Decompose, Deduce, and Dispose: A Memory-Limited Metacognitive Model of Human Problem Solving

Samuel Cheyette[†,1], **Tony Chen**[†,1], **Matthias Hofer**[†,1], **Frederick Callaway**[2], **Neil Bramley**[3], **Joshua Tenenbaum**[1]

[1]Massachusetts Institute of Technology, [2]New York University, [3]University of Edinburgh

[†]*These authors contributed equally to this work.*

{cheyette, thc, mhofer}@mit.edu

Abstract

Many real-world problems are defined by complex systems of interlocking constraints. How people are able to solve these problems with such limited working memory capacity remains poorly understood. We propose a formal model of human problem-solving that uses metacognitive knowledge of its own memory limits and imperfect reasoning to guide subproblem choice. We compare our model to human gameplay in two experiments using a variant of the classic game Minesweeper. In Experiment 1, we find that participants' accuracy was influenced both by the order of subproblems and their ability to externalize intermediate results, indicative of a memory bottleneck in reasoning. In Experiment 2, we used a mouse-tracking paradigm to assess participants' subproblem choice and time allocation. The model captures key patterns of subproblem ordering, error, and time allocation. Our results point toward memory limits and strategies for navigating those limits — including the careful choice of subproblems and memory-offloading — as central elements of human problem-solving.

Keywords: problem solving, planning, logical reasoning, working memory, resource rationality, metacognition

Introduction

Problem-solving in complex domains—whether designing a scientific experiment, diagnosing a disease, or planning a software architecture—can require reasoning through a vast space of possible actions and outcomes. In many cases, the sheer number of possibilities exceeds the limits of what a reasoner can explicitly consider at any given time. Scientists, for instance, must decide which experiments to run without exhaustively simulating all possible outcomes. Programmers must plan software systems without tracking every subroutine. Chess players must navigate ever-branching game trees without computing all possible positions. In each case, the challenge is not just uncertainty about the external world but an internal constraint: the limits of how much information can be actively represented and reasoned over in the mind.

Classical theories of problem-solving, particularly the seminal work of Newell and Simon (1972) have provided a foundational framework for how humans and machines navigate large problem spaces. These models assume that problem-solving proceeds via explicit search over a structured state space, simulating sequences of actions to find those most likely to lead to desirable outcomes. Research in both artificial intelligence and psychology has thus focused on identifying strategies to limit the combinatorial explosion in possible future states, for instance by limiting planning to a few steps ahead (Keramati et al., 2016; Snider et al., 2015), limiting search to the most promising directions (Huys et al., 2012; Sezener, Dezfouli, & Keramati, 2019; van Opheusden et al., 2023), or terminating search when one plan appears much better than any other (Solway & Botvinick, 2015; Zhang, Lipovetzky, & Kemp, 2023).

However, these models typically focus on making efficient use of *computational* resources, and sidestep the memory demands required to represent complex problems, for example by positing that people represent search trees with hundreds of nodes (van Opheusden et al., 2023). Even models that incorporate structured representations and heuristics to decompose problems and manage complexity (Anderson, 2013) require tracking large-scale state representations, such as the entire state of a chessboard across different search trajectories (Newell & Simon, 1972). Yet research in visual object-tracking, numerosity perception, and verbal memory (to name a few) suggest that people can only represent between several to tens of bits at a time (Garner, 1953; Miller, 1956; Sims, 2016; Sims, Jacobs, & Knill, 2012). This gap between idealized problem-solving models and real cognitive constraints raises a key question: how do people structure their reasoning to remain within working memory limits while solving complex problems?

Here we develop an algorithmic-level model of human problem-solving that foregrounds the issue of limited representational capacity, drawing inspiration from classic problem-solving approaches (Newell, Shaw, & Simon, 1959; Newell & Simon, 1972; Anderson, 1993) as well as more recent resource-rational approaches to planning that account for time and memory complexity (Ho et al., 2022; Ho, Cohen, & Griffiths, 2023; Callaway et al., 2022). Our hypothesis is that people manage memory constraints by dynamically structuring their problem-solving process, constructing subproblems *incrementally* and *greedily*, aiming to keep them within their memory limits and moving on to a new subproblem when the complexity becomes unmanageable. Crucially, our model does not assume an abstract cost function for reasoning complexity, as is common in resource-rational models (Callaway et al., 2022; Ho et al., 2022; Bhui, Lai, & Gershman, 2021). Instead, it posits that people use *metacognitive awareness* of a concrete capacity constraint to manage the size and structure of their problem representations.

