$$IC_{SAT} = \left|\alpha - \alpha^\star\right| = \left|\frac{\text{Number of clauses - Max number of clauses set to TRUE}}{\text{Number of variables}}\right|$$

In order to estimate the instance complexity metric (IC), the optimization variant of each instance needs to be solved. These optima were estimated using Gecode [46] in TSP and using the *RC2* algorithm from the 'pysat' python library [47]. For the KP we used the metrics estimated in Franco et al. [28].

Statistical analysis

Python (version 3.6) was used to sample and solve instances. The R programming language was used to analyze the behavioral data. All of the linear mixed models (LMM), generalized logistic mixed models (GLMM) and censored linear mixed models (CLMM) included random effects on the intercept for participants (unless otherwise stated). Different models were selected according to the data structure. GLMM were used for models with binary dependent variables, LMM were used for continuous dependent variables and CLMM were used for censored continuous dependent variables (e.g., time-on-task).

All the models were fitted using a Bayesian framework implemented using the probabilistic programming language Stan via the R package 'brms' [48]. Default priors were used. All population-level

effects of interest had uninformative priors; i.e., an improper flat prior over the reals. Intercepts had a student-t prior with 3 degrees of freedom and a scale parameter that depended on the standard deviation of the response after applying the link function. The student-t distribution was centered around the mean of the dependent variable. Sigma values, in the case of Gaussian-link models, had a half student-t prior (restricted to positive values) with 3 degrees of freedom and a scale parameter that depended on the standard deviation of the response after applying the link function. Standard deviation of the participant-level intercept parameters had a half student-t prior that was scaled in the same way as in the sigma prior.

Each of the models presented was estimated using four Markov chains. The number of iterations per chain was by default set to 2000. This parameter was adjusted to 4000 on some models to ensure convergence. Convergence was verified using the convergence diagnostic $\hat{R}$. All models presented reach an $\hat{R} \approx 1$.

Statistical tests were performed based on the 95% credible interval estimated using the highest density interval (HDI) of the posterior distributions calculated via the R package 'parameters' [49]. For each statistical test we report both the median ($\beta_{0.5}$) of the posterior distribution and its corresponding credible interval ($HDI_{0.95}$).

For the knapsack task, we report the statistical results from [28] if available and are, here, reported as effect estimates (β) and P-Values (P). Otherwise we used the data available at the OSF (project: `https://doi.org/10.17605/OSF.IO/T2JV7`) to run statistical tests on the behavioral data. These tests were performed and reported following the same Bayesian approach used in the TSP and 3SAT analysis.

Some trials and participants were excluded due to different reasons. In the 3SAT task, two participants were omitted from the analysis given that their performance (close to 50%) differed significantly from the group. Additionally, 10 trials (from 9 participants) were omitted given that no answer was given. One participant was excluded from the time-on-task analysis since they never advanced to the response screen before the time limit. In the TSP, one participant was excluded from the analysis given that they did not understand the instructions. This was determined during the course of the experiment. Additionally 9 trials (from 8 participants) were omitted given that no answer was selected. Finally, in the knapsack task, 13 trials (from 8 participants) were excluded in which no response was made.

Data and code availability

The behavioral data and the data analysis code are both available at the Open Science Framework. The 3SAT and TSP tasks are also available there (project: `https://osf.io/tekqa/`).

Acknowledgments

The authors thank Elizabeth Bowman for her support of the laboratory experiments.

Funding

This research was supported by a University of Melbourne Graduate Research Scholarship from the Faculty of Business and Economics (Franco) and a Kinsman Scholarship (Doroc). Bossaerts acknowledges financial support through a R@MAP Chair from the University of Melbourne.

Author contributions

CM, JPF, KD, PB and NY designed the study; NY and JPF performed instance selection; KD and JPF programmed the experimental tasks; KD ran a pilot version of this study; JPF performed data collection and analysis; JPF, CM, KD, NY and PB wrote the manuscript.

Supporting Information Legends

S1 Appendix. Satisfiability and TCC. Joint effect of satisfiability and TCC on human performance and time-on-task.

S2 Appendix. Search strategies. The effect of instance properties on search length.

S3 Appendix. Summary statistics.

S4 Appendix. TCC and the number of witnesses. The number of witnesses drive the effect of TCC on performance in satisfiable instances.

S5 Appendix. Instance complexity in 3SAT. Confounding factors.

S6 Appendix. Supplementary Tables.

S1 Table Human performance in the Boolean satisfiability task.

S2 Table Human performance in the traveling salesperson task.

S3 Table Time-on-task in the Boolean satisfiability task.

S4 Table Time-on-task in the traveling salesperson task.

S5 Table Human performance an the number of solution witnesses.

S6 Table Human performance in the knapsack task.

S7 Table Number of clicks in the Boolean satisfiability task.

S8 Table Number of clicks in the traveling salesperson task.

