CellPress
REVIEWS

Opinion

Computational Complexity and Human Decision-Making

Peter Bossaerts[1,2,*] and Carsten Murawski[1]

The rationality principle postulates that decision-makers always choose the best action available to them. It underlies most modern theories of decision-making. The principle does not take into account the difficulty of finding the best option. Here, we propose that computational complexity theory (CCT) provides a framework for defining and quantifying the difficulty of decisions. We review evidence showing that human decision-making is affected by computational complexity. Building on this evidence, we argue that most models of decision-making, and metacognition, are intractable from a computational perspective. To be plausible, future theories of decision-making will need to take into account both the resources required for implementing the computations implied by the theory, and the resource constraints imposed on the decision-maker by biology.

Trends

New research showing that the quality of human decision-making decreases with the computational complexity of decision problems challenges the core assumption of most models of decision-making: that decision-makers always optimise.

CCT can help explain behavioural biases, such as choice overload and negative elasticity of labour supply.

Integrating CCT with decision theory and neurobiology promises to lay the foundations of a more realistic theory of decision-making and metacognition.

The Rationality Principle

Most modern theories of decision-making, including rational choice theory [1–4], game theory [5], prospect theory [6], as well as many learning models [7], are based on the **rationality principle** (see Glossary): decision-makers are assumed to always choose the best action available to them [8]. Even theories of bounded rationality [9–14] assume that decision-makers optimise, albeit within constraints. These theories do not take into account the difficulty of identifying the best action. A decision situation in which the decision-maker must choose from two available options is not distinguished from one with 2^{100} options. Some approaches have proposed that, when faced with difficult decisions, humans use **heuristics** to make a choice [10,15,16]. However, these approaches treat difficulty only informally. More recent theories, such as bounded optimality [17,18], resource-rational analysis [19], and computational rationality [20,21], address the issue of decision difficulty by considering the computational costs of decisions. These theories also commit to the rationality principle (optimality), assuming that decision-makers optimise at the level of computation, or at least approximate optimality. However, many key issues remain unaddressed. Specifically, it is an open question which dimensions of decision difficulty (computational costs) are relevant to human decision-making, how to quantify these dimensions, or how the brain allocates resources to the decision-making process.

In this article, we suggest that **CCT** [22–24] provides a theoretical framework to render precise what it means for a decision to be difficult. Applying concepts from CCT to decision theory, we show that many existing models of decision-making are implausible; the computations required to implement those models are intractable in the sense that they would require computational resources beyond those available to decision-makers. We propose that future theories of decision-making will need to take into account both the resources required for implementing the computations implied by a theory, and the resource constraints imposed on the decision-maker by biology. Such theories will not only be necessary to develop a more plausible account of human decision-making, but also enable more powerful artificial intelligence.

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

*Correspondence:
peter.bossaerts@unimelb.edu.au
(P. Bossaerts).

Are Current Theories of Decision-Making Feasible?

Consider the example of grocery shopping, a decision task that many people encounter regularly in their every-day life. Theories of choice based on the rationality principle would assume that, from among all goods available in the supermarket, the decision-maker chooses the basket of goods with the highest total utility to them subject to a budget constraint. Let us assume that the decision-maker goes shopping in a supermarket that only stocks 100 different goods. The number of combinations of goods that the decision-maker can form from those 100 goods is approximately 10^{30}. To choose the combination with the highest total utility, the decision-maker would have to check the budget constraint for all of those combinations and find the combination with the highest utility from within those sets within the budget constraint. This computation is intractable even for the world's fastest supercomputer: it would quickly run out of memory (and, if it did have enough memory, the computation would take millions of years). If the supermarket stocked 1000 different goods, the number of possible combinations would be approximately 10^{301}, approximately 10^{220} times more than the estimated number of atoms in the Universe. The number of different goods stocked by an average US supermarket is approximately 40 000, which means that the number of choice sets available to the decision-maker is effectively infinite.

This thought experiment demonstrates in an informal way that current theories of decision-making based on the rationality principle, or optimisation, are infeasible for most decision situations. One may object that this is not the way people shop, but that only addresses the criticism that the theory does not explain what people literally do. Our objection is deeper: people cannot possibly optimise in this example: identifying the best option in most situations is not feasible even in principle. It is not sufficient to argue that people enter a grocery store with a predetermined shopping list: for this list to be consistent with the rationality principle, an intractable optimisation problem would have to be solved in the first place. Of course, how to best formalize feasibility remains an open question. In the next section, we suggest that CCT can provide the foundations for a quantitative framework to make the notion of decision difficulty and feasibility more precise [22–24].

Computational Complexity

CCT is concerned with **computable problems** [25,26]. A problem is considered computable if it can be solved in principle by a computing device, that is, a device that takes as input any precise mathematical statement and, after executing a finite number of steps (**algorithm**), decides whether the statement is true or false. CCT is concerned with the computational resources required to solve computational problems. The **computational complexity** of a problem (that is, an input–output mapping) is defined in terms of the growth of computational resources as a function of the size of the input (number and length of variables) to any algorithm computing a solution [24]. Knowing how computational resource requirements increase provides insights into the inherent complexity of the problem. The resources most commonly studied are time (number of computational operations), referred to as **time complexity**, and space (memory), referred to as **space complexity**. Here, we are primarily concerned with the former.

Four classes of problem relating to time complexity have been of particular interest to computer scientists. The first class is referred to as **P** and contains all (decision) problems that can be solved in an amount of time that grows as a polynomial of the input size, using a deterministic, sequential computer, such as a Turing machine [24]. This means that, for these problems, there exist algorithms whose running time can be upper-bounded by any polynomial function of its input size. Such algorithms are called **efficient**. In practical terms, this means that these problems can typically be solved within a reasonable amount of time. Examples of class P include sorting of arrays, and basic mathematical operations, such as multiplication and (non-integer) linear programming.

Glossary

3-SAT problem: similar to the satisfiability problem but with the number of literals in each clause limited to three at the most.
Algorithm: a well-defined computational procedure (sequence of computational steps) that takes a set of values as input and produces some set of values as output.
Approximation algorithm: an inexact algorithm with an approximation guarantee.
Complexity class: a set of computational problems with similar computational complexity (e.g., P, NP, or NP-complete).
Computable problem: a problem is considered computable if it can be solved in principle by a computing device.
Computational complexity: the (worst-case) growth of computational resources (e.g., time or memory) required for solving a computational problem as a function of the size of the input of the problem.
Computational complexity theory: mathematical framework for classifying computational problems according to their inherent difficulty.
Efficient: an algorithm is called efficient if its running time can be upper-bounded by any polynomial function of its input size.
Heuristic: an inexact algorithm that does not have an approximation guarantee.
Instance complexity: the computational resources required for solving particular instances of a computational problem.
Knapsack problem: a problem to find from a set of items with given values and weights, the subset that maximises total value subject to a total weight constraint.
NP: the class of all (decision) problems for which a given solution can be verified in polynomial time (but for which no polynomial-time algorithm is known to find the solution).
NP-complete: the class of all (decision) problems within NP that are at least as hard as all other problems in NP.
NP-hard: the class of all problems that are at least as hard as the hardest problems in NP.
P: the class of all (decision) problems that can be solved in an amount of time that grows as a polynomial of the input size.

