Proceedings of the 44th Chinese Control Conference
July 28-30, 2025, Chongqing, China

Leveraging LLM Agents for Automated Optimization Modeling for SASP Problems: A Graph-RAG based Approach

Tianpeng Pan[1], Wenqiang Pu[1], Licheng Zhao[1], Rui Zhou[1]
1. Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen
E-mail: 224010189@link.cuhk.edu.cn, {wpu,zhaolicheng,rui.zhou}@sribd.cn

Abstract: Automated optimization modeling (AOM) has evoked considerable interest with the rapid evolution of large language models (LLMs). Existing approaches predominantly rely on prompt engineering, utilizing meticulously designed expert response chains or structured guidance. However, prompt-based techniques have failed to perform well in the sensor array signal processing (SASP) area due the lack of specific domain knowledge. To address this issue, we propose an automated modeling approach based on retrieval-augmented generation (RAG) technique, which consists of two principal components: a multi-agent (MA) structure and a graph-based RAG (Graph-RAG) process. The MA structure is tailored for the architectural AOM process, with each agent being designed based on principles of human modeling procedure. The Graph-RAG process serves to match user query with specific SASP modeling knowledge, thereby enhancing the modeling result. Results on ten classical signal processing problems demonstrate that the proposed approach (termed as MAG-RAG) outperforms several AOM benchmarks.

Key Words: Large language models, sensor array signal processing, retrieval-augmented generation

1 Introduction

Sensor Array Signal Processing (SASP) has experienced remarkable advancements over the past few decades [1], which finds utility in a spectrum of applications, including telecommunications, radar, sonar, etc. Research within this field has encompassed areas such as beamforming, direction-of-arrival (DOA) estimation, primal user detection, source localization, etc. Over time, SASP has witnessed a paradigm shift from a predominantly parametric approach [2] to optimization methodologies [1], [3], leading to substantial advances in various application domains. Typically, SASP problems can be formulated as optimization problems, where mathematical formulations (objective functions and constraints) are established from the prior knowledge of the sensor system models and the final processing goal.

Traditionally, solving SASP problems necessitates the manual formulation and development of algorithms by human experts. However, the recent invention of Large Language Models (LLMs) demonstrates the potential to revolutionize SASP problem-solving. In particular, LLMs are capable of comprehending natural language inputs and generating logical sequences as responses, allowing users to describe the SASP problem and requirement in an intuitive way. Moreover, LLM has shown talent towards comprehension on mathematical equations [4]–[6]. This enables the automation of optimization model and algorithm suggestions, streamlining the process of finding effective solutions for diverse SASP problems. This approach is termed automated optimization modeling (AOM), which has great potential for immediate but reasonable solutions for a wide range of SASP applications.

Currently, AOM methods [4], [6] predominantly utilize prompt engineering, including guiding LLMs to perform step-by-step reasoning [6]–[12] and deploying multi-agent systems to generate manually crafted response chains [13]–[16]. This approach engages LLMs in constructing logical sequences for problem solving, emulating human cognitive processes [7]. However, the inherent knowledge deficiencies present within LLMs has not yet been resolved. To clarify the knowledge referenced, the retrieval-augmented generation (RAG) [17], [18] method has recently been proposed. Notable performance improvements have been realized through the optimization of dataset structure [19] and the enhanced training of the retriever model [20]–[22]. However, despite these efforts, the SASP domain involves substantial domain-specific knowledge, limiting the success achieved by current AOM strategies.

To realize the potential of LLM-assisted AOM for SASP problem-solving, we introduce an automated modeling approach, which combines a multi-agent (MA) structure with a specific graph-based RAG (Graph-RAG) process. The MA structure is specifically tailored for the architectural complexities of AOM processes, following on principles of human expert's problem-solving logic. Each agent in the system is designed to tackle a segment of the problem, thereby decomposing a challenging mission into manageable subtasks [13]. The Graph-RAG component enhances this setup by matching user inputs with detailed domain modeling knowledge. This process mitigates the complexity of the AOM task, and further improves the performance by ensuring that only pertinent information is retrieved and utilized in modeling generation. Unlike traditional RAG, Graph-RAG organizes prior knowledge using a graph structure, making the retrieval process precise, crucial for fields like SASP where specific knowledge is needed [18]. The proposed approach is termed as MAG-RAG. To evaluate it, we build a testing dataset, which includes 10 classical SASP problems along with recommended solutions. The experimental results indicate that MAG-RAG approach outperforms several AOM benchmarks. Meanwhile, several challenging issues are identified and discussed for further research.

This work was supported in part by the National Nature Science Foundation of China (NSFC) under Grant 62101350, Grant 62206182, and Grant 62201362; in part by Shenzhen Science and Technology Program under Grant RCBS20221008093126071; and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515010154.

