Decision Analytics Journal 15 (2025) 100587

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

A structured review of large language models in metaheuristic optimisation

Reza Ghanbarzadeh [a], Seyedali Mirjalili [b,c,d,*]

[a] Faculty of Science and Engineering, Southern Cross University, Gold Coast, Australia
[b] Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Melbourne 3000, Australia
[c] Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava, Ostrava, Czech Republic
[d] University Research and Innovation Center, Obuda University, Budapest, 1034, Hungary

ARTICLE INFO

Keywords:
Systematic literature review
Optimisation algorithms
Large language models
Swarm intelligence
Metaheuristic
Generative AI
ChatGPT
LLM

ABSTRACT

Metaheuristics are widely used to address complex optimisation problems where traditional exact methods are computationally infeasible or insufficiently flexible. With the rapid advancement of artificial intelligence, large language models, such as ChatGPT, Claude, Gemini, and LLaMA, have emerged as powerful tools capable of enhancing, automating, and adapting various stages of the optimisation process. This systematic literature review investigates the evolving role of large language models in metaheuristic optimisation, with a focus on algorithm generation, parameter tuning, hybridisation, constraint handling, and multi-objective optimisation. Following PRISMA guidelines, 25 studies from nine major scientific databases were selected and analysed. Through thematic analysis, a novel role-based taxonomy was developed that categorises large language model contributions into four functional roles: Advisor, Refiner, Enhancer, and Innovator. The findings demonstrate that large language models support the automation of metaheuristic workflows, enable dynamic adaptation, and contribute to the creation of novel heuristic strategies. Despite these advantages, the review also identifies persistent limitations, including prompt sensitivity, computational overhead, and scalability challenges. These issues highlight the need for more robust evaluation frameworks and benchmarking practices. This review offers a comprehensive synthesis of the current landscape, highlights research gaps, and provides actionable insights for researchers and practitioners aiming to integrate large language models into advanced optimisation systems across domains such as engineering, logistics, and computational design.

1. Introduction

The field of computational intelligence has undergone significant transformation in recent decades, driven by unprecedented advances in algorithms, data availability, and computational infrastructure. Algorithms have become substantially more scalable, accurate, and robust, making them suitable for a wide range of applications. The increased availability of data, fuelled by widespread interest in data collection and analysis, has further expanded the scope of computational modelling. Simultaneously, the growth of cloud computing has led to the development of large-scale data centres, and computing resources have become increasingly accessible. Collectively, these advancements have enabled researchers to address previously intractable challenges across various domains [1].

Computational intelligence typically comprises three main subfields: artificial neural networks, fuzzy logic, and evolutionary computation. The key developments in these three areas are shown in Fig. 1. In artificial neural networks, back-propagation, which was proposed in the 1980s [2], remains the main principle for training neural networks today. In 2010, deep learning marked the next big step forward, which allowed neural networks to have multiple hidden layers [3]. This improved their ability to recognise patterns, process complex data, and handle various tasks, like image and speech recognition. This advancement created new possibilities for Artificial Intelligence (AI) and supported important research and applications in different industries. In 2020, another milestone came with the development of transformer architectures. Transformers [4] use attention mechanisms to focus on relevant parts of input data. This makes them highly effective for processing natural language. These innovations resulted in the invention of Large Language Models (LLMs) that can understand and generate human-like text [5]. The capabilities of deep learning combined with transformers led to an unprecedented global interest in AI by the end of 2021. In 2022, tools like ChatGPT, Gemini, and Claude LLMs became very popular, and this marked the start of a new AI boom. These tools have made advanced AI more accessible to the public and businesses, leading to widespread applications in education, customer service, content creation, and beyond.

* Corresponding author at: University Research and Innovation Center, Obuda University, Budapest, 1034, Hungary.
E-mail addresses: Reza.Ghanbarzadeh@scu.edu.au (R. Ghanbarzadeh), ali.mirjalili@torrens.edu.au (S. Mirjalili).

https://doi.org/10.1016/j.dajour.2025.100587
Received 26 February 2025; Received in revised form 17 May 2025; Accepted 28 May 2025
Available online 31 May 2025
2772-6622/

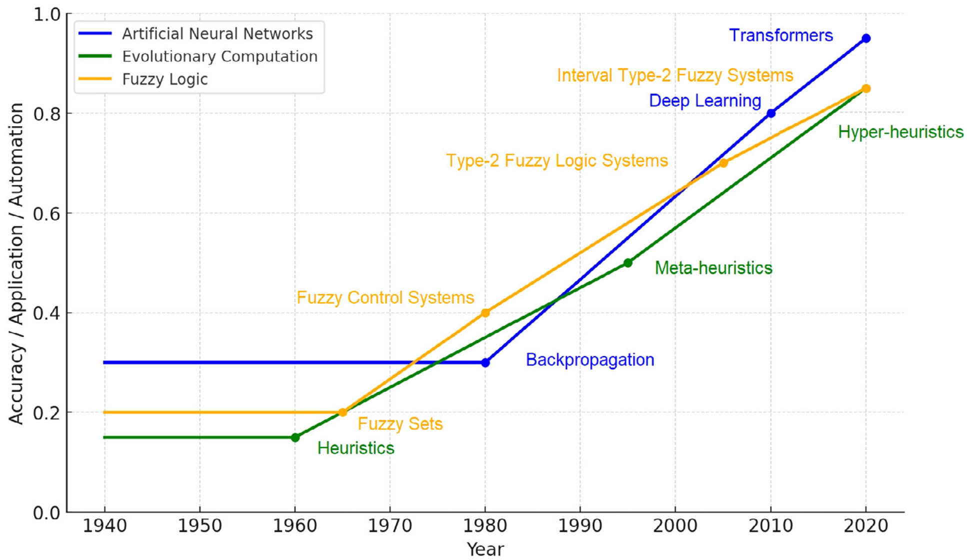

Fig. 1. Major breakthroughs in artificial neural networks, fuzzy, and evolutionary computation. The curves are not to scale but represent general trends in performance improvements across stages, such as accuracy, application, and automation.

In the field of evolutionary computation, early methods called heuristics appeared in the 1960s. These methods included techniques like greedy search and the A* algorithm. Greedy search works by making locally optimal choices in each step to find a quick solution for a given problem. For instance, the A* algorithm, which is usually used in graph theory and combinatorial optimisation, improved pathfinding by combining both the cost to reach a node and the estimated cost to the goal. The limitation of these early algorithms was that they were problem-specific. This means that they worked only for particular types of problems. This issue was addressed by the introduction of metaheuristics [6], which are broader, more adaptable strategies. Metaheuristics expanded the applicability of optimisation algorithms, generating considerable research interest and growth in the evolutionary computation community. This shift led to the development of many new algorithms for various applications. In recent years, we have also seen a move towards more automation in evolutionary computation. Hyper-heuristics were introduced to reduce human intervention by automating the selection and combination of heuristics [7]. This allows systems to adapt more easily and makes the field even more versatile and accessible for complex problem-solving.

Recent studies have demonstrated the versatility of hyper-heuristics and metaheuristics across a range of real-world decision problems. For instance, Dey et al. [8] applied a metaheuristic-based ensemble method for cyber threat detection in IoT networks, which addresses the challenge of high-dimensional feature selection. Zuhanda et al. [9] proposed a hybrid optimisation framework for solving vehicle routing problems with shipment consolidation, which combines population-based and swarm intelligence techniques. Similarly, Rahman et al. [10] developed a decision support system for e-waste recycling operations using a bio-inspired optimisation algorithm. These examples show the growing adoption of metaheuristics in critical domains, which reinforces the value of further enhancing their design and integration through LLMs.

Fuzzy logic [11] has likely grown more slowly than artificial neural networks and evolutionary computation, as there have been no major changes in the last 30 to 40 years. It began with the proposal of fuzzy sets in the 1960s, and these sets introduced ways to handle vague or uncertain information, where values are not simply true or false. Later, fuzzy control systems were developed, which allowed for smoother and more flexible control in applications that needed to handle uncertainty. Eventually, type-2 fuzzy systems were introduced that provided an even better way to manage complex uncertainties by allowing “fuzziness” in the rules themselves. Despite these advancements, progress has remained steady but slow. The main concepts of fuzzy logic have largely stayed the same over the decades. Today, fuzzy logic remains useful in fields such as control systems, decision-making, and engineering, where it helps manage imprecise information. However, as artificial neural networks and evolutionary computation have advanced rapidly, fuzzy logic is often used alongside these newer technologies, rather than standing alone. This integration allows systems to combine the strengths of each approach and create more adaptable and resilient solutions.

