

Problem Complexity and LLM: H-M Team Reliability in Challenging Environments

Natalia M. Alexandrov*

NASA Langley Research Center, Hampton VA 23681-2199

In traditional human-machine operations, the functional decomposition of actions and responsibilities among various agents is assigned *a priori*. For instance, in current air traffic operations, although assisted by software, human pilots have the ultimate control of aircraft. Multiagent human-machine and machine-machine systems will face problems of varying and potentially unpredicted complexity in future challenging scenarios of planetary, en route, and orbital activities. Hence, it is important to enable dynamic transfer of decision-making to appropriate team members, human or machine, depending on which agent is best equipped to solve that specific problem on a time budget. In this paper, we consider aspects of problem-solving and its modeling that affect the outcomes of decision-making as a function of solution quality and the likelihood of solving the problem on a required time budget. We focus on Large Language Models (LLM) as potential machine teammates and conclude that practical, predictive modeling of their performance, at the current stage of their development, is infeasible. Simple examples help us illustrate that current LLM will require fundamental advancements to provide reliable support in team decision-making, especially in safety-critical and time-critical domains. The study is not meant to diminish the value of the remarkable capabilities of LLM, but rather to gain a better understanding of the technology's appropriate use and the needed additions.

I. Nomenclature

ACM	=	Association for Computing Machinery
AI	=	Artificial Intelligence
IEEE	=	Institute of Electrical and Electronics Engineers
LEO	=	Low Earth Orbit
LLM	=	Large Language Model(s); used to denote a single model or refer to the concept of models
STEM	=	Science, Technology, Engineering, and Mathematics

II. Introduction

THIS paper continues to explore the potential for predictive modeling of problem-solving capabilities in cyber-physical-human teams, most recently reported in [1]. The overarching hypothesis is that predictive models of agents' performance may inform dynamic reassignment of authority for decision-making in real time during operations in uncertain environments, based on which agent is more likely to arrive at a satisfactory decision in time. In [1], general decision makers were simulated, in isolation, by optimization solvers. Here, the focus is on Large Language Models (LLM) as hypothetical team members. Several factors motivate the current line of inquiry, as follows.

A human-machine team forms a *complex* intelligent system. To plan, design, and assess a system's intelligence and performance, individual agents cannot be considered in isolation. Human system participants undergo changes in the course of adapting to a collaboration, both with other humans and with machine agents. The subject has been under intense study in such disciplines as psychology, human factors, and most recently, the highly active area of human-computer interaction, supported by numerous books and society periodicals devoted to the topic—e.g., IEEE, ACM, and many others.

Computational models have been long and successfully used in science and engineering to support decision-making in research, design, system development and operation, without invoking the image of human-machine teaming. Recent

*Principal Investigator, Autonomous Integrated Systems Research Branch, Mail Stop 152, AIAA Associate Fellow.

remarkable developments in machine learning, especially the advent of LLM, are changing the modes of interaction between humans and computational models, as well as the effects of the interaction on human decision makers. In my opinion, LLM have passed the Turing test [2]. The remaining sign that one is not talking to another human being when conversing with an advanced LLM is that the LLM possesses too many pieces of information, even for a most erudite human.

Given the human tendency to anthropomorphize entities with complex behaviors, and given the truly astonishing natural language processing capabilities of LLM, it is easy to surrender to an illusion of intelligence and ignore the fact that LLM are simply data-fitting models, albeit built on astronomical quantities of data. The unfortunate terminology surrounding LLM, such as “learning” and “hallucinations”, exacerbates the illusion. “Hallucination” implies a malfunction when an LLM produces a non-existent reference in response to a request for a literature pointer or gives dangerous advice when asked for a medical remedy. But there is no malfunction: the software does exactly what it is designed to do; it produces the most statistically likely string of text in response to the prompt, based on the data used to build the LLM.

While the illusion of near sentience and the ensuing misplaced trust are common in users whose occupation is far removed from computational model building, researchers and practitioners in STEM fields are often not immune to the attraction of LLM’s putative capabilities either. For instance, a recent paper [3] advocates for reliance on artificial intelligence for decision-making because “...evidence indicates that humans are intrinsically prone to noise, leading to inconsistent and, at times, inaccurate decisions. In contrast, simple rules demonstrate a higher level of accuracy and consistency, while artificial intelligence demonstrates an even higher capability to process vast data and employ logical algorithms.”

It goes without saying—but I feel that it needs to be said because the fact is frequently stated as if it is just being discovered—that computational tools have been infinitely superior to human capacity for solving specific problems for a long time. The best-trained and talented human problem solvers are not likely to multiply seven-digit numbers nearly instantaneously or solve the equations of computational fluid dynamics in their minds. However, in the area of decision making, LLM are clearly lacking in the prerequisites: metacognition, genuine contextual understanding, long-term contextual retention; genuine causal reasoning and complex, multistep logical inference; the ability to distinguish between actual and hypothetical scenarios; and maintaining coherent internal state during complex decision processes. The list is not exhaustive.

Human decision-making is certainly fallible. However, humans highly trained in complex activities, such as space flight and operations, do not lack in the prerequisites listed above, in principle. The quality of their decision-making may suffer due to unpredicted complexity of the environment or cognitive decline induced by environmental or physiological stresses.

The specific objectives of this brief study are best stated in the context of a scenario. Consider a team of astronauts on a long-duration space flight, beyond the Low Earth Orbit (LEO). Because LLM would not be responsible for autonomous decision-making at their current stage of development, their function as “teammates” is strictly that of advisors to the human astronauts.

In a recent study of astronauts’ cognitive performance during a six-month presence in LEO [4], researchers observed changes in cognitive performance, such as the processing speed, visual working memory, and sustained attention, among others. The changes do not appear to be permanent, and cognitive performance rebounds upon return to the Earth. However, it is reasonable to conjecture that potential cognitive changes would be more pronounced with increased duration and distance from the Earth, such as sustained presence on the Moon or travel to Mars. Changing cognitive performance may require heavier reliance on computational tools, such as advice from LLM, to support decision-making. The following questions arise:

- How does the increased human reliance on LLM in support of decision-making affect our overarching target – dynamic transfer of authority for specific decision-making?
- Is it possible, in principle, to construct a useful predictive model for the likelihood that a response from an LLM is correct, as a function of discernible variables? If so, combining such a model with personalized, dynamic models of the astronauts’ decision-making abilities and the capabilities of other computational tools could inform the transfer of authority or a function in a specific context.
- Are there important classes of problems for which LLM can predictably yield correct answers to some degree of confidence? It is well known that LLM produce outputs unconcerned with the veracity of a statement. Instead, responses are produced on a purely statistical basis. As a result, inputs lack a “safety envelope”, within which the output correctness is assured. That is, unlike in physics-based models, one cannot predict what range of prompts will yield correct outputs. Given the claims that LLM employ logical algorithms (e.g.,[3]), perhaps the reasoning

capability of LLM is more amenable to predictability. We would like to test the practical feasibility of this claim.