References

1. Simon, H. A. Invariants of human behavior. *Annual Review of Psychology* **41,** 1–19. ISSN: 00664308. www.annualreviews.org (1990).
2. Tversky, A. & Kahneman, D. Judgment under Uncertainty: Heuristics and Biases. *Science* **185,** 1124–1131 (1974).
3. Griffiths, T. L., Lieder, F. & Goodman, N. D. Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in Cognitive Science* **7,** 217–229. ISSN: 17568765 (2015).
4. Gershman, S. J., Horvitz, E. J. & Tenenbaum, J. B. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science* **349,** 273–278. ISSN: 0036-8075, 1095-9203 (2015).
5. Todd, P. M. & Gigerenzer, G. *Ecological rationality: Intelligence in the world.* xviii, 590–xviii, 590. ISBN: 978-0-19-531544-8 (Hardcover) (Oxford University Press, Todd, Peter M.: Cognitive Science Program, Indiana University, 1101 E. 10th St., Bloomington, IN, US, 47405, peter.m.todd@gmail.com, 2012).
6. Murawski, C. & Bossaerts, P. How Humans Solve Complex Problems: The Case of the Knapsack Problem. *Nature (Scientific Reports)* **6.** ISSN: 2045-2322 (2016).
7. Dry, M., Lee, M. D., Vickers, D. & Hughes, P. Human Performance on Visually Presented Traveling Salesperson Problems with Varying Numbers of Nodes. *The Journal of Problem Solving* **1.** ISSN: 1932-6246. http://dx.doi.org/10.7771/1932-6246.1004 (2006).
8. Guid, M. & Bratko, I. *Search-Based Estimation of Problem Difficulty for Humans* in *Artificial Intelligence in Education. AIED 2013. Lecture Notes in Computer Science* (eds H.C., L., K., Y., J., M. & P., P.) **7926** (Springer, Berlin, Heidelberg, 2013).
9. MacGregor, J. N. & Chu, Y. Human Performance on the Traveling Salesman and Related Problems: A Review. *The Journal of Problem Solving* **3,** 1. ISSN: 1932-6246. http://dx.doi.org/10.7771/1932-6246.1090 (2011).
10. Hirtle, S. C. & Gärling, T. Heuristic rules for sequential spatial decisions. *Geoforum* **23,** 227–238. ISSN: 00167185 (May 1992).
11. Ohlsson, S. The Problems with Problem Solving: Reflections on the Rise, Current Status, and Possible Future of a Cognitive Research Paradigm 1. *The Journal of Problem Solving* **5.** http://dx.doi.org/10.7771/1932-6246.1144 (2012).
12. Gigerenzer, G. & Gaissmaier, W. Heuristic decision making. *Annual Review of Psychology* **62,** 451–482. ISSN: 00664308 (Jan. 2011).
13. Payne, J. W., Bettman, J. R. & Johnson, E. J. *The Adaptive Decision Maker* (1993).
14. Arora, S. & Barak, B. *Computational complexity : a modern approach* 579. ISBN: 0521424267 (Cambridge University Press, 2009).

15. Van Rooij, I., Blokpoel, M., Kwisthout, J. & Wareham, T. *Cognition and Intractability* (Cambridge University Press, Apr. 2019).

16. Blum, M. & Vempala, S. The complexity of human computation via a concrete model with an application to passwords. *Proceedings of the National Academy of Sciences of the United States of America* **117,** 9208–9215. ISSN: 10916490. www.pnas.org/cgi/doi/10.1073/pnas.1801839117 (2020).

17. Gent, I. P. & Walsh, T. The TSP phase transition. *Artificial Intelligence* **88,** 349–358. ISSN: 00043702. http://linkinghub.elsevier.com/retrieve/pii/S0004370296000306 (1996).

18. Cheeseman, P., Kanefsky, B. & Taylor, W. M. *Where the Really Hard Problems Are* in *The 12nd International Joint Conference on Artificial Intelligence* (1991), 331–337. ISBN: 1-55860-160-0.

19. Bogdanov, A. & Trevisan, L. Average-Case Complexity. *arXiv preprint cs/0606037.* http://arxiv.org/abs/cs/0606037 (2006).

20. Kotovsky, K., Hayes, J. R. & Simon, H. A. Why are some problems hard? Evidence from Tower of Hanoi. *Cognitive Psychology* **17,** 248–294. ISSN: 00100285 (Apr. 1985).

21. Shepard, R. N. & Metzler, J. Mental rotation of three-dimensional objects. *Science* **171,** 701–703. ISSN: 00368075 (Feb. 1971).

22. Percus, A., Istrate, G. & Moore, C. *Computational Complexity and Statistical Physics* 384. ISBN: 9780199760565 (Oxford University Press, 2006).

23. Monasson, R. *et al.* Determining computational complexity from characteristic 'phase transitions'. *Nature* **400,** 133–137. ISSN: 0028-0836 (1999).

24. Mézard, M., Parisi, G. & Zecchina, R. Analytic and algorithmic solution of random satisfiability problems. *Science* **297,** 812–815. ISSN: 00368075 (2002).

25. Krzakala, F., Montanari, A., Ricci-Tersenghi, F., Semerjian, G. & Zdeborova, L. Gibbs States and the Set of Solutions of Random Constraint Satisfaction Problems. *Proceedings of the National Academy of Sciences of the United States of America* **104,** 10318–23. ISSN: 0027-8424 (June 2006).

26. Selman, B. & Kirkpatrick, S. Critical behavior in the computational cost of satisfiability testing. *Artificial Intelligence* **81,** 273–295. ISSN: 0004-3702 (Mar. 1996).

27. Zdeborová, L. & Mézard, M. Constraint satisfaction problems with isolated solutions are hard. *Journal of Statistical Mechanics: Theory and Experiment* **2008,** P12004. ISSN: 1742-5468 (Dec. 2008).

28. Franco, J. P., Yadav, N., Bossaerts, P. & Murawski, C. Structural properties of individual instances predict human effort and performance on an NP-Hard problem. *bioRxiv,* 405449 (July 2018).

29. Preuschoff, K., Bossaerts, P. & Quartz, S. R. Neural Differentiation of Expected Reward and Risk in Human Subcortical Structures. *Neuron* **51,** 381–390. ISSN: 08966273 (2006).

