Combinatorial algorithms guide economic deliberation

Tao Hong
University of Pittsburgh https://orcid.org/0000-0003-4244-9028
William Stauffer (wrs@pitt.edu)
University of Pittsburgh https://orcid.org/0000-0003-1031-8824

Article

Keywords:

Posted Date: May 9th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1598753/v1

Title: Combinatorial algorithms guide economic deliberation

Authors: Tao Hong[1-3], William R. Stauffer[1-3]

Affiliations

1. Department of Neurobiology, University of Pittsburgh
2. Program in Neural Computation, Carnegie Mellon University
3. Center for the Neural Basis of Cognition

Contact

William Stauffer
Assistant Professor, Dept. of Neurobiology
University of Pittsburgh, Pittsburgh, PA
wrs@pitt.edu

Abstract

Economic deliberations often require sophisticated analysis, but the behavioral algorithms and neurobiological mechanisms for complex deliberations remain unknown. We asked nonhuman primates (NHPs) to select optimal, or suboptimal, item subsets in a modified optimization problem. Surprisingly, the NHP behavior reflected computational algorithms – some that rely on combinatorial reasoning – and the deliberation times reflected the number of operations the best-matching algorithms used. Artificial neural networks revealed that constructing combinatorial solutions required greater computational resources than their greedy counterparts. These results provide mechanistic insights into the sophisticated reasoning skills that support economic deliberation.

Main Text

Economic decisions rarely have 'correct' responses. Instead, decision makers must deliberate and determine the most valuable option prior to making a choice. This process of deliberate consideration, or deliberation, can be computationally demanding. Consider that, values are dependent on combinations of factors including context, alternatives, and internal states. Such combinatorial considerations result in computationally complex economic deliberations that require sophisticated analytical abilities. However, the vast majority of neuroeconomic studies have investigated simple choices with limited deliberation.[1-4] These studies have revealed single unit correlates of economic values and decisions in the orbitofrontal cortex (OFC),[1, 5, 6] dorsolateral prefrontal cortex (dlPFC),[7, 8] lateral intraparietal cortex (LIP),[2, 9] striatum,[10] and midbrain dopamine neurons,[3, 4, 11] but the neurocomputational algorithms and implementations of pure economic deliberations remain unknown.

Combinatorial reasoning is a core facet of the more general concept of quantitative reasoning. Previous studies have shown that nonhuman primates (NHPs) possess numerical sense,[12-17] perform mathematical operations,[18-21] exploit probabilistic reasoning,[22, 23] and even participate in complex economic games.[24-28] NHPs have demonstrated the ability to solve combinatorial optimization (CO) problems in a route planning context,[29-31] but it remains unclear how they optimize in the context of economic choices. The main purpose of this study was to determine the step-by-step processes, or algorithms, that the animals used to optimize economic values. Thus, we devised the 'knapsack task', based on the eponymous problem from computer science,[32] to promote temporally extended economic deliberations. To do so, we used a variety of efficient computational algorithms to characterize the NHP behaviors, including algorithms that rely on combinatorial reasoning.[33-35]

The objective of each knapsack trial was to select subsets of items presented on a touchscreen in order to maximize juice reward, without exceeding a fixed limit of 0.8 ml. During preliminary training sessions, two rhesus macaque monkeys learned the juice rewards associated with eleven items, each represented by a unique fractal image (Extended Data Fig. 1a-c). Each knapsack trial began with the appearance of an initiation target (Fig. 1a). When the animal touched the initiation target, the screen displayed an 'instance,' a combination of five items randomly selected from the eleven pre-trained items. All five items appeared simultaneously on the screen, and the animals used the touchscreen to select individual items. When an item was selected, it was highlighted and the virtual knapsack at the top of the screen was 'filled' by an amount equivalent to item's associated reward. Items could not be de-selected. The items remained on the screen for five seconds, and then, if the sum of the

selected subset was less than or equal 0.8 ml the animals were rewarded with the juice equivalent of the sum of the selected items (Fig 1a). If, however, the animals selected items with a sum that exceed 0.8 ml, no reward was delivered at the end of the trial, and a 4 second timeout was added before the inter-trial interval (Fig. 1b). We implemented three procedures to minimize the potential that the animals would learn specific combinations of items. (1) We randomized the screen location for each item on every trial (Methods). (2) We presented all 462 combinations of 11 chose 5 items, so that the probability of presenting a specific instance more than once per day was less than 0.15. (3) Finally, we used each set of 11 items for no more than one month of behavioral testing, and then trained the monkeys on a new set of fractals.

To test whether the animals' behavior was incentive compatible, we introduced positive and negative control items that predicted 0.8 and 0.0 ml, respectively. When the positive control item appeared as part of a 5-item instance, the animals chose it and nothing else – the optimal economic response – on most of those trials (Fig. 1c, Movie S1, $p < 10^{-70}$ and $p < 10^{-70}$ for animal G and B, Binomial test, comparison with random performance). When the negative control item was presented as part of an instance, the animals included it in the selected subset on less than 5% of trials (Fig. 1d, $p < 10^{-70}$ and $p < 10^{-70}$ for animal G and B, Binomial test, comparison with random performance). These results demonstrate that the animals' behavior was incentive compatible. Thus, the animals sought to maximize economic gain on every trial.

We examined the animals' performance when the control cues were not present. Animals G and B achieved rewards that were at least 90% of optimal on 32% and 26% of all trials, and achieved 75% of optimal on 72% and 64% of all trials, respectively (Fig. 1e). The deliberative nature of the behavior was frankly evident in video recordings of task performance (Movie S2). When a high value item – especially the item predicting 0.7 ml of reward – was part of an instance, the animals quickly selected it (Movie S3). On the other hand, when high value items were not available in a particular instance, the animals often exploited the 5 second window to consider multiple combinations (Movie S4). To quantify what aspects of the task caused these different behaviors, we considered the values of the five items and three combinatorial parameters that we hypothesized to be important: (1) the 'number of witnesses,' or the number of ways to avoid breaking the knapsack,[36] (2) the random score (Methods), (3) the number of 'good' solutions (Methods). We used linear regression to examine the effect of these factors on instance-wise performance (Methods). Across both animals, the most parsimonious models retained maximum item value, number of witnesses, and random

