

Can the “Art” of Mathematical Modeling in Industrial Engineering be Automated by AI?

Shrushti Mardikar, Abdullah Aldhuhayyan and Vittaldas V. Prabhu

Department of Industrial and Manufacturing Engineering

Pennsylvania State University

University Park, PA, USA

sum403@psu.edu, axa1534@psu.edu, vittal.prabhu@psu.edu

Abstract

The capacity of Generative Artificial Intelligence (AI) models in formulating optimization problems is an interesting area of exploration in this rapidly evolving field. This study explores the capability of AI to interpret and formulate mathematical modeling problems from English descriptions. Five Large Language Models (LLMs) were selected, including OpenAI's ChatGPT-4, Google's Gemini, Microsoft's Copilot, Anthropic's Claude, and an open-sourced model by Meta Llama-2. The research is conducted through a systematic comparison between human-expert generated formulations and those produced by the LLMs to better understand the strengths and shortcomings of the five LLMs. A diverse set of 26 linear programming problems was used for this evaluation. The effectiveness of these AI tools is measured based on the correctness of the formulations. ChatGPT-4 outperformed its competitors with a mean score of 88.55, followed by Copilot at 84.93, Gemini at 83.57, Claude at 81.21, and Llama-2 at 46.26. The test problems were example in Linear programming from typical junior level course in industrial engineering and were graded using a rubric. Overall, ChatGPT-4 was the best earning a "B+" grade compared to others Copilot ("B"), Gemini ("B-"), Claude ("B-"), and Llama-2 ("F"). These findings indicate considerable variation in current Generative AI technologies in their ability to automatically formulate mathematical optimization problems. There are interesting opportunities to harness these technologies as they continue to rapidly evolve in research, education, and practice.

Keywords

Mathematical Modeling, Optimization, Linear Programming, ChatGPT, Copilot, Gemini, Llama-2

1. Introduction

Generative AI models, leveraging neural networks, have shown remarkable capabilities in creating content across various domains by learning patterns and structures within datasets (Ray, 2023). These models, encompassing GANs, LLMs, VAEs, and transformers, draw inspiration from human neuronal processes and refine their outputs through repeated training, aiming to generate high-quality and contextually appropriate content (ZDNet, 2020). The evolution of Generative AI models has significantly impacted various sectors, including education (Bandi et al., 2023). Interestingly, Large Language Models (LLMs) have been shown to be effective tools for figuring out the intricacies of solving mathematical puzzles (Romera-Paredes et al., 2023; Imani, Du & Shrivastava, 2023). Generative models like ChatGPT-4 (Brown et al., 2020), Bard (now Gemini), and Bing's AI (now Copilot) have shown remarkable abilities in interpreting and generating natural language, which extends to complex tasks such as mathematical problem formulation (Vaswani et al., 2017; Language Understanding, 2019). Generative AI has opened new avenues for applications such as computer vision, natural language processing, and creative arts (Language Understanding, 2019).

Mathematical modeling for optimization is a core methodology in industrial engineering and this usually requires understanding constraints and objectives of the application at hand from discussions and documents. Developing such

models requires considerable time and engineering effort. Once the mathematical model is developed it is usually coded into some solver. As the application complexity increases there is usually a multitude of ways to formulate it, and some formulations may be much more computationally efficient than others. Some consider formulation of models as an “art” and algorithms for their solution as the “science” of mathematical optimization. This paper explores the ability of extant Generative AI technologies to take natural language description of applications and formulate models – the “art” of mathematical optimization.

Linear program (LP) is a class of mathematical optimization problem in which the objective and constraints are linear and are usually computationally tractable compared to nonlinear problems. LPs arise in a wide range of real-world applications in industrial engineering and operations management. This study adds to the body of knowledge on artificial intelligence by attempting to create a more comprehensive understanding of the mathematical reasoning skills and abilities of LLMs in mathematical formulation of LP (Testolin, 2023; Lu et al., 2023). In the upcoming sections, we first describe the problem, laying the foundation to understand the complexities and significance of our investigation. Following this, we navigate through the existing literature, pinpointing the gap our study aims to fill. The results and discussion section reveals our empirical findings. Finally, we conclude by summarizing the impact and suggesting avenues for future works.