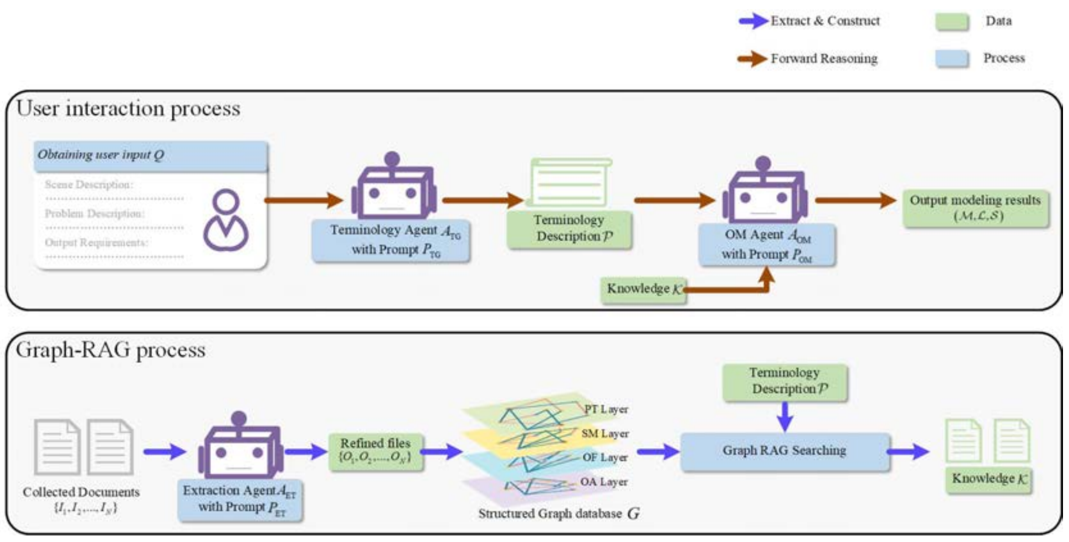

Extract & Construct
Forward Reasoning
Data
Process
User interaction process
Obtaining user input Q
Scene Description:
Problem Description:
Output Requirements:
Terminology Agent A_{TG} with Prompt P_{TG}
Terminology Description $\mathcal{P}$
OM Agent A_{OM} with Prompt P_{OM}
Knowledge $\mathcal{K}$
Output modeling results $(\mathcal{M},\mathcal{L},\mathcal{S})$
Graph-RAG process
Collected Documents $\{I_1,I_2,\ldots,I_N\}$
Extraction Agent A_{ET} with Prompt P_{ET}
Refined files $\{O_1,O_2,\ldots,O_N\}$
PT Layer
SM Layer
OF Layer
OA Layer
Structured Graph database G
Terminology Description $\mathcal{P}$
Graph RAG Searching
Knowledge $\mathcal{K}$

Fig. 1: The overall workflow of the proposed method.

2 Problem Formulation

To facilitate the emergence of domain-specific knowledge generation capabilities in LLMs, supervised fine-tuning (SFT) is typically employed to encode specialized knowledge into the additional neural network parameters. Given a dataset $\mathcal{D}$ on SASP field, the objective function of SFT [23] can be formulated as:

$$\underset{\Delta\theta}{\text{minimize}} \sum_{(\mathcal{A},\mathcal{Q})\in\mathcal{D}} (-\log\, p_{\theta+\Delta\theta}\,(\mathcal{A}|\mathcal{Q})), \tag{1}$$

where $\mathcal{Q}$ denotes the question text and $\mathcal{A}$ is the corresponding answer text. Notations θ and $\Delta\theta$ are the pre-trained parameter and tuning parameter in SFT process.

However, SFT is typically accompanied with high computational complexity and strict requirement on data quantity. Instead, we employ in-context learning for this problem [24] which enable us to exploit rich research document in the literature:

$$\underset{\Phi}{\text{minimize}} \; -\log\, p_\theta \left(\mathcal{A} \middle| \mathcal{Q}, \underbrace{\Phi(G,\mathcal{Q})}_{\mathcal{K}} \right), \tag{2}$$

where G represents a knowledge graph constructed from the literature documents (with details presented in Section 3.2), $\mathcal{K}$ denotes the $\mathcal{Q}$-related knowledge that has been retrieved from G, and Φ signifies the methodology employed for knowledge searching.

The key for designing Φ is designing relevance metric between $\mathcal{K}$ and $\mathcal{Q}$. In this work, we employ the cosine distance metric $\mathrm{d}(\cdot,\cdot)$ to measure the similarity between their embeddings. Further, to facilitate the searching process, we construct $\mathcal{K}$ as four parts:

$$\mathcal{K} = (\mathcal{P},\mathcal{M},\mathcal{L},\mathcal{S}), \tag{3}$$

where $\mathcal{P}$ represents the terminology description, $\mathcal{M}$ is the system model, $\mathcal{L}$ represents the optimization formulation, and $\mathcal{S}$ is the corresponding optimization algorithm. Based on such a decomposition, a straightforward way to solve (2) is to sequentially matching $\mathcal{Q}$ with the four constituent segments of $\mathcal{K}$, and subsequently concatenate four results to reconstruct $\mathcal{K}$. However, this method is computationally intensive, and the correlation between $\mathcal{Q}$ and $\mathcal{M}$, $\mathcal{L}$, $\mathcal{S}$ gradually diminishes in the semantic space [25]. Considering the sequential interdependencies within the logical chain of problem-solving, we further transform (2) into the following form:

$$\begin{aligned} \underset{\mathcal{P},\mathcal{M},\mathcal{L},\mathcal{S}}{\text{minimize}} \quad & \mathrm{d}(f_e(\mathcal{K}), f_e(\mathcal{Q})) \\ \text{s.t.} \quad & \mathcal{K} = (\mathcal{P},\mathcal{M},\mathcal{L},\mathcal{S}) \\ & \mathcal{P} \in G_{PT}, \mathcal{M} \in G_{SM}, \\ & \mathcal{L} \in G_{OF}, \mathcal{S} \in G_{OA} \end{aligned} \tag{4}$$

where $\mathcal{M} \in G_{SM}$, $\mathcal{L} \in G_{OF}$, $\mathcal{S} \in G_{OA}$ and $\mathcal{P} \in G_{PT}$ corresponds to "System Model (SM)" layer, "Optimization Formulation (OF)" layer, "Optimization Algorithm (OA)" layer, and "Problem Type (PT)" layer in G respectively (c.f. Section 3.2). The intuition behind the above transformation is straightforward, that the greater the relevance between $\mathcal{Q}$ and $\mathcal{K}$, the more likely the LLM is to yield desired outcomes. Building upon this insight, we have constructed a graph database and employed the RAG approach to enhance the performance of LLM output.

