

Natural Language to Problem Formulation for the ScheLoc Optimization Problem Using Large Language Models*

Maciej HOJDA and Grzegorz POPEK

Faculty of Computer Science and Telecommunications, Wrocław University of Science and Technology
Janiszewskiego 11/17, 50-372, Wrocław, Poland

ORCID: 0000-0003-3195-4862

ORCID: 0000-0002-3186-507X

Correspondence should be addressed to: Maciej HOJDA, maciej.hojda@pwr.edu.pl

*Presented at the 47th IBIMA International Conference, 29-30 June 2026, Madrid, Spain

Abstract

The manual transition from natural language descriptions to formal mathematical models is a critical yet underappreciated bottleneck in operations research. This paper evaluates the feasibility of using Large Language Models (LLMs), specifically ChatGPT and Gemini, to automate the formulation of the complex Scheduling-Location (ScheLoc) problem. Unlike existing literature that often focuses on basic optimization tasks, this study utilizes five diverse, real-world scenarios ranging from mobile diagnostic units to humanitarian food relief to test model performance. The evaluation employed two query strategies: one assessing independent problem recognition and another providing the specific “ScheLoc” designation as a prompt. Our findings reveal that while LLMs struggle to independently identify the specific problem nomenclature, they demonstrate a robust functional understanding by achieving 100% accuracy in identifying optimization criteria across all tested cases. Furthermore, providing the problem name significantly improved the models’ ability to extract correct variables, domains, and parameters. The results demonstrate that LLMs can effectively handle the heavy lifting of data extraction and initial mathematical formulation, allowing human experts to transition from primary modelers to high-level verifiers. This approach offers a feasible path toward streamlining complex modeling workflows and democratizing expert-level optimization for industries lacking specialized operations research staff.

Keywords: ScheLoc, Optimization Problem Formulation, Data Extraction, Large Language Model, LLM

Introduction

In artificial intelligence, there is a growing need to process texts given in natural language. The use of artificial neural networks, especially the transformer (Vasvani 2017) as the main method of modeling caused explosive growth of language processing methods. Able to be executed in parallel on graphical processing units the transformer networks quickly grew in popularity and, eventually, lead to the creation of what are now known as Large Language Models (LLMs) which are neural networks with learnable parameters ranging in the billions (Zhao 2024). LLMs have quickly become popular due to their unprecedented ability to solve various language processing tasks. New tools such as ChatGPT (Gill and Kaur 2023) quickly took place as personal assistants capable of aiding with virtually every issue the user could face. After the ChatGPT, which was based on the GPT-3 model with over 175 billion of parameters (Brown 2020) more models (Rae 2021, Smith 2022,

Chowdhery 2022, Touvron 2023a, Touvron 2023b, Abhimanyu, 2024) followed back to back with tools such as Microsoft Copilot, Google Gemini, Perplexity and Claude (Evolvagency 2026).

One of the subjects of text processing which is of particular interest is the ability of Large Language Models to handle optimization problems. Given in natural language, optimization problems require expert's knowledge to formulate in a strict, mathematical way. This formulation can then be used to solve the problem using various available tools such as general-purpose solvers or dedicated algorithms. It is crucial however to correctly process the text containing the description of the problem and this time- and effort-consuming task is often underappreciated. To ease the burden of manually formulating problem descriptions it is imperative to provide tools of automation. There are precedents for this type of work, as of 2023 a major nl4opt tournament (Ramamonjison 2023) compared various approaches in a competitive environment. Since then several authors (Tsouros 2023, Zhang 2024) have tackled the problem with varying results. Research on formulating optimization problems with the use of open-source models (Mostajabdaveh 2025) is also underway. The relative simplicity of optimization problems tested in the literature – typically as basic as the knapsack problem – does not give a full view of the modeling challenges that an expert faces when formulating real problems. Our goal is to partially plug this methodological gap by evaluating the capabilities of Large Language Models when processing a more complex problem which is the Scheduling-Location (or ScheLoc) problem (Hennes 2002).

ScheLoc problem

There are many formulations of the ScheLoc problem (Hennes 2002, Filcek 2021, Zhang 2022). We provide one of them as the basis for our considerations. We have a set of tasks $\mathbf{J} = \{1, 2, \dots, J\}$ and a set of known locations $\mathbf{I} = \{1, 2, \dots, I\}$. Each task j is characterized by ready time p_j and transportation speed v_j . Tasks have to be performed by a single machine i each and the execution time is known s_{ij} . We have a vector of binary variables $y = [y_1, y_2, \dots, y_I]$ where each variable y_i takes the value of 1 when the machine i is activated (and 0 if otherwise). Then we have binary variables x_{ijk} representing the schedule, where each takes the value of 1 when the task j is executed on machine i in position k .