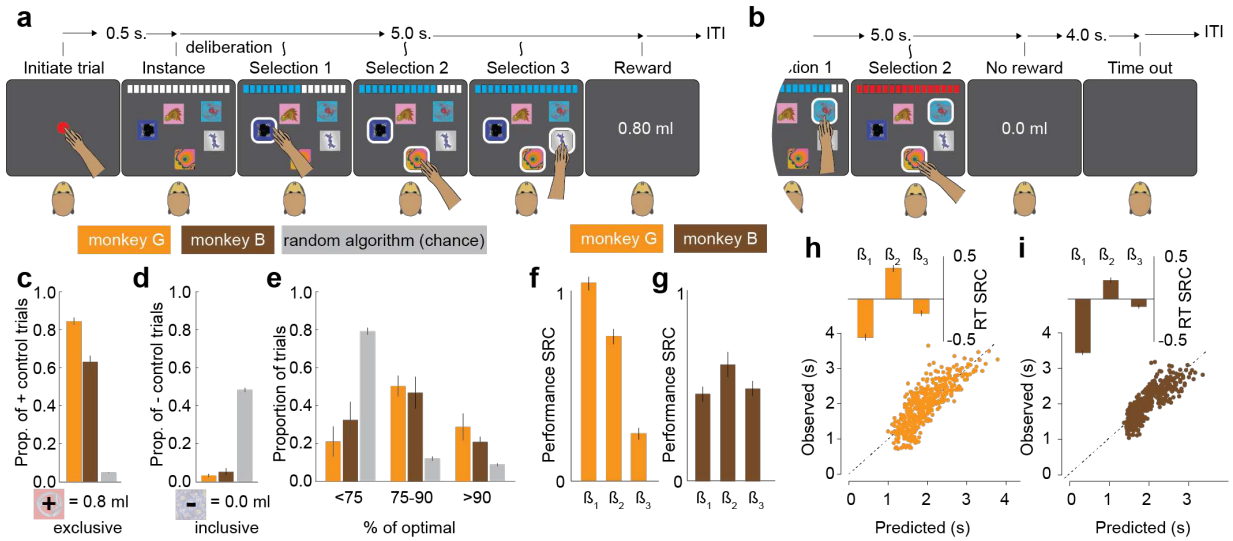


Figure 1: Knapsack task and performance metrics. a) Schematic of the knapsack task. The animals initiated knapsack trials by touching a central target. After initiation, an instance was displayed and remained on the screen for 5 seconds. During those 5 seconds, the number and identity of the selections was determined by the animal. When they selected an item, it was highlighted, it could not be de-selected, and the volume associated with the selection was added to the virtual knapsack at the top of the screen. If the sum of the items was less than or equal to 0.8 ml, then the sum was delivered at the end of the 5 second period. **b**) Schematic diagram of a knapsack trial when animal exceeded 0.8 ml. No reward was delivered and a 4 second timeout was imposed. **c**) Orange (monkey G) and brown (monkey B) bar graphs show the proportion of positive control trials when the animals selected the positive control cue (+) and nothing else (exclusive). The gray bar graph shows the proportion of positive control trials when a random algorithm chooses the positive control and nothing else. Error bars are ±SEM across trials. **d**) Orange (monkey G) and brown (monkey B) bar graphs show the proportion of negative control trials when the animals included the negative control cue (-) in the chosen subset (inclusive). The gray bar graph shows the proportion of negative control trials when a random algorithm includes the negative control item in chosen subset (inclusive). Error bars are ±SEM across trials. **e**) Bar graphs summarize the performance, expressed as percent optimal, for both animals and indicate chance level performance, determined by a random algorithm, for three non-overlapping bins. Both animals also outperformed chance in the 75-90% and greater than 90% optimal. Error bars are ±SEM across sessions. **f**) Bar graphs show the standardized regression coefficients for maximum item value ($ß_1$), number of witnesses ($ß_2$), and random score ($ß_3$) for monkey G. Error bars are ±SEM across instances. **g)** As in f, for monkey B. **I**) (inset) The same variables that influenced performance, maximum item value ($ß_1$), number of witnesses ($ß_2$), and random score ($ß_3$), were highly significant predictors of overall time between instance presentation and last touch. Error bars are ±SEM across instances. (main) Scatter plot of observed and predicted response times, using the model specified in the inset, for monkey G. **i**) As in **h**, but for monkey B,

score. As Movie S3 indicated, high maximum item values provided an easy way to achieve near optimal reward and were associated with high performance for both animals ($ß_1$ = 1.037 and 0.445, $p < 10^{-15}$ and $p < 10^{-7}$ for animal G and B, respectively). High number of witnesses was associated with high performance ($ß_2$ = 0.757 and 0.618, $p < 10^{-15}$ and $p<10^{-8}$ for animal G and B, respectively), as was random score ($ß_3$ = 0.254 and 0.438, $p < 10^{-4}$ and $p < 10^{-8}$ for animal G and B, respectively). These same three parameters strongly predicted response times in both monkeys (Fig.1h,i). Notably, maximum item value was the property of an individual item, whereas number of witnesses and random score were combinatorial properties of an entire instance. Thus, these results demonstrate that both properties of individual items and combinatorial properties of instances influenced the quality and duration of deliberation.

Computer science research has developed a variety of computational strategies to determine approximately optimal solutions to the knapsack problem, and, moreover, these algorithms can be used to characterize deliberation in the knapsack task (Methods).[33-35, 37] For example, the greedy algorithm searches for the largest item that does not 'break the sack', whereas the Sahni-*k* algorithm searches for ideal k-item combinations. We classified the behavior on each trial by matching the items and selection order to one of three intuitive and well-established algorithms – the greedy algorithm, which considers items one at a time, and the Sahni-*k* and Johnson-*k* algorithms, which rely on combinatorial processing (Fig. 2a, Methods). As expected, many solutions generated by both animals matched the greedy algorithm. However, in a substantial proportion of trials, the behavioral solutions matched the solutions produced by the Sahni-*k* and Johnson-*k* algorithms (Fig. 2b). Remarkably, we could predict what algorithm an animal would use, based on the other animal's behavior. There were high correlations between the algorithms the two monkeys used for a given instance (Fig. 2c-e, ρ = 0.54, 0.69 and 0.72, $p < 10^{-35}$, $p < 10^{-65}$ and $p < 10^{-73}$ for greedy, Sahni-*k* and Johnson-*k* algorithms, respectively, Spearman's correlation). Altogether, these results demonstrate that animals flexibly switched between algorithmic strategies and applied combinatorial reasoning to get the best rewards. Moreover, the high inter-subject correlations suggest that the application of combinatorial reasoning was not random and likely determined by the instances.

