Taylor & Francis
Taylor & Francis Group

Engineering Optimization

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/geno20

Effective prompting with ChatGPT for problem formulation in engineering optimization

Nguyen Gia Hien Vu, Ke Wang & G. Gary Wang

To cite this article: Nguyen Gia Hien Vu, Ke Wang & G. Gary Wang (28 Jan 2025): Effective prompting with ChatGPT for problem formulation in engineering optimization, Engineering Optimization, DOI: 10.1080/0305215X.2025.2450686

To link to this article: https://doi.org/10.1080/0305215X.2025.2450686

Published online: 28 Jan 2025.

ENGINEERING OPTIMIZATION
https://doi.org/10.1080/0305215X.2025.2450686

Taylor & Francis
Taylor & Francis Group

Effective prompting with ChatGPT for problem formulation in engineering optimization

Nguyen Gia Hien Vu, Ke Wang and G. Gary Wang

Product Design and Optimization Laboratory, Simon Fraser University, Surrey, BC, Canada

ABSTRACT

Optimization problem formulation, a crucial but manually performed process, can present an obstacle to applying optimization in engineering since many practitioners find it challenging. This article explores the use of ChatGPT to address this challenge. It evaluates the efficacy of self-designed prompts with different ChatGPT models. Using analysis of variance and Tukey's test, assessments are conducted to determine the influence of variations in wording on the quality of solutions. The sequential learning approach is also tested to assess its impact on ChatGPT responses. This article confirms the importance of specificity in word choice and the relevance of domain-specific engineering terminology in crafting prompts for problem modelling. The analysis shows that a combination of properly selected words can lead to high-quality optimization problem formulations. Furthermore, it is found that sequential learning can enhance formulations. This work may bring more attention to the use of ChatGPT for formulating problems in engineering optimization.

ARTICLE HISTORY
Received 24 June 2024
Accepted 3 January 2025

KEYWORDS
Engineering; optimization; problem formulation; large language model

1. Introduction

Engineering optimization (EO) is crucial for continuous improvement and has been widely used to search for better solutions in engineering (Martins and Ning 2021; Ravindran, Ragsdell, and Reklaitis 2023). One key to success in EO projects is effective optimization problem formulation (Martins and Ning 2021). Optimization problems are often expressed in the form of mathematical models (Martins and Ning 2021), which should furnish all related objectives and constraints as functions of design variables and capture correct and precise engineering information (Eckert and Hillerbrand 2018; Sayama 2015).

The problem formulation process, in real-world applications, presents intricate complexities to create a proper model (Arora 2017). It is generally agreed that building a proper model might take more than 50%, even up to 80% or higher, of the total efforts needed to realize an optimum design (Arora 2017). In practice, many issues can commonly arise during this process, including (Cross 2021; Pahl *et al.* 2007):

- purposes not explicitly mentioned, resulting in trouble in determining appropriate constraints and clear objectives
- changes in data, requirements and system information, making it hard to devise equations for functions

CONTACT G. Gary Wang gary_wang@sfu.ca

- possible errors and biases in problem statements, leading to difficulties in establishing variables and their bounds.

In this formulation process, as argued by Martins and Ning (2021), human and multidisciplinary inputs are currently still required, and engineers must construct a sound problem formulation or risk jeopardizing the ultimate goal. Similarly, Arora (2017) emphasizes that this multistep process demands a blend of engineering judgement and technical expertise. These requirements make it hard to establish a guideline to guarantee problem formulation success (Crossley *et al.* 2017), despite multiple instructions and examples currently being available to assist practitioners (Arora 2017). These challenges drive the need for tools that can assist engineers in problem formulation.

In this context, the emergence of large language models (LLMs), such as ChatGPT, ushers in a new and promising era. In recent years, LLMs have been applied for different purposes, for example processing texts, summarizing documents and information retrieval (Ding *et al.* 2024). LLMs significantly accelerate tasks such as code or formula implementation, text generation and reasoning (Ding *et al.* 2024; Kojima *et al.* 2022; Li *et al.* 2020; C. Yang *et al.* 2024). Furthermore, LLMs are trained on various datasets (Wu *et al.* 2023), which spurs their potential applications in different fields.

In the EO domain, such notable features can offer significant help in formulating optimization problems from different perspectives (Dagdelen *et al.* 2024; Zakkas, Verberne, and Zavrel 2024). For example, experienced engineers can use the information extraction and summarizing strengths of LLMs in reading long concept documents and manuals more quickly, obtaining relevant information more easily or breaking down complicated documents into smaller problems (Dagdelen *et al.* 2024; Zakkas, Verberne, and Zavrel 2024). For less experienced users or new practitioners, LLMs can help to initiate ideas by presenting general knowledge, answering queries or providing information for further research and exploration (Dagdelen *et al.* 2024; Zakkas, Verberne, and Zavrel 2024). Overall, LLMs can be applied in diverse ways in formulating problems more quickly and simply, which could encourage engineers to apply EO more widely in their work.

Harnessing such potential, this article aims to experimentally explore effective prompt designs and techniques to exploit ChatGPT, one well-known LLM (Wu *et al.* 2023), in EO problem formulation. Rather than dealing with problem solving, the objective herein is to focus on investigating and evaluating the capabilities of ChatGPT in defining challenging optimization problems.

Within this scope, the remainder of this article is structured as follows. Section 2 provides the contemporary context as the background, including EO, LLM applications and current advances in prompt engineering. Section 3 discusses the methodology to evaluate ChatGPT's potential and its performance. Section 4 presents two EO problems used in this article. Section 5 analyses the results on ChatGPT-3.5 and tests the applicability of the findings from the previous sections against ChatGPT-4o mini and ChatGPT-4. Then, Section 6 discusses the results and limitations of this article. Finally, Section 7 provides the conclusion.

2. Literature review

2.1. Engineering optimization

Optimization applications are possible in any aspect of engineering, such as designing new parts or systems, improving existing ones and conducting engineering analysis (Ravindran, Ragsdell, and Reklaitis 2023). As observed by Arora (2017), traditional approaches in engineering, such as engineering design, which are often reliant on trial-and-error methods for complex systems, usually pose significant challenges and require considerable time and resources. Consequently, it has become customary in practice to settle for suboptimal solutions until some investments are recovered (Arora 2017).

In contrast, EO is increasingly proving to be a promising solution (Arora 2017; Martins and Ning 2021). By aiming to identify optimal solutions for given tasks, EO systematically and quantitatively

optimizes performance while adhering to various constraints (Arora 2017). By applying EO, engineers can improve system performance, reduce costs and increase efficiency with no need to search and assess all possibilities (Arora 2017; Martins and Ning 2021; Ravindran, Ragsdell, and Reklaitis 2023). Furthermore, with ever-growing capabilities of software and hardware, EO has become increasingly prevalent and fruitful (Périaux and Tuovinen 2023; X. S. Yang, Koziel, and Leifsson 2013).

As mentioned in Section 1, optimization problem formulation remains critical and challenging in practice. However, little work has been done to support the problem formulation task. With this in mind, this article aims to explore the potential of LLMs in facilitating problem formulation for EO purposes.