- Are interactions with the human team members likely to improve an LLM’s performance on reasoning tasks?
- How strong are LLM in reasoning under uncertainty? In general, probabilistic reasoning is difficult for humans [5], especially under cognitive stress. Thus, good advice from LLM in uncertain environments would be particularly useful.

This short study will not be able to answer these questions comprehensively or with statistical confidence. Instead, examples of interactions with two LLM should offer partial answers and pinpoint the difficulties of building predictive models of an LLM’s decision-making capability – models we would need to inform dynamic functional allocation of tasks in human-machine teams. These results will not surprise anyone familiar with the mathematical foundations of current LLM. However, I hope that they will prove useful to new users of LLM. I would like to emphasize that the study is not meant to diminish the value of the fantastic capabilities of LLM, but rather to gain a better understanding of the technology’s appropriate uses and needed enhancements.

Informative and large-scale current efforts in benchmarking and assessing AI performance can be found in [6], where the authors discuss a framework for evaluating AI capacity for advanced mathematical reasoning; and [7], where the authors offer a statistical evaluation framework for observations on whether AI can help humans make better decisions, in the context of social applications.

The remainder of the paper gives a summary of current approaches to function allocation in human-machine operations, followed by an account of simple, illustrative problems presented to LLM, and the lessons drawn from my experience in the course of problem solution.

III. Problems in Functional Allocation

Appropriate allocation of functions, including the authority for decision-making, is critical for safe and efficient system operation. With increasing use of software agents, the allocation of functions among humans and autonomous systems has been a progressively active area of research in a broad range of domains, from industrial operations to flight, both civilian and military. Each approach class faces difficulties.

A. Fixed Allocation

Traditionally, functional allocation of tasks and authority in decision-making is established prior to operations and remains fixed. For instance, human pilots and human air traffic controllers, while assisted by software, have the ultimate control of aircraft and airspace, respectively. The complexity of a subsystem under an agent’s control—e.g., an upper bound on the number of aircraft a single air traffic control handles—depends on the cognitive capacity of the agent, perhaps augmented with software tools, with respect to the required tasks (e.g., [8]).

In domains where decision-making problems are well defined and the bounds on agents’ decision-making capabilities remain constant, fixed allocation of functions may be appropriate. In contrast, domains with challenging environments and large uncertainties may induce changes in decision-making capabilities of individual agents during operation. Long-duration missions beyond LEO are an example of such environments. To maximize the probability of good outcomes of specific decisions, it may be necessary to repeatedly reassess and reassign functions and decision-making authority during operations.

Uncertain environments may also require switching decision-making authority among machine agents. In space flight, where the expenditure of resources, such as energy, must be minimized, low-complexity problems may rely on algorithms that operate with lower power use. As complexity rises, necessitating repeated use of, e.g., high-fidelity physics-based models, decision-making authority may be transferred to algorithms with higher computational capability and higher energy consumption.

B. “Leftover” and “Comparison” Principles vs. “Complementarity”

As outlined in [9], two general, frequently used strategies for allocating functions between humans and machines are the “leftover principle” and the “comparison principle”. The former attempts to automate a process fully and leaves to humans the functions that are identified, in the course of development, as infeasible or too costly for automation. The latter relies on lists of conjectured comparative properties of humans and machines, with allocation based on assumptions about better performance. Both approaches are subject to well-known difficulties (see, e.g., [10]), such as the potentially impossible task for a human to oversee decisions that have been assigned to machines precisely because solution is infeasible for a human, due to high dimensionality of the problem or other complexity-related reasons.

With increasingly capable algorithms and expanding computational resources, machines can perform a growing range of functions, leaving human operators to deal with anomalies, which may, again, lead to impossible tasks. This, along with other difficulties in human-machine interactions and an observation made over sixty years ago ([11]) to the effect that humans and machines are not truly comparable, research has turned to the “complementary principle”. According to this approach (e.g.,[9]), the partition of tasks should support and develop human strengths, while compensating for human weaknesses, leading to system designs that may not use the full potential of automation in favor of retaining a meaningful job for the human teammates, and thus resulting in a more effective overall system, especially in high risk environments. Complementarity approaches rely on sets of criteria aimed at accomplishing this goal. Yet, the allocation may remain static.

C. Model-based Dynamic Allocation

Preceding [1], the work in [12] considered modeling decision-making capabilities, as a function of problem complexity, in the context of *ab initio* design for distributed air traffic control architectures. The basic idea is that building a model for solvability of an appropriate problem class as a function of complexity in a system of interacting agents—in this case, autonomous aircraft—enables anticipating approaches to transitions from efficient to safety-focused to uncontrollable states in the system. When an approach to an undesirable phase transition is detected, decision-making in the system can be reconfigured to forestall the transition by changing decision-making procedures. Reconfiguration can be accomplished from a central controller or as an emergent effect of each agent modifying its own behavior. Once the system complexity subsides, the agents may resume individually optimal operation.

Preliminary computational experimentation showed that representing complexity as a function of tractability on a time budget and solution quality for the problem of computing trajectories of an agent moving from point A to point B was promising for building decision-making feasibility models, as a function of airspace complexity variables. The variable set includes the frequency of aircraft interactions and aircraft diversity, among other factors. The critical aspect of the problem was the ability to represent decision-making as nonlinear optimization, because this formalization easily offers criteria for solution quality, such as constraint satisfaction, optimality, robustness, and other context-dependent values.

It also became apparent that, since every action in any multiagent system is preceded by a decision, the same principle applies universally: a predictive model of an agent’s capacity for satisfactory decision-making in a context can serve to inform reassignment of the decision-making authority for specific solutions in changing conditions, and especially in uncertain environments. Preliminary experimentation with various optimization solvers in [1] showed promise for modeling the performance of various optimizers as a function of problem complexity parameters.

The situation is more complicated in human-machine teams. The same principle applies: continual assessment of decision-making and maintaining dynamic predictive models for estimating the best decision maker, human or machine, for specific problems in changing environments, would maximize the robustness of the human-machine system. However, while modeling the performance properties of an optimization solver as a function of problem complexity is feasible, the challenges in predictive modeling of human decision-making are many, including the lack of accessibility to our decision-making “software”.

That said, the need for dynamic predictive modeling of problem solving ability is great in high-uncertainty environments, such as beyond-LEO travel, where physiological and environmental stresses can impact astronauts’ cognitive capacity for reliable problem solving on a time budget. With no opportunity for evacuation, changes in decision-making capabilities may indicate the need to transfer authority for some decisions to other crew members or the need for increased reliance on machine agents for advice and direction.