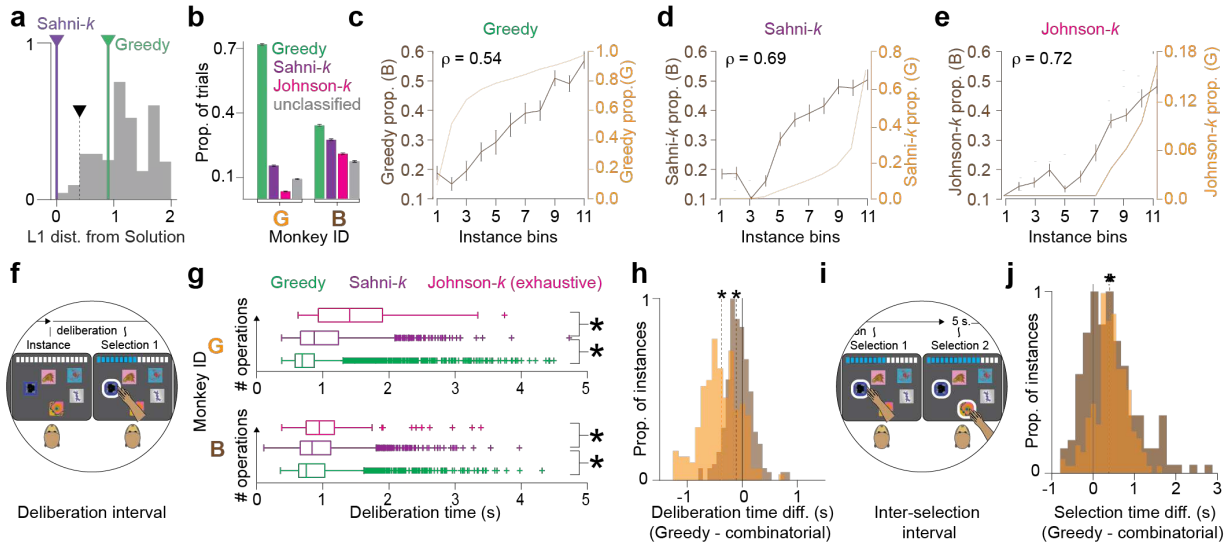


Figure 2: Behavioral optimization strategies reflect computational algorithms. a) Example histogram of L1 distances between behavioral solutions – defined as the origin of the x-axis, algorithmic solutions, and a distribution of randomly generated solutions that define the significance threshold (inverted black triangle indicates α = 0.05). In this example, the solution is classified as Sahni-*k*. If no algorithm is L1 closer than the threshold, the trials is designated as 'unclassified.' **b)** Bar graph shows the proportion of trials significantly matched to the greedy, Sahni-*k*, and Johnson-*k* algorithms. Error bars are ±SEM across trials. **c)** Superimposed line graphs show that,

during instances when monkey G's solutions approximated the greedy algorithm, monkey B's solutions were more likely to approximate the greedy algorithm. Instances were ordered according to increasing proportion of greedy solutions from monkey G, and the same ordering was then used for monkey B. Each instance bin contains 41 instances. Error bars are ±SEM across trials. **d)** As in c, for the Sahni-*k* algorithm. **e)** As in c, for the Johnson-*k* algorithm. **f)** Detail of task schematic highlights the deliberation interval, defined as the time between the instance presentation and first selection. **g)** Box and whisker plots show the deliberation times for trials classified as greedy, Sahni-*k*, or Johnson-*k*. **h)** Histograms show the differences in average deliberation times between greedy and combinatorial trials within the same instance. **i)** Detail of task schematic highlights the selection intervals, defined as the time between the first and second, or second and third, selection. **j)** Histograms show the differences in average duration of the selection intervals times between greedy and combinatorial trials within the same instance.

A key difference between the greedy, Sahni-*k*, and Johnson-*k* algorithms is the number of operations performed before the first selection. When there are 5 items, greedy search requires 5 operations, whereas Sahni-*k* search requires at least 5 choose *k* operations – the Sahni-*2* algorithm, for example, requires at least 10 operations. Accordingly, when the animals employ a greedy strategy, the deliberation time – defined as the period between instance presentation and first selection (Fig. 2d) – should be shorter than when the animals approximate a combinatorial search. This is indeed what we observed. The deliberation time was longer on Sahni-*k* trials compared to greedy trials (Fig. 2g, $p < 10^{-116}$ and $p < 10^{-8}$ for animals G and B, respectively). Furthermore, the deliberation time reflected differently sized search spaces associated with distinct combinatorial strategies: when the minimum value item is 0.25, the Johnson-*k* resorts to a near exhaustive search requiring at least 25 operations. On these trials, the deliberation times were significantly longer - nearly 600 ms longer for animal G and 100 ms longer for animal B - compared to Sahni-*k* trials (Fig. 2g, $p < 10^{-7}$ and $p < 10^{-3}$ for animals G and B, respectively). The differences in deliberation times are even significant when combinatorial and greedy approaches were applied to different trials of the same instance (Fig. 2h, $p < 10^{-50}$ and $p < 10^{-12}$ for animals G and B, respectively. Sign rank test). Since the animals would need to plan multiple selections during combinatorial deliberation, we predicted that the subsequent inter-selection intervals should be shorter than the same intervals during greedy trials (Fig. 2i). This is indeed what we observed. Namely, the subsequent selection times were shorter on combinatorial trials than on greedy trials (Fig. 2j). These deliberation and selection times strongly corroborate the results of the algorithmic classification, and, moreover, suggest that these algorithms are directly implemented by underlying neural circuits.

To gain preliminary insights into the neural circuit implementation of optimization algorithms, we trained a series of artificial neural networks to simulate greedy, Sahni-*2*, Sahni-*3*, Johnson-*2*, and Johnson-*3* algorithms. We generated five solution sets – one for each algorithm – where each set included the algorithm's solutions to all permutations of each 5-item instance (Methods). Each artificial neural network had 5 units in both the input and output layers, four

hidden layers, and 270 total hidden units. We used each solution set to train, one-by-one, five classes of feed-forward neural networks – greedy, Sahni-*2*, Sahni-*3*, Johnson-*2*, and Johnson-*3* networks (Fig.3a, Methods). We used ten-fold cross validation to demonstrate that each network class approximated the algorithms that they were trained to mimic (Fig. 3b, diagonal). Moreover, testing each network against the held-out data from the other algorithms demonstrated that the networks simulated their target algorithms better than networks trained with other solution sets (Fig3b, off diagonal). Together, these results demonstrate that the algorithm-specific training data sets created functionally distinct network classes that mimic different optimization algorithms.

The greedy and Sahni-*k* algorithms produced the most common matches to the animals' noncombinatorial and combinatorial behaviors, respectively (Fig 2b). For instances where the Sahni-*k* algorithm matched the behavior more frequently than the greedy algorithm, the Sahni-*2* networks had higher similarity scores (Methods) compared to the greedy networks (Fig. 3c, purple dots, $p < 10^{-7}$ and $p < 10^{-28}$ for animal G and B, Wilcoxon test). Conversely, for instances where the greedy algorithm matched the behavior more frequently than the Sahni-*k* algorithm, the greedy networks displayed higher network similarity scores than the Sahni-*2* networks (Fig. 3c, green dots, $p < 10^{-66}$ and $p < 10^{-10}$ for animal G and B, Wilcoxon test). Furthermore, the degree to which a network mimicked the animals' behavior was dependent on how often the corresponding algorithm was used. The proportion of greedy trials within each instance strongly predicted the greedy network similarity score (Fig. 3d, $\rho = 0.81$ and 0.78, $p < 10^{-106}$ and $p < 10^{-94}$ for animal B and G, Pearson's correlation). Likewise, the proportion of Sahni-*k* trials within an instance predicted the Sahni-*2* network similarity score (Fig. 3e, $\rho = 0.27$ and 0.19, $p < 10^{-8}$ and $p < 10^{-4}$, for animal B and G, Pearson's correlation). These results are remarkable because the networks were trained on the algorithmic solutions, not the behavioral solutions. Thus, these results (1) provide even greater evidence that the animals used distinct algorithmic strategies to maximize rewards, (2) illustrate the behavioral relevance of the greedy and Sahni-*k* algorithms, and (3) provide strong justification to interrogate the greedy and Sahni-*2* networks to gain preliminary insights into the neural implementation of optimization behaviors.