The second class, **NP**, comprises all those problems for which a given solution can be verified in polynomial time but for which no polynomial-time algorithm is known to find the solution. For this class, no efficient algorithms are known [27]. In practical terms, this means that, while a given solution can be verified quickly, finding this solution might be intractable. Most practical computational problems belong to this class [28]. Problems in class P are often referred to as **tractable** problems, while problems in class NP are called intractable, which means that the computational resources required (e.g., time) to solve these problems are often beyond those available [28]. The class containing the hardest problems in NP is called **NP-complete** [27,29,30]. Examples of this class include the **knapsack problem**, the travelling salesman problem, and the satisfiability problem. There are thousands of other problems of practical relevance that have been shown to be NP-complete [28].

Finally, the class **NP-hard** contains all problems that are at least as hard as the hardest problems in class NP but that are not necessarily contained in class NP [31]. That is, many problems in this class are harder than the hardest problems in NP. For these problems, the time required to solve instances of the problem often increases exponentially in instance size. The shopping problem in the thought experiment above is NP-hard [technically, it is an instance of the 0–1 knapsack (optimisation) problem [32]]. Many models of cognition are also based on computational problems that are NP-hard [33].

To illustrate the distinction between polynomial (which applies to problems in class P) and exponential growth of resources (which applies to some problems in class NP-hard), we consider the following example. Suppose we have two problems whose solutions are deterministic functions of a set of numbers (input). For one of those problems, there exists an algorithm whose resources (e.g., compute time) grow no worse than a quadratic function of the size of the input. If we increased the size of input tenfold from two to 20, the resource requirement would grow from 2^2 to 20^2, that is, a hundredfold increase (10^2). For the other problem, suppose the resources of the best-known algorithm grow as an exponential function of the size of input. In this case, an increase in the size of the set from two to 20 would increase resource requirements from 2^2 to 2^{20}, a factor of 262,144 (10^6). This example makes apparent that, although the distinction between polynomial and exponential growth of resources might appear artificial for small inputs, the difference quickly becomes material.

Empirically, the quantitative differences in computational resources between the classes P and NP are so vast that they can also be considered qualitative differences. It is widely agreed that there are deep structural differences between problems of those classes. Yet, at this stage, it is still an open question whether the classes P and NP are indeed different or the same (Box 1). If the latter turned out to be the case, then we could expect to find efficient algorithms for NP problems, including for tasks such as breaking modern encryption algorithms. However, most computer scientists believe that P does not equal NP; that is, they believe that there are qualitative differences between those classes of problem, which are associated with significant quantitative differences in computational resource requirements [34,35].

Although complexity classes are defined in terms of asymptotic worst-case behaviour [28], there are enormous differences in computational requirements between instances of the same problem. For example, sorting an array of one million integers that are completely out of order might take substantially longer than sorting an array that is already in the desired order. More recently, therefore, interest has focused on understanding the complexity of individual instances of problems, particularly, how **instance complexity** is related to particular properties of instances (Box 1). It has been shown for some computational problems that instance complexity is related to a small number (two–three) of observable instance properties. In the **3-SAT problem**, for example, instance complexity has been associated with the ratio of clauses

Rationality principle: the principle that decision-makers are assumed to always choose the best action available to them.
Sahni-k: a measure of instance complexity for the 0–1 knapsack problem (optimisation version). Sahni-k measures the number of items (k) that already have to be put in the solution before the greedy algorithm can be used on the remaining items to obtain the optimum. It is proportional to computational time and memory required to compute the solution of an instance.
Space complexity: the (worst-case) growth of memory required for solving a computational problem as a function of the size of the input of the problem.
Time complexity: the (worst-case) growth of time required for solving a computational problem as a function of the size of the input of the problem.
Tractable: a computational problem is called tractable if it can be solved by an efficient algorithm.

Box 1. Does P Equal NP?

It has been shown that the quantitative gaps between computational resource requirements of problems in classes P and NP are so vast that they can also be considered qualitative gaps [74]. These gaps are exploited in many areas of everyday life. While multiplication of two integers is in class P, the reverse operation (factoring an integer into primes) is in class NP. This gap between the resource requirements of multiplication and factoring is the basis for most modern cryptography. While it is the case that P is a subset of NP, by definition, it is still an open question whether P is a proper subset of NP (i.e., is strict or not), which is referred to as the 'Does P equal NP?' problem [27]. While most computer scientists believe that the inclusion is strict, that is, $P \neq NP$, nobody has been able to prove it [34,35]. If P does not equal NP, then there is no hope that we will ever be able to find efficient algorithms for NP problems.

Recently, several complexity classes, including NP, have been characterised independently of the notions of computing, algorithm or Turing machine, purely in terms of the type of logic needed to describe them. This new branch of CCT is referred to as 'descriptive complexity' and draws heavily on finite model theory [75–79]. Its results provide additional support to the conjecture that there are inherent structural differences between the different complexity classes.

One criticism often mounted against complexity classes P and NP is that they are defined in terms of how computational resources grow as a function of input size in the worst case. A configuration of a problem with the same input size can have many different values of inputs, referred to as an instance of the problem. The computational resource requirements can vary dramatically across instances, even if they all have the same input size. For example, sorting an array of n integers that is already sorted will typically require fewer resources than sorting an array of the same length but that has maximum entropy (i.e., is perfectly random). Thus, computational resource requirements for particular instances of NP problems can be lower than those for particular instances of P problems. A branch of computational complexity theory examines how resource requirements behave as a function of basic properties of instances. The latter is referred to as 'instance complexity' [36,78,79]. It provides a finer-grained measure of complexity than do complexity classes and might be more useful for the study of human decision-making than are complexity classes.

(statements) to variables (blanks in the statements) of an instance [36]. The study of instance complexity may prove more important for the understanding of human behaviour than the study of complexity classes, although the two are intricately related (Box 1). To summarise, CCT shows that computational problems differ substantially in their computational resource requirements and provides a useful framework to classify these problems according to their resource requirements.