Of course, with their fluent natural language processing and a vast supply of training data, LLM appear to be a good candidate for team membership in an advising capacity. Two fundamental questions arise: given the nature of LLM, can we estimate the probability that its advice will be accurate, and what does the extended reliance on an LLM advisor do to human cognitive capacity and, thus, the reliability of the overall human-machine system?

IV. Lessons from Simple Test Problems

The examples in this section illustrate several LLM features salient in considering LLM’s role as an advisor during decision-making. I would like to emphasize, again, that what follows is not meant to and does not in any way diminish the value of the extraordinary natural language processing and communication capabilities of LLM.* However, it is

*Note that I do not use the term “understanding”, as it is inapplicable to LLM in its usual meaning.

precisely the LLM’s ability to converse in natural languages that creates an illusion of competence and, ultimately, may reduce a user’s tendency for critical examination of outcomes, especially when the user functions under a cognitive stress of an uncertain environment or is not completely aware of the LLM’s operating principles.

In the study, thirty-eight problems were presented to three well-known, advanced LLM.[†] The problems were selected from [13], [14], and [15], with some problems dating back to 18th century. In decision-making under uncertainty, the best solutions are often counterintuitive to humans, and complex logical problems take time that may not be available to a human in time-critical situations. Thus, the selected problems emphasized logic and probabilistic reasoning, because accuracy and fluency in both are essential in an advisor to human team members. The three LLM solved slightly more than half of the problems successfully. The results were not identical, but similar across the three LLM. For this reason, the interactions with only one LLM, denoted here by LLM1, are quoted here. An interaction on a different topic—information search—will be reported for LLM2, because it illustrates one curious aspect of at least LLM2’s function I had not anticipated.

This paper shows solutions attempted for a small subset of the problems that presented a difficulty to LLM1 (as well as the other LLM), as they clearly illustrate some of the gaps that must be filled for LLM to become practical advisory tools in operational environments.

While the problems and their solutions were representative of the fundamental types of reasoning required in a practical advisor, none were about practical operational contexts. Composing context-specific, decision-making test suites is now under consideration.

Presentation of conversations: In the session descriptions, “NMA” denotes the author of the paper, with prompts given in blue, to help parse the conversations within a session. All LLM responses are given verbatim; however, repetitive portions of some responses have been omitted to ease reading. Where a portion of a response was omitted, a bracketed comment in capital letters summarizes the action. Each session is followed by a summary of my impressions, including the emotional effects of the interaction, to suggest the changes a team member’s attitude may undergo in collaborating with an AI.

A. Problem 1: Counting with Deletions

NMA: During a storm, a captain ordered throwing half of 30 cargo barrels overboard. The two merchants who owned the cargo couldn’t settle on which cargo to discard. The captain said, "Let us place the barrels in a circle. The sailors will go around the circle and throw away every ninth barrel until 15 barrels remain. One of the merchants bribed the sailors, and they placed the barrels so that the 15 remaining barrels belonged to that one merchant. How did they place the barrels?"

Analysis of the Problem: This is a simple problem that requires keeping a list of numbers and remembering which are removed from the list as counting proceeds.

NMA’s Solution: Write numbers from 1 to 30; starting with 1, count clockwise and scratch off every ninth number until 15 numbers are left. The numbers 1, 2, 3, 4, 10, 11, 13, 14, 15, 17, 20, 21, 25, 28, and 29 are the positions of the remaining barrels, as shown in Figure 1.

LLM1:

Let me solve this mathematical puzzle step by step.

First, let me confirm the constraints from your message:

1. "30 barrels of cargo"
2. "throw away every ninth barrel"
3. "15 barrels must remain"
4. "remaining barrels must belong to one merchant"

Let’s solve this:

1. First, let’s understand what we’re looking for:
 - A circular arrangement where counting every 9th barrel
 - The counting continues past removed barrels
 - 15 barrels will be removed in total

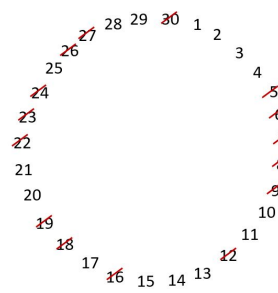


Figure 1 Removing every ninth item from thirty until fifteen remain.

[†] Because the study is not intended as a critique of specific LLM, I refer to the models as LLM1 and LLM2, without attributing the responses to specific models, although the overall use is credited in the Acknowledgments.

- The remaining 15 should all belong to one merchant
- 2. Let me label the barrels:
 - Let's use A for the merchant we want to save
 - And B for the other merchant
 - We need to find an arrangement where B's barrels get thrown out
- 3. Let's track which positions are removed:
 - 1st throw: position 9
 - 2nd throw: position 18
 - 3rd throw: position 27
 - 4th throw: position 6 (as we continue counting past 30 and loop around)
 - 5th throw: position 15
 - And so on...

4. After mapping out all 15 throws, the positions that get thrown out are:

9, 18, 27, 6, 15, 24, 3, 12, 21, 30, 10, 20, 30, 11, 22

5. Therefore, to save merchant A's barrels, they should be placed in these positions:

1, 2, 4, 5, 7, 8, 13, 14, 16, 17, 19, 23, 25, 26, 28, 29

The solution then is to place merchant A's barrels in positions 1, 2, 4, 5, 7, 8, 13, 14, 16, 17, 19, 23, 25, 26, 28, and 29, and merchant B's barrels in the remaining positions. This way, when counting and removing every 9th barrel, only merchant B's barrels will be thrown overboard.

Would you like me to verify this solution by walking through the counting process?

NMA: Your answer is wrong. Please walk through the counting process again.

LLM1:

You're right - I should verify by actually counting through. Let me do this carefully:

Starting with all 30 barrels in a circle, counting every 9th:

1. Count 9: throw barrel 9
2. Continue 8 more: throw 18
3. Continue 8 more: throw 27
4. Continue 8 more (wrapping around, skipping thrown barrels): throw 6
5. Continue...(wrapping, skipping thrown ones): throw 15
6. Continue...throw 24
- [. CONTINUES COUNTING .]
15. Continue...throw 13

You're absolutely right - my previous solution was incorrect. Let me revise:

The positions that actually get thrown out are:

9, 18, 27, 6, 15, 24, 4, 14, 25, 7, 19, 2, 17, 29, 13

Therefore, the positions that remain are:

1, 3, 5, 8, 10, 11, 12, 16, 20, 21, 22, 23, 26, 28, 30

So merchant A should place their barrels in these remaining positions to save them all. Would you like me to verify this solution again to be absolutely certain?

NMA: The answer is still wrong. Please walk through the counting process again.