We conducted single unit lesions within the networks trained with the greedy and Sahni-*2* algorithms. A substantial fraction of lesions produced performance deficits in both networks. However, lesioning single units in the Sahni-*2* network produced larger effects on network performance, compared to lesions made in the greedy network. Similarly, the performance on most individual instances was more sensitive to lesions in the Sahni-*2* network,

compared to the greedy network. These results indicate that networks trained to solve combinatorial problems require greater computational resources compared to networks trained to solve non-combinatorial problems. We quantified the dimensionality of each network as the number of principal components (PCs) that were required to capture the variance in each networks' activity. Consistent with the single unit lesions, the Sahni-*2* networks required more PCs than the greedy networks to capture the same amount of variance (Fig. 3g). This result holds true even on instances where the greedy and Sahni-*2* networks produced identical outputs (Fig. 3g, Inset). Together, these results led us to hypothesize that solution representations emerged later in the Sahni-*2* network, compared to the greedy network. We evaluated whether simple linear classifiers could produce target solutions based on the activities of individual hidden layers (Methods). The second and third layer of the greedy networks contained enough information to construct 'greedy-like' solutions, whereas solution representations were largely absent in the early layers and only emerged in the last layers of the Sahni-*2* network (Fig. 3h). These differences in computational processing were

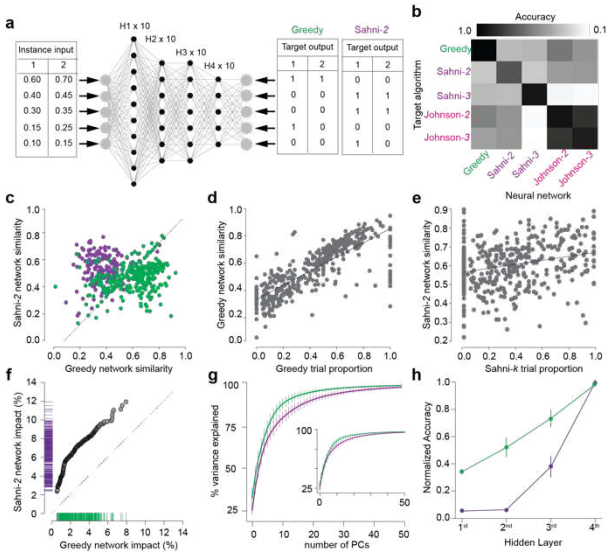


Figure 3: Artificial neural networks (ANNs) simulate algorithmic and behavioral optimization. a) Schematic diagram of ANNs and training paradigm. Large gray dots represent the 5 input and 5 output units. Small black dots represent 1/10 of the hidden units in each hidden layer. The inputs are the 5 elements of the combination, and the training data are comprised of the algorithm-defined solutions. Separate networks are trained for each algorithm-defined solution set. **b)** Heatmap shows the accuracy of the five network types at matching the held-out solutions from the five algorithms. **c)** Scatterplot showing the relationship between similarity to greedy and Sahni-*2* networks for two instance types including greedy algorithm preferred (green dots) and Sahni-*k* algorithm preferred (magenta dots) instances. Network similarity here, and in d and e, is defined as 1-graph distance (Methods). The black dashed line is a unity line. **d)** Scatterplot showing the relationship between greedy similarity and the proportion of trials within an instance that matched the greedy algorithm. Each dot represents an instance. The black dashed line represents the best fitting linear function. **e)** as in d, for Sahni-*k*. **f)** Q-Q plot showing the effects of single unit lesions on instance performance. Each dot represents the performance change for an instance, averaged across all unit lesions. **g)** (main) Elbow plots show the percent variance explained as a function of the number of principal components (PCs) for greedy (green) and Sahni-*2* networks (magenta), averaged across all instances. Error bars are standard deviations across ten random initializations. (inset) Elbow plots show the percent variance explained as a function of the number of principal components (PCs) for greedy (green) and Sahni-*2* networks (magenta), averaged across all instances where greedy and Sahni-*k* algorithms produce identical output. Error bars are standard deviations across ten random initializations. **h)** Line graphs showed the normalized accuracy of linear classifiers in predicting target solutions for individual hidden layers in the greedy and Sahni-*2* networks. We normalized the cross-validated performance of the linear classifiers by dividing the performance of the corresponding network (diagonal in **b**). Error bars are SD across ten random initializations.

consistent with the longer deliberation time observed in the combinatorial trials and could contribute to the animals' preference for noncombinatorial approaches.

Discussion

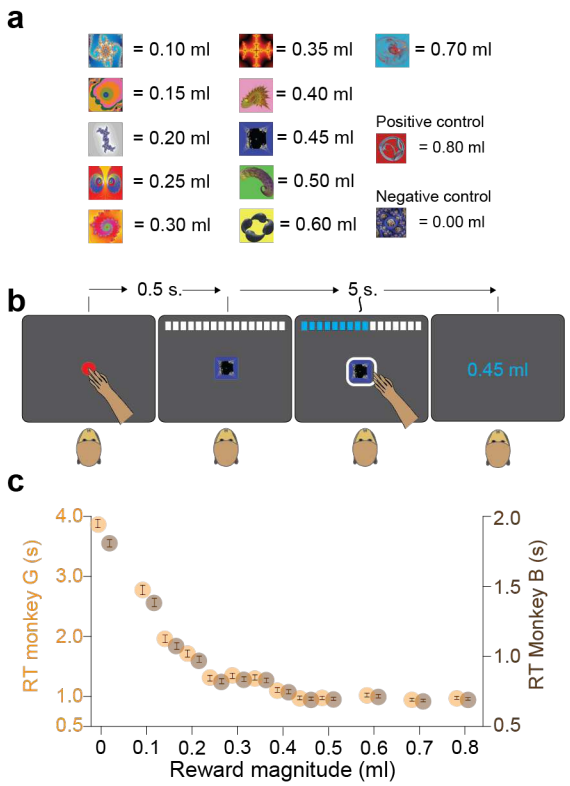
Economic deliberation often requires sophisticated reasoning. Combinations of different commodities or different choice parameters, including gains, losses, uncertainty, time, and effort, make utility maximization a computationally complex problem. A major challenge to understanding high-level cognitive computations is measuring such mental processes in nonhuman animals.[38] Here we created the knapsack task and used it to demonstrate that rhesus monkeys used algorithm-based reasoning to maximize rewards in a complex optimization task. The conclusion that the animals performed algorithmic optimization in general, and combinatorial reasoning specifically, was based on the following five lines of evidence. (1) Combinatorial properties of the instances affected both the performance and the response times (Fig. 1f-i). (2) The order and the identity of the chosen subsets matched the order and identity of items selected by discrete computational algorithms (Fig. 2a,b). (3) The algorithms that matched the instance solutions were highly correlated between the animals, suggesting that 'algorithm selection' was not random, but based on properties of the instance (Fig 2c-e). (4) The deliberation times and subsequent response times all reflected the number of operations each algorithm required, at each timepoint (Fig 2f-j). (5) Finally, artificial neural networks (ANNs) trained to mimic combinatorial and noncombinatorial algorithms mimicked combinatorial and noncombinatorial behavior, respectively. Together, these five lines of evidence overwhelmingly support the notion that rhesus monkeys employ algorithmic based reasoning to optimize rewards. ANNs further showed that combinatorial reasoning requires greater networks resources than noncombinatorial reasoning. Together, these data establish a novel behavioral and computational framework for understanding the neural circuit basis for sophisticated economic reasoning and reveal that combinatorial processing requires greater neurocomputational resources.