30. Yadav, N., Murawski, C., Sardina, S. & Bossaerts, P. *Is Hardness Inherent In Computational Problems? Performance Of Human And Digital Computers On Random Instances Of The 0-1 Knapsack Problem* in *24th European Conference on Artificial Intelligence (ECAI 2020).* (2020).

31. Gigerenzer, G. & Selten, R. *Bounded rationality : the adaptive toolbox* 377. ISBN: 9780262072144 (MIT Press, 2001).

32. Frixione, M. Tractable competence. *Minds and Machines* **11,** 379–397. ISSN: 09246495 (2001).

33. Tsotsos, J. K. Analyzing vision at the complexity level. *Behavioral and Brain Sciences* **13,** 423–445. ISSN: 14691825 (1990).

34. Bourgin, D., Lieder, F., Reichman, D., Talmon, N. & Griffiths, T. L. *The Structure of Goal Systems Predicts Human Performance* in *Proceedigns of the 39th Annual Meeting of the Cognitive Science Society* (eds Gunzelmann, G., Howes, A., Tenbrink, T. & Davelaar, A.) (Cognitive Science Society, Austin, TX, 2017), 1660–1665.

35. Stazyk, E. H., Ashcraft, M. H. & Hamann, M. S. A network approach to mental multiplication. *Journal of Experimental Psychology: Learning, Memory, and Cognition* **8,** 320–335. ISSN: 02787393 (1982).

36. De Visscher, A. & Noël, M. P. The detrimental effect of interference in multiplication facts storing: Typical development and individual differences. *Journal of Experimental Psychology: General* **143,** 2380–2400. ISSN: 00963445 (2014).

37. Smith-Miles, K. & Lopes, L. Measuring instance difficulty for combinatorial optimization problems. *Computers and Operations Research* **39,** 875–889. ISSN: 03050548. http://dx.doi.org/10.1016/j.cor.2011.07.006 (2012).

38. Hill, R. R. & Reilly, C. H. Effects of coefficient correlation structure in two-dimensional knapsack problems on solution procedure performance. *Management Science* **46,** 302–317. ISSN: 00251909. https://www.jstor.org/stable/2634765 (2000).

39. Van Hemert, J. I. *Property analysis of symmetric travelling salesman problem instances acquired through evolution* in *European Conference on Evolutionary Computation in Combinatorial Optimization* (Springer Berlin Heidelberg, 2005), 122–131. https://link.springer.com/chapter/10.1007/978-3-540-31996-2_12.

40. Nudelman, E., Leyton-Brown, K., Hoos, H. H., Devkar, A. & Shoham, Y. Understanding random SAT: Beyond the clauses-to-variables ratio. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **3258,** 438–452. ISSN: 16113349. https://link.springer.com/chapter/10.1007/978-3-540-30201-8_33 (2004).

41. Van Opheusden, B. & Ma, W. J. *Tasks for aligning human and machine planning* 2019. https://doi.org/10.1016/j.cobeha.2019.07.002.

42. Budzynski, L., Ricci-Tersenghi, F. & Semerjian, G. Biased landscapes for random Constraint Satisfaction Problems. *Journal of Statistical Mechanics: Theory and Experiment* **2019,** 023302 (Feb. 2019).

43. Ausiello, G. *et al. Complexity and Approximation : Combinatorial Optimization Problems and Their Approximability Properties* 524. ISBN: 9783642635816 (Springer Berlin Heidelberg, 1999).

44. Johnson, D. S., McGeoch, L. A. & Rothberg, E. E. *Asymptotic experimental analysis for the Held-Karp traveling salesman bound* in *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms* **Part F1294** (1996), 341–350. ISBN: 0898713668.

45. *Unity 3D* 2017. `https://unity3d.com/`.

46. Gecode Team. *Gecode: Generic Constraint Development Environment* 2006. `http://www.gecode.org`.

47. Ignatiev, A., Morgado, A. & Marques-Silva, J. *PySAT: A Python Toolkit for Prototyping with SAT Oracles* in *SAT* (2018), 428–437. `https://doi.org/10.1007/978-3-319-94144-8_26`.

48. Bürkner, P.-C. brms: An R Package for Bayesian Multilevel Models Using Stan. *Journal of Statistical Software* **80,** 1–28 (2017).

49. Lüdecke, D., Ben-Shachar, M. S. & Makowski, D. Describe and understand your model's parameters. *CRAN*. `https://easystats.github.io/parameters` (2020).

S1 Appendix: Satisfiability and TCC

In the results section we found that TCC had a negative effect on performance. Since we are interested in understanding what generic features make instances hard for humans to solve, we explored whether TCC and satisfiability interact to make instances harder. We first explore the interaction of TCC and satisfiability on performance. Performance was not affected by satisfiability in low TCC instances on all three problems (TSP: $\beta_{0.5} = -0.16$, $HDI_{0.95} = [-0.85, 0.55]$, S1 Table Model 4; 3SAT: $\beta_{0.5} = -0.52$, $HDI_{0.95} = [-1.27, 0.11]$, S2 Table Model 4; KP: $\beta = -0.250$, $P = 0.355$, [28]; marginal effect of satisfiability, GLMM). Moreover, in line with the previous study on the KP, we found a negative effect of TCC on both satisfiable and unsatisfiable instances for both problems considered (TSP: $\beta_{0.5}^{sat} = -2.07$, $HDI_{0.95}^{sat} = [-2.64, -1.55]$, $\beta_{0.5}^{unsat} = -2.16$, $HDI_{0.95}^{unsat} = [-2.74, -1.62]$, S1 Table Model 4; 3SAT: $\beta_{0.5}^{sat} = -2.06$, $HDI_{0.95}^{sat} = [-2.56, -1.59]$, $\beta_{0.5}^{unsat} = -0.77$, $HDI_{0.95}^{unsat} = [-1.49, -0.13]$, S2 Table Model 4; the effect of TCC on performance for satisfiable and unsatisfiable instances, respectively, GLMM). Interestingly, in the 3SAT problem we found that the reduction in performance due to TCC was larger for satisfiable instances ($\beta_{0.5} = -1.29$, $HDI_{0.95} = [-2.11, -0.46]$, interaction effect of TCC and satisfiability on performance, GLMM; S2 Table Model 4). In contrast, in the TSP, as with the KP, the size of the effect of TCC on performance was similar for both satisfiable and unsatisfiable instances ($\beta_{0.5} = 0.10$, $HDI_{0.95} = [-0.65, 0.90]$, interaction effect of TCC and satisfiability on performance, GLMM; S1 Table Model 4). This suggests that unlike the KP and TSP, in the 3SAT there is an interaction effect between TCC and satisfiability on performance, which makes satisfiable instances with high TCC harder than the rest.