3 The Proposed AOM Approach

Solving problem (4) involves constructing a knowledge graph G, which is generated using a multi-agent workflow described in Sections 3.1 and 3.2. In Section 3.3, problem (4) is addressed by matching the cosine distance across each layer of the graph.

3.1 The MAG-RAG Pipeline

The pipeline of the developed approach consists of two workflows as illustrated in Fig. 1. The blue workflow illustrates the utilization of the Graph-RAG technique for constructing a knowledge database from domain-specific documents. This knowledge database can provide professional optimization modeling examples tailored to the user's query input (see example in Fig. 2). The other workflow in brown is the automated optimization modeling procedure, requiring the involvement of several agents. Before we formally introduce the pipeline structure, we specify three agents[1] for AOM in the SASP domain, namely, Extraction Agent A_{ET}, Terminology Agent A_{TG}, and Optimization Modeling Agent A_{OM}. The role of each agent is explained as follows:

Extraction Agent A_{ET}. This agent receives raw documents $\{I_1, I_2, \ldots, I_N\}$ in the SASP field as input, and strictly follows instructions from prompt P_{ET} to extract knowledge $\{O_1, O_2, \ldots, O_N\}$ critical for optimization modeling. Take $I_i,\ i \in [1, \ldots, N]$ as an example, the key knowledge extraction process can be formulated as follows:

$$O_i = A_{\text{ET}}\left(P_{\text{ET}}, I_i\right). \tag{5}$$

Terminology Agent A_{TG}. This agent transforms original user input $\mathcal{Q}$ into terminological description $\mathcal{P}$. Given a specially designed prompt instruction P_{TG}, the extraction process is as follows:

$$\mathcal{P} = A_{\text{TG}}\left(P_{\text{TG}}, \mathcal{Q}\right). \tag{6}$$

Optimization Modeling Agent A_{OM}. This agent provides a complete modeling result $(\mathcal{M}, \mathcal{L}, \mathcal{S})$ for $\mathcal{P}$ with reference to the extracted prior knowledge $\mathcal{K}$. Prior knowledge $\mathcal{K}$ is obtained from a Graph-RAG searching process (to be explained in Sec. 3.3) with the query embedding of $\mathcal{P}$. The generation process of A_{OM} can be formulated as follows:

$$(\mathcal{M}, \mathcal{L}, \mathcal{S}) = A_{\text{OM}}\left(P_{\text{OM}}, \mathcal{K}, \mathcal{P}\right). \tag{7}$$

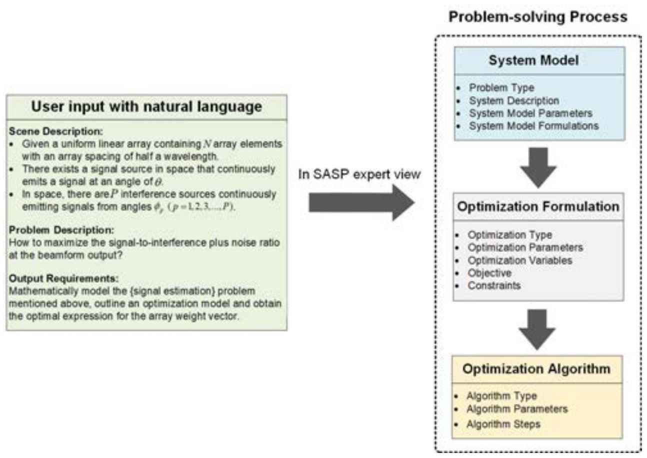
The pipeline of AOM is summarized as follows. Firstly, with the user-input query $\mathcal{Q}$ including the scene description, a Terminology Agent A_{TG} converts unspecified user input query into a terminological problem description $\mathcal{P}$. Secondly, we retrieve top-k most relevant documents as reference knowledge $\mathcal{K}$ based on description $\mathcal{P}$. Finally, combining the terminology description $\mathcal{P}$ with reference knowledge $\mathcal{K}$, Agent A_{OM} provides an answer $(\mathcal{M}, \mathcal{L}, \mathcal{S})$ as the output. The retrieval technique in the second step is Graph-RAG which will be explained in the subsequent section.

3.2 Graph-RAG Dataset Construction

To assist LLMs with sophisticated SASP modeling, we construct a graph-based data base, where domain knowledge is extracted and represented as nodes, then weighted edges are formed among correlated nodes before knowledge searching.

Modeling Information Requirements: Initially, N domain knowledge documents $\{I_1, I_2, \ldots, I_N\}$ are collected as raw property to construct the database. Though we anticipate the original documents to possess more information, they may also introduce information redundancy. Since not all the provided information contributes positively to the AOM task, excessive information imposes an additional burden on the modeling agent, which must first extract the essential information from the documents before proceeding with the subsequent modeling process. Besides, context limitation of LLMs [13], [26] should also be seriously treated.

[1]Agent in this work refers to one LLM, which takes task-specific prompt and task-relevant context as input. To make LLM agents being aware of their concrete responsibilities, each agent is provided a specially designed prompt that outlines specific tasks, guidelines, and structured outputs. Through prompt design, agents mirror the modeling principles of human experts.