We test the model in two experiments using the classic game, Minesweeper, where players deduce the locations of mines based on numerical constraints embedded in a grid. The constraints in Minesweeper form a web of interde-

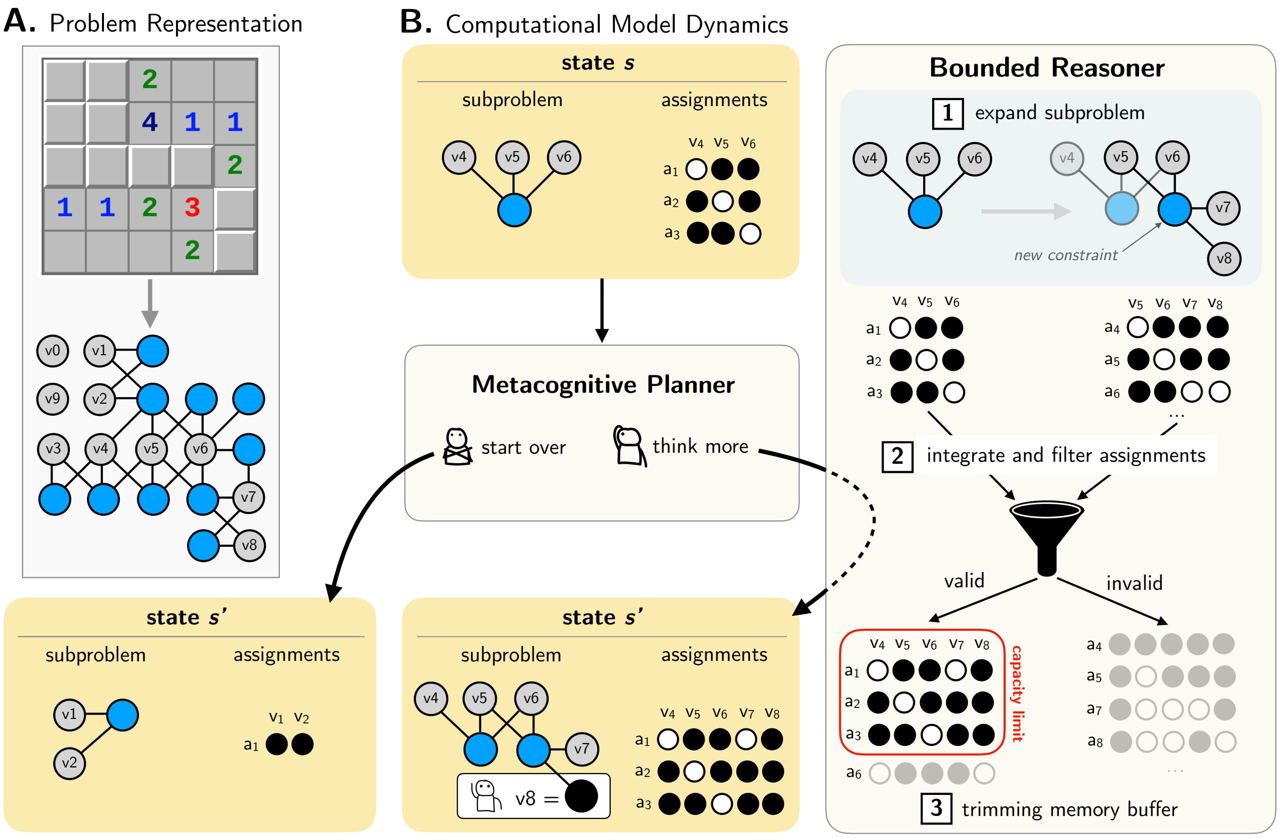

Figure 1: Overview of our capacity-bounded problem-solving model. **(A.)** We represent Minesweeper boards as graphs with variables (grey) and constraints (blue). **(B.)** The model solves problems incrementally by expanding local subproblems. Each state includes a subproblem and a capacity-limited memory buffer tracking hypotheses about variable assignments. A **metacognitive planner** decides whether to continue reasoning or to flush memory and start over by resampling a new constraint. If continuing, a **bounded reasoner** (1) expands the subproblem, (2) integrates and filters assignments, and (3) trims the memory buffer by probabilistically culling assignments that exceed its capacity limit.

pendencies, where deductions made in one region can influence reasoning in another, potentially far away, region. Experiment 1 tests how memory load affects the difficulty of Minesweeper problems. We presented participants with Minesweeper boards in which they had to solve squares in a predetermined order. We manipulated working memory demands in two ways: by restricting access to previously solved squares; and by varying subproblem order such that one order had higher memory demands than the other. As predicted, both manipulations modulated participants' solution rates, validating core assumptions of the model. In Experiment 2, we allowed participants to freely navigate the problem space while tracking their mouse movements. We found that the model captured the order in which people solved subproblems, their time allocation to different parts of the board, and the time and location of errors. Our experimental results highlight memory capacity as an important bottleneck in human reasoning, and our model — which uses an explicit representation of its own memory limits to guide subproblem choice — provides an account of problem decomposition as a rational solution to capacity-limited reasoning.