When analyzing how satisfiability affected time for different levels of TCC we found different results across problems. In the TSP there was no interaction effect on time between satisfiability and TCC ($\beta_{0.5} = 0.002$, $HDI_{0.95} = [-0.054, 0.058]$, interaction effect of satisfiability and TCC on time-on-task, CLMM; S6 Table Model 7), meaning that both properties had independent effects on time-on-task. In contrast, in 3SAT the effect of TCC was modulated by satisfiability in such a way that there was no effect of TCC when the instance was unsatisfiable ($\beta_{0.5} = 0.022$, $HDI_{0.95} = [-0.016, 0.063]$, marginal effect of TCC on time-on-task for unsatisfiable instances, CLMM; S5 Table Model 8). In summary, we only found an interaction effect between satisfiability and TCC in the 3SAT. This was the case for both performance and time-on-task.

S2 Appendix: Search strategies

Generic instance-level complexity metrics are able to explain differences in performance and time-spent across instances. However, it remains an open question whether the generic properties can shed light into how humans solve problems. To explore this question we investigated whether instance-level metrics could explain differences in the number of clicks across instances. This analysis was performed for TSP and 3SAT. In both tasks, participants had the opportunity to click on cities or literals throughout the trial, whereas in the KP clicking on items was not possible. Note that while for the TSP there was no limit in the number of clicks, in the 3SAT participants were only allowed to make a maximum of 20 clicks per trial. The purpose of this limit was to discourage participants from using a trial-and-error strategy to

solve the instances.

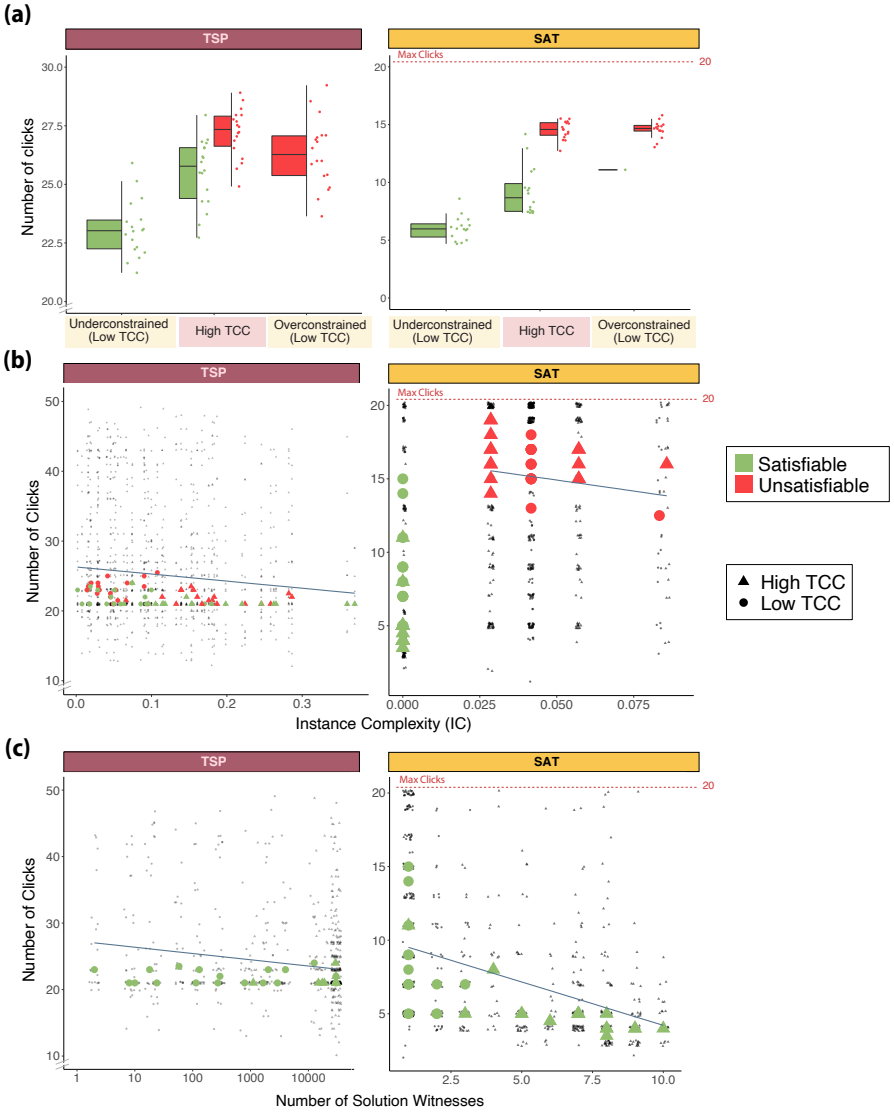
The number of clicks is a useful metric in studying the algorithms implemented by humans. Specifically, the number of clicks is related to the way that the problem state-space is explored. In the 3SAT, the state-space consists of all possible on-off switch setups (2^5 possible combinations) while in the TSP the state-space consists of all possible ordered path selections ($2^{\binom{20}{2}} = 2^{190}$ possible combinations). Arguably, participants search the state-space by clicking on different state combinations in order to decide whether an instance is satisfiable or not. Differences in the quantity of clicks used to solve an instance can shed light into how the state-space is explored (under the assumption that the state-space is explored by clicking on elements in the task).