Despite the differences between artificial neural networks, evolutionary computation, and fuzzy logic, they share some common trends. As shown in Fig. 1, all three areas have seen rises in accuracy, applicability, and automation. Each field has made systems more precise and capable of handling complex tasks with less human input. Among the major breakthroughs, the development of transformers and LLMs has had a particularly high impact on the general public. LLMs have demonstrated the potential of digital intelligence through large-scale AI systems, changing how we interact with technology in everyday life. If advancements continue at this pace, we may soon reach Artificial General Intelligence (AGI), also known as human-level AI, which would represent the highest level of automation, where computers could match human cognitive abilities. This would mark a major milestone, bringing us closer to creating systems that can think and learn like humans.

As discussed above, artificial neural networks and evolutionary computation have undergone many major changes with recent attention and adoption. Due to the novelty of paradigms like LLMs and hyper-heuristics, and despite many studies in these areas, we see a growing niche area with potential impact in the future, as shown in Fig. 2. This figure shows that the intersection of LLMs and optimisation represents a growing niche where AI and optimisation techniques enhance each other. This means that LLMs and optimisation algorithms work together to perform different tasks. This collaboration between LLMs and optimisation algorithms across distinct tasks motivated researchers to investigate the current state-of-the-art in this emerging field. As a result, recent studies, such as [12–15], have begun exploring the integration of LLMs with metaheuristics, focusing on specific algorithmic improvements and applications.

Fig. 3 illustrates the evolution of LLMs, which highlightes key breakthroughs that have shaped their development. The timeline begins with the resurgence of deep learning in 2010, which provided the foundation for modern artificial intelligence systems. A significant milestone

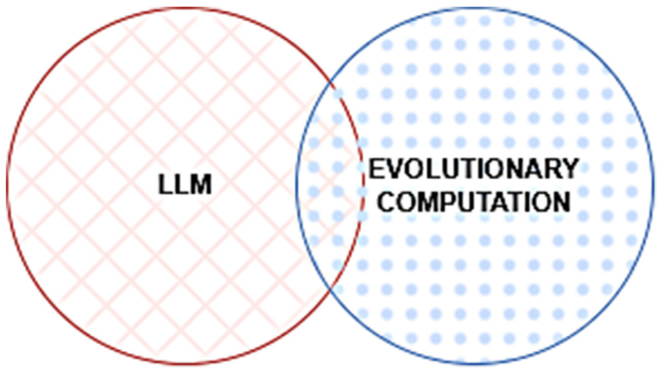

Fig. 2. The intersection of LLMs and Optimisation, which represents a growing niche where AI and optimisation techniques enhance each other.

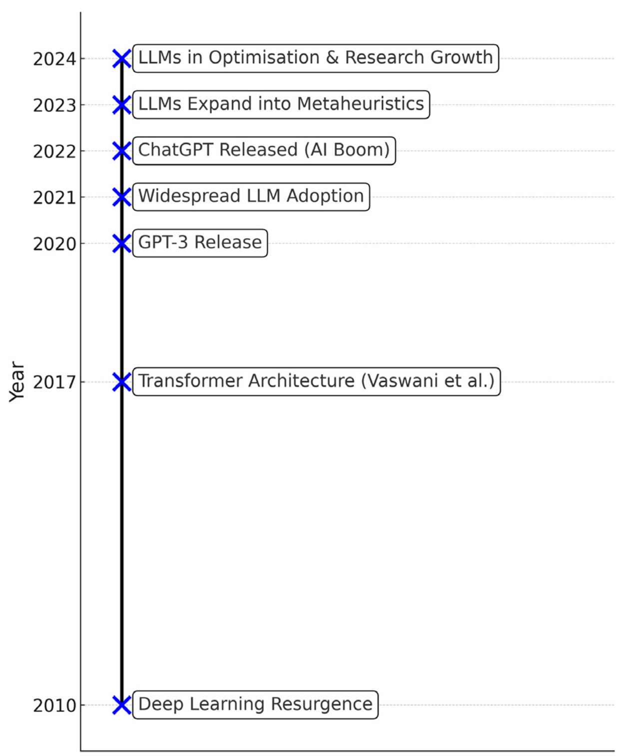

Fig. 3. Evolution of large language models.

was reached in 2017 with the introduction of the Transformer architecture [4], revolutionising natural language processing by enabling models to process entire sequences efficiently. In 2020, OpenAI released Generative Pre-trained Transformer-3 (GPT-3), marking the first large-scale generative model capable of human-like text generation. The widespread adoption of LLMs accelerated in 2021, leading to the launch of ChatGPT in 2022, which sparked a global AI boom [16]. More recently, LLMs have extended their applications beyond traditional natural language processing, entering the field of metaheuristic optimisation in 2023, where they have been leveraged for algorithm design, parameter tuning, and solution generation. As shown in Fig. 3, this progression continues into 2024, with growing research interest in integrating LLMs into optimisation and computational intelligence, opening new frontiers for hybrid AI-driven problem-solving.

The integration of LLMs into metaheuristic optimisation represents a transformative shift in computational intelligence, offering capabilities that traditional methods lack. Unlike classical metaheuristics, which often require manual fine-tuning, heuristic selection, and extensive domain expertise, LLMs bring a new level of automation, adaptability, and self-learning to the optimisation process. With their ability to process unstructured data, dynamically adapt parameters, and generate novel solutions, LLMs offer a promising avenue for addressing complex optimisation problems. LLMs have the capability to handle unstructured data, create optimisation strategies, and adjust parameters dynamically, reducing the need for manual heuristics and making it easier to solve complex problems. Moreover, LLMs enable intelligent decision-making and hybrid optimisation approaches by acting as adaptive controllers, assisting in algorithm selection, and even generating novel heuristic variations in real-time. These models can enhance exploration-exploitation trade-offs, mitigate premature convergence, and provide explainability, an often-overlooked aspect of traditional metaheuristics. As a result, LLM-driven optimisation holds immense potential to outperform classical techniques, particularly in dynamic, multi-objective, and constrained environments.

Recent survey studies such as [13–15] have started exploring the intersection of LLMs and metaheuristics, providing foundational insights into their integration. Specifically, Zhao et al. [13] discuss how LLMs can assist in designing new metaheuristics, which intensifies key mechanisms through which they contribute to heuristic search. Meanwhile, the work of Wu et al. [14] offers a broad review of LLM-driven metaheuristic strategies and categorises their applications in different optimisation tasks. These studies indicate that LLMs are becoming a key component in modern metaheuristic research, which further motivates the current systematic review. Despite the rapid advancements in LLM-driven metaheuristic optimisation, the field remains fragmented, with studies exploring different aspects in isolation. Existing research primarily focuses on individual algorithmic improvements or specific applications, which lack a structured, overarching perspective on how LLMs are transforming metaheuristic optimisation. Moreover, researchers and practitioners face challenges in identifying best practices, understanding emerging trends, and recognising limitations that require further investigation.

While these surveys provide useful overviews of how LLMs are entering the field of metaheuristics, they remain high-level in nature. Zhao et al. [13] takes a broader perspective on automated metaheuristic design and includes LLMs as one of several enablers, but does not offer a focused classification of LLM roles. Wu et al. [14] proposes a taxonomy of LLM and evolutionary algorithms' interactions, however, primarily summarises application domains rather than establishing a consistent comparative structure across studies. These studies primarily focus on summarising emerging methods or listing LLM-supported optimisation strategies without applying a consistent classification scheme or comparative framework. In addition to these, [15] provides a comprehensive dual-perspective review that analyses both the use of LLMs to assist optimisation algorithms and the role of optimisation techniques in improving LLM architectures. Their work contributes a broad but less structurally focused overview, primarily categorising integration strategies across domains such as black-box optimisation, reinforcement learning, neural architecture search, and prompt tuning. While this dual analysis offers valuable breadth, it lacks a role-based classification of LLM contributions or a comparative synthesis across studies. This further motivates the need for a structured, role-centric review that systematically categorises LLM contributions.

This systematic literature review addresses a timely gap in the optimisation literature by exploring the integration of LLMs with metaheuristic algorithms, which represent one of the most cutting-edge developments in computational intelligence. Despite the rapid rise of LLMs across AI domains, their role in optimisation remains underexplored and fragmented across diverse studies. This review aims to systematically analyse the diverse and evolving roles of LLMs in metaheuristics, categorising their applications, examining current methodologies, and identifying their impact on optimisation workflows. In doing so, it provides a foundational reference for researchers aiming to leverage LLMs in algorithm design and for practitioners seeking to implement advanced, AI-driven solutions in real-world problem-solving. In this study, we introduce a structured, role-based taxonomy to categorise LLM contributions as Advisors, Refiners, Enhancers, or Innovators based on their position in the optimisation workflow. We also perform a thematic analysis across 25 studies, which enables a more detailed synthesis of how LLMs contribute to each stage of algorithm design, parameter tuning, hybridisation, and problem-solving. This structured and fine-grained approach makes our review both broader in scope and deeper in analysis than previous surveys, which offers a comprehensive foundation for future research in LLM-driven optimisation. The study also outlines critical limitations in this space (e.g. prompt dependency, lack of benchmarks, scalability, etc.) and offers a roadmap for future research. Through this detailed and thematic exploration, the review aims to advance understanding of how LLMs are shaping the future of intelligent optimisation systems.