Each task has a release date that is calculated by taking the ready time and the transportation time. This is given

$$r_j(x, y) = p_j + \frac{1}{v_j} \sum_{k=1}^J \sum_{i=1}^I y_i x_{jik} d_{ji}$$

as follows

where d_{ij} is a known distance between task's location and the machine location. Let $C_{ik}(x, y)$ be the time the task in position k ends on machine i , then the constraints are as follows

$$\begin{aligned}
\sum_{i=1}^I \sum_{k=1}^J x_{jik} &= 1, \forall j \in \mathbf{J} \\
\sum_{j=1}^J x_{jik} &\leq y_i, \forall i \in \mathbf{I}, \forall k \in \mathbf{J} \\
1 &\leq \sum_{i=1}^I y_i \leq I \\
\sum_{j=1}^J x_{ji(k+1)} - x_{jik} &\geq 0, \forall i \in \mathbf{I}, \forall k \in \mathbf{J} \\
C_{ik}(x, y) &\geq \sum_{j=1}^J r_j(x, y) + s_{ji}, \forall i \in \mathbf{I}, \forall k \in \mathbf{J} \\
C_{ik}(x, y) &\geq C_{i(k-1)}(x, y) + \sum_{j=1}^J s_{ji} x_{jik}, \forall i \in \mathbf{I}, \forall k \in \mathbf{J} \setminus \{1\} \\
y_i &\in \{0, 1\}, \forall i \in \mathbf{I} \\
x_{ijk} &\in \{0, 1\}, \forall j \in \mathbf{J}, \forall i \in \mathbf{I}, \forall k \in \mathbf{J} \\
Q(x, y) &= \max_i \sum_{k=1}^J C_{i,k}(x, y)
\end{aligned}$$

and the quality criterion is the makespan

Means and Methods

With the use of expert's knowledge we have formulated five scenarios of the ScheLoc problem. The scenarios are given as textual descriptions covering five realistic cases. Those are as follows.

- Scenario 1: Mobile Diagnostic Units (Healthcare)
- Scenario 2: Cold Storage Containers (Events/Retail)
- Scenario 3: Firefighting Drone Swarms (Emergency)
- Scenario 4: Field Service Technicians (IoT Infrastructure)
- Scenario 5: Humanitarian Food Relief (Disaster Response)

Those descriptions provide the necessary information to categorize the problem as ScheLoc, they contain descriptions of the variables, descriptions of the constraints and some numerical data. An example is given in Figure 1.

Look, we're having a real tough time with these screening tours. We've got three mobile mammography units roaming the county, but I feel like we're constantly missing our patients. People show up in waves — one day we've got a massive crowd outside a church in some small village on a Monday morning, and then two days later, nobody shows up because everyone's headed to the market in the bigger town. We need to figure out the math here so these buses actually park where the crowds are forming, not just where we guess they'll be. We know where the buses can stop, we just need to pick the right stopping points. Then we get to decide who gets screened at which bus. More people means longer duration but we know the approximate numbers for each group so we know how long it takes for each group and each bus. Groups take between 10 to 20 minutes to form, then it's a matter of walking down to the bus. They move, let's say 4 kilometers per hour on average. The distances between groups and buses are known so that's our timeline right there. Of course the groups cannot be

screened at the same time and we need to do it fast, so the doctors can go home quickly. Let's make sure the last group leaves as early as possible.

Figure 1. The Mobile Diagnostics Units ScheLoc formulation.

We feed the textual descriptions to a LLM while asking a questions to provide various data about the formulation. In the first query we ask about the type of the problem, the variables, the parameters, the constraints and the criteria. We finalize the question by asking to provide a mathematical description (in Latex) of the problem. The second query assumes that the problem is ScheLoc, provides this information to the LLM and asks the remaining questions as in the first query. The queries are provided in Figure 2 and 3.

You are given an optimization problem. Determine what type of problem it is (give it a name). Determine the variables and their domains (discrete versus continuous). Determine the known parameters and their values (if given). Determine the constraints. Determine the criteria. Provide the problem formulation in a mathematical, formal way. Give the formulation in Latex code. The problem is as follows.

Figure 2. Query 1.

You are given Scheduling and Location (ScheLoc) problem. Determine the variables and their domains (discrete versus continuous). Determine the known parameters and their values (if given). Determine the constraints. Determine the criteria. Provide the problem formulation in a mathematical, formal way. Give the formulation in Latex code. The problem is as follows.

Figure 2. Query 2.

We obtain the results and verify them with the use of an expert.

Results

The first fact noticed about the results is that neither LLM recognized the problem as ScheLoc. Most of the time, the problem was recognized as facility location and scheduling problem, which is not too far off.

In one case the LLM (ChatGPT) did not formalize all the constraints and instead provided only a textual description. This is in contradiction to the queries which demanded that problems be formulated in a mathematical, formal way. In some cases the LLM did not recognize that an ordering variable was needed leading to incorrect constraints. The encouraging fact is that the LLMs recognized the criterion perfectly.

We can spot a difference between the capabilities of ChatGPT and Gemini. The first LLM provided better results for cases when the ScheLoc problem was not given explicitly it had however made mistakes on the second query. Gemini was better for the case when ScheLoc was determined, however it failed on many cases when it was the opposite. Furthermore, in two Gemini cases we had to manually correct the latex structure.