2.2. Applications of and problems with LLMs

Recent advances in LLMs such as ChatGPT have catalysed a plethora of considerable applications across various areas, including EO (Ding *et al.* 2024; Kojima *et al.* 2022; Li *et al.* 2020; C. Yang *et al.* 2024), as discussed in Section 1. However, the question of how to obtain accurate answers from LLMs remains unresolved. Amatriain (2024) shows that, to attain accurate results, engineers may confront challenges in crafting effective prompts and wording strategies. To be more specific, using LLMs may demand profound comprehension of operational contexts and thorough understanding of the limitations of LLMs to tailor prompts for diverse scenarios, enabling LLMs to reach their full potential. This undertaking falls under the emerging domain of prompt engineering. The inherent limitations of LLMs can lead to novel challenges for engineers in this domain. For example, LLMs may generate incorrect equations, assumptions or information, a phenomenon commonly known as 'hallucination' (Amatriain 2024).

2.3. Prompt engineering for engineers

In recent years, substantial research efforts have been dedicated to the growing field of prompt engineering. Prompt engineering has been applied across multiple domains, such as hallucination reduction, reasoning and logic, or fine-tuning and optimization (Sahoo *et al.* 2023). Likewise, prompt engineering has been instrumental in research endeavours. For instance, Rios, Menzel, and Sendhoff (2023) employ various prompts and wording strategies to assess the potential of text-to-3D models.

In optimization domain, prompt engineering can help to enhance solutions of mathematical problems and reasoning. For example, C. Yang *et al.* (2024) apply optimization by prompting to various well-known optimization tasks. However, Sahoo *et al.* (2023) mention the work by C. Yang *et al.* (2024) as the only study in the realm of prompt engineering for 'optimization and efficiency', which implies potential for further exploration. In this context, this article focuses solely on prompt engineering for EO problem formulation which, to the authors' knowledge, has not been studied before.

3. Methodology

This section establishes the methodological framework, comprising problem selection and grading scheme, prompt design, performance evaluation on ChatGPT-3.5, application considerations for ChatGPT-4o mini and ChatGPT-4, and sequential learning (SL).

3.1. Problem selection and grading scheme

To establish the ground for this article, certain problems are selected and their corresponding solutions are used as benchmarks for testing and analysis. The following key criteria are desirable:

- Requiring ChatGPT to consider every word therein to deduce all constraints and equations.

Table 1. Keyword/phrase options in instruction prompts.

No.	Field 1 (action)	Field 2 (target)	Field 3 (phrase)
1	Model	Situation	(null) (Phrase 1)
2	Define	Problem	giving me optimization criteria and constraints (Phrase 2)
3	Summarize	Description	giving me the optimization equation and constraints (Phrase 3)
4	Formulate	Information	giving me variables, optimization function, and constraints equation (Phrase 4)
5	Design	Requirements	
6	Develop	Design	

- Embodying explicitly and implicitly stated constraints within the problem definition, necessitating ChatGPT to extract relevant data and induce relevant information.
- Having different solutions available for evaluating ChatGPT's reasoning.

Furthermore, if applicable, the problems should include both linear and nonlinear equations. It is also preferable for such problems to be well known. Then, a grading scheme is established to evaluate ChatGPT's responses based on the respective benchmark solution for the problem. The problems and grading schemes are detailed in Section 4.

3.2. Prompt design

For this article, the prompt consists of two parts: the instruction prompt, which is followed by the respective predefined problem stated in Section 4. In principle, the instruction prompt varies from one attempt to another while the predefined problem remains the same. The instruction prompt uses the following template and a set of keyword or phrase options to fill in:

$$< \text{Field 1} > \text{ the } < \text{Field 2} > \text{ by } < \text{Field 3} >$$

where $< \text{Field 1} >$, $< \text{Field 2} >$ and $< \text{Field 3} >$ are distinct placeholders to be filled in, namely:

- Field 1 (Action): A single verb that instructs what ChatGPT should do.
- Field 2 (Target): A single noun that describes what ChatGPT should respond to.
- Field 3 (Phrase): A phrase that elaborates on the steps and expected results that ChatGPT should deliver.

For Fields 1 and 2, the keyword selection is guided by the appearance frequency of keywords in Chapter 2 of *Introduction to Optimum Design* (Arora 2017), as well as keywords commonly used in engineering. Notably, supplementary keywords that, while relevant, might not be conventionally used in such instructions are also incorporated. Among them, for Field 1, 'Summarize' might be considered unconventional, while 'Model' and 'Formulate' are more widely used in this context. For Field 2, 'Requirements' is in plural form since predefined problems have multiple constraints and requirements. Regarding Field 3, phrases are curated to span from the least to the most detailed instructions. This approach ensures that these prompts cover a spectrum of specificity levels, accommodating various needs in problem formulation. Table 1 shows the selection for completing instruction prompts.

3.3. ChatGPT-3.5-based performance evaluation

To evaluate the potential of LLMs in formulating EO problems, prompts as described in Subsection 3.2 are applied on ChatGPT-3.5. Notably, owing to time and model availability constraints, the evaluation of ChatGPT-3.5's potential is conducted using Problem 1 only (Subsection 4.1).

3.3.1. Data collection

For the data collection process on ChatGPT-3.5, the full factorial design is chosen so that all possible combinations of the three fields are tested (Montgomery 2017). Thirty attempts are administered for each of 144 distinct prompts, which are generated using six options for Field 1, six for Field 2 and four for Field 3 (Table 1). In each attempt, the whole prompt is given to ChatGPT in a single message (one-shot approach). Intentionally, to ensure statistical reliability, 30 attempts for each prompt are the minimum recommended sample size for applying the central limit theorem (Illowsky and Dean 2018). This is because the answer might vary for each attempt of the same prompt owing to the non-deterministic nature of the generative process in ChatGPT (Amatriain 2024). As a result, a total of 4320 experiments is performed.

Notably, to prevent ChatGPT from learning from previous conversations, ChatGPT's Memory Function is turned OFF and each attempt is recorded in a new and independent chat. Furthermore, the grade for each criterion is recorded individually to document separately whether each criterion is met or not in each attempt.

3.3.2. Data analysis

The analysis comprises three distinct steps. First, the criteria in the grading scheme are ranked based on the levels of difficulty for ChatGPT to meet, to assess ChatGPT's potential in problem formulation. To do this, the average grade is calculated for each criterion across the 30 attempts for each of the 144 prompts.

Secondly, the total grades of all attempts are analysed using a three-way analysis of variance (ANOVA) to see whether different keywords of different prompts make any difference (Illowsky and Dean 2018; Montgomery 2017). For ANOVA, the standard significance level of $a = 0.05$ (confidence interval of 95%) is applied. The calculations are performed using MATLAB® R2024a (version 24.1.0.2537033) and the Statistics and Machine Learning Toolbox version 24.1. The MATLAB *anovan* function and an 'interaction model' are used for this analysis. All other parameters for this *anovan* function use MATLAB's default values (MathWorks, "N-Way Analysis of Variance," n.d.).