Model

The overall structure of our model is shown in Figure 1. We adopt the formalism of Constraint Satisfaction Problems (CSP), a well-established framework for representing problems in logical reasoning, deduction, and discrete optimization. Extensive research in AI has focused on developing efficient algorithms and heuristics for solving CSPs (Brailsford, Potts, & Smith, 1999; Russell & Norvig, 2016).

A CSP consists of a graph of two types of nodes: variables $v \in V$ and constraints $c \in C$. For our purposes here, variables take binary values $v \in \{0,1\}$, while constraints enforce relationships between them, expressed as boolean functions $c : \mathrm{Dom}(c) \to \{0,1\}$. The domain of a constraint, Dom, is the set of all possible assignments to the variables it affects. An assignment is a mapping from a set of variables to values, e.g., $(v_1, v_2, v_3) \mapsto (0,1,0)$. Naive CSP solvers evaluate all possible assignments $2^{|V|}$ to find those satisfying all constraints. However, this is computationally expensive and requires holding too many items in memory at once.

Instead, our model operates by iteratively making progress on tractable subproblems. Our approach — inspired by sequential Monte Carlo models of resource-bounded reasoning

(Sanborn, Griffiths, & Navarro, 2010) as well as production rule systems from the problem solving literature (Anderson, Matessa, & Lebiere, 1997; Newell, 1992) — consists of several interacting components: a *bounded reasoner* that noisily updates beliefs about variables and a *metacognitive planner* that chooses what to think about next by simulating constraint evaluations using the bounded reasoner.

Bounded Reasoner. At each step, the bounded reasoner's state s (see yellow rectangles in Figure 1B) includes a subproblem, defined by a subset of variables $V_s \subseteq V$ and constraints $C_s \subseteq C$, and a memory buffer, which tracks possible satisfying assignments A_s for these variables, obtained from reasoning about the constraints. The reasoner also tracks the total runtime t_s, and the *capacity overflow* $C_{o,s}$, which quantifies the amount of information lost thus far due to capacity limits. When a new constraint c is selected for integration (see "Metacognitive Planner" below), the memory buffer is updated through a three-stage process (see bounded reasoner box in Figure 1B): (1) expanding the subproblem by incorporating new variables from the domain of c, (2) combining assignments and filtering by removing assignments inconsistent with $C_s \cup \{c\}$ and (3) culling by trimming the buffer to fit within memory.

After the expansion and filtering, we are left with a new set of assignments A^* that are consistent with the new integrated constraint set containing variables V^*, but may be too large to fit into the memory buffer. We assume that the model's memory buffer information capacity is fixed to a constant L, above which information will likely be lost. $|A^*|$ assignments of $|V^*|$ variables require $\log_2 \binom{2^{|V^*|}}{|A^*|}$ bits to represent; when this complexity exceeds the capacity limit L, we probabilistically cull the set of proposed assignments by dropping each assignment with probability p_L. p_L is chosen to be the minimum probability such that memory capacity is not exceeded:

$$p_L = \min_{0 \le p \le 1} \left[\log_2 \left(\frac{2^{|V^*|}}{(1-p) \cdot |A^*|} \right) \le L \right].$$

The bounded reasoner's state is thus updated as $V_{s'} \leftarrow V^*$, $C_{s'} \leftarrow C_s \cup \{c\}$, and $A_{s'} \leftarrow \{a \in A_s | \texttt{flip}(p_L) = 1\}$.

Notably, this mechanism introduces errors and stochasticity into reasoning, making it possible to arrive at false deductions, simply because the model accidentally forgot about other valid assignments. We assume a simple model of runtime in which the time it takes to integrate the constraint is proportional to the product of the current assignments and the satisfying assignments of the new constraint, so that $t_{s'} \leftarrow t_s + O(|A_s| \cdot |A_c|)$, where A_c denotes the satisfying assignments of the new constraint.

Finally, to account for its own limited memory, the model keeps track of the amount of information it has lost in reasoning (or how much it would likely lose if it integrated a particular new constraint). This is measured as the loss in entropy about possible satisfying assignments from forgetting, which we denote the *capacity overflow* $C_{o,s'} \leftarrow C_{o,s'} + \log_2(|A_{s'}|) - \log_2(|A^*|)$.