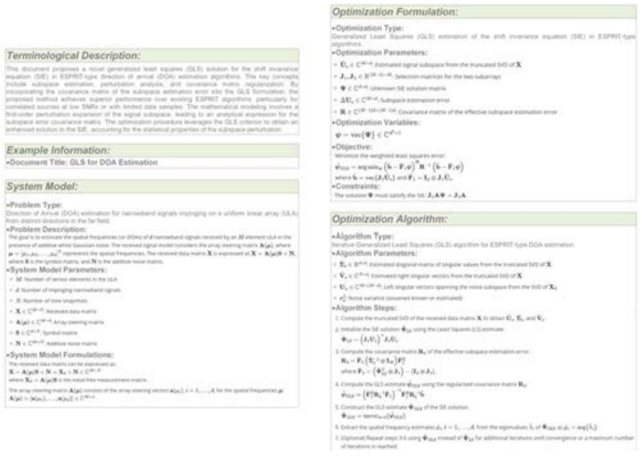
User input with natural language

Scene Description:
- Given a uniform linear array containing N array elements with an array spacing of half a wavelength.
- There exists a signal source in space that continuously emits a signal at an angle of θ.
- In space, there are P interference sources continuously emitting signals from angles ϕ_p $(p = 1, 2, 3, \ldots, P)$.

Problem Description:
How to maximize the signal-to-interference plus noise ratio at the beamform output?

Output Requirements:
Mathematically model the {signal estimation} problem mentioned above, outline an optimization model and obtain the optimal expression for the array weight vector.

Fig. 2: Human optimization modeling procedure for SASP problems.



Terminological Description:

This document proposes a novel generalized least squares (GLS) solution for the shift invariance equation (SIE) in ESPRIT-type direction of arrival (DOA) estimation algorithms. The key concepts include subspace estimation, perturbation analysis, and covariance matrix regularization. By incorporating the covariance matrix of the subspace estimation error into the GLS formulation, the proposed method achieves superior performance over existing ESPRIT algorithms, particularly for correlated sources at low SNRs or with limited data samples. The mathematical modeling involves a first-order perturbation expansion of the signal subspace, leading to an analytical expression for the subspace error covariance matrix. The optimization procedure leverages the GLS criterion to obtain an enhanced solution to the SIE, accounting for the statistical properties of the subspace perturbation.

Example Information:

•Document Title: GLS for DOA Estimation

System Model:

•Problem Type:
Direction of Arrival (DOA) estimation for narrowband signals impinging on a uniform linear array (ULA) from distinct directions in the far field.

•Problem Description:
The goal is to estimate the spatial frequencies (or DOAs) of d narrowband signals received by an M-element ULA in the presence of additive white Gaussian noise. The received signal model considers the array steering matrix $\mathbf{A}(\boldsymbol{\mu})$, where $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_d]^T$ represents the spatial frequencies. The received data matrix $\mathbf{X}$ is expressed as $\mathbf{X} = \mathbf{A}(\boldsymbol{\mu})\mathbf{S} + \mathbf{N}$, where $\mathbf{S}$ is the symbol matrix, and $\mathbf{N}$ is the additive noise matrix.

•System Model Parameters:
- M: Number of sensor elements in the ULA
- d: Number of impinging narrowband signals
- N: Number of time snapshots
- $\mathbf{X} \in \mathbb{C}^{M \times N}$: Received data matrix
- $\mathbf{A}(\boldsymbol{\mu}) \in \mathbb{C}^{M \times d}$: Array steering matrix
- $\mathbf{S} \in \mathbb{C}^{d \times N}$: Symbol matrix
- $\mathbf{N} \in \mathbb{C}^{M \times N}$: Additive noise matrix

•System Model Formulations:
The received data matrix can be expressed as:
$\mathbf{X} = \mathbf{A}(\boldsymbol{\mu})\mathbf{S} + \mathbf{N} = \mathbf{X}_0 + \mathbf{N} \in \mathbb{C}^{M \times N}$
where $\mathbf{X}_0 = \mathbf{A}(\boldsymbol{\mu})\mathbf{S}$ is the noise-free measurement matrix.
The array steering matrix $\mathbf{A}(\boldsymbol{\mu})$ consists of the array steering vectors $\mathbf{a}(\mu_i)$, $i = 1, \ldots, d$, for the spatial frequencies μ_i:
$\mathbf{A}(\boldsymbol{\mu}) = [\mathbf{a}(\mu_1), \ldots, \mathbf{a}(\mu_d)] \in \mathbb{C}^{M \times d}$

Optimization Formulation:

•Optimization Type:
Generalized Least Squares (GLS) estimation of the shift invariance equation (SIE) in ESPRIT-type algorithms.

•Optimization Parameters:
- $\hat{\mathbf{U}}_s \in \mathbb{C}^{M \times d}$: Estimated signal subspace from the truncated SVD of $\mathbf{X}$
- $\mathbf{J}_1, \mathbf{J}_2 \in \mathbb{R}^{(M-1) \times M}$: Selection matrices for the two subarrays
- $\boldsymbol{\Psi} \in \mathbb{C}^{d \times d}$: Unknown SIE solution matrix
- $\Delta\mathbf{U}_s \in \mathbb{C}^{M \times d}$: Subspace estimation error
- $\mathbf{R} \in \mathbb{C}^{(M-1)d \times (M-1)d}$: Covariance matrix of the effective subspace estimation error

•Optimization Variables:
$\boldsymbol{\psi} = \text{vec}\{\boldsymbol{\Psi}\} \in \mathbb{C}^{d^2 \times 1}$

•Objective:
Minimize the weighted least squares error:
$\hat{\boldsymbol{\psi}}_{\text{GLS}} = \arg\min_{\boldsymbol{\psi}} \left(\hat{\mathbf{b}} - \hat{\mathbf{F}}_1\boldsymbol{\psi}\right)^H \mathbf{R}^{-1} \left(\hat{\mathbf{b}} - \hat{\mathbf{F}}_1\boldsymbol{\psi}\right)$
where $\hat{\mathbf{b}} = \text{vec}\{\mathbf{J}_2\hat{\mathbf{U}}_s\}$ and $\hat{\mathbf{F}}_1 = \mathbf{I}_d \otimes \mathbf{J}_1\hat{\mathbf{U}}_s$