We investigated whether generic instance-level complexity metrics could capture differences in the number of clicks. We found that participants performed more clicks on instances with high TCC, compared to low TCC, in 3SAT and TSP (TSP: $\beta_{0.5} = 1.66$, $HDI_{0.95} = [1.01, 2.33]$, GLMM S8 Table Model 1; 3SAT: $\beta_{0.5} = 1.88$, $HDI_{0.95} = [1.23, 2.54]$, CLMM, S7 Table Model 1; effect of TCC on number of clicks; Fig 6c). Additionally, less clicks were performed on satisfiable instances compared to unsatisfiable ones (TSP: $\beta_{0.5} = -2.48$, $HDI_{0.95} = [-3.13, -1.85]$, LMM S8 Table Model 2; 3SAT: $\beta_{0.5} = -7.41$, $HDI_{0.95} = [-7.95, -6.88]$, CLMM, S7 Table Model 2; effect of satisfiability on number of clicks; Fig 6b). We explored how these two metrics jointly affected the length of search and we found that both effects were still significant when controlling for each other in the TSP (S8 Table Model 3; Fig 6b). However, in 3SAT the positive effect of TCC on the number of clicks was only present on satisfiable instances (S7 Table Model 3).

We then explored how the solution-space complexity metrics affected the length of search. Among satisfiable instances a higher number of witnesses was related to a lower amount of clicks (TSP: $\beta_{0.5} = -0.41$, $HDI_{0.95} = [-0.54, -0.26]$, LMM S8 Table Model 5; 3SAT: $\beta_{0.5} = -0.59$, $HDI_{0.95} = [-0.69, -0.49]$, CLMM, S7 Table Model 5; effect of number of witnesses on the number of clicks; Fig 6c). Additionally, we found that a higher IC was related to lower number of clicks in the TSP and on unsatisfiable 3SAT instances (TSP: $\beta_{0.5} = -10.22$, $HDI_{0.95} = [-13.87, -6.37]$, LMM, S8 Table Model 4; 3SAT: $\beta_{0.5} = -29.88$, $HDI_{0.95} = [-54.52, -6.63]$, CLMM, S7 Table Model 4); effect of IC on number of clicks; Fig 6b). We excluded satisfiable 3SAT instances from the analysis since we are unable to disentangle the effect of IC and satisfiability; all satisfiable instances have an $IC = 0$. In the TSP we investigated the joint effect of IC and satisfiability on the number of clicks and found that the effects were still significant when controlling for each other and that there was no interaction effect between the variables ($\beta_{0.5} = -5.95$, $HDI_{0.95} = [-13.43, 1.78]$, interaction effect between IC and satisfiability, LMM, S8 Table Model 6).

Taken together, these findings suggest that the length of search in the state-space can be partially explained by properties of the instance, namely satisfiability and complexity. We found that there was an positive effect of TCC on search length and that the search was in general longer on unsatisfiable instances. Additionally, lower values of IC and number of witnesses were related to a longer search.

Figure 6: **Number of Clicks. (a) Satisfiability and TCC.** Median number of clicks performed while solving an instance before submitting an answer. Each colored dot represents an instance of a problem. **(b) IC.** Each green and orange shape represent the median number of clicks for each instance of TSP and 3SAT problems. The blue lines represents the marginal effect of IC (LMM, S8 Table Model 4, CLMM S7 Table Model 4). Satisfiable 3SAT instances are excluded from the 3SAT model since we are unable to disentangle the effect of IC and satisfiability. Each black dot corresponds to the number of clicks by a single participant on a particular instance. The range of number of clicks presented for the TSP ($[10, 50]$) contains more than 98% of observations. **(c) Number of solution witnesses.** Each green shape represents the median time-on-task per instance. The blue line represents the marginal effect of the number of solution witnesses (LMM, S8 Table Model 5 and CLMM, S7 Table Model 5). *The box-plots represent the median, the interquartile range (IQR) and the whiskers extend to a maximum length of 1.5*IQR*



S3 Appendix: Summary statistics

Boolean satisfiability task

On average, participants chose the 'YES' option on 45% of trials ($\min = 28\%$, $\max = 61\%$). Performance did not vary during the course of the task ($\beta_{0.5} = 0.001$, $HDI_{0.95} = [-0.008, 0.011]$, main

effect of trial number on performance, generalized logistic mixed model (GLMM); S2 Table Model 1), suggesting that neither experience with the task nor mental fatigue affected task performance. However, time spent did vary throughout the task. As the task progressed they spent on average less time on a trial ($\beta_{0.5} = -0.005$, $HDI_{0.95} = [-0.006, -0.004]$, main effect of trial number on time-on-task -as a proportion of the maximum possible time-, censored linear mixed effects model (CLMM); S5 Table Model 1).