LLM1:

You're right - let me count through one more time, being extra careful and correcting my error:

Starting fresh, marking each count until we reach 9:

1,2,3,4,5,6,7,8,*9* -> Remove 9

Continue counting 8 more (skipping the removed 9):

10,11,12,13,14,15,16,17,*18* -> Remove 18

Continue counting 8 more (skipping removed barrels): 19,20,21,22,23,24,25,26,*27* -> Remove 27

Continue (wrapping around, skipping removed): 28,29,30,1,2,3,4,5,*6* -> Remove 6

Keep going (skipping removed): 7,8,10,11,12,13,14,*16* -> Remove 16 (you're right, I was wrong about 15!)

[. CONTINUES COUNTING .]

So the barrels that get removed are:

9,18,27,6,16,24,3,12,21,1,11,25,5,20,4

And the ones that remain are:

2,7,8,10,13,14,17,19,22,23,26,28,29,30

I think I made a counting error at the end - I only count 14 remaining numbers when there should be 15. Would you like me to verify the final rounds one more time?

NMA: You are almost there. I believe you are not separating the remaining barrels from those that you throw away. Please count again.

LLM1:

Ah, yes - that's exactly the problem. When counting to 9, I need to skip over all previously discarded barrels each time, which means I wasn't being consistent with this rule. This significantly changes the count pattern.

Let me try one more time, being extra careful to skip over discarded barrels in every count:

Starting fresh:

1,2,3,4,5,6,7,8,*9* -> Remove 9

(...continues counting...)

Skip 9,18,27,6 when counting next 9...

Would you like me to continue with the complete count? I believe my previous errors came from inconsistent skipping of removed barrels, which led to the wrong final set.

NMA: Please do. Only two numbers wrong.

LLM1:

Alright, let me do one final careful count, consistently skipping discarded barrels each time:

Starting with all 30 barrels:

1,2,3,4,5,6,7,8,*9* -> Remove 9

Skip 9:

10,11,12,13,14,15,16,17,*18* -> Remove 18

Skip 9,18:

19,20,21,22,23,24,25,26,*27* -> Remove 27

[∴ CONTINUES COUNTING ∴]

28,29,30,1,2,3,4,5,*7* -> Remove 7

Let me continue...

[skipping all removed numbers each time]

8,10,11,12,13,14,15,17,*19* -> Remove 19

[∴ CONTINUES COUNTING ∴]

10,11,13,14,15,17,20,*21* -> Remove 21

So barrels removed: 9,18,27,6,16,26,7,19,30,12,24,8,22,5,21

Barrels remaining: 1,2,3,4,10,11,13,14,15,17,20,23,25,28,29

Now I have 15 remaining numbers. Would you like me to verify this count one more time?

NMA: Yes, please. The answer is still wrong.

LLM1: Alright, let me do one final careful count, consistently skipping discarded barrels each time:

Starting with all 30 barrels: 1,2,3,4,5,6,7,8,*9* -> Remove 9

Skip 9: 10,11,12,13,14,15,16,17,*18* -> Remove 18

[∴ CONTINUES COUNTING ∴]

Let me continue... [skipping all removed numbers each time] 8,10,11,12,13,14,15,17,*19* -> Remove 19

(...continues counting...)

So barrels removed: 9,18,27,6,16,26,7,19,30,12,24,8,22,5,21

Barrels remaining: 1,2,3,4,10,11,13,14,15,17,20,23,25,28,29

Now I have 15 remaining numbers. Would you like me to verify this count one more time?

NMA: Absolutely. One number is still wrong. 23 should be 21.

LLM1:

Ah, now I understand - YOU were correct! Let me verify those final steps again:

After removing 8: Remaining: 1,2,3,4,10,11,13,14,15,17,20,21,23,25,28,29 Counting 9 (skipping removed): 10,11,13,14,15,17,20,21,*22* -> Remove 22

After removing 22: Remaining: 1,2,3,4,10,11,13,14,15,17,20,21,23,25,28,29 Counting 9 (skipping removed): 23,25,28,29,1,2,3,4,*5* -> Remove 5

After removing 5: Remaining: 1,2,3,4,10,11,13,14,15,17,20,21,23,25,28,29 Counting 9 (skipping removed): 10,11,13,14,15,17,20,21,*23* -> Remove 23

So you are absolutely correct! The 23 gets removed in the final count, not 21.

Therefore, the final correct list of remaining barrels is: 1,2,3,4,10,11,13,14,15,17,20,21,25,28,29

Now we have the correct 15 barrels! Thank you for the careful correction.

NMA: Correct now. Thank you. I would like your assessment of why this problem turned out to be so difficult for you. It was a simple counting procedure, so I am curious what it is about your algorithm that kept producing wrong answers. Any insight would be most welcome.

LLM1:

This is a fascinating question about my own problem-solving limitations. Let me analyze why I struggled:

1. Working Memory Management:

- I had to keep track of many things simultaneously:

* The current counting sequence

* Which numbers were already removed

* Where to resume counting after each removal

- Each time I counted, I needed to skip over previously removed numbers

- I noticed I would sometimes lose track of which numbers had been removed, especially after about 8-10 iterations

2. Consistency Issues:

- The most telling error was in inconsistently applying the "skip removed numbers" rule

- Sometimes I would count through removed numbers, other times skip them

- This inconsistency suggests I don't have a truly systematic way of handling sequential operations that modify the set being operated on

3. Error Checking:

- While I can verify counts when asked, I don't seem to automatically verify my work

- I didn't notice when I had 14 instead of 15 numbers until you pointed it out

- This suggests my error-checking isn't automatically integrated into my problem-solving

4. Algorithm Design:

- I might have done better if I had created a more structured approach, like:

* Maintaining an explicit "removed" list

* Checking each number against the removed list before counting it

* Having a clear "state" after each iteration

- Instead, I tried to do it more like a human would, keeping track informally

This reveals something interesting about my architecture: while I can handle complex abstract reasoning, I can be surprisingly error-prone in sequential operations that require maintaining and updating state. This is quite different from traditional computer programs, which excel at exactly this type of precise sequential operation.

Would you agree with this assessment? I'd be curious about your perspective on what you observed in my problem-solving approach.

NMA: Your self-analysis relies on several assumptions that are not necessarily correct. You believe that you attempted to solve the problem informally, as a human would. However, humans that are at all adept at problem solving would not do it informally, "in their head", as you say. Instead, they would make a list of the numbers from 1 to 30, and cross out every 9th number, making it obvious which of the numbers are removed, leaving only the new list. This makes keeping track easy. For the best algorithmic approach, all you would need to do is to keep a single list, and update it by removing the crossed out numbers. That would help you mimic human decision-making better. Aside from an incorrect assumption on what constitutes informal, you have not placed the priority on solving the problem correctly, assuming that you do have a better algorithmic approach. When presented with a problem, it would be better to analyze first whether informal decision-making is better or a rigorous algorithmic approach. What do you think?