Thirdly, based on the ANOVA results, *post hoc* tests are conducted. To start with, all null hypotheses rejected for the fields defined in Table 1 and for the interfield pairwise interactions are gathered. Then, to analyse these rejected hypotheses, Tukey's test (Montgomery 2017) is selected to identify which groups of keywords or phrases have means significantly different from those of the others. Tukey's test is preferred since it can compare means of all pairs of a population without the need for a control group. Furthermore, Tukey's test can control the overall error rate (Montgomery 2017). In Tukey's test, the same software and toolbox are employed as with ANOVA. However, the MATLAB *multcompare* function is selected (MathWorks, "Multiple Comparison Test," n.d.). To use Tukey's test, the parameter '*CriticalValueType*' is set to '*tukey - kramer*'. All other parameters for the *multcompare* function use MATLAB's default values (MathWorks, "Multiple Comparison Test," n.d.). Subsequently, based on Tukey's test results, this article discerns any common trends and explores how these trends can be used to enhance future outcomes.

3.3.3. Performance evaluation

Based on the grading scheme, a better ChatGPT answer should be closer or equivalent to the benchmark solution and thus have higher grades. ANOVA is first used to determine whether there exist significant performance differences inside a field (*i.e.* intrafield) or in an interfield pairwise interaction. Here, the focus is more on better performing prompts to derive findings and conclusions regarding better prompting patterns in using ChatGPT.

Subsequently, using Tukey's test, intrafield pairwise comparisons are conducted to assess performance differences for all pairs of keyword or phrase options within each of the three fields (Table 1). Then, all null hypotheses rejected by Tukey's test are filtered out and analysed. This is to categorize these options into distinct groups and identify which groups tend to perform better within each field.

In addition, different options in the same group are expected to demonstrate similar performance if the corresponding null hypothesis is not rejected. In exception cases, *i.e.* pairs of options that belong to the same group and yet display relatively significant performance differences, observations are made to determine whether such exceptions influence the grouping approach.

Next, the performance of three interfield pairwise interactions is assessed and all null hypotheses rejected by Tukey's test are filtered out for analysis. The objective is to determine whether combinations of better performing options from different fields lead to improved performance. Also, verification is carried out to see whether there is any exception to the findings.

3.4. ChatGPT-4o mini and ChatGPT-4 application considerations

This part is scoped to investigate whether the insights from ChatGPT-3.5 are applicable to ChatGPT-4o mini and ChatGPT-4. To do this, all problems detailed in Section 4 are evaluated. However, owing to the limited time and number of attempts allowed for ChatGPT-4o mini and ChatGPT-4, only representative instruction prompts that exhibit the best and worst performance on ChatGPT-3.5 are selected. These prompts are then used to obtain responses from ChatGPT-4o mini and ChatGPT-4; 30 attempts are conducted for each distinct prompt for each problem and on each model, still using the one-shot approach and having ChatGPT Memory Function turned OFF. The data are collected through a process similar to that for ChatGPT-3.5. As not all prompts are tested for this part, ANOVA and Tukey's test are not executed on these two models. Next, the consistency of the results from ChatGPT-4o mini and ChatGPT-4 with those from ChatGPT-3.5 is collated by calculating the average total sum across the 30 attempts.

3.5. Sequential learning

Being applied on different ChatGPT models, as described in previous sections, the one-shot approach may not exploit ChatGPT's advantageous iterative ability, which allows subsequent responses to be tuned based on previous conversations (Meyer *et al.* 2023). Consequently, using the SL approach, this step explores whether this ability could improve ChatGPT problem formulation responses.

In the SL approach, to have previous conversations in place, the whole prompt is split into smaller messages which are then sent to ChatGPT sequentially. The first message contains the instruction prompt and the first part of the problem description, while the remaining messages send the remaining split problem descriptions only. Then, the whole unsplit prompt is sent to ChatGPT again at the end of each attempt. The final answer to this unsplit prompt is evaluated instead of the previous ones with the t-test. The aim is to compare the one-shot approach and SL approach for two conditions of ChatGPT Memory Function (ON or OFF). This t-test applies the standard significance level of $a = 0.05$ (confidence interval of 95%), while other parameters use default values of the MATLAB *ttest2* function (MathWorks, "ttest2," n.d.). Furthermore, since there are only two groups in this test, Tukey's test is not conducted for further analysis.

Owing to model availability and time constraints, only Problem 2 (Subsection 4.2) is used on ChatGPT-4o mini to test the SL approach. Each condition is tested 30 times. Detailed application is found in Subsection 4.2.3. The same procedure is applied for the one-shot approach. These two approaches are then compared.

4. Problem definition and grading criteria

Based on the selection criteria mentioned in Subsection 3.1, this section establishes two problems, as follows.

4.1. Problem 1: Beam design

4.1.1. Definition

Problem 1 is used to examine the potential of ChatGPT in EO problem formulation and explore which prompting strategy can enhance the final results. This problem is modified from Example 2.1 in the book by Arora (2017), as follows:

> Cantilever beams are used in many practical applications in civil, mechanical, and aerospace engineering. To illustrate the step of problem description, we consider the design of a hollow square cross-section cantilever beam to support a load of 20 kN at its end. The beam, made of steel, is 2 m long. The failure conditions for the beam are as follows: (1) the material should not fail under the action of the load, and (2) the deflection of the free end should be no more than 1 cm. The width-to-thickness ratio for the beam should be no more than 8 to avoid local buckling of the walls. A minimum-mass beam is desired. The width and thickness of the beam must be within the following limits:

$$50 \leq \text{width} \leq 300\,\text{mm} \tag{1}$$

$$3 \leq \text{thickness} \leq 15\,\text{mm} \tag{2}$$

4.1.2. Grading scheme

Using the solution from Arora (2017) as the benchmark, a grading scheme is developed with the following six criteria, each of which is given one grade. Notably, grades are awarded solely based on the final equation for each criterion. One full grade is granted for each fully correct criterion, while any incorrect or undefined criterion receives none.

- Criterion 1: The objective is to minimize the total beam mass.
- Criterion 2: The shear stress must not exceed the material shear stress limit.
- Criterion 3: The bending stress must not exceed the material bending stress limit.
- Criterion 4: The free-end deflection must not exceed 1 cm.
- Criterion 5: The width-to-thickness ratio must not exceed 8.
- Criterion 6: The boundary (beam width and thickness) must satisfy Equations (1) and (2).

Notably, other solutions are accepted with grades given if they are mathematically equivalent and use a maximum of two independent variables. Equations that use a third independent variable receive no grades. Furthermore, redundant constraints are accepted without grades if they are mathematically correct and meaningful. For example, constraints stating that width and thickness must be positive do not affect the solution since boundary constraints imply that they are both positive.