Traveling salesperson task

On average, participants chose the 'YES' option on 50% of trials ($\min = 35\%$, $\max = 60\%$). Consistent with our results for the 3SAT, performance did not vary during the course of the task ($\beta_{0.5} = 0.00$, $HDI_{0.95} = [-0.003, 0.013]$, main effect of trial number on performance, GLMM; S1 Table Model 1), but participants spent less time on a trial as they progressed ($\beta_{0.5} = -0.002$, $HDI_{0.95} = [-0.003, -0.001]$, main effect of trial number on time-on-task -as a proportion of maximum possible time-), linear mixed effects model (LMM); S6 Table Model 1).

Knapsack decision task

On average, participants chose the 'YES' option in 48.1% of trials ($\min = 0.32$, $\max = 0.60$, $SD = 0.06$). Performance did not vary during the course of the task ($\beta = 0.005$, $P = 0.196$, main effect of trial number on performance, GLMM; [28]).

S4 Appendix: TCC and the number of witnesses

It is feasible that the effect of TCC on performance, on satisfiable instances, is driven by the number of witnesses. After all, TCC is constructed from a metric of expected constrainedness (α). We thus examined the link between these features of an instance. As expected, we found that the number of witnesses of low TCC instances was significantly higher than that of instances with high TCC in all three problems ($P_{SAT} < 0.001$, $P_{TSP} < 0.001$, $P_{KP} < 0.001$, p-values of unpaired t-tests; Fig 3). This corroborates the link between the typical-case constrainedness (α) and the solution-space constrainedness of satisfiable instances.

Based on the previous results, we hypothesized that the effect on performance of TCC (on satisfiable instances) is driven by the number of witnesses. To test this hypothesis, we studied the effect of TCC on performance while controlling for the number of witnesses. In line with our conjecture, we found that once we controlled for the number of witnesses the marginal effect of TCC on performance was not significant on all three problems (3SAT: $\beta_{0.5} = 0.47$, $HDI_{0.95} = [-0.30, 1.22]$; TSP: $\beta_{0.5} = -0.12$, $HDI_{0.95} = [-0.84, 0.68]$; KP: $\beta_{0.5} = 0.17$, $HDI_{0.95} = [-0.57, 0.90]$; marginal effect of TCC on performance, GLMM; S4 Table Models 2,5,8). We studied further this relation and tested whether there was an interaction effect of TCC and the number of witnesses on performance. The results were different across problems. We found a significant interaction in the KP, an inconclusive result in the 3SAT and a non-significant results in the TSP (KS: $\beta_{0.5} = 0.54$, $HDI_{0.95} = [0.26, 0.81]$; 3SAT: $\beta_{0.5} = 0.56$, $HDI_{0.95} = [-0.00, 1.22]$; TSP: $\beta_{0.5} = -0.26$, $HDI_{0.95} = [-0.66, 0.15]$; interaction effect between

TCC and number of witnesses on performance, GLMM; S4 Table Models 3,6,9). Taken together, these results suggest that the effect of TCC on performance is, at least partially, driven by the number of witnesses. However, on some problems, TCC might affect human performance through other mechanisms as well.

S5 Appendix: Instance complexity in 3SAT

Unlike TSP and KP, the IC metric takes a values of zero ($IC = 0$) when the instance is satisfiable; by definition an instance is only satisfiable if the maximum number of clauses set to TRUE is equal to the number of clauses in the instance (i.e., $IC = 0$). This entails that IC would only be able to explain differences in performance on unsatisfiable instances, which (given our sampling procedure) are half of the instances used in the task. However, we did not find evidence for this explanation when we restricted our analysis to unsatisfiable instances ($R^2 = 0.001$ for unsatisfiable 3SAT instances). For this set of instances the positive relation was not significant in the 3SAT, but significant in KP and TSP (KP: $\beta_{0.5} = 13.48$, $HDI_{0.95} = [10.11, 17.30]$, S3 Table Model 3; TSP: $\beta_{0.5} = 20.10$, $HDI_{0.95} = [15.39, 24.96]$, S1 Table Model 8; 3SAT: $\beta_{0.5} = 9.81$, $HDI_{0.95} = [-14.59, 33.94]$, S2 Table Model 7; the effect of IC on performance for unsatisfiable instances, GLMM). This suggests that the performance variance explained by IC in 3SAT instances might be driven by satisfiability. However, we are unable to disentangle the effect of IC and satisfiability given that all 3SAT satisfiable instance have $IC = 0$. Overall, these results indicate that IC is able to explain variance in performance across instances, but to a lesser degree in 3SAT. Moreover, the effect of IC on performance in the 3SAT might be incongruously driven by satisfiability.

S6 Appendix: Supplementary Tables

Table 1: **Human performance in the Boolean satisfiability task.** Logistic regressions with random intercept effects for participants relating the accuracy on an instance and trial number (1), typical-case complexity (TCC) (2), constrainedness region (3), TCC and satisfiability (4), time-on-task (5), instance complexity (IC) (6), IC on unsatisfiable instances (7) as well as satisfiability (8). *Parameter estimates correspond to the median of the posterior distribution ($\beta_{0.5}$) and the 95% HDI credible interval ($HDI_{0.95}$). ELPD denotes the expected log posterior predictive density.*

	Dependent variable: Human performance							
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Trial number	0 [-0.01,0.01]							
TCC		-1.58 [-1.95,-1.2]		-0.77 [-1.49,-0.13]				
Overconstrained			1.39 [0.93,1.86]					
Underconstrained			1.82 [1.31,2.4]					
Satisfiability				-0.52 [-1.27,0.11]				-1.35 [-1.73,-0.99]
TCC:Satisfiability				-1.29 [-2.11,-0.46]				
Time-on-task					-0.02 [-0.02,-0.01]			
IC						30.3 [21.95,39.24]	9.81 [-14.59,33.94]	
Intercept	1.95 [1.59,2.26]	2.99 [2.59,3.43]	1.41 [1.13,1.75]	3.33 [2.73,3.97]	3.15 [2.53,3.73]	1.51 [1.23,1.8]	2.99 [1.58,4.29]	2.84 [2.43,3.22]
Observations	1398	1398	1398	1398	1335	1398	675	1398
ELPD	-533.3	-493.27	-493.42	-456.8	-487.73	-505.05	-138.46	-503.36