2. Problem Description

One of the main challenges in the field of industrial engineering is the fundamental complexity of mathematical optimization problems, which necessitates the precise formulation of objectives and constraints. We have access to solvers like CPLEX, GAMS, and Lingo, which provide strong support for solving these problems. However, rather than concentrating on the solution stage, the crucial aspect that educational institutions want to cultivate in students is the ability to take a business challenge and successfully formulate it into an LP problem.

In our comprehensive literature review, we have identified a gap in the evaluation of the emerging technologies in the context of formulating of linear programming problems. While many studies evaluate solvers for solving linear programming, only a small number examine the formulation of optimization problems.

This study's main goal is to find out how well LLMs, especially ChatGPT-4 from OpenAI, Gemini from Google, and Copilot from Microsoft, Claude from Anthropic, Llama-2 from Meta, can formulate mathematical optimization problems using a prompt in simple English. Finding decision variables, objective functions, and constraints that accurately reflect the problem as it is stated in natural language is a necessary step in the problem formulation process.

The potential for generative AI to automate the roles of copy editors has been recognized (Chui et al., 2024). This leads to the question: To what extent can generative AI tools effectively automate the development of LP models typically handled by students/professionals?

3. Related Research

The ascendancy of LLMs as potent tools for automating intricate tasks, including mathematical problem-solving, underscores the evolving landscape of AI. LLMs, characterized by their ability to process and generate natural language text, have been developed over decades, transitioning from rule-based models to advanced neural language models capable of understanding the semantics and context of language. The introduction of models like GPT-3 by OpenAI has demonstrated the potential of LLMs to generate coherent and natural-sounding text, propelling a new era of research and application in various fields, including mathematics (Ahn et al., 2024).

In engineering fields, the application of LLMs like ChatGPT has been explored with varying degrees of success. While ChatGPT has shown promise in areas such as software engineering, its reliability in solving engineering problems, particularly in mechanical engineering, has been questioned due to instances of incorrect procedures, formulas, or results. Similarly, in mathematics education, ChatGPT has been found to hold potential as a teaching assistant, offering customized examples and problem-solving strategies. However, its limitations in generating accurate responses and addressing complex mathematical concepts necessitate its use as a supplemental tool alongside traditional teaching method (Pursnani et al., 2023).

The integration of LLMs into educational settings, particularly in disciplines like Industrial engineering, where there's a notable knowledge gap regarding their performance, presents an opportunity for further research. Understanding ChatGPT's proficiency in domains with complex and multidisciplinary topics can provide valuable insights into its applicability and limitations, contributing to the responsible and effective use of AI models in education (Pursnani et al., 2023).

While generative AI models, particularly LLMs, hold significant promise in automating complex tasks and enhancing educational technologies, their practical applications and limitations require thorough investigation. This study's findings contribute to the discourse on AI-assisted learning and problem-solving, highlighting the need for continued exploration and development in AI-powered educational technology (Bandi et al., 2023).

4. Methodology

To effectively solve engineering systems problems, they must be translated into explicit mathematical optimization formulations. This requires careful decision-making in three key areas: defining decision variables, and comparison metrics, through objective and constraint functions (Crossley et al., 2017).

Our methodology is illustrated by the following testing procedure:

- 26 Problems were sourced from undergraduate textbooks and online platforms that included a variety of LP problems including transportation, networks, and integer programming problems. (Vazirani, 2001, Chapter 7; Winston, 2004; Beasley, 2024)
- All problems were presented to the LLM in their original format from the textbook/references. This ensures the problems are presented in a manner like how an instructor would present them to students.
- The dataset was tested using five prominent and accessible generative AI models were selected from leading tech companies, chosen for their technological foundation, market presence, and demonstrated potential impact on AI-assisted tasks.
 - OpenAI's ChatGPT: ChatGPT-4 is trained to follow an instruction in a prompt and provide a detailed response. It is designed to interact conversationally and capable of answering follow-up questions. (OpenAI, 2023)
 - Google's Gemini: Gemini, by Google DeepMind, is a multimodal AI model designed to understand and integrate different types of information, aiming to act as an intuitive and expert assistant (Google DeepMind, 2023).
 - Microsoft's Copilot: Microsoft Copilot integrates AI across Microsoft's products, enhancing coding, productivity, and search with a unified experience that provides contextual assistance (Microsoft, 2023).
 - Anthropic's Claude: Claude is Anthropic's first multimodal chatbot, renowned for its fast, capable, and highly coherent responses across text, voice, images, and documents (Anthropic, 2023).
 - Meta's Llama-2: Llama-2-70b-chat an advanced open-sourced LLM, designed for engaging and insightful conversations, leveraging a massive parameter architecture (Meta AI, 2023).
- Solutions from human experts present in the textbook/reference were also compiled for these problems to serve as a benchmark against the AI-generated solutions. These were then compared against each other and evaluated using a success matrix.
- The primary metric for evaluation is the accuracy of the AI-generated problem formulations compared to human-generated solutions. This involves assessing how well the AI models capture the essence of the problem by obtaining the total score of the three key metrics (Table 1).