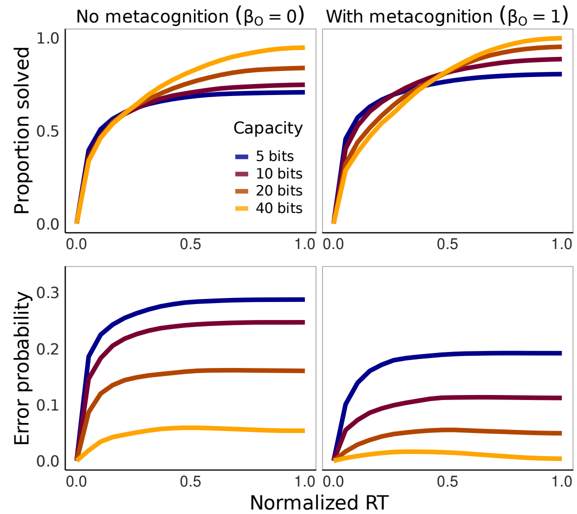

Figure 2: Predictions of the model under different memory capacity limits, showing the proportion of the board solved (top), and the average number of errors (bottom) as a function of runtime, split by β_o. Higher capacity limits allow it to solve more complex subproblems successfully. When $\beta_O = 0$, the model is over-optimistic about its reasoning; when $\beta_O = 1$, the model tries to solve problems within its capacity.

Metacognitive Planner. The metacognitive planner leverages this explicit representation of its own reasoning to guide behavior. At each timestep, it considers two broad classes of actions: integrating a new constraint c, which we label as a_c or flushing the memory buffer and restarting, which we label as $a_\emptyset$. It evaluates potential actions according to their expected value, marginalizing over possible outcomes s' of the action a. The value of integrating a new constraint into its current subproblem is:

$$Q(a_c|s) = E_{s' \sim p(s'|a_c,s)} \left[IG(s,s') - \beta_O [C_{o,s'} - C_{o,s}] - \beta_T [t_{s'} - t_s] \right].$$

$IG(s,s')$ represents the information gain, which we take to be the difference in joint entropy of the two states: $IG(s,s') = \log_2(|A_s|) - \log_2(|A_{s'}|)$, along with any deductions that were made. β_O penalizes the information loss (forgetting valid satisfying assignments), so that higher values of β_O discourage integrating constraints that might exceed the capacity limits of the reasoner. Finally, β_T penalizes the time it took to integrate that constraint. The value of resetting and starting over is simply 0: $Q(a_\emptyset|s) = 0$.

We evaluate this expectation by Monte Carlo integration—drawing N random samples and averaging the results. By tracking both what it knows (solved variables) and what it might have forgotten (discarded assignments), the model can estimate its own likelihood of making errors and adapt its strategy accordingly. At each step, the probability that the model chooses each action is given by softmax choice rule (Luce, 1959), $p(a|s) \propto \exp(Q(a|s)/\tau)$.

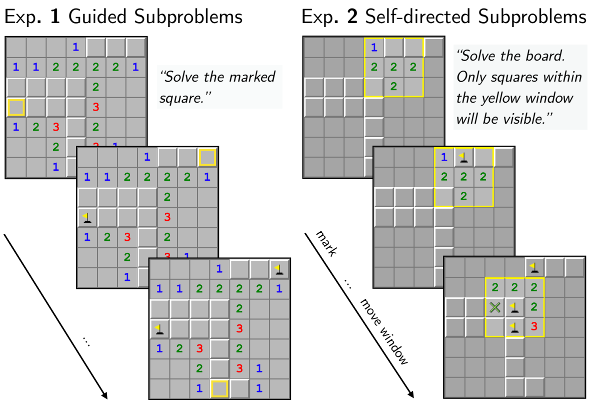

Figure 3: Left: Experiment 1. Participants were shown Minesweeper boards and given squares to solve in a predetermined order. Right: Experiment 2. Participants played an self directed game. They could only see the squares within a 3x3 window that they could move with their mouse.

Figure 2 shows results of simulating the model on the 12 Minesweeper boards used in the experiments under different memory capacity limits and information loss penalties (β_O). As the capacity limit increases, the model can integrate more information without losing track of valid possible assignments, so the proportion of variables it correctly solves correspondingly increases. When it does not account for its own forgetting ($\beta_O = 0$), it is essentially over-optimistic about its reasoning and makes substantially more errors than when its memory errors are penalized ($\beta_O = 1$).

Experiments and Results

Experiment 1: Guided Subproblem Solving

Our account is premised on the idea that memory demands are a significant bottleneck in people's ability to solve hard problems, but that these demands can be mitigated by memory-offloading via externalization and by carefully choosing the order of subproblems. We used two manipulations designed to modulate memory load and, accordingly, subproblem difficulty at different stages of the game. First, we introduce a manipulation in which participants are either allowed ("Regular") or not allowed ("No Marking") to see previously solved variables on the board. Second, we manipulate the order of variables on the board that participants have to solve, with one order designed to have low incremental memory demands over the course of the game ("Curriculum Order") and the other designed to have higher average memory complexity ("Unhelpful Order").

We predict that participants should initially perform equally well in the Regular and No Marking conditions, but should diverge at later stages in solving a board, as those in the No Marking condition do not have the benefit of offloading memory burden onto the environment. We also predict that participants in the Curriculum Order condition should start off with high accuracy and, if they have the benefit of marking (Regular condition), should maintain high accuracy throughout the game. On the other hand, participants in the Unhelpful Order should start off with low accuracy, though they should improve over the course of the game if they are able to offload prior solutions.