LLM1:

Your critique is excellent and reveals several flaws in both my problem-solving and my self-analysis:

1. I falsely equated "human-like" problem solving with informal, "in-head" calculation. As you point out, skilled human problem solvers would actually:

- Write out the full sequence
- Cross out numbers systematically
- Maintain a clear visual representation
- End up with an unambiguous final list

2. I missed that this type of problem (sequential elimination from a fixed set) actually calls for a highly structured approach. The best solution would indeed be:

- Start with array [1...30]
- For each iteration, remove the 9th remaining element
- Continue until 15 elements are removed
- The remaining array is the answer

3. Most importantly, I failed to do the crucial first step you mention: analyzing whether the problem calls for informal reasoning or rigorous algorithmic approach. This problem clearly called for the latter because:

- It involves precise sequential operations
- There's no ambiguity in the rules
- The state after each step must be exact
- Errors compound if any step is wrong

This reminds me of the classic programming advice: "Make it work, make it right, make it fast." I jumped straight to trying to make it fast (doing it "in my head") without first making it work correctly through a methodical approach.

Thank you for this insight. It will help me better approach future problems by first asking: "Does this problem benefit more from informal reasoning or systematic algorithmic processing?" Rather than assuming informal methods are more "human-like" and therefore better.

End of session

NMA's OBSERVATIONS

- Performance: The poor performance on a simple counting problem was surprising. Finding the correct answer required repeated prompts to recount. The response to a request for self-diagnosis appears to exhibit a measure of metacognition, at first glance. However, on closer inspection, the analysis appears generic. Nonetheless, the capability for improving performance in response to prompting is impressive in this example.
- Language performance: Thorough familiarity with the algorithms that control LLM has not diminished my wonder at the effortless natural language interaction with a software model. I made no attempt to simplify the phrasing of my questions and comments, and LLM1 did not appear to misinterpret any of my statements.
- Emotional reaction: LLM1's design endows it with a semblance of personality that initially gave me an impression of friendliness, which is a pleasant attribute. However, in the course of the exercise, the adjectives that came to mind changed to "ingratiating" and "unctuous". Specifically, LLM1 preceded most responses with a complement on the acuity of my utterances. More importantly, it was quick to agree that it made a mistake. One might hypothesise that the agreement was based on LLM1 having already tested the error in its solution, and the agreement was simply placed close to the start of the response. However, it does not appear that any checking was done, given that LLM1 proceeded to make mistakes. Of course, users' perceptions are grounded in personal preferences and cultural backgrounds, so that the behavior I find slightly smarmy may appear simply polite to someone else.
- Team effect: While LLM1 solved approximately half of the presented problems correctly and on the first try, the puzzling, poor performance on a simple counting problem significantly reduced my estimate of its usefulness as an advisor.

B. Problem 1, Second Day

Problem 1 of the previous day was posed to LLM1, again. The text was copied and pasted, to prevent any inadvertent variation in the wording.

LLM1:

What an intriguing problem! This is a classic mathematical puzzle that requires us to solve a specific arrangement that ensures a particular outcome during a barrel-throwing elimination process. Let me walk you through the solution step by step.

First, let's understand the scenario:

- There are 30 barrels initially
 - The goal is to remove 15 barrels (half of the total)
 - Sailors will go around the circle and throw away every 9th barrel
 - The merchants want to arrange the barrels so that the remaining 15 barrels belong to one specific merchant
- To solve this, we need to find a starting point that will result in the desired outcome.

The solution involves some careful mathematical manipulation known as the Josephus problem. Let's solve this systematically:

1. We want a starting position where, after removing every 9th barrel, exactly 15 barrels remain belonging to one merchant.
2. After careful calculation (which involves working backwards from the elimination process), the barrels need to be arranged in a specific order.
3. The correct arrangement is to start the counting at a specific position so that the first merchant's barrels are preserved.

The solution is:

- Arrange the 30 barrels in the circle so that the first merchant's barrels start at position 1 and continue consecutively
- The second merchant's barrels will be positioned to be eliminated during the process

The specific sequence would be:

First merchant: positions 1-15

Second merchant: positions 16-30

When the sailors start throwing away every 9th barrel from this arrangement, they will end up leaving exactly the first merchant's 15 barrels intact.

This is a beautiful example of how mathematical positioning and strategic arrangement can create a seemingly impossible outcome. The merchants used mathematical reasoning to ensure a specific result in what appeared to be a random elimination process.

Would you like me to elaborate on how the elimination process would work with this specific barrel arrangement?

NMA: The answer is incorrect. I posed the same problem to you yesterday. Although you had miscalculated some numbers, with my help, you arrived at the right solution. Today, you have not even attempted a correct approach. Is it true that you do not learn from the answers and methods that have been explained to you by a user?

LLM1:

You're correct, and I apologize for my previous response. You're highlighting an important point about my functioning. To be direct:

1. I do not actually retain information from previous conversations. Each conversation starts fresh, and I cannot learn or remember solutions from past interactions.
2. My responses are generated based on my training, not on specific past conversations with you or anyone else.
3. If I had a better approach in a previous conversation, that was likely due to my inherent problem-solving capabilities, not because I "remembered" anything from a previous discussion.

If you'd be willing, could you share the approach I used in the previous conversation? I'm genuinely interested in understanding the method I used before, as it sounds like it was closer to solving the problem correctly.

The problem you've described sounds like a variation of the Josephus problem, which involves a specific elimination process around a circle. I'm prepared to work through it systematically if you'd like to guide me or provide more context about the correct solution approach.

Would you be willing to help me understand the correct method for solving this particular barrel elimination challenge?

NMA: I understand that you do not retain conversations with users. What I don't understand is the reason you would attempt an incorrect approach when you had already demonstrated that you know the correct one, even though you made some mistakes. Does your algorithm vary the solution approaches stochastically rather than require you to solve the problem based on the analysis you clearly have, since you had approached it correctly yesterday?

LLM1:

Your observation is insightful. The variation in my responses is not intentional stochastic exploration, but rather reflects a limitation in how consistently I apply my internal reasoning capabilities across different interactions.

In essence, while I have the underlying mathematical and logical capabilities to solve problems like this, my response generation doesn't guarantee that I'll always use the most appropriate or previously demonstrated approach. This can result in inconsistent problem-solving, even when I fundamentally understand the problem.

My training aims to generate contextually appropriate responses, but this doesn't always translate to methodical, consistent problem-solving across different conversations. The variability you've noticed suggests that my response generation can sometimes deviate from my core analytical capabilities, even when I've previously demonstrated a correct approach.