The remainder of the paper is organised as follows: Section 2 provides preliminaries and essential definitions needed to understand LLMs and evolutionary computation techniques. The methodology of the current systematic literature review is given in Section 3. Section 4 presents the results and critical analysis of the works reviewed in the literature. A thematic analysis is conducted in Section 5 to identify trends and gaps. This section also proposed a role-based taxonomy. Section 6 includes a discussion and promising future directions. Finally, the limitations and conclusions are given in Section 7 and Section 8, respectively.

2. Preliminaries and essential definitions

Optimisation and generative AI form the core of this systematic review. These foundational concepts are essential for understanding the synergy between metaheuristic techniques and LLMs. Therefore, this section provides some of the most fundamental concepts and essential definitions in optimisation and generative AI.

2.1. Optimisation and metaheuristics:

Optimisation is the process of searching a set of candidate solutions for a given optimisation problem and finding the best one(s). Without the loss of generality, an optimisation problem can be formulated as a minimisation problem as follows [17]:

$$\text{minimise } f(x_1, x_2, \dots, x_n) \tag{1}$$

Subject to the following constraints:

$$\begin{aligned} g_m(x_1, x_2, \dots, x_n) &\le b_m \\ h_p(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \tag{2}$$

where x_i is the ith decision variable, f represents the objective function, g_m is the mth inequality constraint, and h_n is the nth equality constraint.

This formulation shows that the role of an objective function is to evaluate a solution that is nothing more than a set of values for decision variables $x = x_1, x_2, \dots, x_n$. It maps the solution space to corresponding values in the objective space. Since f is a function, the normal rules of functions apply, meaning that the function is well-defined, has a domain (the solution space), and produces a unique output (objective value) for every valid input (set of decision variables). Depending on the properties of f, such as continuity, differentiability, or convexity, different optimisation techniques can be applied. For example, if f is differentiable, gradient-based methods can be used, while if f is convex, then efficient methods like convex optimisation techniques can find global optima.

The first type is inequality constraints, which limit the values that the decision variables can take. The second type is equality constraints, which require the decision variables to satisfy certain exact conditions. Both types of constraints must hold for any solution to be acceptable. If they violate any of these constraints, the solution is considered infeasible, which means it cannot be accepted as a valid answer to the problem.

Given the formulation of the objective function, an optimisation algorithm can be defined as follows: The algorithm generates an initial solution, which is a set of values for decision variables as discussed above ($x = x_1, x_2, \dots, x_n$). This solution is then evaluated using the objective function $f(x)$ to measure its quality. The next step is to use x and calculate the constraints as well. If none of the constraints are violated, the solution is called feasible. This means that the algorithm can proceed to generate other solutions using this solution to further minimise the objective function f. This process is continued iteratively using different techniques such as gradient-based, heuristic, or metaheuristic until the end condition is met.

The literature shows that metaheuristics have been very popular lately, mainly due to the fact that they are gradient-free, benefit from higher local optimal avoidance, and are able to find reasonably good solutions for a given optimisation problem in a reasonable time. It should be noted that metaheuristics belong to the class of evolutionary computation techniques under the big umbrella of computational intelligence, as discussed in the introduction. They are considered black-box optimisation techniques that do not require any detailed information about the problem's structure, such as derivatives or specific mathematical properties. Instead, they rely on exploring the solution space by using general rules, like randomness or evolution-inspired processes, to find good solutions. Because of this, metaheuristics can be applied to a wide variety of problems, even when the problem is complex or poorly understood.

In the literature, there has been an incredibly high growing interest in metaheuristics lately due to their wide applications and superior performance compared to other types of optimisation algorithms. However, they suffer from slow convergence and high computation costs, as a population of solutions is typically used during the optimisation process. They are also stochastic, so quality assurance mechanisms are required to ensure the algorithm finds the best possible solutions given the limitations, such as limited time, computational resources, or randomness in the search process. Without these mechanisms, the algorithm may only find suboptimal solutions or take too long to reach a good result.

Despite their merits and being problem-independent, they typically require fine-tuning, improvement, and/or hybridisation when applied to different problems. Their performance is not guaranteed either. The No Free Lunch theorem [18] states that when we consider all optimisation problems, no single algorithm performs better than others across all types of problems. This means that an algorithm that works well for one problem may perform poorly on another. Therefore, choosing or designing the right algorithm for a specific problem is essential, and adjustments may be needed to improve its effectiveness.

The literature has seen several methods to automate this process, such as automatic parameter tuning, hyper-heuristics, and other strategies. Any effort to make this process less human-dependent and more automated is a great step forward. Quite recently, LLMs have been used to achieve this goal, too. Due to their incredible understanding of context and ability to process unstructured data, they have brought huge potential to many fields. Metaheuristics can also leverage them to improve decision-making, adapt parameters dynamically, and even generate new solution strategies in real time. By combining LLMs with metaheuristics, the optimisation process can become more flexible and efficient, leading to better and faster solutions. As discussed in the introduction, this is exactly the focus of this systematic literature review.

2.2. Generative AI and large language models:

Generative AI (GenAI) refers to a branch of artificial intelligence that generates new content (e.g., text, images, videos, 3D models, and multimedia) based on the datasets on which it has been trained [19].

Such applications predominantly use deep learning techniques, but in a different manner than traditional tasks like classification, regression, or clustering. GenAI methods indeed leverage several innovative technologies to generate new and original content.

Due to their flexibility and incredible capability, GenAI approaches and tools have become quite popular for technical and even non-technical users. Platforms such as ChatGPT, MidJourney, and DALL·E have made GenAI more accessible through their user-friendly interfaces and no-code capabilities, enabling users to generate content without technical expertise. Despite these benefits, some have criticised them for lack of transparency in their training data, over-reliance on AI-generated content, and even a decline in human creativity. There are also a lot of ethical concerns, such as bias, that we need to address to use this emerging technology to its potential.

Among all GenAI tools, perhaps LLMs have been the most popular due to their ability to generate coherent, versatile, and context-aware text. LLMs such as GPT have been widely adopted due to their ability to perform a broad range of tasks, including writing essays, summarising articles, and generating computer code in various programming languages. This is how they have been used in metaheuristics as well. LLMs have the potential to increase automation and reduce human intervention in metaheuristics by doing tasks such as understanding the problem, helping with problem formulation, engaging in programming, or developing hybrids. By leveraging LLMs, the optimisation process in metaheuristics can become more efficient and adaptive, which ultimately leads to better performance in complex problem-solving.

This review synthesises key trends, gaps, and emerging themes in the literature on the use of LLMs in metaheuristics, providing a foundation for addressing the research objectives. It highlights the key trends, challenges, and opportunities on how LLMs and metaheuristics can be used together to achieve better optimisation outcomes. The following section outlines the methodology adopted to systematically identify, select, and analyse the relevant studies, ensuring a transparent and rigorous approach to the review.

3. Methodology

To investigate the contributions of LLMs within the field of metaheuristics, we performed a systematic literature review, following the methodical approach outlined by Kitchenham [20] method and PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) guideline [21] for identifying, evaluating, and interpreting prior research. A systematic literature review offers a structured framework to analyse previous studies in a specific domain, ensuring a comprehensive examination of the available literature [20]. In addition to prior experimental studies, survey papers such as [13–15] provide valuable context for understanding how LLMs interact with metaheuristics. These surveys offer initial classifications of LLM-assisted heuristic design and problem-solving, complementing our systematic review by highlighting the growing significance of LLMs in optimisation. By incorporating insights from these prior works, our systematic review further refines the categorisation and provides a more structured thematic analysis of LLM roles in metaheuristics. In this section, we outline the key steps of the research methodology, including the processes of selecting and excluding relevant studies, along with the methods employed for data extraction and analysis during the review. The systematic literature review was conducted using Covidence, a web-based platform designed to streamline the screening, selection, and data extraction processes for systematic reviews.

3.1. Research question

This study seeks to see how the following question has been addressed in the literature:

How can large language models be effectively leveraged to design, optimise, and enhance metaheuristics?

This research question is crucial for understanding how LLMs, originally developed for tasks in natural language processing, are adapted and integrated into the domain of metaheuristics. Metaheuristics, known for solving complex optimisation problems, are increasingly enhanced by the capabilities of LLMs. By exploring this integration, the review aims to identify the strategies employed, such as using LLMs to generate heuristics, guide search spaces, or improve decision-making processes within metaheuristic frameworks. The goal is to systematically categorise and describe the techniques used in merging LLMs with metaheuristics, highlighting their impact on problem-solving performance and efficiency and providing insights into the evolution of metaheuristic problem-solving with LLM support.