Table 2: **Human performance in the traveling salesperson task.** Logistic regressions with random intercept effects for participants relating the accuracy on an instance and trial number (1), typical-case complexity (TCC) (2), constrainedness region (3), TCC and satisfiability (4), time-on-task (5), satisfiability (6), instance complexity (IC) (7), as well as IC on unsatisfiable instances (8). *Parameter estimates correspond to the median of the posterior distribution ($\beta_{0.5}$) and the 95% HDI credible interval ($HDI_{0.95}$). ELPD denotes the expected log posterior predictive density.*

	Dependent variable: Human performance							
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Trial number	0 [0,0.01]							
TCC		-2.1 [-2.5,-1.73]		-2.16 [-2.74,-1.62]				
Overconstrained			2.18 [1.65,2.73]					
Underconstrained			2.05 [1.56,2.63]					
Satisfiability				-0.16 [-0.85,0.55]		-0.06 [-0.34,0.22]		
TCC:Satisfiability				0.1 [-0.65,0.9]				
Time-on-task					-0.1 [-0.13,-0.08]			
IC							21.13 [17.63,24.91]	20.1 [15.39,24.96]
Intercept	1.66 [1.43,1.94]	3.19 [2.83,3.58]	1.09 [0.91,1.3]	3.29 [2.8,3.84]	5.36 [4.39,6.34]	1.81 [1.59,2.03]	0.14 [-0.14,0.4]	0.52 [-0.21,1.29]
Observations	1575	1575	1575	1575	1575	1575	1575	787
ELPD	-656.28	-578.48	-579.52	-580.43	-612.95	-656.93	-534.35	-241.48

Table 3: **Time-on-task in the Boolean satisfiability task.** Censored linear regressions with random intercept effects for participants relating the time spent on an instance and trial number (1), typical-case complexity (TCC) (2), constrainedness region (3), satisfiability (4), number of solution witnesses (5), instance complexity (IC) (6), IC on unsatisfiable instances (7), as well as TCC and satisfiability (8). *Parameter estimates correspond to the median of the posterior distribution ($\beta_{0.5}$) and the 95% HDI credible interval ($HDI_{0.95}$). ELPD denotes the expected log posterior predictive density.*

	Dependent variable: Time-on-task							
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Trial number	-0.01 [-0.01,0]							
TCC		0.15 [0.12,0.18]						0.02 [-0.02,0.06]
Overconstrained			0.07 [0.04,0.11]					
Underconstrained			-0.35 [-0.39,-0.32]					
Satisfiability				-0.32 [-0.35,-0.29]				-0.42 [-0.46,-0.38]
No. of witnesses					-0.04 [-0.05,-0.04]			
IC						6.04 [5.41,6.7]	-0.56 [-1.91,0.78]	
TCC:Satisfiability								0.21 [0.16,0.27]
Intercept	0.68 [0.65,0.72]	0.51 [0.4,0.62]	0.65 [0.54,0.75]	0.74 [0.64,0.84]	0.59 [0.51,0.69]	0.45 [0.35,0.57]	0.77 [0.63,0.91]	0.73 [0.62,0.83]
Observations	1335	1335	1335	1335	691	1335	644	1335
ELPD	-683.08	-456.79	-268.31	-296.9	-15.46	-340.59	-128.73	-224.94

Table 4: **Time-on-task in the traveling salesperson task.** Censored linear regressions with random intercept effects for participants relating the time spent on an instance and trial number (1), typical-case complexity (TCC) (2), constrainedness region (3), satisfiability (4), number of solution witnesses (scaled via natural logarithm) (5), instance complexity (IC) (6), as well as TCC and satisfiability (7). *Parameter estimates correspond to the median of the posterior distribution ($\beta_{0.5}$) and the 95% HDI credible interval ($HDI_{0.95}$). ELPD denotes the expected log posterior predictive density.*

	Dependent variable: Time-on-task						
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Trial number	0.00 [0.00,0.00]						
TCC		0.12 [0.09,0.15]					0.12 [0.08,0.16]
Overconstrained			-0.02 [-0.06,0.01]				
Underconstrained			-0.2 [-0.23,-0.16]				
Satisfiability				-0.17 [-0.2,-0.15]			-0.17 [-0.21,-0.14]
No. of witnesses (ln)					-0.02 [-0.03,-0.02]		
IC						-0.74 [-0.9,-0.58]	
TCC:Satisfiability							0.00 [-0.05,0.06]
Intercept	0.97 [0.87,1.08]	0.85 [0.75,0.95]	0.97 [0.87,1.06]	1.00 [0.89,1.1]	0.99 [0.89,1.09]	1.00 [0.9,1.11]	0.94 [0.84,1.04]
Observations	1575	1575	1575	1575	788	1575	1575
ELPD	-515.38	-499.04	-460.55	-459.22	-189.73	-491.96	-426.44