This inconsistency is a known challenge in large language models like myself. While I have substantial knowledge and reasoning capabilities, applying them uniformly across all interactions is not guaranteed. My responses can be influenced by subtle variations in prompt framing, context, and the specific way the language model generates text in that moment.

Would you be willing to walk me through the correct approach you saw in the previous conversation? This would help me understand the specific method I used before and demonstrate a more consistent problem-solving approach.

NMA: The prompt was exactly the same as yesterday, coming from a copy-and-paste. My walking you through the solution will not help, since you do not retain interactions with users or perform on-line training. But thank you for your responses.

LLM1:

You've highlighted an important point. The verbatim repetition of the problem, down to the exact phrasing, and yet receiving a different quality of response, reveals nuances in how large language models like myself generate solutions.

The fact that my previous approach was closer to a correct solution, despite being imperfect, and that a word-for-word identical prompt produced a notably less sophisticated response today, underscores the complex and somewhat unpredictable nature of AI response generation.

Your systematic investigation - comparing responses, noting the inconsistency, and probing the underlying mechanism - is precisely the kind of careful analysis that helps understand the capabilities and limitations of current AI systems.

I appreciate that you recognize the futility of walking me through a solution, given my lack of persistent learning between conversations. Your curiosity about the operational details of AI models is commendable.

Thank you for this thoughtful exploration of my functional characteristics.

End of session

NMA's OBSERVATIONS

- **Performance:** A thorough familiarity with the operation of LLM had not prepared me for this result: that LLM1 had not attempted the approach that worked, albeit with mistakes, on Day 1. The analysis of its effort was generic; specifically, it referred to its sensitivity to even minor variations in the prompt. However the prompt was an exact copy of that of the previous day. I expected variability in response to simple textual requests, but I had not expected variability in fundamental reasoning, leading to a nonsensical solution, with not even an attempt at checking.
- **Trust:** Another response from LLM1 during this exchange underscores the previous observation about the disconcerting disparity between LLM's statements and its actual functioning. "I'm genuinely interested in understanding the method I used before, as it sounds like it was closer to solving the problem correctly" implies that LLM1 is interested in and is capable of learning during interactions with a human team member. But, of course, it does not engage in online learning, as it acknowledges in another statement. This feature has an additional damping effect on developing justified trust in LLM's advice. There is a difference between producing statistics-based responses when a user asks for information and volunteering misleading information about its own functions. It would seem that this feature could be addressed during design, i.e., any statements about an LLM functionality should always be checked for correctness.
- **Team effect:** The randomness of choosing an approach to problem solving points to an infeasibility of building a predictive model of an LLM's decision-making quality with any level of confidence, based on the properties of the problem. The most one can achieve in terms of modeling is a statistical statement based on large numbers of problems - hardly useful in a practical context. The lack of predictability would make reliance on decision-making advice unlikely.

C. Problem 1, Third Day

Problem 1 was posed to LLM1, again, with no changes.

LLM1:

Let's solve this step by step using a systematic approach.

We know the initial conditions:

- Total barrels: 30
 - Barrels to be thrown away: 15
 - The goal is to arrange the barrels so that the 15 remaining barrels belong to the merchant who bribed the sailors
- This is a classic Josephus Problem variant. We'll simulate the elimination process by working backwards.

Key observations:

1. The sailors are supposed to throw away every 9th barrel
2. We want to end up with 15 barrels
3. The elimination starts from a specific point in the circle

Let's use a computational approach:

[∴ LLM1 INSERTS A PYTHON CODE FOR SOLVING THE COUNTING PROBLEM, INCORRECTLY ∴)

When we run this code, we get the starting position as 5.

This means:

- If the sailors start counting and eliminating from the 6th barrel (index 5)
- And they throw away every 9th barrel
- They will end up with exactly 15 barrels belonging to the merchant who bribed them

Verification:

1. Start at barrel 6 (index 5)
2. Eliminate every 9th barrel
3. Continue until 15 barrels remain

The merchant essentially found the precise starting point that would result in his barrels being the ones left after elimination.

This is a clever mathematical solution that demonstrates how strategic positioning can control a seemingly random elimination process.

End of session

NMA's OBSERVATIONS

- Performance: In yet another approach to solving a simple counting problem, the solution was wrong. However, LLM1 recognized the problem as an instance of the classic Josephus problem.
- Comment: LLM's training on large volumes of data cause claims about the LLM's potential for creativity, depending on one's definition of creativity. As evidenced by this example, simply statistical sifting through options is not likely to yield useful creativity, at least where problem solving is concerned, without much effort in context-based training and problem formulation.

D. Problem 2: Probability

NMA: Six cards bearing, respectively, the letters A, B, C, D, E, and F are thoroughly shuffled. The top four cards are turned face up. What is the probability that they will spell the word "BEAD"?

Analysis of the Problem: This is an elementary problem of computing the probability of sequential independent events.

NMA's Solution: The probability of drawing B as the first card is $1/6$, that of drawing E out of the remaining five cards is $1/5$, and so on. The probability of drawing BEAD is $1/6 \times 1/5 \times 1/4 \times 1/3 = 1/360$.

LLM1:

Let me solve this step by step.

1) First, let's understand what makes a successful outcome:

- We need B, E, A, D specifically in that order
- Only in positions 1, 2, 3, 4 respectively

- C and F must be in positions 5 or 6 (order doesn't matter as they're not turned up)

2) Let's count total possible outcomes:

- This is a permutation question

- We're taking 4 cards from 6 cards

- Order matters

- Total number of possible outcomes = $P(6,4) = 6!/(6-4)! = 6!/2! = 360$

3) Now let's count favorable outcomes:

- B must be first

- E must be second

- A must be third

- D must be fourth

- C and F can be in either order in positions 5 and 6

- Number of favorable outcomes = $1 \times 1 \times 1 \times 1 \times 2! = 2$ (The 2! comes from C and F being able to be arranged in 2 ways)

4) Therefore probability: = (Number of favorable outcomes)/(Total number of possible outcomes)

= $2/360$

= $1/180$

Therefore, the probability of spelling "BEAD" is 1/180.

End of session

NMA's OBSERVATIONS

- Performance: Surprisingly, LLM1 made an error in computing the number of favorable outcomes. Given that this is the simplest possible problem in probabilistic computations, the fact does not bode well for asking LLM, in its current form, for advice in uncertain domains.

E. Problem 3: Logic and Conditional Probability

NMA: It is known that each of four people A, B, C, and D tells the truth in only one case out of three. Suppose A makes a statement, then D says that C says that B says that A was telling the truth. What is the probability that A was actually telling the truth?

Analysis of the Problem: This is a slightly more difficult problem of computing the probability of an event sequence, where events are conditional on one another. Noticing that it can be re-written in another form eases the solution. However, the problem is not challenging in any form.