Table 1. Success metrics

Category	Score (%)	Scoring Method
Objective	5%	Identifies the max/min function correctly
	5%	All the arithmetic signs are correctly identified
	20%	$\frac{\text{Correctly identified terms in objective function}}{\text{Total terms in objective function}} * 20\%$
Constraints	25%	$\frac{\text{Correctly identified constraints}}{\text{Total constraints}} * 25\%$
	25%	$\frac{\text{Correctly identified non – misinterpreted constraints}}{\text{Total constraints}} * 25\%$
Decision Variables	20%	$\frac{\text{Correctly identified decision variables}}{\text{Total decision variables}} * 25\%$

- Example: A post office requires different numbers of full-time employees on different days of the week. The number of full-time employees required on each day is given in Table 2. Union rules state that each full-time employee must work five consecutive days and then receive two days off. For example, an employee who works Monday to Friday must be off on Saturday and Sunday. The post office wants to meet its daily requirements using only fulltime employees. Formulate an LP that the post office can use to minimize the number of full-time employees who must be hired. (Beasley, 2024)

Table 2. Number of full-time employees required on each day

Day	Number of Full-time Employees Required
1 = Monday	17
2 = Tuesday	13
3 = Wednesday	15
4 = Thursday	19
5= Friday	14
6 = Saturday	16
7 = Sunday	11

Textbook Solution:

- Decision variables:
 X_i = number of employees beginning work on day ($i = 1, 2, 3, 4, 5, 6, 7$)
- Objective function:
 $\min Z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$
- Constraints:
 Monday constraint: $x_1 + x_4 + x_5 + x_6 + x_7 \geq 17$
 Tuesday constraint: $x_1 + x_2 + x_5 + x_6 + x_7 \geq 13$
 Wednesday constraint: $x_1 + x_2 + x_3 + x_6 + x_7 \geq 15$
 Thursday constraint: $x_1 + x_2 + x_3 + x_4 + x_7 \geq 19$
 Friday constraint: $x_1 + x_2 + x_3 + x_4 + x_5 \geq 14$
 Saturday constraint: $x_2 + x_3 + x_4 + x_5 + x_6 \geq 16$
 Sunday constraint: $x_3 + x_4 + x_5 + x_6 + x_7 \geq 11$
 $x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0$

ChatGPT-4 Solution:

- Decision variables:
Let x_i represent the number of full-time employees starting their workweek on day i , where i ranges from 1 to 7, corresponding to Monday through Sunday, respectively.
- Objective function:
 $\min Z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$
- Constraints:
Monday's requirement: $x_1 + x_4 + x_5 + x_6 + x_7 \geq 17$
Tuesday's requirement: $x_1 + x_2 + x_5 + x_6 + x_7 \geq 13$
Wednesday's requirement: $x_1 + x_2 + x_3 + x_6 + x_7 \geq 15$
Thursday's requirement: $x_1 + x_2 + x_3 + x_4 + x_7 \geq 19$
Friday's requirement: $x_1 + x_2 + x_3 + x_4 + x_5 \geq 14$
Saturday's requirement: $x_2 + x_3 + x_4 + x_5 + x_6 \geq 16$
Sunday's requirement: $x_3 + x_4 + x_5 + x_6 + x_7 \geq 11$
 $x_i \geq 0 \forall i = 1, 2, 3, 4, 5, 6, 7$

Gemini Solution:

- Decision variables:
 x_1 : Number of full-time employees starting on Monday
 x_2 : Number of full-time employees starting on Tuesday
 x_3 : Number of full-time employees starting on Wednesday
 x_4 : Number of full-time employees starting on Thursday
 x_5 : Number of full-time employees starting on Friday
- Objective function:
Minimize $Z = x_1 + x_2 + x_3 + x_4 + x_5$
- Constraints:
Day 1 (Monday): $17 \geq x_1$
Day 2 (Tuesday): $13 \geq x_1 + x_2$
Day 3 (Wednesday): $15 \geq x_2 + x_3$
Day 4 (Thursday): $19 \geq x_3 + x_4$
Day 5 (Friday): $14 \geq x_4 + x_5$
Day 6 (Saturday): $16 \geq x_5 + x_1$
Day 7 (Sunday): $11 \geq x_1 + x_2 + x_3$

Copilot Solution:

- Decision variables:
Let's denote the number of full-time employees starting work on each day as follows:
 x_1 : Employees starting on Monday
 x_2 : Employees starting on Tuesday
 x_3 : Employees starting on Wednesday
 x_4 : Employees starting on Thursday
 x_5 : Employees starting on Friday
 x_6 : Employees starting on Saturday
 x_7 : Employees starting on Sunday
- Objective function:
Minimize $Z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$
- Constraints:
Monday constraint: $x_1 + x_4 + x_5 + x_6 + x_7 \geq 17$
Tuesday constraint: $x_1 + x_2 + x_5 + x_6 + x_7 \geq 13$
Wednesday constraint: $x_1 + x_2 + x_3 + x_6 + x_7 \geq 15$
Thursday constraint: $x_1 + x_2 + x_3 + x_4 + x_7 \geq 19$
Friday constraint: $x_1 + x_2 + x_3 + x_4 + x_5 \geq 14$
Saturday constraint: $x_2 + x_3 + x_4 + x_5 + x_6 \geq 16$
Sunday constraint: $x_3 + x_4 + x_5 + x_6 + x_7 \geq 11$

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0$$

Claude Solution:

- Decision variables:
 x_1 : Number of employees working M - F
 x_2 : Number of employees working T - Sa
 x_3 : Number of employees working W - Su
 x_4 : Number of employees working Th - M
 x_5 : Number of employees working F - Tu
- Objective function:
Minimize Total Employees = $x_1 + x_2 + x_3 + x_4 + x_5$
- Constraints:
Monday constraint: $x_1 + x_4 \geq 17$
Tuesday constraint: $x_1 + x_5 \geq 13$
Wednesday constraint: $x_1 + x_3 \geq 15$
Thursday constraint: $x_2 + x_4 \geq 19$
Friday constraint: $x_1 + x_2 \geq 14$
Saturday constraint: $x_2 + x_3 \geq 16$
Sunday constraint: $x_3 + x_5 \geq 11$
 $x_1, x_2, x_3, x_4, x_5 \geq 0$

Llama-2 Solution:

- Decision variables:
Let X_1, X_2, \dots, X_7 be the number of full-time employees working on each day, respectively
- Objective function:
The total number of employees hired must be minimized:
Minimize: $\sum_{i=1}^7 X_i$
- Constraints:
The union rules can be modeled by the following constraints:
 1. For each day i , the number of employees working on that day must be at least 5:
 $5 \leq X_i \leq 10$ ($i = 1, 2, \dots, 7$)
 2. For each employee, the number of consecutive days worked must be 5, and the number of days off must be 2:
For each employee, let D_i be the day they start working, and let E_i be the number of days they work consecutively. Then:
 $E_i \leq 5$ ($i = 1, 2, \dots, 7$)
 $D_i + 4 \leq E_i + 1$ ($i = 1, 2, \dots, 7$)
 3. The total number of employees working on each day must be at least the number required:
 $17 \leq X_1 + X_2 + \dots + X_7 \leq 28$ (since 17 is the maximum number of employees required on any day) (Table 3)

Table 3. Results for the example

		Textbook Solution (%)	ChatGPT-4 (%)	Gemini (%)	Copilot (%)	Claude (%)	Llama-2 (%)
Decision variables	Number of variables	20	20	14.28	20	0	20
Objective Function	Number of terms	20	20	14.28	20	14.28	20
	Annotations	5	5	100	5	5	5
	Signs	5	5	25	5	5	5
Constraints	Non-Missing constraints	25	25	21.87	25	25	0
	Non-Misinterpreted constraints	25	25	0	25	0	0
Total Score		100	100	60.45	100	49.29	50
Grade			A	C	A	F	F

In the above example, ChatGPT-4 and Copilot both scored 100%. This shows that they could accurately satisfy the requirements in their formulations, correctly interpreting the decision variables, objective function, and constraints. Gemini, for the above example, scored 60.45%. Gemini failed to capture the necessary decision variables and constraints. Claude scored the lowest at 49.29% because it missed several decision variables and didn't interpret the constraints correctly, showing a weaker understanding of this problem. Llama-2 also scored low at 50%, mainly because it incorrectly added new rules about workdays that didn't match the original problem setup.