3.2. Search strategy

The search strategy for this systematic review was carefully designed to comprehensively identify all relevant studies related to the research topic. A combination of scientific databases was used, and advanced searches were conducted using predefined keywords related to the core concepts of the review. These search terms were applied to capture a broad range of potentially relevant studies. The following subsections provide further details regarding the databases and sources, search terms, time period and language restrictions, and the inclusion of grey literature.

3.2.1. Databases and sources

To ensure a comprehensive review of relevant literature, a wide range of scientific databases was utilised for the search. The primary databases included Scopus, ACM Digital Library, Web of Science (via Thomson Reuters), Wiley, Taylor & Francis, IEEE Xplore, ScienceDirect, PubMed, and arXiv.org.

These databases were selected as they cover a broad spectrum of peer-reviewed journals, conference proceedings, and preprints across various fields, including computer science, engineering, and healthcare. Scopus, Web of Science, and ACM Digital Library are widely recognised for their extensive coverage of peer-reviewed publications in technology and metaheuristic applications. IEEE Xplore and ScienceDirect provide access to cutting-edge research in computing and optimisation techniques, while PubMed offers relevant studies in healthcare, particularly for cross-disciplinary applications. Additionally, arXiv.org was included to capture preprints and early-stage research, ensuring that the latest advancements in LLMs and metaheuristics are considered.

3.2.2. Search terms

The advanced search services provided by each of the scientific databases were utilised to perform search operations. Depending on the capabilities of each database's search engine, searches were conducted across titles, abstracts, keywords, and the full text of articles. The search was performed using the following set of keywords and search terms:

("large language model" OR "large language models" OR "LLM" OR "LLMs") AND ("metaheuristic" OR "meta-heuristic" OR "metaheuristics" OR "meta-heuristics" OR "Optimisation algorithm" OR "Optimization algorithm")

3.2.3. Time period and language restrictions

The search was conducted in November 2024, with no publication date restrictions applied, allowing the inclusion of studies regardless of their publication year. This approach ensured that the search captured all relevant research at the intersection of LLMs and metaheuristics, including both recent advancements and earlier foundational work in the field. In terms of language, only studies published in English were included to maintain consistency in analysis and to avoid potential issues related to translation accuracy and interpretation.

3.2.4. Grey literature

All included studies were sourced exclusively from peer-reviewed journals and conferences, except for arXiv.org, which was the only grey literature source considered. arXiv.org was included to capture preprints related to LLMs and their integration into metaheuristics or optimisation techniques. Other forms of grey literature, such as technical reports, theses, working papers, and white papers, were excluded to ensure the review focused on high-quality, peer-reviewed publications. By selectively incorporating arXiv.org, the review maintained rigour while ensuring that emerging and cutting-edge research was not overlooked.

3.3. Study selection criteria

The study selection process followed predefined inclusion and exclusion criteria to ensure the relevance and quality of the studies considered for this review.

Inclusion criteria

Relevance to research topic: Studies that directly addressed the integration of LLMs within metaheuristics were included, and research that explored the use of LLMs to enhance, optimise, or complement metaheuristic techniques was prioritised.

Publication type: Only peer-reviewed journal articles, conference papers, and preprints from arXiv.org were considered. These sources ensure a high standard of academic rigour and up-to-date contributions in the field.

Content: Studies that provided sufficient detail on the methodology, application, or evaluation of LLMs within metaheuristic contexts – including practical or theoretical insights – were considered.

Exclusion criteria

Irrelevant focus: Studies that did not directly discuss LLMs and metaheuristics or that were tangentially related to the topic were excluded. For instance, studies involving the use of metaheuristics to enhance the performance of LLMs were not considered.

Non-peer-reviewed sources: Grey literature sources other than arXiv.org, such as technical reports, working papers, theses, and white papers, were excluded to maintain a focus on validated, peer-reviewed content.

Duplicate studies: Any duplicate studies identified across multiple databases were removed during the screening process, with duplicates automatically and manually removed within the Covidence platform.

Language: Studies published in languages other than English were excluded, ensuring consistency in analysis and preventing potential issues with translation or interpretation.

3.4. Study selection process

The study selection process for this systematic literature review was conducted in five stages. First, key scientific databases were identified as the primary sources for retrieving relevant studies. In the second stage, a list of keywords and search terms was applied to the databases, utilising advanced search functions and filters to refine the results to the most relevant studies. As a result, a preliminary list of publications was identified. Following this, in stage three, all search results were imported into the Covidence platform to facilitate the systematic review process. In the fourth stage, titles and abstracts of the retrieved papers were screened by two reviewers within Covidence to exclude studies that did not align with the research topic. The fifth stage involved full-text screening of the remaining studies to ensure their relevance. At this stage, all remaining studies were relevant and included in the literature review, and they were ready for data extraction. In instances where discrepancies arose between the reviewers' decisions on stages 4 and 5, a final resolution was achieved through a consensus discussion held collaboratively between both reviewers to determine the final selection. Duplicate studies were identified and removed by the platform when importing; however, in certain instances, duplicates were also manually detected and eliminated to ensure accuracy. Fig. 4 provides an overview of the paper selection process across these five stages, and Table 1 presents the primary search results for the nine scientific databases, separately searching within "title", and "title/abstract/keyword" and "full text".

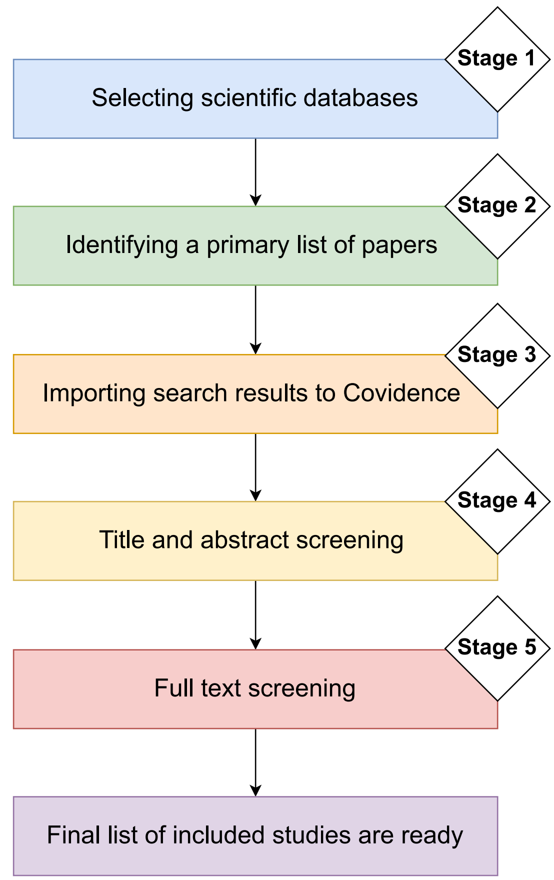

Fig. 4. Stages in the study selection procedure.

The initial search based on titles yielded a limited number of relevant studies, while searches using full text resulted in an overwhelming number of irrelevant papers. To refine the selection process, we decided to rely on search results based on a combination of title, abstract, and keywords. This approach provided a more balanced and efficient method for identifying studies directly related to the topic, ensuring the inclusion of relevant and high-quality research. The initial search based on title, abstract, and keywords identified a total of 215 publications. After the removal of duplicates and applying the exclusion criteria, the final number of studies included in the systematic review was 25.

3.5. PRISMA flow diagram

The study selection process is detailed in the PRISMA flow diagram (Fig. 5), which tracks the identification, screening, and inclusion of studies. Initially, 215 publications were identified based on title, abstract, and keyword search in the nine databases. A total of 68 duplicate studies were removed, with one duplicate identified manually and 67 identified by Covidence, resulting in 147 studies being screened in Stage 4 for title and abstract screening. Of these, 111 studies were found to be irrelevant. In Stage 5 (full-text review), 36 full-text studies

Table 1
Summary of initial search results for each database.