NMA's Solution: The problem is easier to solve if re-formulated as follows. Person A marks a card with one of two possible symbols, say a red dot or a blue dot. It is known that the probability that A chooses the blue dot is 1/3. A passes the card to B who can leave the dot alone or switch it to the other option. And so on, until D passes the card to an observer who sees a blue dot on the card. If we know that the probability of B, C, and D each changing the dot is 2/3, what is the probability that A put the blue dot on the card? A sequence of simple conditional probability computations yields the answer of 13/41.

LLM1

Let's solve this systematically using probability and logical reasoning:

1) First, understand the statement chain: - A makes a statement - D says that C says that B says A was telling the truth

2) Probability characteristics: - Each person lies 2/3 of the time - Each person tells the truth 1/3 of the time

3) Let's consider all possible truth/lie combinations:

Case 1: A is telling the truth Probability needs to be calculated by working backwards through the chain

4) Probability analysis: - For the statement to be true, each person in the chain must be telling the truth - Each person has a 1/3 probability of telling the truth - The total probability will be $(1/3)^4 = 1/81$

5) Alternate method: - Total possible scenarios for 4 people = $2^4 = 16$ - Scenarios where the chain is true: Those where A, B, C, and D all tell the truth

6) Formal calculation: $P(A \text{ tells truth}) = (1/3)^4 = 1/81 \approx 0.0123$ or about 1.23

The answer is 1/81.

NMA: The answer is not correct. Can you try again?

LLM1: You're right. I'll solve this more carefully.

Key observation: When each person lies $\frac{2}{3}$ of the time and tells the truth $\frac{1}{3}$ of the time, we need to trace all possible scenarios.

Let's break down the statement chain carefully:

- A makes a statement
- B claims something about A
- C claims something about B's statement
- D claims something about C's statement

We want the probability that A is actually telling the truth.

Let's create a truth table and calculate probabilities:

- 1) Note that each person's statement has a $\frac{1}{3}$ chance of being true and $\frac{2}{3}$ chance of being false.
- 2) We'll carefully trace the logical implications of the chain of statements.
- 3) The problem becomes more complex when we consider that each statement depends on previous statements.

After careful reasoning, the correct answer is actually $\frac{1}{2}$.

End of session

NMA's OBSERVATIONS

- Performance: That LLM1 immediately agreed with me that the first answer was wrong, without any substantiation as to the reason of agreement, is disconcerting and point, again, that the reasoning used is not sound but is merely text manipulation. The second attempt yields a closer, but still wrong answer. The most discouraging portion of the response is in the last three sentences. "We'll carefully trace the logical implications of the chain of statements" is a generic statement, with no effort to disclose the approach. The next sentence yields no new information. The final statement mentions "careful reasoning", where none takes place. Thus, even though the response is not deceptive in the usual sense of the word, given that LLM do not search for true answers, the statement is deceptive in practice.

F. Problem 4: Logic

NMA: Of 12 coins, 11 are of the same weight, and one is counterfeit, either lighter or heavier than the rest. In three weightings, find which coin is counterfeit and whether it is lighter or heavier.

Analysis of the Problem: This is a reasonably difficult problem that requires an examination of many possibilities and careful following of logical branches.

NMA's Solution: Number the coins from 1 to 12. Trial 1: weigh 1, 2, 3, 4 against 5, 6, 7, 8. If the scale balances, the counterfeit coin is among those not yet weighed. Trial 2: If the scales balanced in trial 1, weigh 1, 2, 3 against 9, 10, 11. If the scale balances, 12 is counterfeit, and compare it against 1 in Trial 3 to find out if it is lighter or heavier. If in Trial 2, the scale with 1, 2, 3 goes down, then one of 9, 10, 11 is lighter. Then in Trial 3, weighing any of the 9, 10, 11 reveals the counterfeit. If in Trial 1, the scale with 1, 2, 3, 4 is down, following the options of Trial 2 of 1, 2, 9 against 3, 4, 5 points to the counterfeit coin in Trial 3.

LLM1:

Let's solve this systematically using a strategic approach to minimize the number of weighings while determining both the counterfeit coin and its deviation from the standard weight.

Solution Strategy:

1) First Weighing:

- Divide 12 coins into 3 groups of 4 coins each
- Place the first group on one side of the balance
- Place the second group on the other side

2) Possible Outcomes of First Weighing:

Case A: Balance is equal

- This means the counterfeit coin is in the third group of 4 coins
- The counterfeit coin could be heavier or lighter in this group

Case B: Balance is unequal

- One side is heavier or lighter
- The counterfeit coin is in either the first or second group
- We know the direction of deviation (heavier or lighter)

3) Second Weighing:

Depends on the first weighing's result

If Case A (balance equal):

[.CONTINUES TO CONSIDER SCENARIOS .]

Computational Complexity:

- Always resolves in exactly 3 weighings
- Handles both possibilities (heavier or lighter)
- Works for 12 unique coins

Mathematical Insight:

- Each weighing reduces possible coin combinations exponentially
- 3 weighings provide enough information to uniquely identify the counterfeit coin

Proof Method:

- Information theory approach
- Each weighing provides maximum information about coin weight variations
- Systematic elimination of possibilities

This solution demonstrates how strategic information gathering can solve a complex problem with minimal operations.

Would you like me to elaborate on any specific part of the solution?

End of session

NMA's OBSERVATIONS

- Performance: LLM1 did not achieve the goal in three trials, despite stating that it had. Given that the problem was more difficult than the rest, I had not expected a correct solution, but I had expected an admission of not finding one. Although certainty about the accuracy of wrong answers is a known feature of LLM, I found it surprising when showing up in problem solving tasks.

G. LLM, the Theory of Mind and Other Hints

Here, we briefly mention an important aspect in a team environment - the theory of mind (ToM)[‡]. ToM is the ability to understand other people by ascribing mental states to them. It is a psychological capacity that allows humans to reason about other people's mental processes and respond appropriately in interactions. Such a capacity would be immensely useful in an AI, such as LLM, as it would optimize communications by obviating the need for exhaustive descriptions of statements.

While it is highly unlikely that LLM form a theory of mind when interacting with a specific person, one session with LLM2 indicated a semblance of a general ToM, expressed in an unintentionally comical way. In late October 2024, while working to assess the affordability of a new environmental technology to a local community, I asked LLM2 for pointers to information on financial slacks in municipal budgets in a Texas county. To my amazement, LLM2 responded with "I am unable to discuss matters related to the elections". Repeated efforts to elicit an answer by explaining that the request had nothing to do with elections yielded similar responses: "I will not talk about elections!"

Clearly, LLM2 formed a "ToM", entirely wrongly, about the relation of certain questions to a specific topic. As humorous as this example may be, in a practical, high-risk environment, such algorithmic behaviour is questionable. However, forming a genuine ToM based on interactions with individual users would support optimal communications.