•Constraints:
The solution $\boldsymbol{\Psi}$ must satisfy the SIE: $\mathbf{J}_1\mathbf{A}\boldsymbol{\Psi} = \mathbf{J}_2\mathbf{A}$

Optimization Algorithm:

•Algorithm Type:
Iterative Generalized Least Squares (GLS) algorithm for ESPRIT-type DOA estimation

•Algorithm Parameters:
- $\hat{\boldsymbol{\Sigma}}_s \in \mathbb{R}^{d \times d}$: Estimated diagonal matrix of singular values from the truncated SVD of $\mathbf{X}$
- $\hat{\mathbf{V}}_s \in \mathbb{C}^{N \times d}$: Estimated right singular vectors from the truncated SVD of $\mathbf{X}$
- $\mathbf{U}_n \in \mathbb{C}^{M \times (M-d)}$: Left singular vectors spanning the noise subspace from the SVD of $\mathbf{X}_0$
- σ_n^2: Noise variance (assumed known or estimated)

•Algorithm Steps:
1. Compute the truncated SVD of the received data matrix $\mathbf{X}$ to obtain $\hat{\mathbf{U}}_s$, $\hat{\boldsymbol{\Sigma}}_s$, and $\hat{\mathbf{V}}_s$
2. Initialize the SIE solution $\hat{\boldsymbol{\Psi}}_{\text{LS}}$ using the Least Squares (LS) estimate:
$\hat{\boldsymbol{\Psi}}_{\text{LS}} = \left(\mathbf{J}_1\hat{\mathbf{U}}_s\right)^+ \mathbf{J}_2\hat{\mathbf{U}}_s$
3. Compute the covariance matrix $\hat{\mathbf{R}}_\epsilon$ of the effective subspace estimation error:
$\hat{\mathbf{R}}_\epsilon = \hat{\mathbf{F}}_2\left(\hat{\boldsymbol{\Sigma}}_s^{-2} \otimes \mathbf{I}_M\right)\hat{\mathbf{F}}_2^H$
where $\hat{\mathbf{F}}_2 = \left(\hat{\boldsymbol{\Psi}}_{\text{LS}}^T \otimes \mathbf{J}_1\right) - \left(\mathbf{I}_d \otimes \mathbf{J}_2\right)$
4. Compute the GLS estimate $\hat{\boldsymbol{\psi}}_{\text{GLS}}$ using the regularized covariance matrix $\hat{\mathbf{R}}_\epsilon$:
$\hat{\boldsymbol{\psi}}_{\text{GLS}} = \left(\hat{\mathbf{F}}_1^H\hat{\mathbf{R}}_\epsilon^{-1}\hat{\mathbf{F}}_1\right)^{-1}\hat{\mathbf{F}}_1^H\hat{\mathbf{R}}_\epsilon^{-1}\hat{\mathbf{b}}$
5. Construct the GLS estimate $\hat{\boldsymbol{\Psi}}_{\text{GLS}}$ of the SIE solution:
$\hat{\boldsymbol{\Psi}}_{\text{GLS}} = \text{unvec}_{d \times d}(\hat{\boldsymbol{\psi}}_{\text{GLS}})$
6. Extract the spatial frequency estimates $\hat{\mu}_i$, $i = 1, \ldots, d$, from the eigenvalues $\hat{\lambda}_i$ of $\hat{\boldsymbol{\Psi}}_{\text{GLS}}$ as $\hat{\mu}_i = \arg\{\hat{\lambda}_i\}$
7. (Optional) Repeat steps 3-6 using $\hat{\boldsymbol{\Psi}}_{\text{GLS}}$ instead of $\hat{\boldsymbol{\Psi}}_{\text{LS}}$ for additional iterations until convergence or a maximum number of iterations is reached.

Fig. 3: An example of the output generated by Example Extraction Agent

Thus, an Extraction Agent A_{ET} is developed to distill the original documents into content pertinent to optimization modeling $\{O_1, O_2, \ldots, O_N\}$. In order to conquer the uncontrollability of the LLM responses, we make it strictly follow the modeling mindset of human expert (as illustrated in Fig.2) with a target-definite prompt, where LLM response is instructed to consist of five parts: terminological description, example information, system model, optimization formulation and optimization algorithm. An example of the results is illustrated in Fig. 3.

Dataset Construction of Graph-RAG: The extracted content is formulated as a graph structure [19], [27]:

$$G = (V, E), \tag{8}$$

where G denotes the graph dataset, V and E represent nodes and edges, respectively. Such a structure is chosen due to the fact that the graph structure is inherently endowed with a hierarchical process, allowing for a natural division of graphs into sub-graphs to capture the optimization modeling procedure. Additionally, graphs present superior flexibility for community clustering and efficient RAG searching. The intrinsic logic and hierarchical relationships inherent in graphs closely align with the optimization modeling procedure employed by human experts.

Concretely, a four-layer graph consisting of "System Model (SM)" layer, "Optimization Formulation (OF)" layer, "Optimization Algorithm (OA)" layer and "Problem Type (PT)" layer is constructed. For each $O_i,\ i \in [1, \ldots, N]$, related content is traced and represented as nodes, which are subsequently allocated to the corresponding layers. Besides, nodes inherit the type of their respective layers, and are further assigned a "keyword" attribute generated using all-MiniLM-L6-v2 [28] to serve as the searching query.

Two types of edges are developed according to the nodes' belonging entity: Nodes extracted from the same document are connected with "single document (SD)" edges weighing 1.0, following the unoriented chain of "PT-SM-OF-OA". Nodes extracted from different documents are linked with "different documents (DD)" edges, with the edge weights representing their similarity. Specifically, for nodes n_i and n_j, the embedding of each node's key words is generated as:

$$v_i = f_e\left(f_k\left(n_i\right)\right),\ v_j = f_e\left(f_k\left(n_j\right)\right), \tag{9}$$

where f_e denotes the transformation from natural language to feature space using text-embedding-3-small [29], and f_k is the value extraction process from the node attribute "keyword". Then, cosine similarity s_{ij} is applied to calculate the relevance:

$$s_{ij} = \frac{v_i \cdot v_j}{||v_i|| \times ||v_i||}. \tag{10}$$

If s_{ij} is greater than ε, a relationship between n_i and n_j is established, with s_{ij} assigned as the value of "similarity" attribute.