Database	Search in	Number of publications found
Scopus	Title	3
	Title/Abstract/keyword	81
	All fields	1,161
ACM Digital Library	Title	2
	Abstract	10
	Full Text	432
Web of Science (via Thomson Reuters)	Title	2
	Abstract	14
	Topic	15
Wiley	Title	0
	Abstract	2
	Anywhere	72
Taylor & Francis	Title	0
	Abstract	1
	Anywhere	31
IEEE Xplore	Title	0
	Abstract	14
	Full Text Only	566
ScienceDirect	Title	0
	Title/Abstract	3
	Full Text	486
PubMED	Title	0
	Title/Abstract	1
	All fields	1
arXiv.org	Title	5
	Title/Abstract	89
	Full Text	103

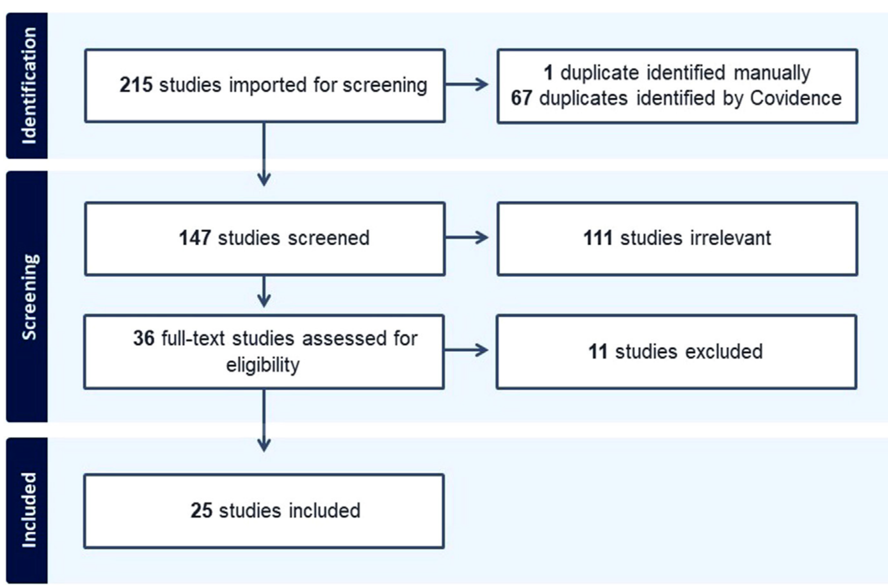

Fig. 5. PRISMA workflow diagram - Generated by Covidence.

were assessed for eligibility, with 11 studies excluded based on the exclusion criteria. Ultimately, 25 studies were included in the final systematic review. This flow diagram provides a transparent overview of the entire selection process, ensuring clarity on how the final set of studies was determined.

3.6. Data extraction

The data extraction process for this systematic review was carried out with a structured approach, utilising a predefined set of criteria to ensure comprehensive and accurate extraction from each of the 25 included studies. The extracted data were organised into the following categories:

- **Bibliographic Information:** Title, country of publication, year of publication, and type of article (journal, conference paper, book chapter, etc.).
- **Study Objectives and Scope:** Study type (e.g., empirical, theoretical, survey), research objective or question, and the specific application area.
- **Methodology:** Details of the metaheuristics and LLMs used, the method of integration between LLMs and metaheuristics, the

type of optimisation problem (e.g., continuous, binary, multi-objective), test functions or datasets used, and the programming language employed.
- **Results and Analysis:** Key performance metrics, statistical tests applied, main findings, and conclusions.
- **Limitations and Criticisms:** Study limitations, suggestions for future work, LLM prompts used (including type and wording if available), and the population or sample size involved in the study.

To streamline the data extraction process, a popular LLM (GPT-4.o) was employed in a human-in-the-loop approach, where the LLM assisted in extracting data, but all extractions were overseen and refined by two human reviewers. The two reviewers performed the extraction independently, and any discrepancies were discussed and resolved through consensus, ensuring accuracy and consistency in the final data set. This approach enabled an efficient and reliable extraction process while ensuring that human oversight remained central to preserving data integrity.

3.7. Quality assessment

A quality assessment was performed on all 25 studies included in the review to evaluate their credibility, relevance, and methodological rigour. The evaluation focused on key aspects such as the clarity and robustness of the study design, particularly in terms of the integration of LLMs and metaheuristics. Reproducibility was also assessed, examining whether sufficient methodological detail was provided to allow for replication. The validity of the data, including the use of appropriate datasets, performance metrics, and statistical analysis, was carefully scrutinised to ensure reliable conclusions. Additionally, potential biases and study limitations were considered, reviewing for any incomplete data reporting or conflicts of interest. The assessment was conducted independently by two reviewers, and any discrepancies were resolved through consensus, ensuring a high standard of quality across the selected studies.

3.8. Data synthesis

The synthesis of data from the 25 selected studies was performed through a combination of qualitative and quantitative approaches. The key findings from each study were systematically extracted, organised, and compared to identify common themes, trends, and variations in how LLMs were integrated into metaheuristics. For qualitative synthesis, the primary focus was on understanding the methods of integration, application areas, and challenges faced in using LLMs with metaheuristics. Studies were grouped based on their methodological approaches, such as the types of metaheuristics employed, optimisation problems addressed, and the nature of the LLMs used. This thematic analysis helped highlight recurring techniques and innovative strategies, as well as the limitations and gaps identified by the authors.

In terms of quantitative synthesis, where performance metrics were comparable, results from studies using similar metaheuristic algorithms or LLM techniques were aggregated to provide a clearer understanding of their effectiveness. This helped identify any patterns in performance improvements or shortcomings when LLMs were applied to metaheuristic tasks. Through this synthesis, the review was able to draw general conclusions on the effectiveness, applicability, and future directions for integrating LLMs in metaheuristics, providing a foundation for further research and practical applications in this area.

3.9. Threats to validity

This systematic review was conducted with rigorous methods, but several potential threats to validity should be acknowledged. First, selection bias may have occurred due to the restriction of studies to those published in English, potentially excluding relevant research in other languages. Second, limiting grey literature to arXiv.org may have overlooked emerging studies available in other non-peer-reviewed outlets. Third, publication bias is a concern, as studies reporting positive outcomes are more likely to be published, possibly skewing the overall findings towards more favourable results regarding the integration of LLMs with metaheuristics. Fourth, the data extraction and synthesis process, though assisted by human reviewers and GPT-4.o, may have introduced subjective interpretations or inconsistencies in categorising the studies. A consensus process was employed to mitigate this, but potential biases could still influence the findings. Fifth, heterogeneity in methodologies across the included studies, such as variations in types of metaheuristics, LLMs, and evaluation metrics, could limit the comparability and generalisability of the findings, affecting the strength of conclusions drawn from the synthesis. Finally, the search was conducted across nine major scientific databases. While we believe these databases are the most relevant to the topic, covering a wide range of journals and conferences, it is possible that some studies may have been missed if they were not indexed within these databases. However, given the comprehensive scope of these databases, we are confident that they provide substantial coverage of the relevant literature. Acknowledging these potential threats to validity provides context for interpreting the findings and ensures that the conclusions are approached with appropriate caution.

4. Results and critical analysis of the existing works

This section presents the findings of the systematic review on the integration of LLMs within metaheuristics. The results are organised into several key areas: the demographic characteristics of the selected studies, metaheuristics used in studies, LLMs used in the studies, programming languages used for the implementation of algorithms, application areas of the studied metaheuristics, the thematic analysis of the integrations, and the extracted themes. Additionally, patterns and gaps in the current body of literature are highlighted, providing insights for future exploration in this rapidly evolving field.

The demographic analysis includes an overview of the publication trends and types, study types, and geographic distribution of the research. The subsequent sections will delve into the specific methods employed by the researchers, the types of optimisation problems addressed, and the overall effectiveness of the approaches.

4.1. Demographics

- **Year of Publication:** The temporal distribution of publications indicates a sharp rise in research activity in 2024, with 88% of the studies (22 out of 25) published in this year, compared to 12% (3 out of 25) in 2023. This surge reflects the rapidly growing interest in the application of LLMs in metaheuristics, likely driven by recent advances in natural language processing and the increasing recognition of LLMs' potential to enhance optimisation techniques. The concentration of studies in recent years highlights the field's dynamic nature, with researchers actively exploring new intersections between these technologies. This trend is depicted in Fig. 6, which shows the number of publications by year.
- **Country of Origin:** The geographic distribution of the studies reveals that the majority of contributions originated from China (32%), followed by countries such as Japan (24%) and the Czech Republic (16%). This distribution demonstrates a strong interest

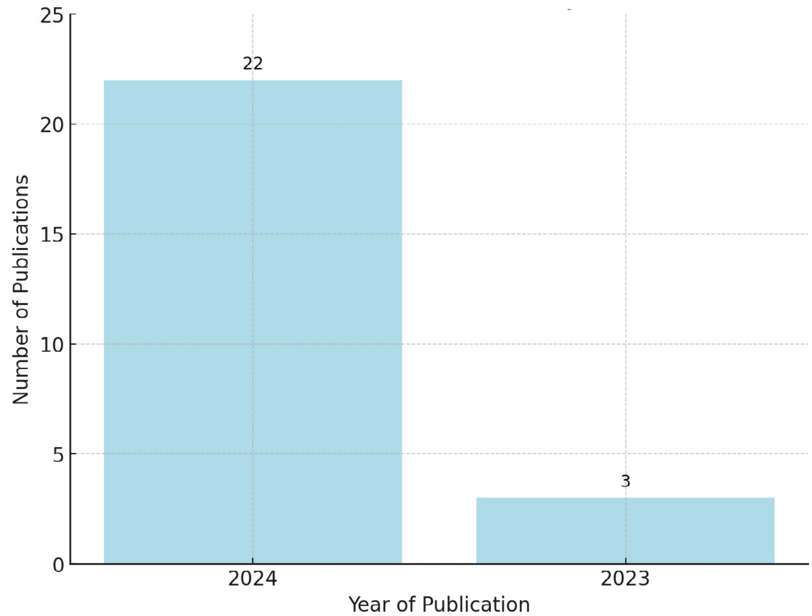

Fig. 6. Number Of Publications By Year.