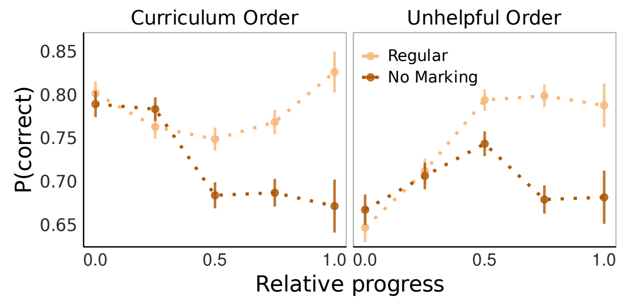

Figure 4: Results from Experiment 1, showing binned accuracy as a function of progress in the game, grouped by whether participants were shown the results of previously-attempted squares. The Curriculum Order trials are shown on the left and the Unhelpful Order is shown on the right.

Stimuli Participants played on 12 boards that were generated with a logic-based solver of Minesweeper. Boards were always 7x7 cells with 10 hidden mines and partially solved—most of the squares were revealed and only between 12 and 19 squares were unrevealed, left to be solved by the subject. Boards were generated with the constraint that they had a unique solution and varied in difficulty and complexity.

Procedure We recruited 114 participants on Prolific, 11 of whom were excluded for getting less than 50% correct across games or for having average response times of less than 1 second. Participants were briefed on the game of Minesweeper, randomly assigned to either the Regular or No Marking condition, and asked to play all 12 boards in random order—half of which, chosen at random, needed to be played in the Curriculum Order, and the other half in the more difficult Unhelpful Order. At each step, a square on the board would be highlighted, and the participant was told to identify whether that square contained a mine or not. The correct answer would then be revealed, and the participant would be queried for another square. The trial ended after the participant had given a response for every unknown square. Participants could win a point for each correct answer, and lose a point for an incorrect answers, and received \$0.02 performance reward per bonus point they collected in this way at the end of the game.

Results Figure 4 shows participants' accuracy as a function of progress in the game in each condition. The results confirmed our main hypotheses: participants started off with higher accuracy in the Curriculum Order condition and only maintained (or increased) in accuracy in either ordering if they could externalize intermediate results. We ran a mixed-effects logistic regression predicting accuracy from relative progress, difficulty, and condition, with random subject and stimulus effects. Participants in the *Unhelpful Order* condition initially had significantly lower accuracy compared to the *Curriculum Order* condition ($\beta = -0.77$, $p < .001$), but their

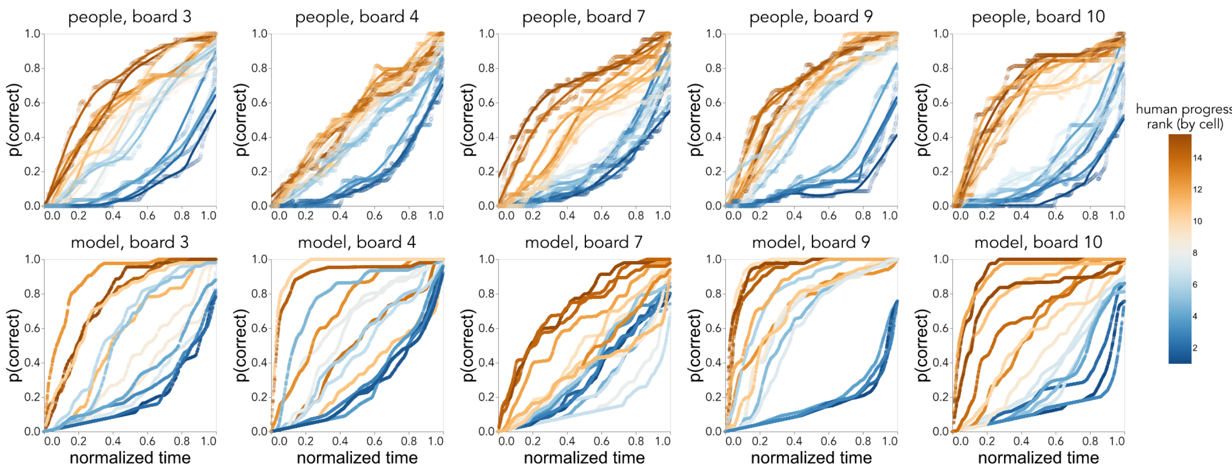

Figure 5: The probability of a square being marked correctly by humans (top) and our model (bottom) at a given point in time, for five out of the twelve boards we test in Experiment 2. Each curve is colored by the rank of human accuracy on that square halfway through the game.

accuracy improved over time ($\beta = 1.24$, $p < .001$). Externalization (*No Marking* condition) did not significantly impact initial accuracy ($\beta = 0.16$, $p = .36$), but it interacted with relative progress ($\beta = -0.73$, $p < .001$), indicating that participants in the No Marking condition declined in accuracy as the game continued. None of the other main effects or interactions were significant ($ps > 0.2$).