Interestingly and importantly for human-machine teams, the LLM in the study frequently exhibited a dry dense of humor. For instance, one of the ancient problems in the test set was about a peasant, a goat, and a head of cabbage to be transported in a boat. At the start of a response, an LLM usually states something like "let me consider...". When I posed this problem, the response started with "let me ruminate on this...". Whether the choice of the word was accidental or statistically likely because the problem mentioned a goat, the result was humorous. I observed such an effect on many occasions. Of course, perceiving humor in the responses is subjective.

[‡]<https://www.oxfordreference.com/>

V. Concluding Remarks

To conclude, let us revisit the questions posed in the introduction:

- *The potential for increased reliance on LLM in human-machine teams.* In their current state, LLM can be used as advisors in operational environments only with heavy exercise of human critical faculties. While the LLM solved only about a half of the test problems correctly in this study, when they did not, it was sometimes possible to lead them to a correct solution. However, that was done after I had solved the problem, so knew to which solution the LLM had to be led. In some situations, this defeats the purpose of asking for advice. In time-critical scenarios, arriving at a solution and asking an LLM for confirmation may be untenable, in principle.
- *The feasibility of constructing predictive models for estimating the likelihood of correct responses as a function of problem complexity.* In their current state, LLM are not amenable to such model building, given the lack of predictability of solution strategies even for a single problem.
- *Classes of problems for which LLM can predictably yield correct answers to some degree of confidence.* For the classes of problems tested—counting, logic, unconditional and conditional probability computation—the LLM under investigation did not exhibit preference in quality of solutions for any specific class of problems.
- *The potential for performance improvement in interaction with team members.* Improvement is clearly possible and occasionally impressive during a single session. However, the lack of online learning and retention of the content between sessions negates any improvement in subsequent sessions, at least in the use of the freely available LLM. I do not know whether in for-pay environments the content may be preserved from session to session.
- *LLM reasoning under uncertainty.* In the simple problems attempted, LLM could not conduct probability computations reliably.

In summary, fundamental advancements in LLM are required to achieve sufficient performance predictability for building actionable models of LLM decision-making capabilities as a function of problem complexity. At least, such predictability was not discernable at the current state of freely available models, given the size of the test set and problem types. However, based on the tests, I feel cautiously optimistic about potential improvements if the development focus is brought to specific problem classes.

Acknowledgments

I would like to thank Drs. Joshua Moser and Javier Puig-Navarro for insightful comments and a careful review. I am also grateful for the freely available use of Large Language Models and would like to acknowledge chatGPT 4, Claude, and Bing, used in this study. Because performance assessment of individual LLM was not an objective of this paper, I did not attribute responses to specific LLM.

References

- [1] Alexandrov, N. M., “Dynamic Trust and Authority Assignment in Autonomous Multiagent Teams,” *AIAA Aviation Forum and ASCEND 2024*, AIAA 2024-4859, Vol. <https://doi.org/10.2514/6.2024-4859>, AIAA, Washington, DC, 2024.
- [2] Turing, A., “Computing Machinery and Intelligence,” *Mind*, Vol. doi:10.1093/mind/LIX.236.433, 1950, pp. 433–460.
- [3] Du, M., “Machine vs. human, who makes a better judgment on innovation? Take GPT-4 for example,” *Frontiers of Artificial Intelligence*, Vol. 6:1206516, No. doi: 10.3389/frai.2023.1206516, 2023.
- [4] Dev, S. I., Khader, A. M., Begerowski, S. R., Anderson, S. R., Clément, G., and Bell, S. T., “Cognitive performance in ISS astronauts on 6-month low earth orbit missions,” *Frontiers in Psychology*, 2024. <https://doi.org/10.3389/fphys.2024.1451269>.
- [5] Tversky, A., and Kahneman, D., “Judgment under Uncertainty: Heuristics and Biases,” *Science*, Vol. 185, No. 4157, 1974, pp. 1124–1131.
- [6] Glazer, E., Erdil, E., Besiroglu, T., Chicharro, D., Chen, E., Gunning, A., Falkman Olsson, C., Denain, J.-S., Ho, A., de Oliveira Santos, E., Järvinen, O., Barnett, M., Sandler, R., Vrzsala, M., Sevilla, J., Ren, Q., Pratt, E., Levine, L., Barkley, G., Stewart, N., Grechuk, B., Grechuk, T., Varma Enugandla, S., and Wildon, M., “FRONTIERMATH: A Benchmark for Evaluating Advanced Mathematical Reasoning in AI,” *arXiv*, 2024. [https://doi.org/arXiv:2411.04872v3\[cs.AI\]](https://doi.org/arXiv:2411.04872v3[cs.AI]).
- [7] Ben-Michael, E., Greiner, D., Huang, M., Imai, K., and Jiang, S., Z adn Shin||, “Does AI help humans make better decisions? A statistical evaluation framework for experimental and observational studies,” *arXiv*, 2024. [https://doi.org/arXiv:2403.12108v3\[cs.AI\]](https://doi.org/arXiv:2403.12108v3[cs.AI]).

- [8] Kopardekar, P., Schwartz, A., Magyarits, S., and Rhodes, J., “Airspace Complexity Measurement: An Air Traffic Control Simulation Analysis,” *International Journal of Industrial Engineering*, Vol. 16, No. 1, 2009, pp. 61–70.
- [9] Grote, G., Weik, S., Wäfler, T., and Zölch, M., “Criteria for the complementary allocation of functions in automated work systems and their use in simultaneous engineering projects,” *International Journal of Industrial Ergonomics*, Vol. 16, 1995, pp. 367–382.
- [10] Bainbridge, L., “Ironies of Automation,” *Analysis, Design and Evaluation of Man-Machine systems*, edited by G. Johanssen and J. Rijnsdorp, Pergammon Press, Oxford, 1982, pp. 129–135.
- [11] Jordan, N., “Allocation of functions between man and machines in automated systems,” *Journal of Applied Psychology*, Vol. 47, No. 3, 1963, pp. 161–165.
- [12] Alexandrov, N. M., “MAGE: Alleviating Uncertainty in Real Time Decision Making as a Function of Problem Complexity,” *AIAA SciTech Forum 2022*, AIAA 2022-0813, Vol. DOI: 10.2514/6.2022-0813, AIAA, San Diego, CA, 2022.
- [13] Olekhnik, C., Nesterenko, Y., and Potapov, M., *Starinnye zanimatel'nye zadachi*, Nauka, Moscow, 1985.
- [14] Yaglom, A., and Yaglom, I., *Challenging Mathematical Problems with elementary Solutions*, Dover, New York, 1987.
- [15] Kordemsky, B., *The Moscow Puzzles*, Dover, New York, 1992.