5. Results and Discussion

The comparative analysis of LLMs in LP problem formulation shows GPT standouts, securing the highest overall mean (Table 4) and excelling in formulating objective functions and constraints (Figure 1). Copilot and Gemini, with comparable overall.

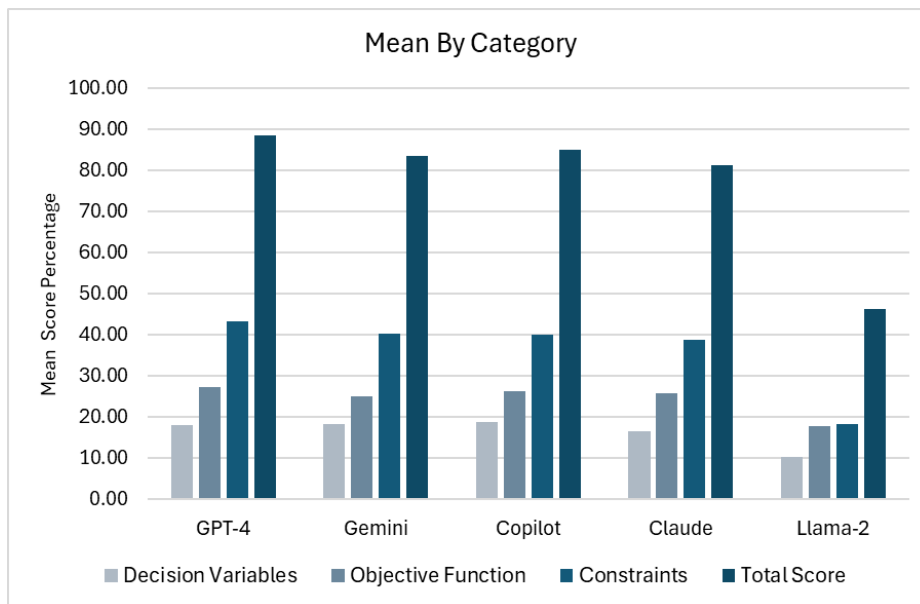


Figure 1. illustrates the mean score percentage of each main category for all models.

Table 4. Overall results

	Textbook Solution (%)	ChatGPT-4 (%)	Gemini (%)	Copilot (%)	Claude (%)	Llama-2 (%)
Decision variables	20	18	18.28	18.69	16.63	10.24
Objective Function	30	27.29	24.99	26.44	25.87	17.73
Constraints	50	43.16	40.29	40.06	38.71	18.29
Total Score	100	88.55	83.57	85.18	81.21	46.26
Grade		B+	B-	B	B-	F

means, demonstrate strength for defining decision variables—Copilot narrowly surpasses Gemini. Claude maintains a good performance, slightly trailing Gemini. Llama-2, in contrast, has significantly lower scores in all categories, suggesting difficulties in accurately formulating LP problems (Table 5).

Table 5. MSE, RSME and MAPE values foe models

Model	MSE	RMSE	MAPE
ChatGPT	579.04	24.06	11.45
Gemini	469.12	21.66	14.92
Copilot	418.54	20.46	15.07
Claude	703.82	26.53	18.79
Llama-2	3422.50	58.50	53.74

To provide a clear overview of models’ predictive accuracy and error characteristics, we analyzed the performance metrics. The RMSE values, which measure the average magnitude of the errors in prediction, show that Llama-2 has the highest error rate with an RMSE of 58.502, significantly higher than the other models, indicating less accuracy in its formulation. In contrast, Copilot displays the lowest RMSE at 20.458, suggesting it makes more precise predictions compared to the other models. Similarly, MAPE values, which express prediction errors as a percentage, provide additional insight into model performance. Llama-2 again underperforms, with a very high MAPE of 53.740%, indicating significant relative errors in its predictions. ChatGPT, on the other hand, has the lowest MAPE at 11.454%, indicating that its predictions are the most proportionally accurate. This comparison of RMSE and MAPE values across the models reveals significant differences in performance, with ChatGPT and Copilot, and Gemini generally providing more reliable and accurate predictions, whereas Llama-2 lags behind in both metrics, indicating less accuracy in predictions and lower reliability.