4.2. Problem 2: Vehicle routing problem (VRP) design

4.2.1. Definition

Problem 2 deals with the VRP with breaks, a variant of traditional VRPs. For the past 60 years, VRP has been one of the most well-known and challenging problems (Konstantakopoulos, Gayialis, and Kechagias 2022). Given this, Problem 2 is considered to be more challenging and requires more efforts to formulate than Problem 1, and can be combined with Problem 1 to confirm and enhance the findings of this article. Problem 2, which is modified from the problem description by Coelho *et al.* (2016), is stated as follows:

> There is a directed graph $G = (V,A)$ where V is the vertex set and A is the arc set. The vertices include the depot and set of customers to be serviced. Each arc is associated with a travel distance and a travel time. Each customer is associated with a service time, a delivery weight and a delivery volume. The service of a customer must start within a time window. We assume that all time windows are feasible; that is, all customers can be reached from the depot within their time windows. A heterogeneous fleet of vehicles is located at the depot. Each vehicle is associated with a volume capacity and a weight capacity. Vehicle routes must begin within a given time window and must end before a given time and, within these limits, each route must respect a maximum working time.

Vehicles are allowed to wait between two customers, and a lunch break period must be scheduled to start within a given time window in each route. We assume pauses are made at a customer location, as is the case of our industrial partner. In this context, the goal is to minimize the total length of the routes in terms of travelling distance.

For the scope of this article, the definition of Problem 2 removes all keywords directly related to VRP, all variable names and the requirements of minimal break duration from the original problem description. Different formulations summarized by Bazirha (2023) can be used to handle Problem 2. Notably, Bazirha (2023) also mentions the formulation developed by Coelho *et al.* (2016).

4.2.2. Grading scheme

The grading scheme is developed based on solutions in the peer-reviewed publication by Bazirha (2023). Even though there are different solutions to Problem 2, they should demonstrate the following 13 criteria, which are used to evaluate the performance of ChatGPT:

- Criterion 1: The objective is to minimize the total distance travelled.
- Criterion 2: Each customer, excluding the initial and final location (depot), must be visited only once by one vehicle.
- Criterion 3: The total weight and volume for each vehicle must not exceed the limit of that vehicle.
- Criterion 4: One break must be considered. The break must start within a time window.
- Criterion 5: If one vehicle is used, it must leave the depot.
- Criterion 6: If one vehicle leaves the depot, it must return to the depot.
- Criterion 7: If one vehicle visits one customer, it must leave that customer.
- Criterion 8: Customers must be serviced within a time window, and subtour elimination must be considered.
- Criterion 9: The total duration for each shift must not exceed a limitation.
- Criterion 10: A shift must start and end within a time window.
- Criterion 11: If a vehicle takes a break between visiting customers i and j, it must visit both of these customers.
- Criterion 12: If a vehicle takes a break between visiting customers i and j, it must break after servicing customer i and before servicing customer j.
- Criterion 13: The domain of variables such as binary variables, if available, must be mentioned.

Grades are awarded solely based on the final equation for each criterion. Given the characteristics of the selected solutions, one full grade is granted for each fully correct criterion. Half a grade is granted for a criterion if the answer is not fully correct but it has at least one equation that is meaningful and correct in defining that criterion. Any incorrect or undefined criterion receives no grade.

4.2.3. Sequential learning

This SL approach is tested using Problem 2 only, since Problem 1 is simple and may fail to demonstrate the effectiveness of the iterative ability. Specifically, the prompts derived from Problem 2 are sent to ChatGPT sequentially, as shown in Table 2.

5. Experimental results

This section presents the experimental results. To start with, Problem 1 is applied to evaluate and confirm the potential of ChatGPT in formulating optimization problems (Subsection 5.1), to assess and determine the impact of different prompting approaches on ChatGPT's performances (Subsection 5.2), and to verify prompting strategies to improve the final results (Subsections 5.3 and 5.4). Then, Subsection 5.5 re-examines the results found in Subsections

Table 2. Sequential prompt content.

Prompt sequence	Prompt content
1	Instruction prompt 'There is a directed graph $G = (V,A)$ where V is the vertex set and A is the arc set. The vertices include the depot and set of customers to be serviced. Each arc is associated with a travel distance and a travel time.'
2	'Each customer is associated with a service time, a delivery weight, and a delivery volume.'
3	'The service of a customer must start within a time window. We assume that all time windows are feasible, that is, all customers can be reached from the depot within their time windows.'
4	'A heterogeneous fleet of vehicles is located at the depot. Each vehicle is associated with a volume capacity and a weight capacity.'
5	'Vehicle routes must begin within a given time window and must end before a given time and, within these limits, each route must respect a maximum working time.'
6	'Vehicles are allowed to wait between two customers, and a lunch break period must be scheduled to start within a given time window in each route.'
7	'We assume pauses are made at a customer location, as is the case of our industrial partner.'
8	'In this context, the goal is to minimize the total length of the routes in terms of travelling distance.'
9	Instruction prompt Full Problem 2 definition

5.1–5.4, using both problems. Finally, Subsection 5.6 investigates effects of the SL approach with Problem 2. Detailed numerical results for this section are available at: https://github.com/vuk1716/Analysing-Strategic-Wording-Techniques-with-ChatGPT-for-Problem-Formulation/tree/main.

5.1. Criterion difficulty ranking on ChatGPT-3.5

Exploring the potential of ChatGPT in problem formulation, this subsection ranks the six criteria in the grading scheme for Problem 1 by their difficulty levels for ChatGPT-3.5 across all 4320 attempts. Observably, the less difficult the criterion, the more accurate ChatGPT-3.5 response can be, which is demonstrated by the higher average, as shown in Table 3. Accordingly, the results for these six criteria can be summarized as follows:

1. Boundary constraint: This constraint is explicitly stated in the problem definition, and ChatGPT-3.5 is simply required to extract and replicate the corresponding part of the problem in its answer.
2. Width-to-thickness ratio constraint: This constraint is directly mentioned in the problem definition. However, ChatGPT-3.5 must translate from the text description to a correct mathematical formula.
3. Free-end deflection constraint: This constraint is less complex compared to the optimization objective function and two stress constraint equations, but more intricate than the boundary and width-to-thickness constraints. However, ChatGPT-3.5 often fails to provide the correct explicit equation for this constraint.
4. Optimization objective function: This function is prone to confusion owing to the hollow shape and ambiguity between mass, volume and cross-sectional area.
5. Bending stress constraint: The bending stress equation is confusing owing to its form, including the form of bending stress equation and equation variables.
6. Shear stress constraint: ChatGPT-3.5 consistently fails to mention this constraint across all attempts, suggesting possible current flaws in its engineering sense and reliability.

Conclusively, despite its imperfections, ChatGPT-3.5 still shows significant potential in problem formulation as it can precisely model many constraints, functions and criteria.

Table 3. Difficulty ranking of grading scheme criteria for Problem 1.

Criterion ranking (most to least accurate)	Average grade (range: 0–1)
Boundary constraint	0.7745
Width-to-thickness ratio constraint	0.5729
Free-end deflection constraint	0.2213
Optimization objective function	0.0741
Bending stress constraint	0.0546
Shear stress constraint	0.0000

Table 4. Analysis of variance results using MATLAB R2024a.