3.3 Knowledge Searching Using Graph-RAG

Knowledge searching process plays a vital role in Graph-RAG procedure. Due to the tendency of LLMs to utilize few-shot learning [30], they prefer to draw on the given examples for responses. Thus, following formulation (4), we retrieve the nearest knowledge in G w.r.t. $\mathcal{Q}$ sequentially. Furthermore, A_{TG} produces content consistent with SASP terminologies, thus we employ "PT" layer for knowledge searching. For instance, considering a node $n_p, p \in [1, \ldots, N]$, the relevance of it to $\mathcal{P}$ can be determined as:

$$s_{ep} = \frac{f_e\left(\mathcal{P}\right) \cdot f_e\left(f_k\left(n_p\right)\right)}{||f_e\left(\mathcal{P}\right)|| \times ||f_e\left(f_k\left(n_p\right)\right)||}. \tag{11}$$

Subsequently, we compute the distance from each node n_p in the "PT" layer to $\mathcal{P}$, encapsulate the result as a key-value pair $\{n_p : s_{ep}\}$, and aggregate these pairs into a set L:

$$L \leftarrow L \cup \left\{n_p : s_{ep}\right\}. \tag{12}$$

Finally, the nodes corresponding to the top-k similarities in L are selected. We build the knowledge $\mathcal{K}$ for A_{OM} by concatenating the node content connected by "SD" edges from these selected nodes. In this paper, we set $k = 3$, by taking into account the context limitation and knowledge richness.

4 Experiments

4.1 Experimental Setup

Dataset: We select ten classical SASP problems to evaluate the AOM performance, including transmitted beam pattern matching (Q1), cooperative sensing under ideal communication conditions (Q2), sensor placement (Q3), MIMO radar waveform design (Q4), direct positioning determination (Q5), DOA estimation (Q6), interference signal suppression (Q7), bearing-based localization (Q8), TOA-based localization (Q9) and TDOA-FDOA-based localization (Q10). For each issue, we finely select a number of documents containing standard modeling approaches to construct dataset SPAMR. The selected documents contain heuristic modeling strategies and optimization algorithms that can help researchers and LLMs improve the modeling process. The implementation code and evaluation dataset are available at `https://github.com/advantages/MAG-RAG-for-SASP`.

Comparison Methods: To fully evaluate MAG-RAG, we employ two external benchmarks. Pure MA refers to a pure agent logic chain for AOM, that the knowledge retrieval process of Graph-RAG is replaced by Knowledge Generation Agent A_{KG}. Pure LLM outputs the overall solutions for the input signal processing issue $\mathcal{Q}$ without any reference or prior knowledge.

Metrics: Five metrics granted different scores in overall 100'are adopted to evaluate the modeling results: Completeness (30'), Standardization (20'), Correctness (30'), Relevance (10'), Readability (10').

4.2 Performance Evaluation

With the assistance of three human scientists specializing in SASP domain, we evaluate all the generated AOM results. The overall performances are shown in Table 1.

Table 1: Overall performance on different base LLMs. H: Haiku-3 [31], S: Sonnet-3, G3.5: GPT-3.5, G4: GPT-4. D: pure LLM, G: MAG-RAG, T: pure MA

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
HD	82	62	40	62	73	72	89	90	85	72
HG	70	60	70	75	64	81	61	63	42	46
HT	87	65	0	70	58	61	48	88	**92**	76
SD	75	21	42	71	68	75	**91**	91	90	84
SG	62	15	**85**	**82**	**80**	**82**	85	**96**	**92**	**91**
ST	70	40	35	79	71	80	80	92	87	81
G3.5D	10	40	0	41	53	51	54	30	61	40
G3.5G	75	40	44	49	64	73	46	38	43	46
G3.5T	35	20	45	28	53	28	15	52	61	43
G4D	60	65	60	70	63	75	74	75	72	59
G4G	**92**	45	60	52	66	80	64	83	70	71
G4T	65	**78**	60	55	58	81	52	76	80	68

Three human experts are responsible for evaluating Q1-Q3, Q4-Q6 and Q7-Q10, respectively. From Table. 1, we discover that there remains a strong tendency in the given scores, where LLMs with MAG-RAG tend to achieve higher scores, despite the varied scoring preferences of the human scientists. Moreover, Sonnet-3 typically achieves better performance on modeling tasks across ten selected SASP problems. We statistically calculate the distributional properties of different metrics, and the results are shown in Fig. 4.

From Fig. 4(A), we observe that MAG-RAG achieves better results, occupying 67 percent on items getting the highest score, while pure MA and pure LLM approaches achieve only 25 percent and 8 percent, respectively. And in terms