Experiment 2: Self-directed Subproblem Selection

While Experiment 1 established that memory constraints significantly impact problem-solving and that structuring subproblems in memory-efficient ways improves accuracy, it leaves open the question of *how* people choose subproblems. We designed a free-play variant of our task (see Figure 3, right), where participants could determine their own solution order, unconstrained by predetermined subproblem sequences. We used a process-tracing paradigm inspired by prior work on planning and problem-solving strategies (Jain et al., 2023; Callaway et al., 2022), in which numbered squares were only visible within a 3×3 spotlight window. To see other parts of the board, participants had to move the window. This allowed us to collect fine-grained data on where participants directed their attention, which subproblems they engaged with, and when they abandoned or restructured their problem-solving strategy.

Procedure We recruited 55 participants on Prolific. Participants were briefed on the game and the spotlight mechanic, and then proceeded to play the same twelve boards used in Experiment 1, with the same performance bonus mechanics. We excluded all trials for which participants performed below chance, which in our boards is equal to the number of mines divided by the number of unmarked squares.

Results Overall accuracy was high: participants achieved an average board accuracy of 0.92 on each trial, and took on average 141 seconds to finish each board. To ensure the dynamics of human and model marking are comparable, we normalize trial runtimes across human and model runs to lie between 0 and 1. Using the PyBADS package (Singh & Acerbi, 2024; Acerbi & Ma, 2017), we fit our model to the joint probability of marking a square correctly at a given time step. Letting $0 < t_1 < t_2 < \ldots < t_N \leq 1$ denote the times of actions taken by participants, and letting $m_k(t), c_k(t)$ be binary variables that denote whether square k is marked and marked correctly, respectively, at time t, the likelihood of the parameters θ is a product of Bernoulli probabilities: $L(\theta) = \prod_i \prod_k P(c_k(t_i)|m_k(t_i),\theta)P(m_k(t_i)|\theta)$. To account for the effects of guessing, we included a noise parameter ε, which we fit in addition to the other parameters.

For the full model, the best fitting parameters were a capacity limit of $L = 13.8$ bits, metacognitive and time weights of $\beta_O = 0.71, \beta_T = 0.65$, and a softmax temperature of $\tau = 0.71$. In addition to our `full` model, we consider three ablations. The `unbounded` model has infinite memory, the `no rt` model fixes $\beta_T = 0$, and the `no metacognition` model fixes $\beta_O = 0$. We report the cross validated log likelihoods for each model, obtained by splitting the dataset in half, fitting parameters on one half, and evaluating the likelihood on the held out data. The `full` model performs the best, followed by the `no rt`, `no metacognition`, and `unbounded memory` model (see Table 1).

Model	ΔLL_{CV}
Full Model	0
No RT ($\beta_T = 0$)	-2,331
No Metacognition ($\beta_O = 0$)	-9,915
Unbounded Memory ($L = \infty$)	-30,544

Table 1: The change in cross validated log likelihood for Experiment 2 when ablating model parameters.

Figure 5 shows the probability that people (top) and the model (bottom) marked squares correctly over time, with each curve representing a different variable. The color cor-

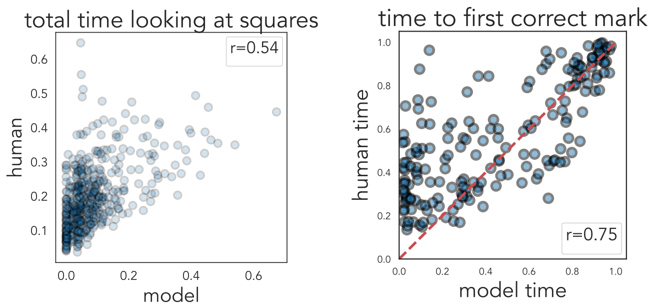

Figure 6: Left: the total amount of time spent considering a square. For humans, this was measured as the total time a square was highlighted by the spotlight window. For the model, this was the total amount of time that the square was held in the model's subproblem context. Right: time to first correct mark, defined as the smallest time for which the probability of marking the square correctly exceeds 0.5.

responds to the square's rank probability of being correctly solved by humans in the middle of the game ($t = 0.5$). The figure highlights that the model captures the dynamics of human marking well over time: cells people solved correctly later in the game were also ones that the model found difficult or did not attempt until late. We find a high average alignment between the model and people in the average number of squares marked ($r = 0.99$), number of correctly solved squares ($r = 0.98$), and average number of errors ($r = 0.87$) at a given point in time. We also see alignment at the level of individual squares, specifically in the probability of marking ($r = 0.77$), marking correctly ($r = 0.78$), and making an error ($r = 0.49$) at a given time point for a given square.