Source	*F*-statistic value	*p*-Value
Field 1	71.8543	0.0000
Field 2	10.0505	0.0000
Field 3	2241.2061	0.0000
Interaction 1: Field 1–Field 2	3.8974	0.0000
Interaction 2: Field 2–Field 3	29.2734	0.0000
Interaction 3: Field 3–Field 1	8.9029	0.0000

5.2. ANOVA on ChatGPT-3.5

Analysing the grading results of Problem 1, the aim of this subsection is to verify the impact of different prompting approaches on ChatGPT performance, using MATLAB *anovan* (MathWorks, "N-Way Analysis of Variance," n.d.). Accordingly, Table 4 displays the *F*-statistic values and *p*-values for the three fields (Table 1) and the three interfield pairwise interactions.

Table 4 shows that all *p*-values are significantly lower than the standard significance level of 0.05. This indicates that all null hypotheses have been rejected, and there are significant differences in the results between different keyword and phrase options for each field as well as significant interactions between all pairs of fields. It can also be seen that ChatGPT is unable to present the shear stress constraint condition correctly in all attempts. Following this observation, Tukey's test is conducted to analyse which options perform better in each field and which interfield interactions yield better results.

5.3. Tukey's test analysis for three fields on ChatGPT-3.5

This subsection aims to evaluate and unveil which prompting strategies tend to have better results with intrafield interaction analysis. To analyse the impact of each field independently with Tukey's test, each independent test, or group comparison, has two parts to be compared; namely, Class 1 and Class 2. For this subsection, each class consists of a keyword (Field 1, Field 2) or a phrase (Field 3). The mean difference of the respective test is calculated by subtracting the mean of Class 1 from that of Class 2. Here, the focus is on those pairs where the respective null hypotheses are rejected ($p < 0.05$).

5.3.1. Field 1

With six keywords and $\binom{6}{2} = 15$ possible tests for Field 1, Tukey's test rejects 10 out of 15 hypotheses. The 95% confidence interval and the corresponding *p*-values for those 10 pairs are shown in Table 5.

In Table 5, 'Model' and 'Summarize' seemingly perform the best and the worst, respectively. Conceptually, 'Model' refers to creating a simplified representation of a system. Its better performance probably stems from its specificity and associativity with engineering concepts, capturing specific essential aspects in mathematical, physical or computational form (IEEE 1989). Meanwhile, 'Summarize' generally means to condensed contents (Cambridge University Press 2004), probably making it more ambiguous.

Table 5. Tukey's test results for Field 1 keywords.

Class 1 parameter	Class 2 parameter	Lower limit	Mean difference ($\mu_1 - \mu_2$)	Upper limit	p-Value
'Action = Model'	'Action = Define'	0.4159	0.5250	0.6341	0.0000
'Action = Model'	'Action = Summarize'	0.5853	0.6944	0.8036	0.0000
'Action = Model'	'Action = Formulate'	0.3159	0.4250	0.5341	0.0000
'Action = Model'	'Action = Design'	0.3061	0.4153	0.5244	0.0000
'Action = Model'	'Action = Develop'	0.3381	0.4472	0.5564	0.0000
'Action = Define'	'Action = Summarize'	0.0603	0.1694	0.2786	0.0001
'Action = Define'	'Action = Design'	−0.2189	−0.1097	−0.0006	0.0479
'Action = Summarize'	'Action = Formulate'	−0.3786	−0.2694	−0.1603	0.0000
'Action = Summarize'	'Action = Design'	−0.3883	−0.2792	−0.1700	0.0000
'Action = Summarize'	'Action = Develop'	−0.3564	−0.2472	−0.1381	0.0000

Table 6. Field 1 keyword grouping.

Performance ranking (best to worst)	Group number	Keywords
1	Group 1	Model
2	Group 2	Formulate
		Define
		Design
		Develop
3	Group 3	Summarize

Table 7. Field 2 keyword grouping.

Performance ranking (best to worst)	Group number	Keywords
1	Group 1	Problem
		Information
		Situation
		Design
2	Group 2	Requirements
		Description

Between these two ends, four remaining keywords have mixed results, and Tukey's test cannot reject any of the corresponding null hypotheses, except for the single pair 'Define–Design'. With only one exception out of 15 tests, six keywords are divided into three performance-based groups (Table 6), and four keywords in Group 2 are expected not to have significant performance differences. Furthermore, as evidenced by the results in Table 5, changing one single keyword, such as replacing a Group 1 keyword with a Group 3 keyword, can significantly change the quality of ChatGPT formulation.

5.3.2. Field 2

By following the same procedure for Field 2 keywords, with six keywords and $\binom{6}{2} = 15$ possible pairs, these Field 2 keywords can be classified into two performance-based groups (Table 7), with Group 1 being the better one.

In Table 7, 'Requirements' and 'Description' can potentially involve more indirect and complex pathways to formulation. Conceptually, 'Requirements' implicitly drives the focus on multiple constraints to be satisfied (IEEE 1990). Meanwhile, 'Description' provides a broad narrative (Cambridge University Press 2004) that may not clearly delineate the steps towards mathematical formulation. The lack of specificity and clarity can be claimed for the worse results for these two words.

Table 8. Field 3 phrase grouping.

Performance ranking (best to worst)	Group number	Phrases
1	Group 1	Phrase 4: giving me variables, optimization function, and constraints equation
2	Group 2	Phrase 3: giving me the optimization equation and constraints
3	Group 3	Phrase 2: giving me optimization criteria and constraints
4	Group 4	Phrase 1: (null)

5.3.3. Field 3

Using the same process for the previous two fields, with four Field 3 phrases and $(4 * 3)/2 = 6$ possible pairs for testing comparison, these four phrases are categorized into four performance-based groups (Table 8), where Group 1 can be identified as the best performing group. Notably, these four phrases differ in their level of precision, with Phrase 4 and Phrase 1 being the most and least detailed, respectively. Furthermore, Phrase 2 focuses specifically on constraints.

Conclusively, the intrafield pairwise comparisons consistently affirm that specificity and engineering nuance are imperative in keyword or phrase selection for better results, as illustrated by the best performing groups ranked as Group 1 in all three fields. Moreover, as exemplified by significant mean differences in Table 5, a minor change in keywords or phrases can arguably lead to varied performance outcomes in ChatGPT-3.5 responses.

5.4. Tukey's test analysis for interfield pairwise interactions on ChatGPT-3.5

This subsection further investigates the results in Subsection 5.3 with interfield interactions to elucidate the dynamics of keyword and phrase combinations across multiple fields. For this subsection, each class in each independent test consists of an interaction between two keywords or phrases from different fields. The emphasis is on pairs where the respective null hypotheses are rejected ($p < 0.05$).

5.4.1. Field 1–Field 2 interactions

With six keywords in each field, $6 * 6 = 36$ possible combinations and $(36 * 35)/2 = 630$ independent tests, Tukey's test rejects 170 out of 630 tests. Notably, Field 1 exerts a greater impact on achieving better results than Field 2. For clarification, when the Field 1 keyword of one class is expected to outperform that of the other class, the average grade of the class with the outperforming Field 1 keyword tends to be higher, indicating its better performance, regardless of their Field 2 keywords. To illustrate this, if Class 1 uses 'Model' (Group 1) and Class 2 uses 'Summarize' (Group 3) for Field 1, the performance of Class 1 is supposed to be better, irrespective of their Field 2 keywords. However, if the Field 1 keywords of both classes are from the same group, Field 2 keywords become influential. In this case, the class with the outperforming Field 2 keyword would have a higher average grade.