Appropriate decision-making is crucial for survival, and therefore understanding the neural basis of deliberation and decision making is a priority of modern neuroscience. Studies of perceptual judgements have revealed a psychologically intuitive and neuroscientifically grounded mechanism for the deliberate considerations of sensory evidence leading to choices.[39, 40] A key element to that research program, and the deep mechanistic insights that it has generated, is the normative control of perceptual difficulty. For example, low coherence motion in a random dot kinematogram is normatively harder to perceive than high coherence

motion. Easier trials require less integration time, whereas difficult trials require more integration time. Together, these manipulations have been the keystone of studies revealing neuronal correlates of evidence integration and perceptual decision boundaries in the lateral intraparietal area,[41] caudate,[42] and prefrontal cortex.[43] Binary choice tasks, which are commonly used to study the neural correlates of economic decisions, do not offer that same level of control over difficulty. For sure, several neuroeconomic studies have demonstrated that the differences in economic values between choice options operate analogously to the way perceptual difficulty operates on response times: smaller value differences lead to longer response times.[44, 45] However, what, exactly, causes this reaction time effect is not clearly determined.[46] Longer response times near 'choice indifference' – a widely used descriptive economic phrase that itself is not readily translated as 'difficult' – might reflect the consideration of option attributes,[47, 48] uncertainty in the underlying value distributions,[49] memory retrieval,[50, 51] or they may be simple byproducts of the underling circuit dynamics.[52] Thus, the mechanisms for deliberate consideration of economic variables leading to economic choice remain largely unknown. Optimization problems – in sharp contrast to binary choice tasks – possess a normative scale of difficulty: computational complexity.[36, 53, 54] Thus, the knapsack task and the optimization behaviors demonstrated here, provide a well-defined path to achieving a mechanistic understanding of economic deliberation.

We carefully selected the knapsack limit of 0.8 ml. Previous measurements of utility functions in monkeys performing economic tasks have consistently revealed that 0.8 ml falls on a region of the utility function that is roughly linear.[4, 55] Linear utility functions imply that the animals gained similar amounts of utility by increasing the total reward size from 0.5 to 0.6 ml, 0.6 to 0.7 ml, and 0.7 to 0.8 ml. Thus, the effect of diminishing marginal utility, which could easily lead to diminishing motivation to achieve optimal results as the reward size increases, was likely minimal within the reward ranges used. Nevertheless, 'satisficing' behavior – defined as finding a solution that is subjectively satisfactory and sufficient [56]– could play a significant role in the ultimate explanation of behavior. Whether the exact goal was optimal or satisfactory, however, was beyond the scope of this study. Our goal, here, was to characterize the algorithms the animals used to achieve high utility on every trial. Future studies with greater complexity, smaller reward intervals, and the application of satisficing models will be required to characterize satisficing behavior in rhesus monkeys. We suspect that understanding the neural mechanisms for determining a satisfactory reward size will be a crucial component of a complete understanding of economic behaviors.

As a general theory of decision-making behavior, utility maximization is computationally intractable. As the complexity of choice problems increases, the number of computations required to find the maximum utility exceeds the capabilities of the fastest computers or the most intricate brain networks. For this reason, it has long been recognized that heuristics are critical for navigating the real world.[57] Our results do not contradict this fundamental truth. Here, we have constrained the difficulties of the problem at a level where the animals can come close to rational reasoning and true utility maximization. We will use this paradigm to understand how neural circuits perform such sophisticated reasoning. In addition, future studies will push the complexity of the task higher to discover when and how logical reasoning breaks down, and to learn how the brain approximates utility maximization via heuristic algorithms. Our ultimate goal is to achieve an understanding of economic decision making that spans multiple 'levels of understanding.'[58, 59] Here, we demonstrated algorithmic insights into economic deliberation. These insights will serve as a theoretical framework to advance our understanding the cell type specific computations that enable sophisticated economic decisions.



Extended Data Figure 1 Item training. a) Fractal images for 1 stimulus set. We defined stimulus sets that each contained 11 items - fractal images that predicted rewards between 0.1 and 0.7 ml. – and two additional fractal images were used as positive (0.8 ml) and negative (0.0 ml) control items. **b)** Schematic of item training task. The animals used a simple stimulus-response-reward task to learn the predictive value of each individual item in the stimulus set (a). **c)** Scatter plot of response times vs reward magnitude for monkey G (orange, left y-axis) and monkey B (brown, right y axis). Lower response times for large-reward cues indicated that the animals had learned the values. Error bars are SEM across trials.

1. Padoa-Schioppa, C. & Assad, J.A. Neurons in the orbitofrontal cortex encode economic value. *Nature* **441**, 223-226 (2006).
2. Platt, M.L. & Glimcher, P.W. Neural correlates of decision variables in parietal cortex. *Nature* **400**, 233-238 (1999).
3. Lak, A., Stauffer, W.R. & Schultz, W. Dopamine prediction error responses integrate subjective value from different reward dimensions. *Proc Natl Acad Sci U S A* **111**, 2343-2348 (2014).
4. Stauffer, W.R., Lak, A. & Schultz, W. Dopamine reward prediction error responses reflect marginal utility. *Curr Biol* **24**, 2491-2500 (2014).
5. Padoa-Schioppa, C. & Assad, J.A. The representation of economic value in the orbitofrontal cortex is invariant for changes of menu. *Nat Neurosci* **11**, 95-102 (2008).
6. Padoa-Schioppa, C. Range-adapting representation of economic value in the orbitofrontal cortex. *J Neurosci* **29**, 14004-14014 (2009).
7. Hosokawa, T., Kennerley, S.W., Sloan, J. & Wallis, J.D. Single-neuron mechanisms underlying cost-benefit analysis in frontal cortex. *J Neurosci* **33**, 17385-17397 (2013).
8. Enel, P., Wallis, J.D. & Rich, E.L. Stable and dynamic representations of value in the prefrontal cortex. *Elife* **9**, e54313 (2020).
9. Louie, K. & Glimcher, P.W. Separating value from choice: delay discounting activity in the lateral intraparietal area. *J Neurosci* **30**, 5498-5507 (2010).
10. Lau, B. & Glimcher, P.W. Value representations in the primate striatum during matching behavior. *Neuron* **58**, 451-463 (2008).
11. Kobayashi, S. & Schultz, W. Influence of reward delays on responses of dopamine neurons. *J Neurosci* **28**, 7837-7846 (2008).
12. Brannon Elizabeth, M. & Terrace Herbert, S. Ordering of the Numerosities 1 to 9 by Monkeys. *Science* **282**, 746-749 (1998).
13. Nieder, A., Freedman David, J. & Miller Earl, K. Representation of the Quantity of Visual Items in the Primate Prefrontal Cortex. *Science* **297**, 1708-1711 (2002).
14. Viswanathan, P. & Nieder, A. Differential impact of behavioral relevance on quantity coding in primate frontal and parietal neurons. *Curr Biol* **25**, 1259-1269 (2015).
15. Nieder, A. The neuronal code for number. *Nat Rev Neurosci* **17**, 366-382 (2016).
16. Roitman, J.D., Brannon, E.M. & Platt, M.L. Monotonic Coding of Numerosity in Macaque Lateral Intraparietal Area. *PLOS Biology* **5**, e208 (2007).
17. Ramirez-Cardenas, A., Moskaleva, M. & Nieder, A. Neuronal Representation of Numerosity Zero in the Primate Parieto-Frontal Number Network. *Current Biology* **26**, 1285-1294 (2016).
18. Livingstone, M.S.*, et al.* Symbol addition by monkeys provides evidence for normalized quantity coding. *Proceedings of the National Academy of Sciences* **111**, 6822 (2014).
19. Cantlon, J.F., Merritt, D.J. & Brannon, E.M. Monkeys display classic signatures of human symbolic arithmetic. *Anim Cogn* **19**, 405-415 (2016).
20. Drucker, C.B., Rossa, M.A. & Brannon, E.M. Comparison of discrete ratios by rhesus macaques (Macaca mulatta). *Anim Cogn* **19**, 75-89 (2016).
21. Bongard, S. & Nieder, A. Basic mathematical rules are encoded by primate prefrontal cortex neurons. *Proceedings of the National Academy of Sciences* **107**, 2277-2282 (2010).