Computational Complexity and Decision Theory

What are the implications of CCT for the decision sciences? Most modern theories of decision-making, implicitly or explicitly, take a computational approach [33,37]. They represent behaviour as the outcome of computational problems. Rational choice theory, for example, interprets behaviour as the outcome of an optimisation problem (maximisation relative to a set of preferences or utilities) [1,4,8]. However, these theories do not explain how decision-makers implement the optimisation problem or, indeed, whether finding the optimal solution is feasible even in principle [33]. As our thought experiment above shows, optimisation is often computationally intractable in the sense that the computational resources required to compute the optimum by far exceed the resources available to decision-makers [38]. This means that expected utility maximisation may be intractable from a computational perspective in some, and possibly many, cases [39]. The same is true for Bayesian belief updating, which is NP-hard in many cases [40]. Satisficing, where decision-makers do not compute the highest-utility solution but keep looking for an option until they have found one that achieves at least a given utility level, was proposed as a heuristic in response to cognitive limitations of decision-makers [41]. It can be regarded as a sequence of instances of the knapsack (decision) problem, which is NP-complete. Indeed, many models of human decision-making require computational resources, as identified by CCT, that are well beyond those available to decision-makers, thus rendering these models implausible from a computational point of view [33]. The same has been shown for models at lower levels of cognition, particularly visual perception [42,43].

Several challenges might be raised in response to this claim. First, many decision scientists take the stance that models of decision-making describe observed behaviour but have no ambition to explain it [44,45]. In this view, the rationality principle, on which many of these models are based, is an assumption through which behaviour is interpreted and not a mechanistic description of the decision-making process. Decision-makers are merely assumed to behave 'as if' they solved an optimisation problem [45] (Box 2).

This includes many psychological models of choice. Prospect theory is a prominent example [6]. It summarises, through maximisation of a single-dimensional value function, several properties of choice under uncertainty including the tendency to frame choices in terms of a manipulable reference point, the creation of the perception of gains (outcomes above the reference point) and losses (outcomes below the reference point), and the overweighing of small probabilities of large losses. The theory expresses features of observed human choices under uncertainty in terms of optimisation of a value function that exhibits all these features. Prospect theory is also an 'as if' model: it does not claim that decision-makers literally compute the optimisation function it postulates.

Such models can have a valuable role in characterising the computations involved in decision-making, often referred to as 'computational models' in the tradition of Marr [46]. However, while 'as if' models do not make any explicit assumptions about the computations underlying observed behaviour, and indeed want to be agnostic about them, these models have implicit computational requirements. If a decision-maker's choice is interpreted as the best option available, according to the model, then the decision-maker must have been able to identify this option as the best option, at least in principle (Box 2). This means that the decision-maker must have had the computational resources available to find the best option. However, many 'as if' models are NP-hard [38–40]. By assuming that decision-makers behaved 'as if' they solved an NP-hard problem every time they make a choice, the proponents of these models effectively assume that decision-makers can efficiently solve these problems. This is equivalent to assuming that P equals NP, at least for those decision-makers, which, based on the current state of knowledge, is not plausible [35] (Box 3). Thus, such models should always be

Box 2. 'As If' Models

Many models of human decision-making, including almost all economic models, are defined at the level of behaviour. They map observable properties of a decision situation into choices between available decision options. The restriction of those models to observable behaviour has been justified by the fact that the mechanism by which a decision-maker arrives at the action, such as the underlying mental or computational process, is typically not observable [80].

The most representative example of this class of models is revealed preference theory [1,2,81] and its close cousin utility theory (including prospect theory) [6]. Both characterise actions (choices) purely in terms of latent preferences or utilities. In this theory, preferences and utilities are 'observationally equivalent' with choices. Importantly, no causation of choices by utility or preferences is assumed or implied by the theory. To infer preferences or utilities from observed behaviour, the latter is assumed to be the outcome of an optimisation process. Specifically, observed choices are assumed to represent the highest-utility option available to the decision-maker in a given decision frame. Optimisation follows from the fundamental assumptions (axioms) of the theory, such as that choices satisfy transitivity and completeness. The assumptions are justified either with a psychological or an economic argument. In the psychological argument, decision-makers are assumed to be intentional agents whose aim it is to achieve the best outcome available to them [52]. In the economic argument, the assumptions are justified with the claim that in a competitive economy, agents who violate them would not survive [80].

What is missing is a justification of these assumptions from the perspective of computational complexity theory. Here, the question is whether it is plausible to assume that decision-makers will always be able to identify the most preferred or highest-utility option. At present, the theories assume that decision-makers can somehow do this, without specifying how they do it. This ignores the computational resources required for the task. A realistic theory of decision-making needs to explain under which conditions a decision-maker can be expected to behave in a way consistent with the assumptions and, therefore, needs to take into account the computational complexity of the decision task.

Box 3. The Church–Turing Thesis

The use of concepts from computer science in the description of the human brain and behaviour is justified by the Church–Turing thesis (CTT) [25,26,82]. The latter is concerned with the notion of 'effective' or mechanical procedures (algorithms) in logic and mathematics. Intuitively, the thesis states that a computational task is (Turing) computable if it is possible to specify a sequence of basic mechanical instructions that will result in the completion of the task once the instructions have been carried out by a (Turing) machine, a general model of computation. It implies that a Turing machine can compute any Turing computable task, that is, any (Turing) computable task can be simulated by a Turing machine. An extended version of CTT (complexity-theoretic CTT) conjectures that any effectively computable mathematical function that is 'hard' to compute for a Turing machine (in the sense of requiring a large amount of computational resources), is also hard for any other model of computation [83]. The CTT is a deep and perhaps unsolvable fundamental question. Today, it is widely believed to hold, whereas the extended version of it is contentious [84]. However, it should be pointed out that it is possible that certain human tasks, such as creativity, are incomparable with the type of mathematical functions to which CTT applies (Turing-computable functions) [74]. Similarly, it is conceivable that there are certain tasks that the brain can compute faster than a Turing machine, which would be a violation of the extended CTT [74]. By contrast, it has also been suggested that CTT is too liberal to be of practical use for the explanation of human cognition [33].

supplemented by models that take computation into account explicitly, which are often referred to as 'algorithmic models' [46].

Another challenge could be levelled based on studies of the neural signals associated with decision-making. Many studies have documented the existence of so-called 'value' (or utility) signals. These studies have been interpreted as prima facie evidence that the brain does compute the variables required for optimisation models of choice, which in turn is interpreted as evidence that the brain literally performs optimisation [47,48]. This inference is not valid, however, at least not universally. Almost all studies of value signals used decision situations with a small number of simple decision options (typically two), with each option having only a few decision-relevant properties (typically one or two). The computational requirements to choose the best option from two available options might well be within the computational resources of the brain. However, few naturalistic decision situations feature only two options with two decision-relevant properties, such as value and risk. It is not clear whether these same signals, and the underlying neural processes, would also occur in computationally more complex decision scenarios. It is conceivable that different computational processes are used in more complex scenarios. Indeed, there is evidence that human decision-makers switch computational processes when the computational complexity of the decision scenario increases (e.g., through an increase in the number of states to consider) [49].