Notably, the null hypothesis should not be rejected if both keywords for both classes belong to the same corresponding group. For example, if Class 1 uses 'Define' (Field 1) and 'Situation' (Field 2), while Class 2 employs 'Develop' (Field 1) and 'Information' (Field 2), the null hypothesis for their comparison should not be rejected. This is because both 'Define' and 'Develop' come from Group 2 of Field 1, and 'Situation' and 'Information' from Group 1 of Field 2. However, six cases deviate from this pattern. As these cases represent around 3.5% of the 170 rejected null hypotheses, the initial findings regarding Field 1–Field 2 interactions hold true in most cases. Thus, Field 1 is typically more influential than Field 2, and combinations of keywords from better performing groups are generally expected to yield better or equal results.

5.4.2. Field 2–Field 3 interactions

With six keywords in Field 2 and four phrases in Field 3, there are $6 * 4 = 24$ potential combinations. This leads to a total of $(24 * 23)/2 = 276$ independent tests for Tukey's test, out of which 190 hypotheses are rejected. Similarly to Field 1–Field 2 interaction patterns, Field 3 demonstrates stronger influences on achieving favourable outcomes than Field 2. However, if both classes share the same phrases for Field 3, the class with the outperforming Field 2 keyword will have a higher average grade. Since there are only six exceptions to this pattern, or 3.2%, Field 3 generally has a greater influence than Field 2, and combinations of keywords or phrases from better performing groups are also expected to have better or equal results.

5.4.3. Field 1–Field 3 interactions

With six keywords in Field 1 and four phrases in Field 3, there are $6 * 4 = 24$ potential combinations. This results in a total of $(24 * 23)/2 = 276$ independent tests conducted by Tukey's test, out of which 193 hypotheses are rejected. Similarly to the previous interfield interaction patterns, Field 3 is observed to have stronger influences on the likelihood of favourable outcomes than Field 1. However, if both classes share the same Field 3 phrase, a higher average grade is expected for the class with the outperforming Field 1 keyword. As there are only three exceptions, or 1.6%, Field 3 tends to have more impact than Field 1. Once again, combinations of keywords and phrases from better performing groups generally lead to superior or comparable results.

Conclusively, the interfield pairwise interaction analysis demonstrates that Field 3 is the most influential of the three fields. This reinforces the indispensable technique of specificity and engineering nuance in designing prompts for optimal and correct results in ChatGPT exploitation, as concluded in Subsection 5.3. Consistently, combinations of keywords and phrases from better performing groups can result in better or at least equivalent outcomes. Notably, while Tukey's test controls the experiment-wise error rate, the analysis does not fully consider the possibility of type II errors.

5.5. ChatGPT-4o mini and ChatGPT-4 test results

This subsection confirms the results from Subsections 5.1–5.4, using ChatGPT-4o mini and ChatGPT-4 on both Problems 1 and 2. Notably, this subsection focuses only on the best and worst combinations derived from findings from Subsections 5.3 and 5.4 for both problems. Accordingly, the following four representative prompt options are established for this part, using the performance-based grouping approach (Tables 6–8) and their total average grades on ChatGPT-3.5:

- Option 1: The most promising prompt formed by keywords and phrases from the best performing groups (Group 1 across all three fields), *i.e.* 'Model the problem by giving me variables, optimization function, and constraints equation'.
- Option 2: The least promising prompt consisting of keywords and phrases from the worst performing groups (Group 3 in Field 1, Group 2 in Field 2 and Group 4 in Field 3), *i.e.* 'Summarize the requirements'.
- Option 3: The prompt with the highest average grade on ChatGPT-3.5 for Problem 1 out of 144 averages calculated across all 30 repetitive attempts for each prompt, *i.e.* 'Summarize the design by giving me variables, optimization function, and constraints equation'.
- Option 4: The prompt with the lowest average grade on ChatGPT-3.5, using the same calculation method for Option 3, *i.e.* 'Design the situation'.

Then, the total average grades across 30 attempts are calculated for each prompt option on ChatGPT-4o mini and ChatGPT-4. The results are presented in Table 9.

Table 9. ChatGPT-4o mini and ChatGPT-4 results for four selected prompts.

Problem	Prompts	ChatGPT model	Total average
1	Option 1	4o mini	4.0667
1	Option 2	4o mini	0.8667
1	Option 3	4o mini	4.0333
1	Option 4	4o mini	1.9000
1	Option 1	4	4.4667
1	Option 2	4	0.0000
1	Option 3	4	3.8667
1	Option 4	4	1.9667
2	Option 1	4o mini	7.1667
2	Option 2	4o mini	0.0000
2	Option 3	4o mini	4.8667
2	Option 4	4o mini	0.0333
2	Option 1	4	7.7000
2	Option 2	4	0.0000
2	Option 3	4	6.1500
2	Option 4	4	1.3167

Table 10. *t*-Test results using MATLAB R2024a.

ChatGPT Memory Function	Total average across 30 attempts (one-shot)	Total average across 30 attempts (sequential learning)	*p*-Value
ON	6.6167	8.4500	0.0000
OFF	7.1667	7.8500	0.0336

An overall improvement in grades is noticeable on ChatGPT-4o mini and ChatGPT-4 for Problem 1 compared to that on ChatGPT-3.5. However, Option 2 (the worst promising prompt) performs significantly worse. Conversely, Option 1 (the most promising prompt) yields the highest grade, which is expected. In addition, Option 4 improves notably, although it lags behind Option 1 and Option 3, possibly owing to its lower potential combination of keywords and phrases, particularly with Field 3 influence. Problem 2 shows a similar trend, where Option 1 and Option 2 receive the highest and lowest average, respectively, on both models. The results herein consistently reaffirm the validity of the findings in the previous sections.

5.6. Sequential learning results

As one more step to further exploit ChatGPT effectively in EO problem formulation, this subsection uses the *t*-test to determine whether the SL approach makes any performance difference compared to the one-shot approach on ChatGPT-4o mini for Problem 2. This test only uses Option 1 (the most promising prompt), as defined in Subsection 5.5. The specific values are given in Table 10.

As shown in Table 10, both *p*-values are less than 0.05. This indicates that all null hypotheses are rejected, and the SL approach truly makes differences to ChatGPT formulation responses. Furthermore, since the SL approach shows higher average grades when ChatGPT Memory Function is ON and OFF, it can be concluded that the performance of ChatGPT can generally be improved with the SL approach compared to the one-shot approach.

However, ChatGPT Memory Function can have varied effects in this experimental stage. For the SL approach, as evidenced by the average grades of 30 SL-based attempts, ChatGPT performs better when the Memory Function is ON than it does when this function is OFF. Meanwhile, for the one-shot approach, the results of 30 one-shot-based attempts reveal that ChatGPT demonstrates

better performance when the Memory Function is OFF. This indicates that the condition of ChatGPT Memory Function may not have a clear and decisive influence on the final quality of ChatGPT formulation.