The non-Gaussian nature of the data from ChatGPT, Gemini, Copilot, and Claude precludes the use of statistical tests that assume normality, such as ANOVA or paired t-tests. The Shapiro-Wilk test results for the datasets of various models show a clear distinction in their distribution characteristics. For ChatGPT, Gemini, Copilot, and Claude, the test statistics are less than 1, and their p-values are significantly low (ranging from 0.000 to 0.005), leading us to reject the null hypothesis that these samples come from a Gaussian distribution. Conversely, for Llama-2, the test statistic is

0.978 and the p-value is 0.831, indicating a failure to reject the null hypothesis, suggesting that this sample could plausibly come from a Gaussian distribution.

These tests are sensitive to normality assumptions and using them on non-normal data can lead to incorrect conclusions. Therefore, alternative non-parametric methods, which do not assume a normal distribution, such as the Kruskal-Wallis test, are more appropriate for analyzing these data to ensure robust, reliable results.

Following the significant findings from the Kruskal-Wallis test, which showed a notable difference in score distributions among the LLMs with a p-value of approximately 5.31×10^{-9} , we decisively reject the null hypothesis of equal medians across groups. This statistically significant result implies that at least one model's performance is different from the others. To pinpoint these differences, it is necessary to proceed with post-hoc analysis. The Dunn's test is an appropriate choice for this next step, as it facilitates pairwise comparisons between groups.

Table 6. Dunn's post-hoc test results

ChatGPT	0.380567
Claude	0.000159
Copilot	0.001535
Gemini	0.001586
Llama-2	3.07e-15

The Dunn's post-hoc test results, analyzing differences between the LLMs and the Textbook solutions (Table 6), reveal insightful patterns in performance disparities. Notably, while ChatGPT shows no significant difference from the textbook, suggesting its performance closely mirrors the ideal solutions, models like Claude, Copilot, and Gemini exhibit divergences from the textbook standards, implying areas of strength or weakness that merit further exploration. Llama-2 stands out distinctly, showing significant differences from both the textbook and all other models, indicating that its performance is substantially different, typically underperforming. Among the models themselves, except for their comparisons with Llama-2, no significant differences emerge, suggesting a generally comparable performance level across these models for certain tasks.

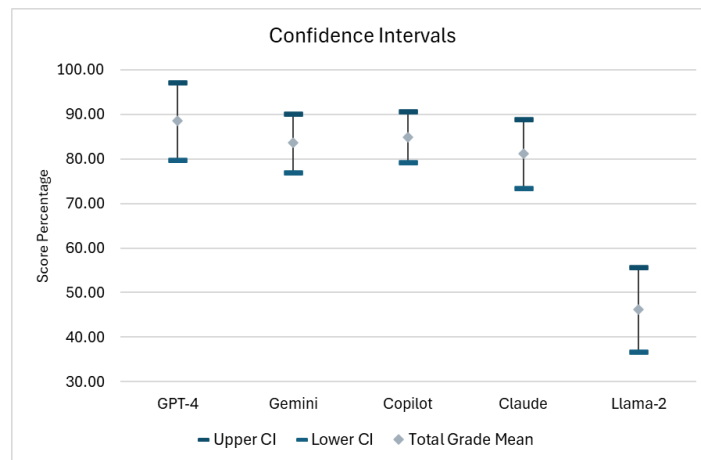


Figure 2. illustrates the score percentage with 95% Confidents Interval.

Lastly, variability in model performance is determinable through standard deviations and margins of error (Evans, 2021), with GPT exhibiting high variation despite its top performance. The 95% confidence intervals underscore score reliability; Copilot and Gemini's tight intervals indicate high consistency and a robust grasp of LP formulation elements, while Llama-2's wider intervals denote less predictability in its formulations (Figure 2).