Another, possibly more likely, explanation of the discovery of value signals in the brain is that most of these signals are in fact neural correlates of other variables related to the decision process, such as attention, salience, or subthreshold premotor activation. These variables are highly correlated with value signals [37]. In fact, a recent neuroimaging study of a decision task demonstrated that neural signals in 30% of the brain appeared to be value signals [50]. Some of these signals were abstract in the sense that they were not correlated with particular stimulus properties or motor response, and were unlikely to be indirect consequences of reward, such as increased arousal or attention [50]. This could be interpreted as evidence that the brain implements something like 'parallel computing', but does not imply that expected utility maximisation is computationally tractable for the brain in general.

A third challenge is deeper and concerns the relevance of CCT for human cognition and behaviour. CCT is based on a mathematical model of computation. It is an open question: (i) whether this model applies to the human brain; and (ii) whether it is useful for our understanding of human behaviour [33,51]. These questions have occupied computer scientists, philosophers, psychologists, and neuroscientists since the initial work of Turing [26] and Church [25] during the 1930s (Box 3) [51–54]. They remain unresolved empirical questions [51], to which we now turn.

Does Computational Complexity Theory Apply to Human Decision-Making?

The effects of computational complexity on human decision-making can be tested empirically. In the following, we report results from several studies that investigated the relation between computational complexity and human behaviour in a canonical computational problem, the 0–1 knapsack problem (KP). In this problem, the decision-maker is given a set of items with different weights and values and has to find the subset of items with the highest total value, subject to a weight constraint (capacity). The problem is a constrained optimisation problem and is similar to many models of human decision-making [32,38,55]. In the studies, participants were asked to solve several different instances. These instances differed in the properties of the items (values and weights) as well as the weight constraint. While the instances contained a similar number of items, they varied in their computational complexity, that is, the amount of time and memory required by computer algorithms to solve them. The KP is NP-hard: while it is easy, from a computational perspective, to verify that a given subset of items has a given total value and total weight, it is hard to determine whether this subset is the optimum.

The studies found that participants' ability to find the optimal subset of items decreased rapidly as the computational complexity of instances increased [38,55]. Behaviour was particularly well described by one measure of difficulty, **Sahni-k** (Figure 1A). This metric is proportional to the amount of time and the amount of memory a computer algorithm requires to solve an instance. The finding shows that CCT, based on an abstract model of computation, also applies to humans. The fact that both humans and electronic computers are sensitive to the same metric of instance complexity corroborates the conjecture that the theory of computation may be universal (Box 3). At a minimum, it means that constraints identified by CCT also apply to biological organisms, such as humans.

In these studies, humans were rewarded according to their performance, which was measured by the proximity of their solution to the optimal solution. It was found that humans spent more effort and, in consequence, earned more money on average on more difficult instances (Figure 1B). This behaviour is hard to reconcile with models of optimisation: economic theory would predict that participants would spend less effort on hard instances and, hence, earn less. The observed increase in effort with difficulty may be consistent with satisficing if humans set a fixed value target to be attained irrespective of the effort it required [41]. For more difficult KP instances, more effort is required to reach the target.

Interestingly, novice taxi drivers in New York City evidently exhibit the same counter-intuitive effort–difficulty relationship, a paradox in economics. Taxi drivers work longer on days when it is harder to spot potential passengers, despite never reaching the payoffs achieved on days when the marginal return to effort is much higher [56]. Psychological reasons have been given for this phenomenon, such as trying to meet performance targets [57]. However, we would suggest that these taxi drivers are merely trying to solve an NP-hard computational problem (a variant of the travelling salesman problem: they need to find the shortest route from one potential passenger to another). In their solution attempts, taxi drivers spend more effort on days when the problem is tougher. One would expect the effect to disappear as taxi drivers learn to navigate their surroundings better and, indeed, experienced taxi drivers work less on days with fewer potential passengers [58]. Unfortunately, the data of this study do not allow a clean analysis of the effect of computational complexity on effort because confounding factors might have been at work (e.g., slower days might have been associated with lower driver opportunity costs). By contrast, in other studies of the KP, researchers used controlled experimentation, which allowed the effect of computational complexity to be isolated.

There is one distinctly puzzling aspect of this finding: by choosing to spend more effort on instances with higher Sahni-k (higher difficulty), the participants revealed that they sensed