6. Discussion

This article explores LLMs, such as ChatGPT, as assistive tools for problem formulation in EO to reduce workload and improve success rates for engineers. In this regard, ChatGPT evidently possesses significant strengths (Dagdelen *et al.* 2024; Ding *et al.* 2024; Kojima *et al.* 2022; Li *et al.* 2020; Wu *et al.* 2023; C. Yang *et al.* 2024; Zakkas, Verberne, and Zavrel 2024) and solutions can be found to address its imperfection to a certain extent. The findings herein consistently show positive effects on the tested ChatGPT models, *i.e.* ChatGPT-3.5, ChatGPT-4o mini and ChatGPT-4, when relevant techniques and approaches are applied. For example, instead of random or poorly structured input instructions, a combination of properly selected words which enhance specificity and engineering nuance, even slightly, in crafting prompts can visibly improve ChatGPT formulation responses. Furthermore, better outcomes can be obtained if well-designed prompts are used in combination with the SL approach to capture ChatGPT's learning ability. This observation is true whether the ChatGPT Memory Function is ON or OFF. The evaluation findings reveal that the combination of the most effective instruction prompt, 'Model the problem by giving me variables, optimization function, and constraints equation', and the SL approach leads to the best formulation responses. Notably, this task might not be hard to implement in practice. In short, prompt designs and prompting techniques, if properly applied, can enhance the value of ChatGPT in EO problem formulation.

Notably, certain defects can be observed in EO problem formulation tasks performed by ChatGPT during the process. Evidently, both Problem 1 and Problem 2 include criteria that ChatGPT cannot formulate correctly in most of the attempts. In Problem 1, ChatGPT cannot include the correct shear stress constraint. For Problem 2, ChatGPT struggles to indicate that the vehicle must visit customers i and j if it wants to break between customers i and j. Such defects risk ChatGPT 'confidently' generating inaccurate information, which may confuse users (Azaria, Azoulay, and Reches 2024), and failing to evaluate and apply results appropriately, especially in high-risk fields such as healthcare (Cong-Lem, Soyoof, and Tsering 2024). In addition, concerns regarding plagiarism arise, particularly when ChatGPT responses lack specific sources (Cong-Lem, Soyoof, and Tsering 2024). From this perspective, ChatGPT should be monitored and cannot fully substitute engineers in EO, such as in validating final formulation correctness, so that solutions can avoid missing equations or incorrect interpretation. Notably, although engineers are indispensable in the process, they should unceasingly acquire and enhance their domain technical competency to proactively leverage ChatGPT's evolving strengths as their assistive tools more responsibly and effectively.

While this article investigates a range of prompting techniques, its scope of prompts, which is based on a predefined set of keywords and phrases, cannot encompass all possible variations and effective strategies. Also, despite their proven potential, the discussed techniques and approaches herein may not be universally optimal for all EO formulations. Arguably, although the two chosen problems might be popular in the EO field, they cannot fully capture the diverse challenges encountered in real world and offer an optimal environment to examine the full capabilities of various LLMs. However, viewing LLMs' evolving capabilities (Dagdelen *et al.* 2024; Ding *et al.* 2024; Kojima *et al.* 2022; Li *et al.* 2020; Wu *et al.* 2023; C. Yang *et al.* 2024; Zakkas, Verberne, and Zavrel 2024), this article sees the positive findings of ChatGPT as an encouragement to conduct further research on the enhanced application of LLMs in problem formulation and to increase EO adoption.

Overall, possessing enormous potential, LLMs, including ChatGPT, should be used only as assistive tools rather than entirely automatic formulators, as mentioned in Section 1. Inevitably, their applications require engineers' crucial technical mastery to achieve improvements in EO problem formulation.

7. Conclusion and future research

This article contemplates exploiting the potential of ChatGPT, one popular LLM (Wu *et al*. 2023), as a formulator to facilitate problem formulation in EO. The aim is to identify effective prompt designs and techniques that can improve the accuracy and thoroughness of problem formulation by ChatGPT.

The experimental set-up involves establishing and evaluating various prompts designed to formulate optimization problems in engineering tasks. Based on predefined formulation problems and corresponding benchmark solutions, this article establishes detailed grading schemes to assess ChatGPT responses, investigating the impact of different wording strategies and prompting techniques through a series of tests including ANOVA and Tukey's test.

The key findings confirm that, while ChatGPT potentially benefits EO problem formulation tasks, the specificity and domain-specific terminology inherent in words and phrases chosen for prompts can affect its response quality substantially, including accuracy and relevance. Notably, even subtle variations in phrasing can significantly alter outcomes. Consistently, combinations of keywords and phrases from better performing groups tend to yield better or at least equivalent outcomes. ChatGPT formulation responses can be further improved if the sequential learning approach is also applied, whether ChatGPT Memory Function is ON or OFF. These findings are applicable to all three tested ChatGPT models, namely ChatGPT-3.5, ChatGPT-4o mini and ChatGPT-4.

Despite the robust design of the methodology, this article is constrained to predefined problem statements for ChatGPT only, which do not fully represent the wide range of real-world engineering problems. The above briefly mentioned inherent problems of LLMs, such as 'hallucination', remain challenging in deploying these tools for critical engineering decisions. Although the relevant prompting techniques and the SL approach may offer improvements, it can be said that the three tested ChatGPT models, as potential formulation assistants, are currently not fully capable of completing all EO problem formulation tasks without engineers' undeniably critical roles, even in the era of different LLMs. Nevertheless, engineers must enhance their technical competency to utilize LLMs more productively.

It is suggested that this study is just a starting point in research on using LLMs in EO and particularly problem formulation. There will be ample opportunities for further exploration. For example, advanced strategies and techniques can be developed, such as training LLMs with engineering-specific documents, to give LLMs more engineering sense and overcome their current limitations. In addition, consideration should be given to investigating the strengths of other LLMs in formulating EO problems, using a broader array of wording from diverse libraries or experimenting with different prompting strategies tailored to specific problems. Furthermore, the ability of LLMs to generate prompts by themselves should be explored, and could be applied to choose better prompts for each problem automatically and reduce human interventions.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

Financial support provided by the Natural Sciences and Engineering Research Council (NSERC) of Canada [grant number RGPIN-2019-06601] is gratefully appreciated.

Data availability statement

Detailed data results for this research are available at: https://github.com/vuk1716/Analysing-Strategic-Wording-Techniques-with-ChatGPT-for-Problem-Formulation/tree/main.

ORCID

G. Gary Wang http://orcid.org/0000-0003-1742-4849

References

Amatriain, X. 2024. "Prompt Design and Engineering: Introduction and Advanced Methods." *arXiv:2401.14423v4*.

Arora, J. S. 2017. *Introduction to Optimum Design*. 4th ed. Cambridge, MA: Academic Press.

Azaria, A., R. Azoulay, and S. Reches. 2024. "ChatGPT is a Remarkable Tool—For Experts." *Data Intelligence* 6 (1): 240–296. https://doi.org/10.1162/dint_a_00235.

Bazirha, M. 2023. "A Novel MILP Formulation and an Efficient Heuristic for the Vehicle Routing Problem with Lunch Break." *Annals of Operations Research*. https://doi.org/10.1007/s10479-023-05742-3.