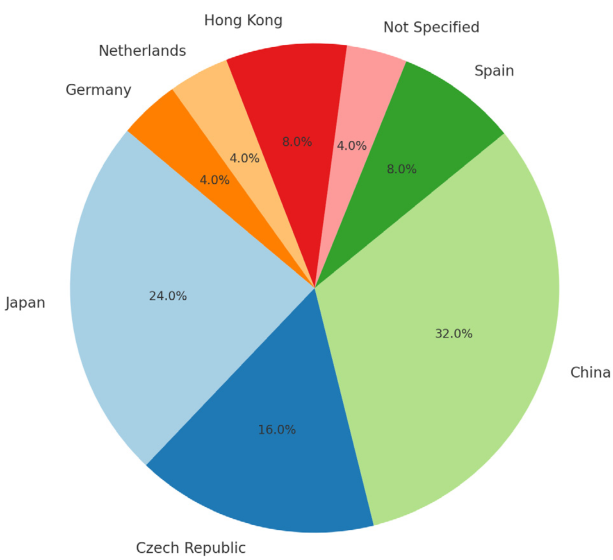

Fig. 7. Distribution Of Studies By Country.

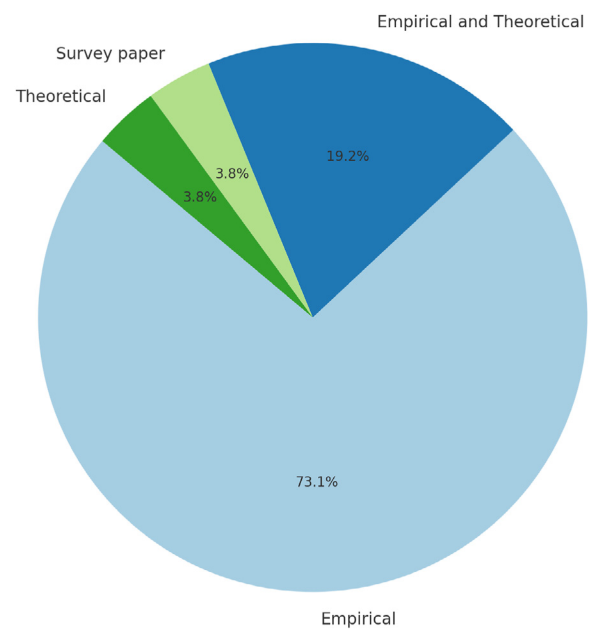

Fig. 8. Distribution of Study Types in LLM-Metaheuristics Research.

in Asia and Europe, where research institutions are heavily investing in AI and computational optimisation. The prominence of China, in particular, aligns with the country's broader push towards leadership in AI research. The presence of research from various regions underscores the global relevance of integrating LLMs with metaheuristics, although some studies had unspecified country origins, reflecting a possible gap in the reported metadata. The breakdown of studies by country is illustrated in Fig. 7 and Table 2, demonstrating the global distribution of research efforts.

- **Study Type:** The dominance of empirical studies suggests that much of the current research focuses on the practical application and experimental validation of LLMs in metaheuristics. Empirical studies are critical in this field, as they provide concrete evidence of how LLMs can enhance optimisation processes. The scarcity of theoretical or survey-based studies may indicate a need for further foundational research to develop a more comprehensive theoretical framework or for broader reviews of the existing literature to identify research gaps. This emphasis on empirical research is reflected in Fig. 8, which illustrates the proportion of study types.

Table 2
Country of origin of research studies.

Country	Number of studies	Percentage (%)
China	8	32.0
Japan	6	24.0
Czech Republic	4	16.0
Spain	2	8.0
Hong Kong	2	8.0
Netherlands	1	4.0
Germany	1	4.0
Not Specified	1	4.0

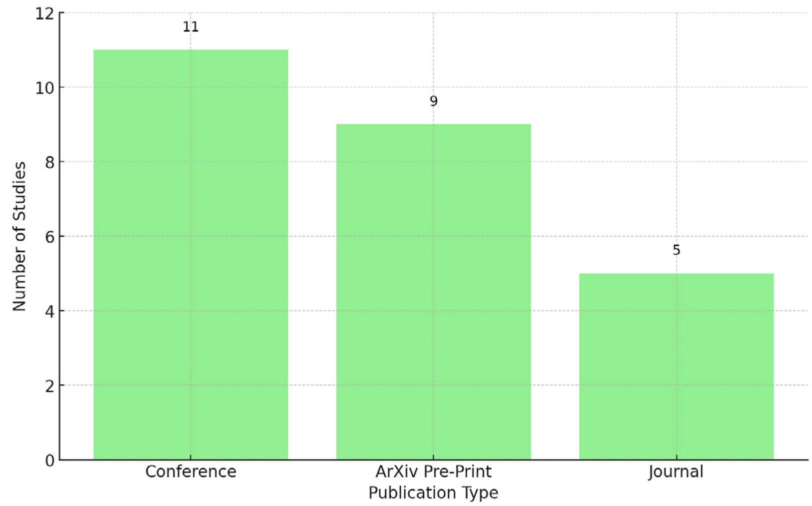

Fig. 9. Distribution Of Studies By Publication Type.

Table 3
Metaheuristic categories across reviewed studies.

Category	Number of studies
Evolutionary Algorithms	9
Swarm Intelligence Algorithms	15
Bayesian and Statistical Techniques	2
Hybrid and Reflective Algorithms	11
Derivative-Free and Custom Algorithms	4

- **Publication Type:** The types of publications highlight the diverse platforms through which findings are disseminated. A substantial portion of the studies appeared in conference proceedings and journal articles, which are traditional outlets for peer-reviewed research. Notably, preprints from arXiv.org accounted for 36.0% of the total publications. This high percentage of preprints points to the fast-evolving nature of this research area, where researchers seek to quickly share their findings with the community before formal peer review. The balance between peer-reviewed and preprint publications reflects both the rigour and immediacy with which this field is advancing. This distribution is represented in Fig. 9, which provides an overview of the publication types across the studies.

4.2. Metaheuristics used in studies

The analysis of metaheuristic algorithms across the 25 reviewed studies reveals a consistent reliance on five core categories: swarm intelligence algorithms (15 studies), hybrid and reflective algorithms (11), evolutionary algorithms (9), derivative-free and custom algorithms (4), and Bayesian and statistical techniques (2). These categories reflect the varied strategies adopted to address complex optimisation problems, with LLMs integrated at different levels to enhance adaptability, automate decision-making, or generate novel heuristics. Swarm intelligence algorithms emerged as the most frequently applied, likely due to their population-based design and ease of modular augmentation via LLM prompts. Hybrid and reflective approaches were also prominent, indicating a growing shift towards embedding LLMs directly into the optimisation loop, as adaptive agents, heuristic generators, or co-optimisers. Evolutionary algorithms also featured strongly, capitalising on their population-based search processes and the flexibility of LLMs to dynamically generate or adapt evolutionary operators such as crossover and mutation. Although fewer in number, the studies employing derivative-free and Bayesian techniques demonstrate innovative uses of LLMs in surrogate modelling, hyperparameter tuning, and black-box optimisation. This classification, detailed in Tables 3 and 4, highlights the breadth of LLM-metaheuristic integration and signals a clear movement towards more autonomous and flexible LLM-guided optimisation frameworks. Each category is examined in detail in the following subsections.

Evolutionary algorithms

Evolutionary algorithms represent one of the most widely adopted metaheuristic categories in the reviewed studies because of their flexibility, scalability, and effectiveness in solving complex, high-dimensional optimisation problems. These algorithms are inspired by natural evolutionary processes such as selection, crossover, and mutation and are particularly well suited for problems where the objective function is non-differentiable or lacks mathematical structure. Within the 25 studies analysed, several evolutionary variants were utilised, including the classical Genetic Algorithm, Differential Evolution, and more advanced forms such as the Covariance Matrix Adaptation Evolution Strategy, Self-Adaptive Evolution Strategies, and Biased Random Key Genetic Algorithm [46]. In addition, a number of studies developed custom evolutionary algorithms, often leveraging LLMs to define new operators, heuristics, or decision rules. Across these variants, LLMs were integrated to assist with tasks such as operator design, mutation strategy refinement, dynamic parameter tuning, and even solution interpretation. This integration introduces a new level of autonomy and responsiveness, enabling evolutionary algorithms to adapt on the fly and operate with reduced manual intervention.