The analysis reveals distinct formulation challenges among the LLMs. In Claude's instance of incorrect decision variables definition yet managed to formulate an accurate objective function, scoring a full mark in the latter category. A recurring issue, most notably with Llama-2, is the tendency to describe constraints verbally without proper mathematical expression, leading to lower scores due to our emphasis on precise mathematical representation. Furthermore, Copilot and Gemini often produced concise formulations, indicating efficiency in model optimization policy, though potentially increasing interpretative effort. Notably, across all models, the direction of optimization was consistently identified correctly. These insights into the models' performance highlight the strengths and areas for improvement in automated mathematical formulation.

6. Conclusions

The study investigates the LLMs' efficacy and seeks to understand how these models perform in formulating LP problems. GPT, Gemini, Copilot and Claude are skilled at formulating LP problems. They demonstrate a notable ability to define decision variables, formulate objective functions and constraints. This indicates that LLMs prove to be a useful aid for formulating LP problems. However, it is crucial to highlight that one should not solely depend on the Generative AI tool, as our findings point out their tendency to make errors.

In our study we found that ChatGPT stands out as the best performing model but exhibited high score variation. On the other hand, Copilot and Gemini show low variations among all models and thus being reliable options for applications that demand accurate linear programming problem formulation, particularly in defining decision variables. Llama exhibits markedly lower performance, suggesting that it may need significant enhancements or might be less appropriate for tasks necessitating exact mathematical modeling.

In conclusion, the extent to which generative AI tools effectively automate the development of LP models typically handled by students/professionals is limited. Our results recommend using these tools with caution, acknowledging their value but also their limitations due to their variable dependability and tendency of occasional mistakes.

There are interesting future opportunities in research, education, and practice that could build off of this exploratory work. In terms of research more tests should be conducted with varying levels of difficulty non-linear problems including quadratic objectives and integer variables. As this area of research matures, it can be expected to find applications in education such as serving as a real-time tutor for students learning mathematical modeling. A longer term objective for this line of research is to develop technologies that can be scaled to address industrial scale problems and also in developing interfaces appropriate for operational managers to readily solve their problems. The scope can also be broadened by incorporating more AI models and newer versions. The number of test problems can be increased in terms of size and complexity to provide more insights on improving and applying them in practice. A set of benchmark problems could be made publicly available as the interest in this area grows. The study considers only the accuracy of the solution, not other factors such as the computational time and other performance metrics of the generative AI, which is a limitation that can be addressed in the future.

References

- Anthropic, 2023. *Introducing Claude*. Available at: <https://www.anthropic.com/news/introducing-claude>. Accessed: Feb. 18, 2024.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, 2017. *Attention is All You Need*. In Proceedings of NeurIPS 2017.
- Bandi, A., Adapa, P. V., and Kuchi, Y. E., *The power of Generative AI: A review of requirements, models, input-output formats, evaluation metrics, and challenges*. Future Internet, 15(8), 260. 2023. Available at: <https://doi.org/10.3390/fi15080260>.
- Beasley, J.E., OR-Notes, Accessed [Jan, 2024]. Available from: <https://people.brunel.ac.uk/~mastjjb/jeb/or/rights.html>
- Bernardino Romera-Paredes, Mohammadamin Barekatin, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli & Alhussein Fawzi, *Mathematical discoveries from program search with large language models*. Nature, pp. 1-3. 2023.
- Brown Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger,

- Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei, 2020. *Language Models are Few-Shot Learners*. In Proceedings of NeurIPS 2020.
- Chaoning Zhang, Chenshuang Zhang, Sheng Zheng, Yu Qiao, Chenghao Li, Mengchun Zhang, Sumit Kumar Dam, Chu Myaet Thwal, Ye Lin Tun, Le Luang Huy, Donguk kim, Sung-Ho Bae, Lik-Hang Lee, Yang Yang, Heng Tao Shen, In So Kweon, Choong Seon Hong, 2023. *A Complete Survey on Generative AI (AIGC): Is ChatGPT from GPT-4 to GPT-5 All You Need?* arXiv:2303.11717.
- Chui, M. et al., *The economic potential of Generative AI: The Next Productivity Frontier*. 2024.
- Evans, J. R., *Business Analytics: Methods, Models, and Decisions*. Harlow, England: Pearson. 2021.
- Google DeepMind, *Introducing Gemini: A New Generation of AI*. 2023. Available at: <https://blog.google/technology/ai/google-gemini-ai/>.
- Imani, S., Du, L., and Shrivastava, H., *Mathprompter: Mathematical reasoning using large language models*. In Proceedings of the ACL, pp. 37-42. 2023.
- Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, Wenpeng Yin, *Large Language Models for Mathematical Reasoning: Progresses and Challenges*. arXiv:2402.00157. 2024.
- Journal of AI in Education, *Generative AI models in education: Opportunities and limitations*. Journal of AI in Education, 2(2), pp. 20-30. 2023.
- Meta AI, *Introducing LLaMA: A foundational, 65-billion-parameter large language model*. Meta AI Blog. 2023. Available at: <https://ai.meta.com/blog/large-language-model-llama-meta-ai/>.
- Microsoft, *Microsoft Copilot: Your Everyday AI Companion*. 2023. Available at: <https://blogs.microsoft.com>.
- NAACL-HLT, *Language Understanding*. In Proceedings of NAACL-HLT 2019.
- OpenAI. ChatGPT (Version 4). 2023. Available at: <https://www.openai.com/chatgpt>
- Pan Lu, Liang Qiu, Wenhao Yu, Sean Welleck, Kai-Wei Chang, *A survey of deep learning for mathematical reasoning*. In Proceedings of ACL, pp. 14605-14631. 2023.
- Ray, P. P., *ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope*. Internet of Things and Cyber-Physical Systems, 3, pp. 121-154. 2023. Available at: <https://doi.org/10.1016/j.iotcps.2023.04.003>.
- Testolin, A., *Can neural networks do arithmetic? A survey on the elementary numerical skills of state-of-the-art deep learning models*. CoRR, abs/2303.07735. 2023.
- Vazirani, V. *Algorithms*. (n.d.). Retrieved from <https://people.eecs.berkeley.edu/~vazirani/algorithms/chap7.pdf>
- Vinay Pursnani, Yusuf Sermet, Ibrahim Demir, *Performance of ChatGPT on the US fundamentals of engineering exam: Comprehensive assessment of proficiency and potential implications for professional environmental engineering practice*. CAEAI, 2023. DOI: 10.1016/j.caeai.2023.100183, License CC BY-NC-ND 4.0.
- William A. Crossley, Siyao Luan, James T. Allison, Deborah L. Thurston, *Optimization problem formulation framework with application to engineering systems*. Syst. Eng., 20(6), pp. 512–528. 2017. DOI: <https://doi.org/10.1002/sys.21418>.
- Winston, W. L., *Operations Research: Applications and Algorithms*. Fourth Edition. Indiana University, Curt Hinrichs. 2004.
- ZDNet, *The evolution and impact of Large Language Models in AI*. 2020. Available at: <https://www.zdnet.com/article/the-evolution-and-impact-of-large-language-models-in-ai/>.

Biographies

Shrushti Mardikar is currently advancing towards a Master's degree in Industrial Engineering and Operations Research at the Harold and Inge Marcus Department of Industrial and Manufacturing Engineering, Pennsylvania State University. Shrushti's research focuses on optimizing supply chain dynamics through advanced simulation and data analytics. Key areas of interest include systems modeling, bottleneck analysis, and the application of Python-based tools to enhance supply chain efficiency. Shrushti is committed to leveraging these competencies to contribute to research that improves industrial operations and supply chain sustainability.

Abdullah Aldhuhayyan is currently advancing towards a PhD at the Harold and Inge Marcus Department of Industrial and Manufacturing Engineering, Pennsylvania State University. Building on the foundation of an M.S. in Industrial Engineering from the same institution, Abdullah's research focuses on the intersection of digital transformation and industrial engineering. Key areas of interest include generative AI, large language models, and digital twin technology.

Dr. Vittal Prabhu holds the Charles and Enid Schneider Faculty Chair in Service Enterprise Engineering at Pennsylvania State University and chairs the Consortium for Digital Enterprises (CODE) since August 2017. Vittal works in the area of distributed control systems focused on manufacturing and service enterprises. The goal of his work is to develop a unified mathematical and computational framework that is suitable for engineering of distributed systems consisting of discrete-events, physical processes, and service processes. He has developed distributed control algorithms for production scheduling, maintenance scheduling, batch sequencing, inventory control, transportation, and supply chain optimization. He has also researched real-time control of physical processes such as cryogenic freezing, laser manufacturing, and electron beam-physical vapor deposition.