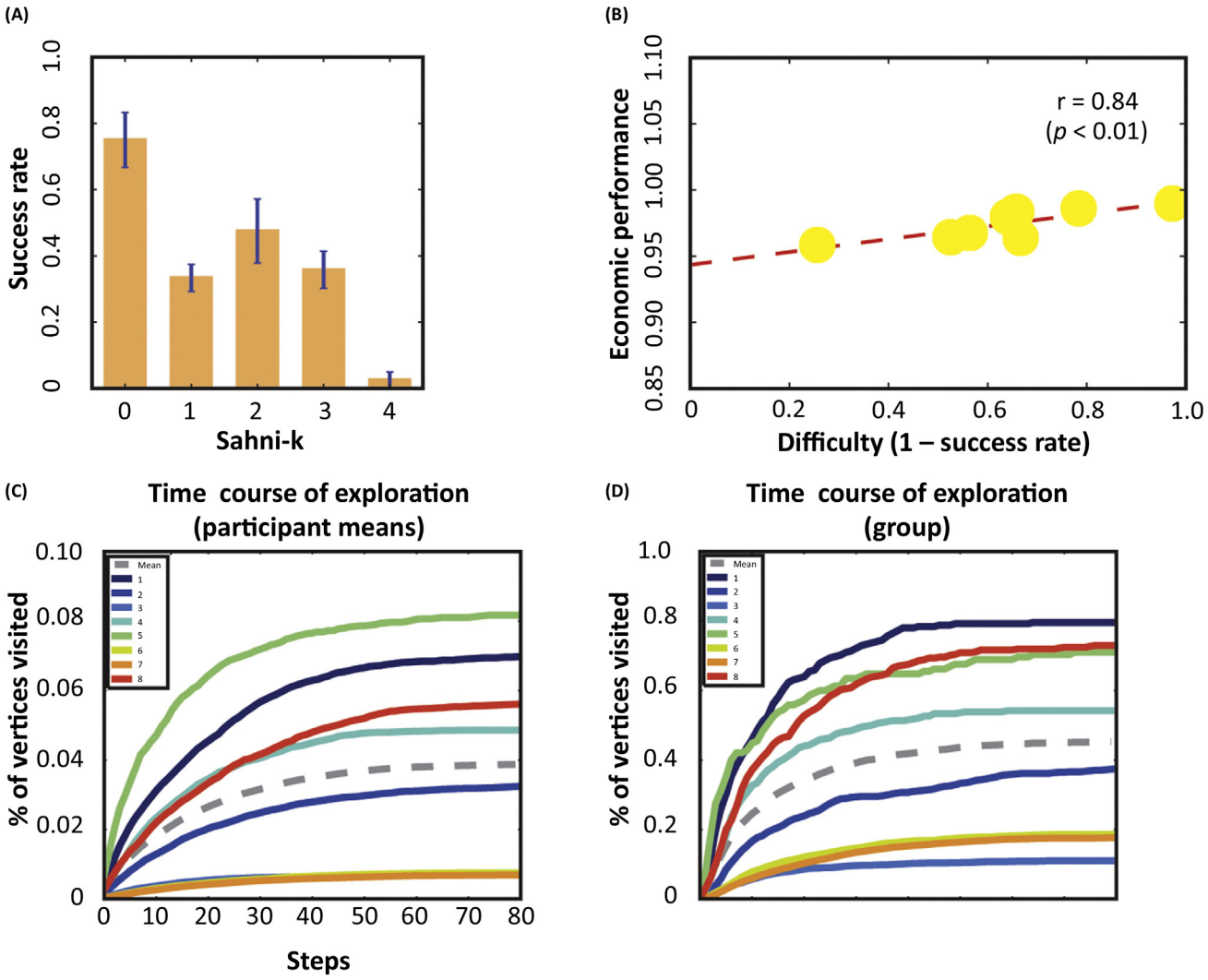

Figure 1. The Relation between Instance Complexity and Human Behaviour. Several studies have investigated the relation between instance complexity and human behaviour in the 0–1 knapsack problem, a canonical computational problem closely related to many models of human decision-making. In the task, participants need to find, from a set of items with different values and weights, the subset of items with the highest total value, given a total weight limit. The instances that the participants were asked to solve varied in the degree of computational complexity, that is, in the amounts of computational steps and memory that algorithms require to solve them. (A) Participants' ability to find the solution of an instance was negatively related with Sahni-k, a measure of instance complexity proportional with compute time and memory required to solve the instance. (B) Participants earned more money on average on instances that were more difficult. (C) Mean proportion of search space (considering only full knapsacks) explored by individual participants in different instances. (D) Proportion of search space explored by all participants across different instances. Reproduced from [38].

whether a particular instance was more difficult. This is paradoxical because all known computer algorithms need to solve an instance to compute the complexity of an instance. By contrast, many of the participants, never solved the instances and, if they did, they were generally not able to tell that they did (this may reflect the very nature of NP-hard problems) [38]. It is an open question how participants were able to detect instance complexity and subsequently adjust effort. This may be an example of a computational task that the human brain can do but for which no computer algorithm is currently known.

Participants exhibited a striking diversity in how they approached solving the KP instances (Figure 1C,D). They appeared to use different search strategies (algorithms) for different

instances of the same problem. Moreover, there was little overlap in participants' search paths within instances (the sequence of candidate solutions considered) [38]. The latter would make information sharing beneficial. Therefore, in another study, participants were incentivised to share information during the search with other participants through a market mechanism [55]. It was found that, in this setting, more participants found the optimal solution compared with a setting in which they had to work in isolation. This happened despite the fact that market prices never provided enough information to infer the optimal solution (but prices in combination with trading volume signalled valuable information to participants).

Market efficiency, a core concept in financial economics, states that prices will reflect all the available information [59]. In the experiments described above, there were always participants who were able to compute the optimal solution, so this information was available. Therefore according to the principle of market efficiency, prices should have reflected the solution. However, they did not, casting doubt on the principle [55]. This suggests that the computational complexity affects decision-making at the level of not only the individual decision-maker, but also the market. Among others, this is highly relevant for the use of prediction markets as problem-solving tools: these markets may only work for problems of low computational complexity.

Human Computational Resources and Metadecision-Making

Assuming that human computational resources are limited, the capacity to make a decision depends on both the resources required to make the decision and the resources available to the decision-maker (Figure 2, Key Figure). Computational resource requirements can be identified by CCT. However, it is equally important to understand resource availability.

Cognitive resources have not only intrinsic costs (e.g., in the form of metabolic cost of firing spikes), but also opportunity costs [7,60,61]. The latter arise because most cognitive resources are shared across domains and processes, both lower-level functions, such as the visual system, and high-level, executive function, including working memory, attention, and the central executive [62,63]. All of these processes are crucially involved in decision-making. Given that many cognitive processes depend on these resources, their capacity and availability are heavily dependent on context [61]. For example, it has been shown that acute stress reduces working memory capacity, in turn weakening deliberative, model-based choice [64]. Different tasks involving executive function cross-influence each other and cross-predict performance [65–67].

It is an open question how limited cognitive resources are allocated to tasks, a problem referred to as 'cognitive control' or 'metadecision-making' [7,61,68]. Most models of cognitive control are framed in terms of optimisation: decision-makers optimally allocate resources to tasks, trading off (expected) rewards and costs [69–72]. Here, too, the issue of computational plausibility arises: many of those models are based on computational problems of high computational complexity. Therefore, these models may also explicitly or implicitly require enormous computational resources, even in cases of relatively low-level tasks. For example, optimal allocation of attention is an NP-hard problem [38]. Thus, decision-makers face high computational complexity at the level of not only single decision tasks, but also metacognition.

Cognitive control presents another problem. Existing models assume that cognitive resources are allocated based on known (expected) rewards of tasks and costs of resources [7,61,68]. However, the computations of either (expected) rewards to be gained from engaging in a particular task or (expected) costs incurred might be NP-hard computational problems themselves. For example, in the knapsack problem discussed above, all known computer algorithms to compute the computational resources to solve a particular instance of the problem need to solve the instance to do so, which is an NP-hard problem. The same is true for