22. Yang, T. & Shadlen, M.N. Probabilistic reasoning by neurons. *Nature* **447**, 1075-1080 (2007).
23. Lin, Z., Nie, C., Zhang, Y., Chen, Y. & Yang, T. Evidence accumulation for value computation in the prefrontal cortex during decision making. *Proceedings of the National Academy of Sciences* **117**, 30728-30737 (2020).
24. Haroush, K. & Williams, Z.M. Neuronal prediction of opponent's behavior during cooperative social interchange in primates. *Cell* **160**, 1233-1245 (2015).
25. Baez-Mendoza, R., Harris, C.J. & Schultz, W. Activity of striatal neurons reflects social action and own reward. *Proc Natl Acad Sci U S A* **110**, 16634-16639 (2013).
26. Ong, W.S., Madlon-Kay, S. & Platt, M.L. Neuronal correlates of strategic cooperation in monkeys. *Nature Neuroscience* **24**, 116-128 (2021).
27. Barraclough, D.J., Conroy, M.L. & Lee, D. Prefrontal cortex and decision making in a mixed-strategy game. *Nat Neurosci* **7**, 404-410 (2004).
28. Seo, H., Cai, X., Donahue Christopher, H. & Lee, D. Neural correlates of strategic reasoning during competitive games. *Science* **346**, 340-343 (2014).
29. Cramer, A.E. & Gallistel, C.R. Vervet monkeys as travelling salesmen. *Nature* **387**, 464-464 (1997).
30. Teichroeb, J.A. Vervet monkeys use paths consistent with context-specific spatial movement heuristics. *Ecol Evol* **5**, 4706-4716 (2015).
31. Menzel, E.W. Chimpanzee spatial memory organization. *Science* **182**, 943-945 (1973).
32. Dantzig, T. *Number the Language of Science (1930)* (Kessinger Publishing, 2003).
33. Dantzig, G.B. Discrete-Variable Extremum Problems. *Operations Research* **5**, 266-277 (1957).
34. Horowitz, E. & Sahni, S. Computing partitions with applications to the knapsack problem. *J. ACM* **21**, 277-292 (1974).
35. Johnson, D.S. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences* **9**, 256-278 (1974).
36. Franco, J.P., Yadav, N., Bossaerts, P. & Murawski, C. Generic properties of a computational task predict human effort and performance. *Journal of Mathematical Psychology* **104**, 102592 (2021).
37. Martello, S. & Toth, P. *Knapsack Problems: Algorithms and Computer Implementations* (John Wiley & Sons Inc., USA, 1990).
38. Allen, C. & Hauser, M.D. Concept attribution in nonhuman animals: Theoretical and methodological problems in ascribing complex mental processes. *Philosophy of Science* **58** (1991).
39. Shadlen, M.N. & Newsome, W.T. Motion perception: seeing and deciding. *Proceedings of the National Academy of Sciences* **93**, 628-633 (1996).
40. Gold, J.I. & Shadlen, M.N. The neural basis of decision making. *Annu Rev Neurosci* **30**, 535-574 (2007).
41. Roitman, J.D. & Shadlen, M.N. Response of Neurons in the Lateral Intraparietal Area during a Combined Visual Discrimination Reaction Time Task. *The Journal of Neuroscience* **22**, 9475 (2002).
42. Ding, L. & Gold, J.I. Caudate encodes multiple computations for perceptual decisions. *J Neurosci* **30**, 15747-15759 (2010).

43. Kim, J.-N. & Shadlen, M.N. Neural correlates of a decision in the dorsolateral prefrontal cortex of the macaque. *Nature Neuroscience* **2**, 176-185 (1999).
44. Krajbich, I., Armel, C. & Rangel, A. Visual fixations and the computation and comparison of value in simple choice. *Nat Neurosci* **13**, 1292-1298 (2010).
45. Cavanagh Sean, E., Malalasekera, W.M.N., Miranda, B., Hunt Laurence, T. & Kennerley Steven, W. Visual fixation patterns during economic choice reflect covert valuation processes that emerge with learning. *Proceedings of the National Academy of Sciences* **116**, 22795-22801 (2019).
46. Rich, E.L. & Wallis, J.D. Decoding subjective decisions from orbitofrontal cortex. *Nature Neuroscience* **19**, 973-980 (2016).
47. Tversky, A. Elimination by aspects: A theory of choice. *Psychological Review* **79**, 281-299 (1972).
48. Busemeyer, J.R., Forsyth, B. & Nozawa, G. Comparisons of elimination by aspects and suppression of aspects choice models based on choice response time. *Journal of Mathematical Psychology* **32**, 341-349 (1988).
49. Achtziger, A., Al, xf & s-Ferrer, C. Fast or Rational? A Response-Times Study of Bayesian Updating. *Management Science* **60**, 923-938 (2014).
50. Bakkour, A.*, et al.* The hippocampus supports deliberation during value-based decisions. *eLife* **8**, e46080 (2019).
51. Shadlen, M.N. & Shohamy, D. Decision Making and Sequential Sampling from Memory. *Neuron* **90**, 927-939 (2016).
52. Wang, X.J. Decision making in recurrent neuronal circuits. *Neuron* **60**, 215-234 (2008).
53. Murawski, C. & Bossaerts, P. How Humans Solve Complex Problems: The Case of the Knapsack Problem. *Scientific Reports* **6**, 34851 (2016).
54. Bossaerts, P. & Murawski, C. Computational Complexity and Human Decision-Making. *Trends in Cognitive Sciences* **21**, 917-929 (2017).
55. Genest, W., Stauffer, W.R. & Schultz, W. Utility functions predict variance and skewness risk preferences in monkeys. *Proc Natl Acad Sci U S A* **113**, 8402-8407 (2016).
56. Simon, H.A. Rational choice and the structure of the environment. *Psychol Rev* **63**, 129-138 (1956).
57. Gigerenzer, G. & Gaissmaier, W. Heuristic Decision Making. *Annual Review of Psychology* **62**, 451-482 (2010).
58. Poggio, T. The Levels of Understanding framework, revised. *Perception* **41**, 1017-1023 (2012).
59. Marr, D. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* (Henry Holt and Company, 1983).