Finally, our model also captures the dynamics of participants' time allocation and attention. We compute the total average time that a participant's window spotlight illuminated each square; for the model, we computed the total amount of time that a constraint or variable was active in the current subproblem context. We find a correlation of ($r = 0.54$) for the total time the model and humans consider a square. We also find a correlation of ($r = 0.75$) for the time to first correct mark, defined as the smallest time for which the probability of marking the square correctly exceeds 0.5.

Discussion and Future Work

In this paper, we asked how people learn and reason in settings that involve various interlocking constraints given a limited ability to keep track of more than a few bits of information at a time. We developed a model that solves constraint satisfaction problems under a fixed memory budget, using metacognitive awareness of its own limits and imperfect reasoning to decide whether to consider a new piece of information or to "give up" on a current reasoning path and start on a new one. In Experiment 1, we found that participants performed significantly better when problem sequences were structured to reduce memory load. Their ability to solve hard problems was influenced both by the order of subproblems and their ability to externalize solutions. In Experiment 2, in which participants played freely, the model closely matched human time allocation and error patterns. Ablating model parameters significantly worsened the fit on a held-out dataset, indicating that memory limits and metacognition about those limits were both necessary to capture human performance.

Our findings add to a long line of research demonstrating that problem decomposition is a core strategy in human reasoning, spanning domains from navigation (Balaguer et al., 2016; Wiener & Mallot, 2003; Gordon, Chuang, & Pezzulo, 2025) to problem solving (Ward & Allport, 1997; Cooper & Shallice, 2006). By providing a concrete implementation of subproblem selection in a logical reasoning task, our model extends previous resource-rational accounts (Correa et al., 2023; Binder et al., 2023; Ho et al., 2022), which focus on the tradeoff between planning cost and subgoal complexity but do not provide a process-level account of subproblem choice. Our findings suggest that people implicitly optimize this tradeoff by choosing subproblems to stay within cognitive limits. An implication of our approach is that strategies may be dynamically shaped by the interaction between internal cognitive constraints and external task structure. This adaptability highlights a distinction between descriptive accounts of problem decomposition and mechanistic models that specify how subgoals are selected in real time.

The measure of "capacity overflow" that the model tracks can be understood as a particular instantiation of a metacognitive confidence judgment (Fleming, 2024). Critically, confidence judgments can function not only as a retrospective assessment of accuracy but also as a prospective estimate of future success — in our case, in reasoning. On our account, people decide when to continue integrating evidence versus when to give up on a reasoning path based on how many valid possibilities they are missing. By monitoring information lost in reasoning, then, problem-solvers are able to anticipate potential failures based on perceived memory deficiencies and adjust what they think about accordingly. While we assumed here that people have a perfect estimate of the information lost in reasoning, people's estimates of their own certainty are likely subject to noise and bias themselves (Li & Ma, 2020).

A limitation of the model as formulated in this paper is that it is explicitly designed for a "perfect information" setting—uncertainty derives solely from its inability to represent all available information. However, this need not be the case: the information lost in reasoning is simply realized in terms of subjective entropy about the true state of the environment. Since the model updates its reasoning process based on the expected impact of information gain and information loss, it can handle probabilistic evidence obtained from the environment analogously. In this way, our model can actually be viewed as a unifying framework for incorporating *deductive* and *inductive* inference, with imperfect information deriving either from noise in memory or the environment. An important direction for future work will be to actually test this account in settings with probabilistic evidence.

References

Acerbi, L., & Ma, W. J. (2017). Practical bayesian optimization for model fitting with bayesian adaptive direct search. *Advances in neural information processing systems*, *30*.

Anderson, J. R. (1993). Problem solving and learning. *American psychologist*, *48*(1), 35.

Anderson, J. R. (2013). *The adaptive character of thought*. Psychology Press.

Anderson, J. R., Matessa, M., & Lebiere, C. (1997). Act-r: A theory of higher level cognition and its relation to visual attention. *Human–Computer Interaction*, *12*(4), 439–462.

Balaguer, J., Spiers, H., Hassabis, D., & Summerfield, C. (2016). Neural mechanisms of hierarchical planning in a virtual subway network. *Neuron*, *90*(4), 893–903.

Bhui, R., Lai, L., & Gershman, S. J. (2021). Resource-rational decision making. *Current Opinion in Behavioral Sciences*, *41*, 15–21.

Binder, F. J., Mattar, M. G., Kirsh, D., & Fan, J. E. (2023). Humans choose visual subgoals to reduce cognitive cost. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 45).

Brailsford, S. C., Potts, C. N., & Smith, B. M. (1999). Constraint satisfaction problems: Algorithms and applications. *European journal of operational research*, *119*(3), 557–581.

Callaway, F., van Opheusden, B., Gul, S., Das, P., Krueger, P. M., Griffiths, T. L., & Lieder, F. (2022). Rational use of cognitive resources in human planning. *Nature Human Behaviour*, *6*(8), 1112–1125.