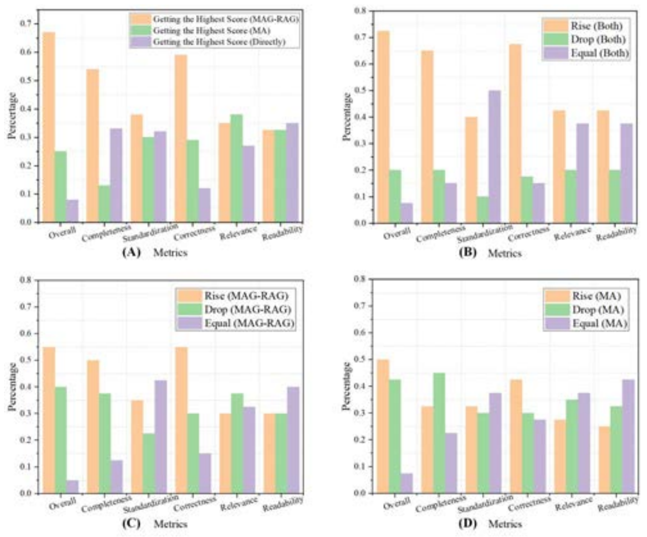


Fig. 4: Statistical results on overall scores. (A) Frequency of different methods achieving the highest scores across various metrics. (B) Frequency of scoring gains (positive, negative and no gain) obtained with prior knowledge (both pure MA and MAG-RAG) compared to pure LLM. (C) Frequency of scoring gains obtained with MAG-RAG compared to pure LLM. (D) Frequency of scoring gains obtained with pure MA compared to pure LLM.

of completeness and correctness, MAG-RAG similarly far outperforms the comparison benchmarks, demonstrating that the knowledge extracted using a specially designed graph database has a positive effect on modeling.

In viewing of standardization, relevance and readability, three methodologies perform similarly. This phenomenon is caused by the fact that LLM itself performs well in content formation, and the insertion of prior knowledge primarily aims to promote the optimization modeling.

From Fig. 4 (B), we conclude that the utilization of prior knowledge for specific problems can indeed have a positive impact on modeling results, with the percentage of improved scores significantly exceeding that of declined scores. And in the cases where scores decreased, we statistically discover that four out of the eight samples had reduced scores originating from Q7. By referencing scientists' advice on Q7, directly invoking LLMs usually achieves higher results (89, 91, 54, 74), aligning with the expectations of different base LLMs. However, with the insertion of extracted prior knowledge, LLMs may additionally introduce incorrect constraints or miss key steps in algorithms.

In Fig. 4 (C) and (D), we find that LLMs augmented with prior knowledge generally yield lower performance in terms of readability and contextual relevance compared to directly employing LLMs. Considering the auto-regression process in LLMs, this phenomenon may be attributed to the following factors: During the process of knowledge integration, the attention mechanism in Transformers often struggles to allocate attention weights appropriately, leading to biases in comprehension.

Subsequently, we devise experiments to validate the intuition behind the transformation from (2) to (4). Given that the models utilized are all commercial LLMs with closed

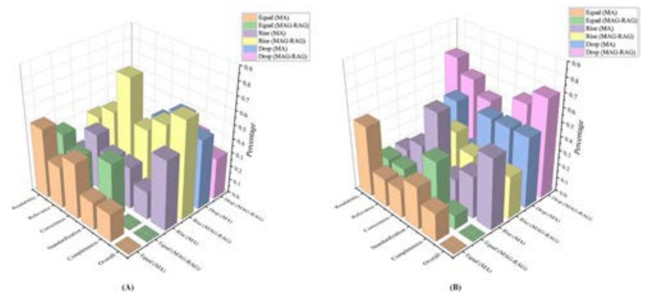


Fig. 5: Statistical results of the scores. (A) Performance of claude-3-sonnet-20240229. (B) Performance of claude-3-haiku-202440307

sources, we are unable to precisely determine the probability distribution of $\mathcal{A}$. Consequently, we have resorted to statistical methods as a substitute for this purpose. To enhance the persuasiveness of our findings, we select LLMs from different companies, one with superior performance and another with comparatively weaker capabilities, to evaluate their performance on Q6. Each LLM was tasked with generating ten responses—five of which were informed by more relevant corpora as $\mathcal{K}$, and the remaining five by less relevant corpora. In particular, we employ the relevance metric ϕ defined as follows to substantiate the mentioned intuition:

$$\phi = \frac{1}{C(5,2)} \sum_{i=1}^{4} \sum_{j=i+1}^{5} \sum_{\substack{\mathcal{C}_i=\{\mathcal{M}_i,\mathcal{L}_i,\mathcal{S}_i\} \\ \mathcal{C}_j=\{\mathcal{M}_j,\mathcal{L}_j,\mathcal{S}_j\}}} \mathrm{d}(f_e(\mathcal{C}_{ik}), f_e(\mathcal{C}_{jk})). \tag{13}$$

The results are documented in Table 2. As evidenced, our hypothesis holds true irrespective of whether the LLM is of high or low performance. This substantiates the assertion that the accuracy of reference knowledge exerts a significant influence on the performance of LLMs.

Table 2: Evaluated performance. S: Sonnet-3, G3.5: GPT-3.5, G: MAG-RAG, H: high relevance of $\mathcal{K}$ towards Q6, L: low relevance

	value of ϕ	relevance of $\mathcal{K}$
SG-H	0.92	0.78
SG-L	0.46	0.35
G3.5G-H	0.67	0.69
G3.5G-L	0.32	0.29

To explore the domain knowledge interpretation capability bias towards LLMs, we set up an ablation test. We choose the weaker model haiku in claude-3 as well as the stronger model sonnet for comparison. The results are displayed in Fig. 5.

The results show that sonnet outperforms haiku, particularly with MAG-RAG, where sonnet achieves significant gains while haiku's performance declines sharply. The decline in haiku can be attributed to two main factors. One is that Haiku's reliance on prior knowledge leads to inaccuracies in understanding SASP issues, as it neglects actual configurations in $\mathcal{P}$ and overly depends on extracted knowledge. The other is that Haiku's modeling logic and algorithm choices are suboptimal, lacking the ability to filter out noisy knowledge.

In conclusion, long-context learning for LLMs remains an unresolved and underexplored challenge. MAG-RAG, leveraging multi-agent chains and graph-based knowledge search, delivers more reliable modeling results for high-capability LLMs.