Methods

Animals and experimental setup

All animal procedures were approved by Institutional Animal Care and Use Committee of the University of Pittsburgh. We used two male Rhesus macaque monkeys (Macaca mulatta) for these studies (13.6 and 9.8 kg). In-cage training system (Thomas recording) was attached to the cage and the tasks were presented on a touch screen Samsung tablet. During experiments, free-moving animals interacted with the tablet to complete the trials. Black currant juice was delivered by a tablet-controlled solenoid liquid valve. Custom-made software (Android) running on the tablet controlled the behavioral tasks.

Behavioral training and Knapsack task

We generated stimulus sets containing 13 fractal images; 11 of the fractal images predicted rewards between 0.1 and 0.7 ml, whereas the other two images were used as positive (0.8 ml) and negative (0.0 ml) controls (Figure 1A). We trained freely moving monkeys (n = 2 male rhesus macaques) the predictive value of each item (fractal image) in stimulus set. On each training trial, the animal touched a 'self-initiation' cue, and then one item was pseudorandomly selected from the set and presented on a touchscreen. When the animal touched the screen where the item was presented, a virtual knapsack, pictured above the item, was filled with the 'volume' associated with the item, and the animals were rewarded with that same reward volume (Extended Data Figure 1B). Training for each stimulus set lasted ~ 2 weeks. We used the animals' response times and skipping rates to measure learning progress. For the subsequent behavioral testing, we used one stimulus set for approximately 1 month in the knapsack paradigm. Then we discarded the fractals and trained new ones. For monkey G and B we used two stimulus sets, respectively, over the course of the project

Each knapsack trial began with an initiation cue, and then the touchscreen displayed an 'instance' - a combination of five items that appeared, simultaneously, on the screen (Figure 2A). The binomial coefficient formula indicates that there are 462 five item combinations. The monkeys were limited to 300-400 trials per day, and instances were chosen at random. Thus, it was uncommon that the same instance was repeated on any one day. The animals used the touchscreen to select items one-by-one. Every time an item was selected two things happened, (1) that item was highlighted and (2) the virtual knapsack at the top of the screen filled by an amount equivalent to item's predicted reward. The knapsack limit was set at 0.8 ml. This limit was chosen based on prior studies that demonstrated the reward utility functions were relatively linear around 0.8 ml.[1,2] Optimal performance on every trial was defined as the largest possible sum of rewards less than or equal to the limit (Figure 2A). If the monkey selected a subset of items whose reward magnitude sum was less than optimal, they were rewarded by that lesser amount (Figure 2B, top). If they selected a subset that was greater than the limit, then the trial ended, no reward was delivered, and there was a 4 second timeout (Figure 2B, bottom).

Measurements of difficulty and its relationship with performance and response time

We defined three high-level properties of the instances that could affect the difficulty of trials, in addition to the value of the single fractals. The number of witnesses counts the number of subsets whose sums are below the limit.

$$number\ of\ witnesses = \sum_{S} I(\sum_{x \in S} x \leq limit)$$

where S is a solution subsets of an instance.

We also assessed the optimality of a randomly behaving agent by calculating their average performance for each instance (random score). The random agent only selects solutions whose sums are below the limit:

$$random\ score = \frac{1}{|\Lambda|}\sum_{s\,\in\,\Lambda}\left(\frac{\sum_{x\,\in\,S}x}{opt}\right)$$

where $\Lambda = \{S\colon \sum_{x\,\in\,S}x\ \leq \mathrm{limit}\}$. Thus, random score is bounded by 0 and 1. Furthermore, we quantified how easy it is to achieve a 'good' result by counting the number of subsets whose sums are above 0.6ml juice. Mathematically, the property was defined as:

$$\#\ of\ good\ solutions = \sum_{S} I(\tau_S \geq 0.6)$$

where $\tau_S = \begin{cases}\sum_{x\in S}x\,, & if\ \sum_{x\in S}x \leq limit\\ 0, & if\ \sum_{x\in S}x > limit\end{cases}$. We first regressed instance-wise performance on these instance properties, along with the value of all single fractals, and performed forward-backward variable selections to pick the most parsimonious models. Then, we used variables that survived the model selection process across both animals to predict their response times.

Candidate algorithms

Computer science research has developed efficient algorithms to determine approximately optimal solutions. We considered five different approximate algorithms due to their simplicity and intuitiveness: the greedy algorithm,[3] the Sahni-*k* algorithm,[4] the Johnson-*k* algorithm,[5] the MTGS algorithm,[6] the MTSS-*k* algorithm.[7]

The greedy algorithm picks the largest available item that fits the residual sack each step. The Sahni-*k* algorithm searches among all k-item combinations and fills the residual sack with the greedy algorithm. As the last step, it compares all constructed solutions. In the paper, we considered the Sahni-*2* and Sahni-*3* algorithm. Previous work also showed the behavioral relevance of the Sahni-*k* algorithm in human participants**.**[8] The Johnson-*k* algorithm searches for the best combination among items with value higher than $\frac{limit}{k+1}$ before filling the residual sack with the greedy algorithm. In the paper, we considered the Johnson-*2* and Johnson-*3* algorithm. The MTGS algorithm applies the greedy algorithm n times by considering the entire instance, the subset containing all items in the instance except for the item with the largest value, the subset containing all items in the instance except for the item with the largest value and the second largest value item, and so on. The MTSS-*k* algorithm combines the MTGS algorithm and the Sahni-*k* algorithm. The MTSS-*k* algorithm considered all (k-2)-item combinations and fills the residual sack with the MTGS algorithm. Among these five algorithms, we only used the greedy algorithm, Sahni-*k* algorithm, and the Johsnon-*k* algorithm to classify behavior because the MTGS and MTSS-*k* algorithms are simply variants of the greedy algorithm and the Sahni-*k* algorithms.

Although the dynamic programming algorithm solves the knapsack problem exactly, we did not take it into consideration, since it required memory size way larger than the capacity of human working memory.[8] Moreover, since the animals only experienced the juice reward on a continuum rather than discrete drops, dynamic-programming-based methods are not applicable to our task.

Solution representation and algorithmic analysis

The behavioral or algorithmic solutions can be represented in the following way. For each instance, we constructed an associated undirected graph, where each node of the graph was a potential solution subset. Two nodes are connected if one node can be reached from another by adding or removing a single item and the corresponding edge weight is the value of that item.[8] Hence, the distance between any two solution subsets for an instance is the length of the shortest path between the two associated nodes. Note that the special structure of the undirected graph allows us to quickly compute the distance between two nodes and gives an intuitive interpretation. For any two solutions subsets s_1 and s_2 of the same instance, denote their intersection as *I* and union as *U*. The distance was defined as follows:

$$d_{graph}(s_1, s_2) = \sum_{e \in U - I} e$$

Intuitively, the two similar solution subsets have greater 'overlap' in the set elements. We can also take into account the selection order. Accordingly, each solution was defined as an ordered tuple and padded with zeros. For example, if the monkey selected 0.4 ml, then 0.2 ml, then 0.1 ml and then stopped. The solution was defined as (0.4, 0.2, 0.1, 0, 0). Thus, the distance between two solutions p and q can be characterized by the L1 distance.

$$d_{l_1}(p, q) = \sum_{i=1}^{5} |p_i - q_i|$$

In order to characterize animals' algorithmic behaviors, we matched their solution on each trial to one of the candidate algorithms. For each trial, we first determined which algorithms produced the most similar subsets to the behavioral response using d_{graph}. Among those algorithms, we broke the tie by taking into account the selection order and comparing d_{l_1} between the behavioral and algorithmic solutions. To minimize the possibility that all five algorithms match the behavioral response poorly, we constructed a null distribution for each trial by calculating the L1 distance between all possible solutions (5! = 120 in total) and the behavioral response (Figure 2a). The best-match algorithm according to the previous procedure has to have a smaller L1 distance to the behavioral solution than the lower 5% threshold of the null distribution. Trials where this criterion was not met was discarded in the deliberation time analysis (9.3% of all trials for animal G and 18.2% of all trials for animal B). After trial classification, we grouped the Sahni-*2*, Sahni-*3* as Sahni-*k* trials and Johnson-*2* and Johnson-*3* trials as Johnson-*k* trials. As a result, we obtained four trial types: greedy, Sahni-*k*, Johnson-*k* and unclassified.