Cooper, R. P., & Shallice, T. (2006). Hierarchical schemas and goals in the control of sequential behavior. *Psychological Review*.

Correa, C. G., Ho, M. K., Callaway, F., Daw, N. D., & Griffiths, T. L. (2023). Humans decompose tasks by trading off utility and computational cost. *PLOS Computational Biology*, *19*(6), e1011087.

Fleming, S. M. (2024). Metacognition and confidence: A review and synthesis. *Annual Review of Psychology*, *75*(1), 241–268.

Garner, W. R. (1953). An informational analysis of absolute judgments of loudness. *Journal of experimental psychology*, *46*(5), 373.

Gordon, J. R., Chuang, J., & Pezzulo, G. (2025). Gaze dynamics prior to navigation support hierarchical planning. *bioRxiv*, 2025–01.

Ho, M. K., Abel, D., Correa, C. G., Littman, M. L., Cohen, J. D., & Griffiths, T. L. (2022). People construct simplified mental representations to plan. *Nature*, *606*(7912), 129–136.

Ho, M. K., Cohen, J. D., & Griffiths, T. L. (2023). Rational simplification and rigidity in human planning. *PsyArXiv*.

Huys, Q. J., Eshel, N., O'Nions, E., Sheridan, L., Dayan, P., & Roiser, J. P. (2012). Bonsai trees in your head: how the pavlovian system sculpts goal-directed choices by pruning decision trees. *PLoS computational biology*, *8*(3), e1002410.

Jain, Y. R., Callaway, F., Griffiths, T. L., Dayan, P., He, R., Krueger, P. M., & Lieder, F. (2023). A computational process-tracing method for measuring people's planning strategies and how they change over time. *Behavior Research Methods*, *55*(4), 2037–2079.

Keramati, M., Smittenaar, P., Dolan, R. J., & Dayan, P. (2016). Adaptive integration of habits into depth-limited planning defines a habitual-goal–directed spectrum. *Proceedings of the National Academy of Sciences*, *113*(45), 12868–12873.

Li, H.-H., & Ma, W. J. (2020). Confidence reports in decision-making with multiple alternatives violate the bayesian confidence hypothesis. *Nature communications*, *11*(1), 2004.

Luce, R. D. (1959). *Individual choice behavior* (Vol. 4). Wiley New York.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, *63*(2), 81.

Newell, A. (1992). Soar as a unified theory of cognition: Issues and explanations. *Behavioral and Brain Sciences*, *15*(3), 464–492.

Newell, A., Shaw, J. C., & Simon, H. A. (1959). Report on a general problem solving program. In *Ifip congress* (Vol. 256, p. 64).

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Prentice-Hall.

Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: a modern approach*. Pearson.

Sanborn, A. N., Griffiths, T. L., & Navarro, D. J. (2010). Rational approximations to rational models: Alternative algorithms for category learning. *Psychological review*, *117*(4), 1144–1144.

Sezener, C. E., Dezfouli, A., & Keramati, M. (2019). Optimizing the depth and the direction of prospective planning using information values. *PLOS Computational Biology*, *15*(3), 1–21.

Sims, C. R. (2016). Rate–distortion theory and human perception. *Cognition*, *152*, 181–198.

Sims, C. R., Jacobs, R. A., & Knill, D. C. (2012). An ideal observer analysis of visual working memory. *Psychological review*, *119*(4), 807.

Singh, G. S., & Acerbi, L. (2024). Pybads: Fast and robust black-box optimization in python. *Journal of Open Source Software*, *9*(94), 5694. Retrieved from `https://doi.org/10.21105/joss.05694` doi: 10.21105/joss.05694

Snider, J., Lee, D., Poizner, H., & Gepshtein, S. (2015, September). Prospective Optimization with Limited Resources. *PLOS Computational Biology*, *11*(9), e1004501.

Solway, A., & Botvinick, M. M. (2015). Evidence integration in model-based tree search. *Proceedings of the National Academy of Sciences*, *112*(37), 11708–11713.

van Opheusden, B., Kuperwajs, I., Galbiati, G., Bnaya, Z., Li, Y., & Ma, W. J. (2023). Expertise increases planning depth in human gameplay. *Nature*, *618*(7967), 1000–1005.

Ward, G., & Allport, A. (1997). Planning and problem solving using the five disc tower of london task. *The Quarterly Journal of Experimental Psychology Section A*, *50*(1), 49–78.

Wiener, J. M., & Mallot, H. A. (2003). 'fine-to-coarse' route planning and navigation in regionalized environments. *Spatial cognition and computation*, *3*(4), 331–358.

Zhang, C., Lipovetzky, N., & Kemp, C. (2023). Comparing AI Planning Algorithms with Humans on the Tower of London Task. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *45*(45).