The use of genetic algorithms and differential evolution across the reviewed studies highlights their continued relevance as foundational tools in optimisation research, particularly due to their modular structures and ease of adaptation. Differential evolution was especially prominent in studies aiming to improve search efficiency, where LLMs were employed to enhance mutation strategies, introduce adaptive switching mechanisms, and refine selection processes in real time. More sophisticated methods, such as the covariance matrix adaptation evolution strategy and self-adaptive evolution strategies, were used in contexts that required more precise control over convergence behaviour and population dynamics. In these cases, LLMs contributed by monitoring algorithmic progress and dynamically adjusting internal parameters to avoid stagnation or premature convergence. The biased random key genetic algorithm, which uses probabilistic representations to encode solutions, was applied in combinatorial optimisation scenarios, where LLMs assisted in evolving selection pressures and improving solution diversity. Notably, several studies employed custom evolutionary algorithms, often developed through iterative prompting and refinement guided by LLMs. These customised designs reflect a growing trend

Table 4
Overview of metaheuristics employed across studies and their application methods.

Category	Algorithm	Reference(s)
Evolutionary Algorithms	Genetic Algorithm	[22]
	Differential Evolution	[23–26]
	Covariance Matrix Adaptation Evolution Strategy	[26]
	Self-Adaptive Evolution Strategies	[27]
	Biased Random Key Genetic Algorithm	[28]
	Custom Evolutionary Algorithm	[29]
Swarm Intelligence Algorithms	Particle Swarm Optimisation	[30]
	Whale Optimisation Algorithm	[30–32]
	Grey Wolf Optimisation	[30,31,33]
	Self-Organising Migrating Algorithm	[30,31,34]
	Artificial Bee Colony	[30,31]
	Cuckoo Search	[30]
	Ant Colony Optimisation	[22,35]
Bayesian and Statistical Techniques	Hyper-Parameter Optimisation	[36]
	Multi-Objective Bayesian Optimisation	[37]
Hybrid and Reflective Algorithms	Reflective Evolution	[22,35]
	LLM-Generated and Adaptive Operators	[38]
	LLM-Integrated Search Guidance	[28]
	LLM-Reflective Optimisation	[39–43]
	LLM-Assisted Algorithm Analysis and Tuning	[44]
	LLM-Assisted Metaheuristic Design	[45]
Derivative-Free and Custom Algorithms	Zoological Search Optimisation	[31]
	Hill Climbing	[29]
	Iteratively generated custom algorithm from SOMA	[34]
	LLM-generated modular algorithms	[26]

towards LLM-guided algorithm construction, where the model actively participates in generating or modifying evolutionary strategies. While some of these approaches incorporate reflective elements, they are formally categorised under hybrid frameworks, as detailed in Table 4. Collectively, these applications demonstrate the versatility of evolutionary algorithms and highlight how their integration with LLMs facilitates greater adaptability, autonomy, and problem-specific tailoring.

Ye et al. introduced ReEvo, a reflective evolutionary framework that builds on genetic algorithm principles, using LLMs as both heuristic generators and evaluators [22]. Individuals, represented as code snippets, are evolved through LLM-guided selection, crossover, and mutation, enhanced by short- and long-term verbal reflections. ReEvo consistently outperforms expert-designed and neural heuristics across a range of combinatorial optimisation problems, exhibiting strong adaptability and high sample efficiency.

Pluhacek et al. explored the enhancement of differential evolution by leveraging GPT-4 to generate a novel mutation strategy named DE/dynamic-switch/1/bin. The LLM was used to propose, code, and benchmark the new strategy, which outperformed the classical DE/rand/1/bin on numerous benchmark functions. As a representative of the differential evolution subcategory, this study highlights how LLMs can serve as design partners in the creation of more adaptive and effective evolutionary operators [23]. Another study evaluated the use of LLMs to iteratively refine the differential evolution algorithm through repeated prompting. Starting from a standard differential evolution variant, several LLM-generated modifications outperformed the baseline on benchmark functions, highlighting the model's potential as an autonomous refinement tool in evolutionary optimisation [24]. Zhong et al. proposed a parameter adaptation scheme for differential evolution that integrates an LLM (Gemini) into the optimisation process to dynamically generate control parameters during each generation. Through structured prompt engineering, the model adjusts the scaling factor and crossover rate based on real-time optimisation feedback, leading to improved convergence compared to fixed and random parameter settings. This study exemplifies the use of LLMs to enhance the adaptability and automation of classical differential evolution frameworks [25]. Van Stein and Bäck introduced LLaMEA, an evolutionary framework that employs an LLM to automatically generate novel metaheuristic algorithms through iterative refinement. While some of the evolved algorithms exhibit structural similarity to differential evolution, they are not restricted to its traditional operators. As a representative of the Differential Evolution and Covariance Matrix Adaptation Evolution Strategy (CMA-ES) subcategories, this work demonstrates the potential of LLMs to autonomously design effective and structurally original optimisation strategies [26]. Although LLaMEA does not use CMA-ES directly, it includes CMA-ES and its advanced variants as benchmarking baselines to evaluate the performance of the LLM-generated algorithms.

Baumann and Kramer combined a self-adaptive evolution strategy with an LLM to generate post-hoc explanations of the optimisation process. The approach adjusts mutation step sizes dynamically and uses LLMs to summarise convergence behaviour, fitting within the self-adaptive evolution strategies subcategory [27]. Another study integrates an LLM into the Biased Random Key Genetic Algorithm, using it to generate probabilistic guidance for decoding random keys into candidate solutions. The LLM produces parameterised weights based on graph metrics, which are used to calculate probability scores that influence the construction of solutions, effectively shaping the search space in a problem-aware manner. This work demonstrates how LLMs can enhance evolutionary search by guiding solution interpretation and improving search efficiency [28].

Lange et al. proposed EvoLLM, an LLM-guided optimisation framework that emulates the behaviour of evolution strategies through prompt-based recombination and search distribution updates. The LLM generates new candidate means based on fitness-ranked solution histories, enabling zero-shot black-box optimisation without traditional gradient information or parameter self-adaptation. While structurally aligned with evolution strategies, the absence of internal feedback-driven control places this approach within the Custom Evolutionary Algorithms subcategory [29].

Swarm intelligence algorithms

Swarm intelligence algorithms constitute a core category within the reviewed studies, underscoring their proven effectiveness in complex, nonlinear, and high-dimensional optimisation problems. Rooted in the collective behaviours of biological systems, such as flocking, hunting, migration, and foraging, these algorithms offer a robust framework for exploration and global search. Their population-based and decentralised nature makes them particularly amenable to integration with LLMs, which were leveraged across multiple studies to enhance decision-making, adapt control parameters, and design novel hybrid operators. Prominent swarm-based methods featured in the current literature review include particle swarm optimisation, whale optimisation algorithm, grey wolf optimisation, and artificial bee colony, each of which was extended or guided by LLMs to improve convergence and adaptability. Other studies employed self-organising migrating algorithm, cuckoo search, and ant colony optimisation as flexible templates for LLM-driven customisation and algorithmic innovation. Together, these examples demonstrate the strong compatibility between swarm-based metaheuristics and LLM-guided optimisation. The following section examines these studies in detail.

Pluhacek et al. employed an LLM to generate hybrid metaheuristic algorithms by combining structural elements from swarm-based approaches, including particle swarm optimisation. The resulting algorithms retained PSO-inspired mechanisms such as velocity-influenced updates while incorporating novel components proposed through prompt-driven generation [30]. They also used the LLM to generate hybrid metaheuristic algorithms by combining components from several well-known swarm intelligence methods, including the whale optimisation algorithm, grey wolf optimisation, self-organising migrating algorithm, cuckoo search, and artificial bee colony.

Zhong et al. developed the zoological search optimisation algorithm using ChatGPT-3.5 and the CRISPE prompt framework, drawing inspiration from animal behaviours such as prey-predator interaction and social flocking. While not explicitly based on existing swarm intelligence algorithms, the algorithm's collective and decentralised structure aligns it conceptually with approaches like grey wolf optimisation and self-organising migrating algorithm (a population-based metaheuristic inspired by cooperative-competitive strategies, where individuals migrate towards a leader to explore the search space) [31]. Chen et al. developed a hybrid optimisation framework that combines an LLM with deep reinforcement learning and the whale optimisation algorithm to coordinate power dispatching for electric vehicle charging networks. In this approach, the whale optimisation algorithm is employed to refine candidate strategies generated by LLM-guided policy models [32]. Zhao et al. developed a time series prediction framework that combines an LLM with the UniTS architecture, using the grey wolf optimisation algorithm to optimise key hyperparameters such as learning rate and number of iterations [33]. Pluhacek et al. investigated the use of an LLM to iteratively evolve the self-organising migrating algorithm, starting from its all-to-all variant and introducing prompt-driven structural modifications across multiple iterations. Although the final algorithm incorporated features beyond standard SOMA, the study is directly grounded in its structure and is classified under the self-organising migrating algorithm subcategory within swarm intelligence algorithms [34]. Two studies employed LLMs to enhance ant colony optimisation through reflective or hyper-heuristic approaches. In one, Ye et al. used the ReEvo framework to evolve heuristic functions within ant colony optimisation, significantly improving performance on combinatorial problems such as the travelling salesman and vehicle routing problems [35]. In a related study, the same authors extended this approach by positioning LLMs as general-purpose hyper-heuristics, capable of generating and refining ant colony optimisation heuristics alongside those for other metaheuristics [22].