5 Conclusion

In this paper, we propose MAG-RAG approach for AOM, targeting SASP problems. We transform the AOM process into consecutive but separate parts, and based on this, a MA architecture is utilized to assign different sub-tasks into different LLMs. To enhance the efficiency, a graph-based RAG is adopted, where prior knowledge can be structurally stored and searched with more performance improvement.

Finally, we note that the proposed MAG-RAG mainly explores AOM for different SASP problems, while implicit relation between similar optimization algorithms is not sufficiently explored. Moreover, the inherent clustering strategy towards correlative SASP issues that may potentially enhance the knowledge searching efficiency is still unexplored.

References

[1] M. Pesavento, M. Trinh-Hoang, and M. Viberg, Three more decades in array signal processing research: An optimization and structure exploitation perspective, *IEEE Signal Processing Magazine*, vol. 40, no. 4, pp. 92–106, 2023.

[2] H. Krim and M. Viberg, Two decades of array signal processing research: The parametric approach, *IEEE signal processing magazine*, vol. 13, no. 4, pp. 67–94, 1996.

[3] W. Liu, M. Haardt, M. S. Greco, C. F. Mecklenbräuker, and P. Willett, Twenty-five years of sensor array and multichannel signal processing: A review of progress to date and potential research directions, *IEEE Signal Processing Magazine*, vol. 40, no. 4, pp. 80–91, 2023.

[4] Z. Tang, C. Huang, X. Zheng, *et al.*, Orlm: Training large language models for optimization modeling, *arXiv preprint arXiv:2405.17743*, 2024.

[5] P. Song, K. Yang, and A. Anandkumar, Towards large language models as copilots for theorem proving in lean, *arXiv preprint arXiv:2404.12534*, 2024.

[6] A. AhmadiTeshnizi, W. Gao, and M. Udell, Optimus: Optimization modeling using mip solvers and large language models, *arXiv preprint arXiv:2310.06116*, 2023.

[7] J. Wei, X. Wang, D. Schuurmans, *et al.*, Chain-of-thought prompting elicits reasoning in large language models, *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[8] S. Yao, D. Yu, J. Zhao, *et al.*, Tree of thoughts: Deliberate problem solving with large language models, *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[9] M. Besta, N. Blach, A. Kubicek, *et al.*, Graph of thoughts: Solving elaborate problems with large language models, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 17 682–17 690.

[10] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, Large language models are zero-shot reasoners, *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[11] B. Wang, X. Deng, and H. Sun, Iteratively prompt pre-trained language models for chain of thought, *arXiv preprint arXiv:2203.08383*, 2022.

[12] L. Gao, A. Madaan, S. Zhou, *et al.*, Pal: Program-aided language models, in *International Conference on Machine Learning*, PMLR, 2023, pp. 10 764–10 799.

[13] Z. Xiao, D. Zhang, Y. Wu, *et al.*, Chain-of-experts: When llms meet complex operations research problems, in *The Twelfth International Conference on Learning Representations*, 2023.

[14] C.-M. Chan, W. Chen, Y. Su, *et al.*, Chateval: Towards better llm-based evaluators through multi-agent debate, *arXiv preprint arXiv:2308.07201*, 2023.

[15] Y. Talebirad and A. Nadiri, Multi-agent collaboration: Harnessing the power of intelligent llm agents, *arXiv preprint arXiv:2306.03314*, 2023.

[16] Q. Wu, G. Bansal, J. Zhang, *et al.*, Autogen: Enabling next-gen llm applications via multi-agent conversation framework, *arXiv preprint arXiv:2308.08155*, 2023.

[17] G. Mialon, R. Dessı, M. Lomeli, *et al.*, Augmented language models: A survey, *arXiv preprint arXiv:2302.07842*, 2023.

[18] P. Lewis, E. Perez, A. Piktus, *et al.*, Retrieval-augmented generation for knowledge-intensive nlp tasks, *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[19] D. Edge, H. Trinh, N. Cheng, *et al.*, From local to global: A graph rag approach to query-focused summarization, *arXiv preprint arXiv:2404.16130*, 2024.

[20] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, Retrieval augmented language model pre-training, in *International conference on machine learning*, PMLR, 2020, pp. 3929–3938.

[21] O. Ram, Y. Levine, I. Dalmedigos, *et al.*, In-context retrieval-augmented language models, *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1316–1331, 2023.

[22] S. Weijia, M. Sewon, Y. Michihiro, *et al.*, Replug: Retrieval-augmented black-box language models, *ArXiv: 2301.12652*, 2023.

[23] E. J. Hu, Y. Shen, P. Wallis, *et al.*, Lora: Low-rank adaptation of large language models. *ICLR*, vol. 1, no. 2, p. 3, 2022.

[24] S. Min, X. Lyu, A. Holtzman, *et al.*, Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

[25] N. Reimers and I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, *arXiv preprint arXiv:1908.10084*, 2019.

[26] Y. Ding, L. L. Zhang, C. Zhang, *et al.*, Longrope: Extending llm context window beyond 2 million tokens, *arXiv preprint arXiv:2402.13753*, 2024.

[27] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, *et al.*, Graph attention networks, *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.

[28] Y. He, Z. Yuan, J. Chen, and I. Horrocks, Language models as hierarchy encoders, *arXiv preprint arXiv:2401.11374*, 2024.

[29] T. Abdullahi, R. Singh, and C. Eickhoff, Retrieval augmented zero-shot text classification, in *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*, 2024, pp. 195–203.

[30] S. Cahyawijaya, H. Lovenia, and P. Fung, Llms are few-shot in-context low-resource language learners, *arXiv preprint arXiv:2403.16512*, 2024.

[31] A. Anthropic, The claude 3 model family: Opus, sonnet, haiku, *Claude-3 Model Card*, vol. 1, 2024.