Using Large Language Models for Optimization Modeling

Mathematical optimization is crucial for organizations to reduce costs, improve customer satisfaction, and mitigate risks. Many real-life problems can be modeled and solved using advanced optimization techniques, aiming to optimize an objective function within a set of constraints. However, converting natural language descriptions into mathematical models is challenging, and most decision-makers lack the necessary skills, often requiring OR experts who may not always be available.

Large Language Models (LLMs) have made AI skills accessible to almost anyone through natural language, revolutionizing various fields, including software development with tools like GitHub Copilot. Recent AI advances in LLMs have led to new research combining AI with mathematical optimization. This aims to develop optimization modeling copilots, enabling decision-makers to interact with AI in natural language, leveraging their knowledge to create and solve specific optimization models.

LLMs excel in natural language processing, allowing them to interpret optimization problems described in everyday language. They can identify and parse key components such as objectives, constraints, variables, and parameters, making mathematical optimization more accessible to a broader range of decision-makers.

Besides understanding natural language instructions, LLMs can recognize patterns from in-context demonstrations. This ability has been utilized to develop new algorithms and improve existing heuristics for various combinatorial optimization problems, such as TSP and online bin packing. LLMs can generate heuristics or rules of thumb to enhance the search process in large solution spaces. For instance, they can improve metaheuristics by devising pattern-based subroutines and enhancing existing algorithms by suggesting improvements in code efficiency, logic, and structure. This significantly accelerates the development process, from conceptualization to implementation, reducing the time-to-solution.

Many research papers now explore using LLMs to build and solve mathematical optimization models through natural language interaction with decision-makers and enhancing existing algorithms. However, the results are still far from being applicable in real-world settings due to relatively low accuracy and a lack of effective strategies for formulating and solving large-scale, complex problems. The challenge is to develop new LLM-driven techniques to translate and solve large-scale optimization problems, create strategies to handle extensive input data, and address integer programming issues with many variables.

Within the scope of this challenge, we seek to gain a thorough and multifaceted understanding of the topic. This includes, but is not limited to, exploring various dimensions and perspectives, such as:

1. We are seeking innovative ideas for integrating LLMs into the optimization-solving process. This includes exploring new methods for incorporating LLMs into heuristics or exact algorithms and addressing all aspects related to the different stages of finding optimal solutions. Additionally, a literature review of recent papers on cutting-edge approaches in the field of LLM utilization is desired.

2. Proposing novel ways to leverage LLMs in enhancing solution algorithms by introducing new subroutines or refining existing ones. LLMs can generate heuristics to

streamline searches in extensive solution spaces and optimize algorithms by enhancing code efficiency and logic. This expedites development and minimizes time-to-solution. Noteworthy examples can be referenced in [5,6,7,8,9].

3. Formulating mathematical models from natural language descriptions and rendering them into specific formats, such as LaTeX or Pyomo, is essential, particularly within the context of the NL4Opt competition. LLMs excel in generating precise mathematical expressions that effectively encapsulate the optimization problem described. Once these expressions are formulated, LLMs adeptly translate them into well-structured modeling code, ensuring syntactical correctness and readability. Notably, this capability has been demonstrated in [3,4].
4. Analyzing solutions for optimization problems and proposing ways to improve results via changing input data or constraints in natural language format. Natural language interpreter will be very useful for users who do not have experience in math modeling, it can improve the user interaction experience and promote this process to the new level with better performance [1,2].

We are open to any new ideas, suggestions, or innovative approaches you may have to tackle this challenge. Your expertise and input could significantly strengthen our collaboration and develop fruitful partnerships in the future.

References

1. Li, Beibin, Mellou, K., Zhang, Bo, Pathuri, J., Menache, I. Large Language Models for Supply Chain Optimization. (2023).
2. Slack, D., Krishna, S., Lakkaraju, H. *et al.* Explaining machine learning models with interactive natural language conversations using TalkToModel. *Nat, Mach, Intell,* **5**, 873–883 (2023).
3. A. AhmadiTeshnizi, W. Gao, M. Udell. OptiMUS: Optimization Modeling Using MIP Solvers and large language models. 2023.
4. S. Wasserkrug et al. From Large Language Models and Optimization to Decision Optimization CoPilot: A Research Manifesto. 2024.
5. Romera-Paredes, B., Barekatain, M., Novikov, A. *et al.* Mathematical discoveries from program search with large language models. *Nature* 625, 468–475 (2024).
6. Fei Liu, Xialiang Tong et al. An Example of Evolutionary Computation + Large Language Model Beating Human: Design of Efficient Guided Local Search. 2024
7. H. Ye, J. Wang et al. ReEvo: Large Language Models as Hyper-Heuristics with Reflective Evolution. 2024
8. C. Yang et al. Large Language Models as Optimizers. Google DeepMind.
9. E. Meyerson. Language Model Crossover: Variation through Few-Shot Prompting. CarperAI, Cognizant AI Labs, American University. https://arxiv.org/pdf/2302.12170.pdf