We provide concise results in the following Table 1.

Table 1. Miss and hit results for the ChatGPT and Gemini LLMs

	Query 1 ChatGPT	Query 1 Gemini	Query 2 ChatGPT	Query 2 Gemini
Correct variable descriptions	1, 0, 1, 1, 0	0, 1, 1, 1, 1	1, 1, 1, 1, 1	1, 1, 1, 1, 1
Correct variable domains	1, 1, 1, 1, 1	1, 1, 1, 1, 1	1, 1, 1, 1, 1	1, 1, 1, 1, 1
Correct variables	0, 1, 1, 1, 1	0, 0, 0, 1, 1	1, 0, 1, 1, 1	1, 1, 1, 1, 1
Correct parameters	1, 1, 1, 1, 1	0, 1, 0, 1, 1	1, 0, 1, 1, 1	1, 1, 1, 1, 1
Correct constraints	0, 1, 1, 1, 1	0, 0, 0, 1, 0	1, 0, 1, 1, 1	1, 1, 1, 1, 1
Correct criterion	1, 1, 1, 1, 1	1, 1, 1, 1, 1	1, 1, 1, 1, 1	1, 1, 1, 1, 1

Conclusions

The results of this study confirm that utilizing Large Language Models (LLMs) to bridge the gap between natural language and formal optimization is not just a theoretical exercise but a feasible solution for complex problems like ScheLoc. While the models did not independently recognize the specific “ScheLoc” nomenclature, they successfully identified the core facility location and scheduling components, demonstrating a functional understanding of the problem's underlying architecture. The perfect accuracy in identifying optimization criteria across all five realistic scenarios—ranging from mobile healthcare units to firefighting drone swarms—further underscores the technical feasibility of this automated approach. Even with minor manual corrections required for LaTeX syntax or complex ordering constraints, the heavy lifting of data extraction and initial formulation was handled effectively by the AI.

The advantages of this application are manifold, primarily in reducing the time-consuming and often underappreciated burden of manual problem formulation. By moving beyond simple “knapsack” problems to address multifaceted, real-world logistics, LLMs allow human experts to transition from the role of primary modeler to that of a high-level verifier. This shift significantly streamlines the workflow, enabling faster deployment of optimization solutions in critical fields such as disaster response and IoT infrastructure. Ultimately, as LLM capabilities in tools like ChatGPT and Gemini continue to sharpen, the potential to democratize expert-level mathematical formulation through simple natural language becomes a tangible reality for industries that lack specialized operations research staff.

References

- Abhimanyu D, et al (2024) “The Llama 3 Herd of Models” arXiv:2407.21783
- Brown T, et al (2020) “Language Models are Few-Shot Learners” arXiv:2005.14165
- Chowdhery A, et al (2022) “PaLM: Scaling Language Modeling with Pathways” arXiv:2204.02311
- Evolvagency (2026) <https://evolvagency.io/learn/generative-search/llm-statistics-2026> [accessed: 04/05/2026]
- Filcek G, et al (2021) “An evolutionary algorithm for joint bi-criteria location-scheduling problem”, *International Journal of Industrial Engineering Computations*, 12 (2021) 159–176
- Hennes H, et al (2002) “Integrated scheduling and location models: Single machine makespan problems,” *Fachbereich Mathematik, Univ. Kaiserslautern, Kaiserslautern, Germany*, Rep. 82/2002
- Lima, et al (2025) “Toward a Trustworthy Optimization Modeling Agent via Verifiable Synthetic Data Generation” arXiv:2508.03117v1
- Mostajabdaveh M, et al (2025) “Evaluating LLM Reasoning in the Operations Research Domain with ORQA” arXiv:2412.17874v2
- Rae J, et al (2021) “Scaling Language Models: Methods, Analysis & Insights from Training Gopher” arXiv:2112.11446
- Ramamonjison, et al (2023) “NL4Opt Competition: Formulating Optimization Problems Based on Their Natural Language Descriptions” arXiv:2303.08233v2
- Smith S, et al (2022) “Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model” arXiv:2201.11990
- Touvron H, et al (2023a) “LLaMA: Open and Efficient Foundation Language Models” arXiv:2302.13971
- Touvron H, et al (2023b) “Llama 2: Open Foundation and Fine-Tuned Chat Models” arXiv:2307.09288
- Tsouros, et al (2023) “Holy Grail 2.0; From Natural Language to Constraint Models” arXiv:2308.01589v1
- Vaswani A, et al (2017) “Attention Is All You Need”, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 5998-6008; arXiv:1706.03762
- Zhang, et al (2022) “Exact and matheuristic methods for the parallel machine scheduling and location problem with delivery time and due date” *Computers & Operations Research* 147 (2022) 105936
- Zhang, et al (2024) “Solving General Natural-Language-Description Optimization Problems with Large Language Models”, *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 483–490
- Zhao W, et al (2024) “A Survey of Large Language Models” arXiv:2303.18223v14