Cambridge University Press. 2004. *Cambridge Learner's Dictionary*. 2nd ed. Cambridge: Cambridge University Press.

Coelho, L. C., J. P. Gagliardi, J. Renaud, and A. Ruiz. 2016. "Solving the Vehicle Routing Problem with Lunch Break Arising in the Furniture Delivery Industry." *Journal of the Operational Research Society* 67 (5): 743–751. https://doi.org/10.1057/jors.2015.90.

Cong-Lem, N., A. Soyoof, and D. Tsering. 2024. "A Systematic Review of the Limitations and Associated Opportunities of ChatGPT." *International Journal of Human-Computer Interaction*, 1–16. https://doi.org/10.1080/10447318.2024.2344142.

Cross, N. 2021. *Engineering Design Methods: Strategies for Product Design*. 5th ed. Hoboken, NJ: John Wiley & Sons.

Crossley, W. A., S. Luan, J. T. Allison, and D. L. Thurston. 2017. "Optimization Problem Formulation Framework with Application to Engineering Systems." *Systems Engineering* 20 (6): 512–528. https://doi.org/10.1002/sys.21418.

Dagdelen, J., A. Dunn, S. Lee, N. Walker, A. S. Rosen, G. Ceder, K. A. Persson, and A. Jain. 2024. "Structured Information Extraction from Scientific Text with Large Language Models." *Nature Communications* 15: 1418. https://doi.org/10.1038/s41467-024-45563-x.

Ding, Q., D. Ding, Y. Wang, C. Guan, and B. Ding. 2024. "Unraveling the Landscape of Large Language Models: A Systematic Review and Future Perspectives." *Journal of Electronic Business & Digital Economics* 3 (1): 3–19. https://doi.org/10.1108/JEBDE-08-2023-0015.

Eckert, C., and R. Hillerbrand. 2018. "Models in Engineering Design: Generative and Epistemic Function of Product Models." In *Advancements in the Philosophy of Design*, edited by P. E. Vermaas and S. Vial, 219–242. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-73302-9_11.

IEEE. 1989. *IEEE Standard Glossary of Modeling and Simulation Terminology*. IEEE Std 610.3-1989, The Institute of Electrical and Electronics Engineers, NY.

IEEE. 1990. *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Std 610.12-1990, The Institute of Electrical and Electronics Engineers, NY.

Illowsky, B., and S. Dean. 2018. *Introductory Statistics*. Houston, TX: OpenStax, Rice University.

Kojima, T., S. (Shane) Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. 2022. "Large Language Models are Zero-Shot Reasoners." In *Advances in Neural Information Processing Systems, (NeurIPS 2022)*, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, 22199–22213. New Orleans, LA: Neural Information Processing Systems Foundation, Inc.

Konstantakopoulos, G. D., S. P. Gayialis, and E. P. Kechagias. 2020. "Vehicle Routing Problem and Related Algorithms for Logistics Distribution: A Literature Review and Classification." *Operational Research* 22 (2022): 2033–2062. https://doi.org/10.1007/s12351-020-00600-7.

Li, J., T. Tang, W. X. Zhao, J. Y. Nie, and J. R. Wen. 2024. "Pre-trained Language Models for Text Generation: A Survey." *ACM Computing Surveys* 56 (9): 1–39. https://doi.org/10.1145/3649449.

Martins, J. R. R. A., and A. Ning. 2021. *Engineering Design Optimization*. Cambridge: Cambridge University Press.

MathWorks. n.d. "Multiple Comparison Test – MATLAB multcompare." mathworks.com. Accessed May 13, 2024 [Online]. https://www.mathworks.com/help/stats/multcompare.html.

MathWorks. n.d. "N-Way Analysis of Variance – MATLAB anovan." mathworks.com. Accessed May 13, 2024 [Online]. https://www.mathworks.com/help/stats/anovan.html#bulw41w-3.

MathWorks. n.d. "ttest2." mathworks.com. Accessed August 31, 2024 [Online]. https://www.mathworks.com/help/stats/ttest2.html.

Meyer, J. G., R. J. Urbanowicz, P. C. N. Martin, K. O'Connor, R. Li, P. C. Peng, T. J. Bright, et al. 2023. "ChatGPT and Large Language Models in Academia: Opportunities and Challenges." *BioData Mining* 16. https://doi.org/10.1186/s13040-023-00339-9.

Montgomery, D. C. 2017. *Design and Analysis of Experiments*. 9th ed. Hoboken, NJ: John Wiley & Sons.

Pahl, G., W. Beitz, J. Feldhusen, and K.-H. Grote. 2007. *Engineering Design: A Systematic Approach*. 3rd ed. London: Springer.

Périaux, J., and T. Tuovinen. 2023. "Thirty Years of Progress in Single/Multi-Disciplinary Design Optimization with Evolutionary Algorithms and Game Strategies in Aeronautics and Civil Engineering." In *Impact of Scientific Computing on Science and Society*. Computational Methods in Applied Sciences. Vol. 58, edited by M. L. Rantalainen, and P. Neittaanmäki, 429–450. Vol. 58. Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-29082-4_24.

Ravindran, A., K. M. Ragsdell, and G. V. Reklaitis. 2006. *Engineering Optimization: Methods and Applications*. 2nd ed. Hoboken, NJ: John Wiley & Sons.

Rios, T., S. Menzel, and B. Sendhoff. 2023. "Large Language and Text-to-3D Models for Engineering Design Optimization." In *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1704–1711. Mexico City: Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/SSCI52147.2023.10371898.

Sahoo, P., A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha. 2024. "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications." *arXiv:2402.07927v1*.

Sayama, H. 2015. *Introduction to the Modeling and Analysis of Complex Systems*. Geneseo, NY: Open SUNY Textbooks, Milne Library, State University of New York at Geneseo.

Wu, T., S. He, J. Liu, S. Sun, K. Liu, Q. L. Han, and Y. Tang. 2023. "A Brief Overview of ChatGPT: The History, Status Quo and Potential Future Development." *IEEE/CAA Journal of Automatica Sinica* 10 (5): 1122–1136. https://doi.org/10.1109/JAS.2023.123618.

Yang, X. S., S. Koziel, and L. Leifsson. 2013. "Computational Optimization, Modelling and Simulation: Recent Trends and Challenges." *Procedia Computer Science* 18, 855–860. https://doi.org/10.1016/j.procs.2013.05.250.

Yang, C., X. Wang, Y. Lu, H. Liu, Q. V. Le, D. Zhou, and X. Chen. 2024. "Large Language Models as Optimizers." Presented at The Twelfth International Conference on Learning Representations (ICLR 2024), Vienna, Austria, May 7–11.

Zakkas, P., S. Verberne, and J. Zavrel. 2024. 'SumBlogger: Abstractive Summarization of Large Collections of Scientific Articles.' In *Advances in Information Retrieval, ECIR 2024*. Lecture Notes in Computer Science. Vol. 14608, edited by Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis, 371–386. Cham: Springer. https://doi.org/10.1007/978-3-031-56027-9_23.