Key Figure

Computational Complexity Theory and Human Decision-Making

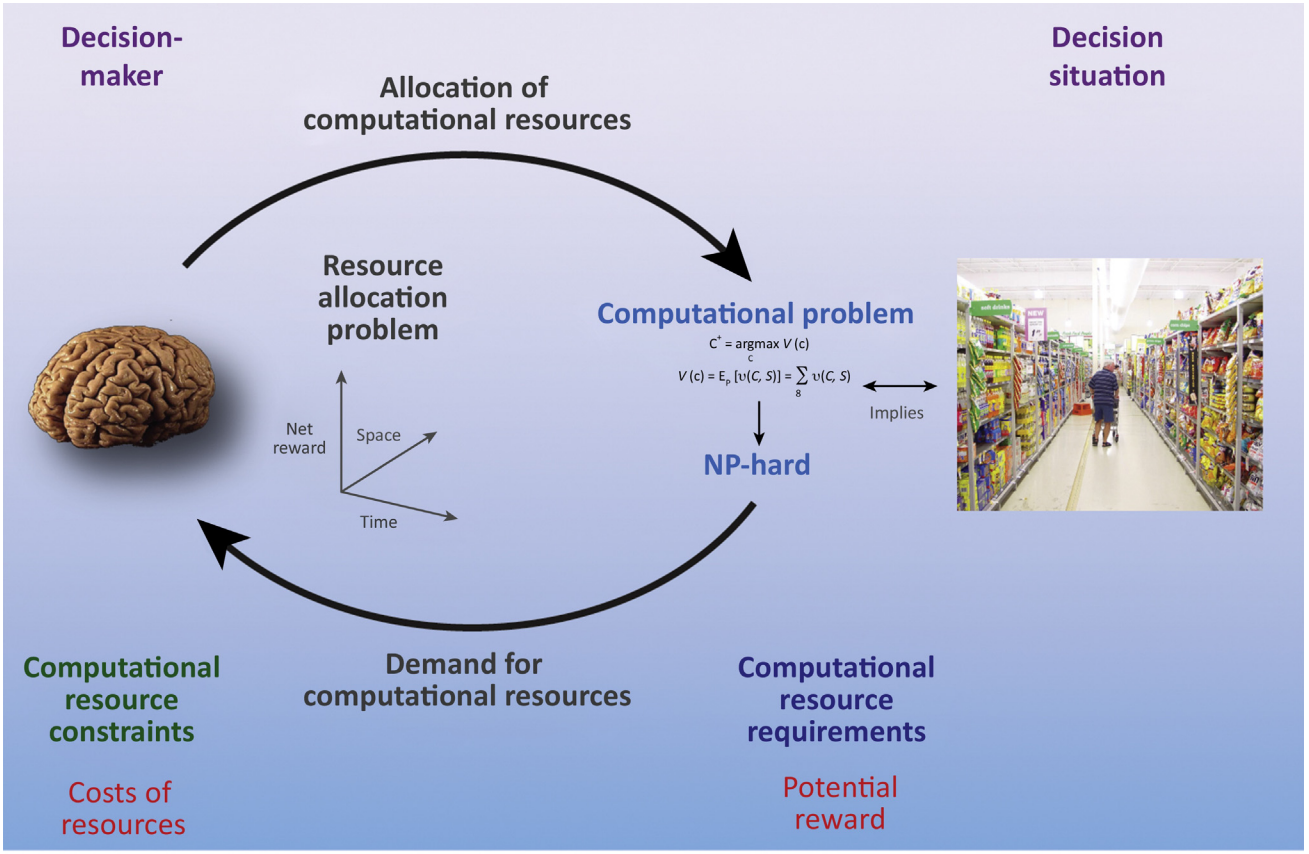

Figure 2. The decision sciences typically represent decision situations as computational problems (e.g., utility maximisation). Solving these computational problems requires computational resources, which can be quantified using the framework of computational complexity theory. Thus, computational models can be categorised according to complexity classes. Many optimisation models belong to the complexity class NP-hard, which means that computing the solutions of these problems may be intractable. When approaching a decision situation, the decision-maker needs to allocate (limited) computational resources (e.g., time or memory) to the decision task to solve the computational problem. The set of computational problems that humans can solve at any time is constrained by the amount of computational resources available. The problem of allocating resources to problems may itself be a difficult computational problem. It is an open question how organisms detect the computational requirements (complexity) of a decision task and how they allocate resources to formulating a decision (cognitive control and metadecision-making).

computing the maximum reward available in an instance. Yet, in the study described above, it was found that participants appeared to be able to adjust not only effort spent, but also their search strategies, based on instance complexity [38]. However, we do not know how participants achieved this. In particular, we do not know whether participants knew instance complexity before they started solving an instance and whether this determined their effort level at that point (which is what would be required by existing cognitive control models), or whether they adjusted effort dynamically while they were trying to solve the instance.

In addition, cognitive control itself requires energy and imposes a cost [68]. Thus, cognitive control could easily end in an infinite regress [61]. Instead of one-off, static optimisation,

cognitive control, and indeed decision-making, might be based on dynamic, hierarchical, recurrent processes that continuously update expected levels of reward and cost of tasks, and that allocate resources dynamically based on those updates [37]. Cognitive control has been related to an extensive network of brain regions, including dorsal cingulate and dorsolateral prefrontal cortices, which in turn are connected with a network of regions intricately linked to decision-making, such as medial prefrontal cortex [68]. However, details of the architecture and functioning of this network are only beginning to emerge. CCT can provide a much-needed framework to characterise cognitive resources required for cognitive control, providing crucial insights into not only decision-making, but also cognition more generally.

Concluding Remarks: Towards Computationally Plausible Models of Human Decision-Making

Most modern theories of decision-making assume, implicitly or explicitly, that decision-makers optimise [8]. The same is the case for many models of metadecision-making and cognitive control [61,68]. These models all assume, explicitly or implicitly, that decision-makers are always able to find the best option available to them. We argue that these models are implausible for most decision situations because they would require computational resources, in the form of time (energy), memory, and others, that are beyond those available to decision-makers.

Existing work has shown that human decision quality is strongly affected by instance complexity in canonical decision tasks [38,55]. These findings may explain many behavioural biases documented in the literature, such as choice overload, present bias or the disposition effect [38]. Indeed, heuristics, which are usually provided as an explanation of cognitive biases, could be regarded as an effective response of an organism to cognitive resource constraints (Box 4). Future work will need to identify the dimensions of instance complexity that affect human

Box 4. The Computational Complexity of Every-Day Life

What is the computational complexity of problems encountered in every-day life? It turns out that many every-day problems, such as attention allocation, time management, and many financial problems, are in class NP [38] (see Box 1 in the main text).

However, it might be the case that most instances of the problems encountered in real life have low complexity. This idea is captured by the notion of average case complexity [24,85]. Unfortunately, it has become known that even the average instance of many problems is hard. The study of computational complexity of instances of naturally occurring problems is an active area of research [86].

A related issue concerns the hardness of **approximation algorithms**. We refer to an algorithm that is guaranteed to find the optimal solution of a problem as an exact algorithm. While we may not be able to find efficient exact algorithms for a problem, we might still be able to find efficient approximation algorithms, that is, algorithms that are inexact in the sense that they are not guaranteed to find the optimal solutions but that are guaranteed to closely approximate solutions. Obtaining an approximate solution to a problem might be much easier than computing the exact solution, but it might be good enough for a particular application. If such approximation algorithms existed for all NP problems, then the distinction of P versus NP would be less significant for practical purposes.

This is not the case, however. An important theorem in CCT, the PCP Theorem [87], shows that, in many cases, computing approximate solutions is as hard as computing exact solutions [24]. For decision theory, this means that bounded rationality may be no easier than full rationality.

In the decision sciences, it is often assumed that humans use heuristics to navigate the complexity of decisions [11,15]. To separate the notion of heuristic from that of approximation algorithm, we define the former as an inexact algorithm that does not have an approximation guarantee, that is, it is not known how closely it will approximate the solution [33].

Little attention has been paid to the computational complexity of heuristics. There is no reason to believe, *ex ante*, that all heuristics are easy from a computational perspective and, indeed, many heuristics may be computationally complex for many instances. Worse, at present, we do not have any way to tell whether a given heuristic is easy for a given instance (see 'Does computational complexity theory apply to human decision-making?' in the main text).

decision-making capacity, which may overlap with those relevant for electronic computers. Among others, this work will also contribute to a deeper understanding of the differences between human computing and computing of electronic computers.