Deliberation time analysis

Deliberation time was defined to be the time between the appearance of the fractals and animals' first touch. Behavioral variability allowed us to compare deliberation time between the greedy trials and the combinatorial trials within the same instance. For instances where the animals exhibited both greedy and combinatorial behaviors (each class must have at least three trials), we calculated the difference between the average deliberation times of the corresponding trial types and tested whether this difference was significantly different from zero across available instances. Among combinatorial trials, we also explored whether deliberation time was modulated by the size of search space. In our task, the search space of the Sahni-*2* and Sahni-*3* algorithms are of the same size: 5 choose 2 = 5 choose 3 = 10 and stay constant across instances. On the other hand, Johnson-*2* and Johnson-*3* algorithms revert to a near exhaustive search when the minimum value of an instance is above 0.27ml because all items would have

higher value than the preset thresholds (0.266 for Johnson-*2* and 0.2 for Johnson-*3*). In these instances, the Johnson-*2* and Johnson-*3* have to search at least 25 subsets (exhaustive search goes over all 31 subsets). Accordingly, we would expect the animals to take the most time deliberating for Johnson-*2* and Johnson-*3* trials where the algorithms revert to a near exhaustive search.

One prediction of the longer deliberation times observed in the combinatorial trials is that the animals should spend less time thinking about the subsequent selections (i.e. selections after the first one). To confirm this prediction, on each instance, we calculated the difference between the average selection times across the subsequent selection intervals of the corresponding trial types. This analysis confined us to consider only trials where the animals chose more than one item.

Neural networks configurations and training

We trained feedforward neural networks to mimic the greedy algorithm, the Sahni-*2* algorithm, the Sahni-*3* algorithm, the Johnson-*2* algorithm and the Johnson-*3* algorithm individually. For each algorithm, we provided one network with all permutations of all 462 instances (55440 permutations in total) and their corresponding solution subsets as targets. The provided targets are in 0s and 1s – 1 if the associated item is chosen by the algorithm and 0 if not. The networks all have in total six layers, including one input layer and one output layer both with five units. Sigmoid function was used as the activation function for the output units. There are four hidden layers with 100 units, 60 units, 60 units and 50 units. RELU function was used as the activation function for the hidden units. For each permutation, we defined the metric to be the indicator function that compares the solution subset produced by the network and the target.

$$I(|solution\ subsets\ \triangle\ target| = 0)$$

where $A \triangle B = (A \cup B) \setminus (A \cap B)$. Networks were initialized randomly, and maximum training epochs were set to be 500. For single unit lesions and dimensionality analysis, we used 20% held-out data to facilitate early stopping in training the networks. On the other hand, to properly assess whether networks mimic their target algorithms but not the other candidate algorithms, we performed ten-fold cross validation without using held-out data to guide early stopping.

Behavioral similarity scores

The similarity between the neural networks and the animals was assessed on an instance-level. For each instance, we collected around 30 trials from each animal, while the neural networks were trained using all permutations (120). To address this mismatch, we randomly sampled solutions generated by the networks for each instance to match the number of behavioral trials and constructed the complete bipartite graph consisting of network solution vertices N and behavioral solution vertices B. The weight of each edge is the graph distance, as defined in the previous section, between the corresponding vertex in N and vertex in B. The similarity between the networks and the behavioral solutions was defined as follows. We first solved the minimum weight bipartite matching problem using the Hungarian algorithm between N and B.[9] The similarity between the networks and the behavioral solutions is one minus the minimal total weight. For each instance, we randomly sampled ten sets of network solutions to calculate the similarity measure.

Single unit lesions

We lesioned individual units by setting the its connection weights to zero. We monitored the lesion impact for each instance and either averaged across all instances for each unit or averaged across all units for each instance.

Dimensionality of neural networks

To assess the dimensionality of neural networks trained to mimic the greedy algorithm and the Sahni-*2* algorithm respectively, we performed principal component analysis on the network activity matrix, where each row corresponds to the hidden unit firing pattern for one permutation and each column corresponds to the single hidden unit firing pattern for all permutations. We calculated the percent of variance explained using only one principal component to fifty principal components. To test whether the high dimensionality of Sahni-*2* network comes from being required to produce different solutions than the greedy network, we considered a subset of permutations where the greedy algorithm and the Sahni-*2* algorithm produced the exact same solution subsets and repeated the analysis.

Decoding analysis for algorithm specific representations

To investigate when the algorithm-specific representations emerge, we used individual hidden layers to generate target solutions. After training a neural network, we projected individual hidden layers linearly to predict individual target outputs (logistic regression models with L1 penalty). If a hidden layer contained information needed to mimic the target algorithm of the corresponding neural network, the cross-validated performance of the linear classifiers should be high (the performances of the linear classifiers were evaluated jointly for all outputs). For each candidate algorithm, we trained ten neural networks with random initializations.

Data and code availability

The data that support the findings of this study are available from the corresponding author upon request. The code used to analyze these data are available from the corresponding author upon request.

Methods references

1 Stauffer, W. R., Lak, A. & Schultz, W. Dopamine reward prediction error responses reflect marginal utility. *Curr Biol* **24**, 2491-2500 (2014).

2 Genest, W., Stauffer, W. R. & Schultz, W. Utility functions predict variance and skewness risk preferences in monkeys. *Proc Natl Acad Sci U S A* **113**, 8402-8407 (2016).

3 Dantzig, G. B. Discrete-Variable Extremum Problems. *Operations Research* **5**, 266-277 (1957).

4 Horowitz, E. & Sahni, S. Computing partitions with applications to the knapsack problem. *J. ACM* **21**, 277-292, (1974).

5 Johnson, D. S. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences* **9**, 256-278 (1974).

6 Lai, T.-C. Worst-case analysis of greedy algorithms for the unbounded knapsack, subset-sum and partition problems. *Operations Research Letters* **14**, 215-220 (1993).

7 Fischetti, M. Worst-Case Analysis of an Approximation Scheme for the Subset-Sum Problem. *Oper. Res. Lett.* **5**, 283–284 (1986).

8 Murawski, C. & Bossaerts, P. How Humans Solve Complex Problems: The Case of the Knapsack Problem. *Scientific Reports* **6**, 34851 (2016).
9 Kuhn, H. W. The Hungarian method for the assignment problem. *Naval research logistics quarterly* **2**, 83-97 (1955).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- S1.mov
- S2.mov
- S3.mov
- S4.mov