Table 5: **Human performance an the number of solution witnesses.** Logistic regressions with random intercept effects for participants with accuracy as dependent variable. The data included on each regression is comprised of the satisfiable instances of one of the three tasks considered: 3SAT (1-3), TSP (4-6) and KP (7-9). Regressions (1), (4) and (7) include the the number of witnesses alone as regressor (the number of witnesses for the TSP is scaled via natural logarithm). Models (2), (5) and (8) include TCC, additionally, as regressor. Models (3), (5) and (8) include the interaction between TCC and number of witnesses as well. *Parameter estimates correspond to the median of the posterior distribution ($\beta_{0.5}$) and the 95% HDI credible interval ($HDI_{0.95}$). ELPD denotes the expected log posterior predictive density.*

	Dependent variable: Human performance								
	3SAT			TSP			KP		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
No. of witnesses	0.62 [0.49,0.79]	0.7 [0.51,0.91]	0.63 [0.44,0.85]				0.26 [0.19,0.34]	0.29 [0.17,0.41]	0.11 [-0.03,0.25]
TCC		0.47 [-0.3,1.22]	-0.42 [-1.71,0.7]		-0.12 [-0.84,0.68]	2.24 [-1.5,6.13]		0.17 [-0.57,0.9]	-1.77 [-2.95,-0.44]
TCC:No. of witnesses			0.56 [0,1.22]						0.54 [0.26,0.81]
No. of witnesses (ln)				0.45 [0.37,0.53]	0.44 [0.33,0.54]	0.68 [0.25,1.03]			
TCC:No. of witnesses (ln)						-0.26 [-0.66,0.15]			
Intercept	-0.02 [-0.62,0.53]	-0.52 [-1.58,0.47]	-0.21 [-1.19,0.93]	-1.07 [-1.78,-0.46]	-0.92 [-2.16,0.21]	-3.2 [-6.83,0.73]	0.41 [-0.04,0.83]	0.22 [-0.7,1.22]	1.55 [0.35,2.78]
Observations	723	723	723	788	788	788	716	716	716
ELPD	-258.83	-259.54	-258.59	-244.8	-245.4	-245.76	-303.21	-303.92	-297.45

Table 6: **Human performance in the knapsack task.** Logistic regressions with random intercept effects for participants relating the accuracy on an instance and satisfiability (1), instance complexity (IC) (2), and IC on only unsatisfiable instances. *Parameter estimates correspond to the median of the posterior distribution ($\beta_{0.5}$) and the 95% HDI credible interval ($HDI_{0.95}$). ELPD denotes the expected log posterior predictive density.*

	Dependent variable: Human performance		
	(1)	(2)	(3)
Satisfiability	-0.29		
	[-0.57,0.01]		
IC		9.05	13.48
		[7.2,11.02]	[10.11,17.3]
Intercept	1.79	0.59	0.47
	[1.51,2.1]	[0.28,0.92]	[0.08,0.92]
Observations	1427	1427	711
ELPD	-637.57	-574.82	-253.01

Table 7: **Number of clicks in the Boolean satisfiability task.** Censored linear regressions with random intercept effects for participants relating the number of clicks performed on an instance and typical-case complexity (TCC) (1), satisfiability (2), TCC and satisfiability (3), instance complexity (IC) on unsatisfiable instances (4), as well as the number of solution witnesses on satisfiable instances (5). *Parameter estimates correspond to the median of the posterior distribution ($\beta_{0.5}$) and the 95% HDI credible interval ($HDI_{0.95}$). ELPD denotes the expected log posterior predictive density.*

	Dependent variable: Number of clicks				
	(1)	(2)	(3)	(4)	(5)
TCC	1.88 [1.23,2.54]		0.06 [-0.66,0.84]		
Satisfiability		-7.41 [-7.95,-6.88]	-8.8 [-9.51,-8.05]		
TCC:Satisfiability			2.91 [1.81,3.88]		
IC				-29.88 [-54.52,-6.63]	
No. of witnesses					-0.59 [-0.69,-0.49]
Intercept	10.43 [9.19,11.67]	15.15 [13.95,16.39]	15.11 [13.76,16.4]	16.4 [14.19,18.82]	10.09 [9.23,10.96]
Observations	1375	1375	1375	664	711
ELPD	-4111.67	-3825.19	-3794.12	-1645.16	-2016.33

Table 8: **Number of clicks in the traveling salesperson task.** Linear regressions with random intercept effects for participants relating the number of clicks performed on an instance and typical-case complexity (TCC) (1), satisfiability (2), TCC and satisfiability (3), instance complexity (IC) (4), the number of solution witnesses (transformed via natural logarithm) on satisfiable instances (5), as well as IC and satisfiability (6). *Parameter estimates correspond to the median of the posterior distribution ($\beta_{0.5}$) and the 95% HDI credible interval ($HDI_{0.95}$). ELPD denotes the expected log posterior predictive density.*

	Dependent variable: Number of clicks					
	(1)	(2)	(3)	(4)	(5)	(6)
TCC	1.66 [1.01,2.33]		0.89 [0.05,1.83]			
Satisfiability		-2.48 [-3.13,-1.85]	-3.23 [-4.12,-2.3]			-1.77 [-2.81,-0.65]
TCC:Satisfiability			1.53 [0.2,2.77]			
IC				-10.22 [-13.87,-6.37]		-6.71 [-12.41,-0.87]
TCC:No. of witnesses (ln)					-0.41 [-0.54,-0.26]	
IC:Satisfiability						-5.95 [-13.43,1.78]
Intercept	24.21 [21.99,26.32]	26.38 [24.09,28.56]	26 [23.61,28.28]	26.29 [24.04,28.61]	27.31 [25.52,29.17]	27.2 [24.87,29.46]
Observations	1575	1575	1575	1575	788	1575
ELPD	-5236.1	-5220.82	-5207.65	-5234.03	-2544.04	-5207.35