In addition to understanding computational resource requirements, a model of human decision-making will also need to take into account the resources available to the decision-maker in various decision situations (Figure 2; see Outstanding Questions). The latter is complicated by the fact that resource availability appears to be highly context-dependent [68]. In such a framework, decision-making is inseparable from metacognition (cognitive control), because the latter controls resource allocation, a crucial input in the decision-making process.

We hope that our argument has made clear the need for decision theorists, neurobiologists, and computer scientists to join forces in the quest to decipher how humans choose. Their insights will benefit not only the psychological and social sciences, but also the life sciences and medicine (mental disorders), and ultimately computer science (artificial intelligence and human–robot interactions) [73].

Outstanding Questions

Can the brain detect computational complexity? How?

How does the brain adapt to computational complexity?

Which resources does the brain use for problem solving, and what are the binding constraints?

How does the brain allocate resources when confronted with a particular problem?

Which approaches does the brain use to solve computational problems?

Acknowledgements

The authors would like to thank Juan Pablo Franco Ulla, Nitin Yadav, as well as three anonymous reviewers for their comments and suggestions, which have substantially improved the manuscript. P.B. received financial support from the University of Melbourne Research at Melbourne Accelerator Program. C.M. received financial support from the University of Melbourne Faculty of Business and Economics Strategic Initiatives Fund.

References

1. Samuelson, P.A. (1938) A note on the pure theory of consumer's behaviour. *Economica* 5, 61–71
2. Samuelson, P.A. (1948) Consumption theory in terms of revealed preference. *Economica* 15, 243–253
3. Houthakker, H.S. (1950) Revealed preference and the utility function. *Economica* 17, 159–174
4. Mas-Colell, A. *et al.* (1995) *Microeconomic Theory,* Oxford University Press
5. Nash, J.F. (1950) Equilibrium points in n-person games. *Proc. Natl. Acad. Sci. U. S. A.* 36, 48–49
6. Kahneman, D. and Tversky, A. (1979) Prospect theory – analysis of decision under risk. *Econometrica* 47, 263–291
7. Dayan, P. (2012) How to set the switches on this thing. *Curr. Opin. Neurobiol.* 22, 1068–1074
8. Blume, L.E. and Easley, D. (2008) Rationality. In *New Palgrave Dictionary of Economics* (Durlauf, S. and Blume, L.E., eds), pp. 884–893, Palgrave Macmillan
9. Simon, H.A. (1955) A behavioral model of rational choice. *Q. J. Econ.* 69, 99–118
10. Gigerenzer, G. and Selten, R. (2002) *Bounded Rationality: The Adaptive Toolbox,* MIT Press
11. Gigerenzer, G. and Brighton, H. (2009) Homo heuristicus: why biased minds make better inferences. *Top. Cogn. Sci.* 1, 107–143
12. Lieder, F. and Griffiths, T.L. (2015) When to use which heuristic: a rational solution to the strategy selection problem. In *Proceedings of the 37th Annual Conference of the Cognitive Science Society* (Noelle, D.C. *et al.*, eds), pp. 1362–1367, Cognitive Science Society
13. Gabaix, X. *et al.* (2006) Costly information acquisition: experimental analysis of a boundedly rational model. *Am. Econ. Rev.* 96, 1043–1068
14. Sims, C.A. (2003) Implications of rational inattention. *J. Monetary Econ.* 50, 665–690
15. Tversky, A. and Kahneman, D. (1974) Judgment under uncertainty: heuristics and biases. *Science* 185, 1124–1131
16. Payne, J.W. *et al.* (1993) *The Adaptive Decision Maker,* Cambridge University Press
17. Horvitz, E.J. (1987) Reasoning about beliefs and actions under computational resource constraints. In *Proceedings of the Third Workshop on Uncertainty in Artificial Intelligence* (Lemmer, J. *et al.*, eds), pp. 429–447, AUAI Press
18. Russell, S.J. and Subramanian, D. (1995) Provably bounded-optimal agents. *J. Artif. Intell. Res.* 2, 575–609
19. Griffiths, T.L. *et al.* (2015) Rational use of cognitive resources: levels of analysis between the computational and the algorithmic. *Top. Cogn. Sci.* 7, 217–229
20. Lewis, R.L. *et al.* (2014) Computational rationality: linking mechanism and behavior through bounded utility maximization. *Top. Cogn. Sci.* 6, 279–311
21. Gershman, S.J. *et al.* (2015) Computational rationality: a converging paradigm for intelligence in brains, minds, and machines. *Science* 349, 273–278
22. Hartmanis, J. and Stearns, R.E. (1965) On the computational complexity of algorithms. *Trans. Am. Math. Soc.* 117, 285–306
23. Cook, S.A. (1983) An overview of computational complexity. *Commun. ACM* 26, 400–408
24. Arora, S. and Barak, B. (2009) *Computational Complexity: A Modern Approach,* Cambridge University Press
25. Church, A. (1936) An unsolvable problem of elementary number theory. *Am. J. Math.* 58, 345–363
26. Turing, A.M. (1937) On computable numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* 42, 230–265
27. Cook, S.A. (1971) The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on the Theory of Computing* (Harrison, M.A., ed.), pp. 151–158, ACM
28. Currin, A. *et al.* (2017) Computing exponentially faster: implementing a non-deterministic universal Turing machine using DNA. *J. R. Soc. Interface* 14, 20160990
29. Karp, R.M. (1972) Reducibility among combinatorial problems. In *Complexity of Computer Computations* (Miller, R.E. and Thatcher, J.W., eds), pp. 85–103, Springer
30. Trakhtenbrot, B.A. (1984) A survey of Russian approaches to Perebor (brute-force searches) algorithms. *IEEE Ann. Hist. Comput.* 6, 384–400

31. Knuth, D.E. (1974) Postscript about NP-hard problems. *ACM SIGACT News* 6, 15–16
32. Pisinger, D. *et al.* (2004) *Knapsack Problems,* Springer
33. van Rooij, I. (2008) The tractable cognition thesis. *Cogn. Sci.* 32, 939–984
34. Fortnow, L. and Homer, S. (2003) A short history of computational complexity. *Bull. EATCS* 26, 400–408
35. Fortnow, L. (2009) The status of the P versus NP problem. *Commun. ACM* 52, 78–86
36. Achlioptas, D. *et al.* (2005) Rigorous location of phase transitions in hard optimization problems. *Nature* 435, 759–764
37. Hunt, L.T. and Hayden, B.Y. (2017) A distributed, hierarchical and recurrent framework for reward-based choice. *Nat. Rev. Neurosci.* 18, 172–182
38. Murawski, C. and Bossaerts, P. (2016) How humans solve complex problems: the case of the knapsack problem. *Sci. Rep.* 6, 34851
39. Gershman, S. and Wilson, R. (2010) The neural costs of optimal control. *Adv. Neural Inf. Process. Syst.* 23, 4167
40. Cooper, G.F. (1990) The computational complexity of probabilistic inference using Bayesian belief networks. *Artif. Intell.* 42, 393–405
41. Simon, H.A. (1956) Rational choice and the structure of the environment. *Psychol. Rev.* 63, 129–138
42. Tsotsos, J.K. (1988) How does human vision beat the computational complexity of visual perception. In *Computational Processes in Human Vision: An Interdisciplinary Perspective* (Pylyshyn, Z., ed.), pp. 286–338, Ablex Press
43. Tsotsos, J.K. (1993) The role of computational complexity in perceptual theory. *Adv. Psychol.* 99, 261–296
44. Gul, F. and Pesendorfer, W. (2008) The case for mindless economics. In *The Foundations of Positive and Normative Economics* (Caplin, A. and Schotter, A., eds), pp. 3–39, Oxford University Press
45. Glimcher, P.W. (2010) *Principles of Neuroeconomic Analysis,* Oxford University Press
46. Marr, D. (1982) *Vision,* W.H. Freeman
47. Schultz, W. (2015) Neuronal reward and decision signals: from theories to data. *Physiol. Rev.* 95, 853–951
48. Schultz, W. *et al.* (2017) The phasic dopamine signal maturing: from reward via behavioural activation to formal economic utility. *Curr. Opin. Neurobiol.* 43, 139–148
49. d'Acremont, M. and Bossaerts, P. (2008) Neurobiological studies of risk assessment: a comparison of expected utility and mean-variance approaches. *Cogn. Affect. Behav. Neurosci.* 8, 363–374
50. Vickery, T.J. *et al.* (2011) Ubiquity and specificity of reinforcement signals throughout the human brain. *Neuron* 72, 166–177
51. Piccinini, G. (2007) Computationalism, the Church–Turing thesis, and the Church–Turing fallacy. *Synthese* 154, 97–120
52. Dennett, D.C. (1978) *Brainstorms,* MIT Press
53. Churchland, P.M. and Churchland, P.S. (1990) Could a machine think? *Sci. Am.* 262, 32–37
54. Copeland, B.J. (2002) Hypercomputation. *Minds Machines* 12, 461–502
55. Meloso, D. *et al.* (2009) Promoting intellectual discovery: patents versus markets. *Science* 323, 1335–1339
56. Camerer, C. *et al.* (1997) Labor supply of New York City cabdrivers: one day at a time. *Q. J. Econ.* 112, 407–441
57. Crawford, V.P. and Meng, J. (2011) New York City cab drivers' labor supply revisited: reference-dependent preferences with rational-expectations targets for hours and income. *Am. Econ. Rev.* 101, 1912–1932
58. Farber, H.S. (2015) Why you can't find a taxi in the rain and other labor supply lessons from cab drivers. *Q. J. Econ.* 130, 1975–2026
59. Fama, E.F. (1965) The behavior of stock-market prices. *J. Bus.* 38, 34–105
60. Kool, W. and Botvinick, M. (2013) The intrinsic cost of cognitive control. *Behav. Brain Sci.* 36, 697–698
61. Boureau, Y.-L. *et al.* (2015) Deciding how to decide: self-control and meta-decision making. *Trends Cogn. Sci.* 19, 700–710
62. Baddeley, A. (2012) Working memory: theories, models, and controversies. *Annu. Rev. Psychol.* 63, 1–29
63. Hofmann, W. *et al.* (2012) Executive functions and self-regulation. *Trends Cogn. Sci.* 16, 174–180
64. Otto, A.R. *et al.* (2013) Working-memory capacity protects model-based learning from stress. *Proc. Natl. Acad. Sci. U. S. A.* 110, 20941–20946
65. Schmeichel, B.J. (2007) Attention control, memory updating, and emotion regulation temporarily reduce the capacity for executive control. *J. Exp. Psychol. Gen.* 136, 241–255
66. Otto, A.R. *et al.* (2014) Cognitive control predicts use of model-based reinforcement learning. *J. Cogn. Neurosci.* 27, 319–333
67. Schmeichel, B.J. *et al.* (2008) Working memory capacity and the self-regulation of emotional expression and experience. *J. Pers. Soc. Psychol.* 95, 1526–1540
68. Botvinick, M.M. and Cohen, J.D. (2014) The computational and neural basis of cognitive control: charted territory and new frontiers. *Cogn. Sci.* 38, 1249–1285
69. Shenhav, A. *et al.* (2013) The expected value of control: an integrative theory of anterior cingulate cortex function. *Neuron* 79, 217–240
70. Westbrook, A. and Braver, T.S. (2015) Cognitive effort: a neuroeconomic approach. *Cogn. Affect. Behav. Neurosci.* 15, 395–415
71. Keramati, M. *et al.* (2011) Speed/accuracy trade-off between the habitual and the goal-directed processes. *PLoS Comput. Biol.* 7, e1002055
72. Howes, A. *et al.* (2009) Rational adaptation under task and processing constraints: implications for testing theories of cognition and action. *Psychol. Rev.* 116, 717–751
73. Lake, B.M. *et al.* (2016) Building machines that learn and think like people. *Behav. Brain Sci.* 24, 1–101
74. Aaronson, S. (2011) Why philosophers should care about computational complexity. *arXiv* 1108.1791
75. Fagin, R. (1974) Generalized first-order spectra and polynomial-time recognizable sets. In *Complexity of Computation* (Karp, R., ed.), pp. 43–73, SIAM-AMS
76. Immerman, N. (1987) Languages that capture complexity classes. *SIAM J. Comput.* 16, 760–778
77. Immerman, N. (1999) *Descriptive Complexity,* Springer
78. Anderson, P.W. (1999) Computing solving problems in finite time. *Nature* 400, 115–116
79. Fu, Y. and Anderson, P.W. (1986) Application of statistical mechanics to NP-complete problems in combinatorial optimisation. *J. Phys. A Math. Gen.* 19, 1605
80. Friedman, M. (1953) The methodology of positive economics. In *Essays in Positive Economics* (Friedman, M., ed.), pp. 3–43, University Of Chicago Press
81. Samuelson, P.A. (1936) A note on measurement of utility. *Rev. Econ. Stud.* 4, 155–161
82. Kleene, S.C. (1936) General recursive functions of natural numbers. *Math. Ann.* 112, 727–742
83. Bernstein, E. and Vazirani, U. (1997) Quantum complexity theory. *SIAM J. Comput.* 26, 1411–1473
84. Valiant, L.G. (2013) *Probably Approximately Correct,* Basic Books
85. Levin, L.A. (2006) Average case complete problems. *SIAM J. Comput.* 15, 285–286
86. Livne, N. (2006) All natural NPC problems have average-case complete versions. *SIAM J. Comput.* 15, 285–286
87. Arora, S. *et al.* (1998) Proof verification and the hardness of approximation problems. *J. ACM* 45, 501–555