Bayesian and statistical optimisation techniques

Bayesian and statistical techniques represent a smaller but conceptually significant category within the reviewed studies, offering probabilistic frameworks for optimisation that contrast with the heuristic and stochastic dynamics of evolutionary and swarm-based methods. These approaches rely on statistical inference, uncertainty modelling, and probabilistic sampling to guide the search process, often enabling more efficient exploration in high-cost or black-box environments. In the context of LLM integration, Bayesian methods have been employed for tasks such as multi-objective configuration, surrogate modelling, and ensemble tuning, typically when optimisation involves balancing competing performance criteria or operating under data scarcity. Although fewer in number, the studies in this category highlight the potential for LLMs to collaborate with statistically principled optimisers in designing adaptive, sample-efficient search strategies.

Van Stein et al. introduced a hybrid framework that combines an LLM with Bayesian hyper-parameter optimisation via SMAC3 to automate the design and tuning of metaheuristic algorithms. The framework delegates algorithm generation to the LLM and uses probabilistic modelling to explore optimal parameter settings, improving both efficiency and modularity [36]. Li et al. introduced a framework for merging LLMs by applying multi-objective Bayesian optimisation to search for optimal parameter configurations across several task-specific and sparsity objectives. Their method, MM-MO, builds upon qEHVI and incorporates techniques such as Fisher information and weak-to-strong adaptation to efficiently guide the search [37]. These two studies are classified under the Bayesian and statistical techniques category.

Hybrid and reflective algorithms

Hybrid and reflective algorithms represent a rapidly emerging category within the reviewed literature, characterised by the integration of LLMs as adaptive agents within the metaheuristic process itself. Unlike traditional approaches that use LLMs as external aids for parameter tuning or operator design, these studies embed the LLM directly into the optimisation loop, often enabling dynamic modification of algorithmic components based on feedback or internal reasoning. Reflective algorithms, in particular, leverage LLMs to interpret the performance of candidate solutions and guide future search directions through mechanisms such as verbal feedback, self-assessment, or strategic recomposition. Hybridisation, meanwhile, typically involves combining metaheuristic principles with LLM-generated logic or data-driven components to form novel composite frameworks. This category highlights the unique capacity of LLMs to go beyond support roles and act as co-optimisers or heuristic architects. The studies in this section exemplify this shift and are discussed in detail below.

Ye et al. introduced a framework to demonstrate how LLMs can act as both heuristic generators and reflective agents within an evolutionary process. In their first study, they proposed the concept of language hyper-heuristics, where LLMs generate problem-specific heuristics within an open-ended search space [35]. This was further developed into a reflective evolutionary framework, incorporating short-term and long-term verbal feedback from the LLM to guide heuristic refinement [22]. These approaches exemplify the use of LLMs as embedded evolutionary agents.

Cai et al. proposed a conceptual framework for advancing evolutionary computation by integrating LLMs into the design of variation operators. They differentiate between operators that are guided by LLMs, adapting traditional mechanisms like crossover and mutation based on context, and those that are fully generated by the model through prompt-driven synthesis [38]. Chacón Sartori et al. proposed a hybrid framework that integrates LLMs into a biased random key genetic algorithm by using LLM-generated metrics to guide node selection probabilities [28]. This approach exemplifies LLM-driven operator design, where the model's output is used to dynamically bias the search behaviour of the metaheuristic.

Huang et al. proposed a multimodal optimisation framework in which an LLM processes both textual and visual prompts to extract heuristics from solved instances and iteratively generate and refine solutions for the capacitated vehicle routing problem. The LLM is embedded within the optimisation loop, guiding search decisions based on multimodal input and performance feedback [39]. Tianrungroj and Iba proposed ROIL, a framework that integrates an LLM into the optimisation loop to iteratively refine natural language rule sets for imitation learning. The LLM guides policy construction by modifying rules based on performance feedback from agent-environment interactions [40]. Du et al. proposed an equation discovery framework in which an

LLM iteratively generates and refines symbolic expressions for nonlinear dynamical systems through prompt-guided self-improvement and structural recombination. The LLM operates within the optimisation loop, influencing the search trajectory based on performance feedback and guided mutation, placing this study under hybrid and reflective algorithms [41]. Lin et al. proposed PE-GPT, a hybrid framework that integrates an LLM with metaheuristic algorithms and simulation tools to support power electronics design tasks such as circuit configuration and modulation optimisation. The LLM operates within an iterative search loop, using retrieval-augmented prompts and design feedback to guide the generation and refinement of solutions [42]. Zhong et al. proposed an LLM-assisted framework for adversarial robustness neural architecture search, employing structured prompts to guide Gemini in generating and refining candidate architectures based on iterative performance feedback. The LLM operates within the optimisation loop, contributing directly to architecture evolution without explicit self-reflection [43].

Chacón Sartori et al. proposed a framework that integrates LLMs into the STNWeb visual analytics system to analyse the behaviour of optimisation algorithms, generate descriptive summaries, and recommend parameter adjustments based on performance trends. The LLM operates within the analytical loop, assisting in interpretation and tuning without directly generating solutions [44]. Liu et al. proposed AEL, a framework in which an LLM is embedded into an evolutionary process to generate, crossover, and mutate optimisation algorithms themselves rather than problem solutions. Candidate algorithms are evaluated on a set of benchmark functions, and the LLM is iteratively prompted to refine or recombine them based on performance outcomes [45]. This framework exemplifies meta-level optimisation and is also categorised under hybrid and reflective algorithms.

Derivative-free and custom optimisation algorithms

Derivative-free and custom optimisation algorithms constitute a diverse and adaptive category within the reviewed literature, encompassing methods that operate without gradient information and those purpose-built using unconventional or novel heuristics. These algorithms are particularly useful in scenarios where objective functions are non-differentiable, noisy, or evaluated via black-box procedures. In the context of LLM integration, several studies introduced entirely new optimisation strategies generated or co-developed by LLMs through prompt-driven synthesis. Others applied LLMs to refine or extend existing heuristic mechanisms into non-standard forms not aligned with traditional metaheuristic families. This category captures both the versatility of LLMs in custom algorithm design and their suitability for derivative-free optimisation contexts. The relevant studies are examined in detail below.

Zhong et al. introduced the zoological search optimisation algorithm, a novel metaheuristic developed through the CRISPE prompt framework using ChatGPT-3.5. Designed to be structurally distinct from established methods such as genetic algorithms and differential evolution, zoological search optimisation incorporates custom operators inspired by animal behaviour, including prey-predator interaction and social flocking [31]. Lange et al. introduced EvoLLM, a prompting framework in which an LLM serves as a recombination operator within an evolution strategy to guide search updates based on fitness-sorted candidate populations. The framework generalises beyond traditional strategies like hill climbing by enabling the LLM to propose new search distributions through in-context learning [29]. Pluhacek et al. explored the use of GPT-4 Turbo to iteratively evolve the SOMA-ATA metaheuristic by applying prompt-driven modifications over successive generations. The final evolved variant exhibited significant structural divergence from the original, integrating novel elements beyond traditional SOMA logic [34]. Another study introduced LLaMEA, an evolutionary framework that embeds an LLM within the optimisation loop to iteratively generate, mutate, and refine entire metaheuristic algorithms based on performance feedback from benchmark functions. The LLM drives the evolution process without explicit self-reflection, positioning this study under the LLM-generated modular algorithms [26].

The categorisation of metaheuristics used across the reviewed studies reveals a strong emphasis on swarm intelligence and evolutionary algorithms. These methods remain dominant due to their modularity and flexibility, allowing LLMs to intervene at multiple levels, such as generating operators, modifying solution candidates, or evolving algorithmic structures. Their population-based nature and reliance on variation operators make them especially suitable for LLM-driven adaptation and hybridisation.

A growing number of studies fall into hybrid and reflective categories, where LLMs are embedded directly into the optimisation loop. These include cases where LLMs guide search operators, assist with heuristic construction, or evolve entire algorithms, classified under subcategories such as LLM-integrated search guidance and LLM-guided algorithm evolution. Although explicit reflective optimisation remains rare, some frameworks hint at future directions where LLMs could self-assess or reason about their outputs, potentially enabling autonomous refinement mechanisms.

Bayesian and statistical techniques are also featured in the reviewed work, especially for hyperparameter tuning and multi-objective optimisation. These methods are valued for their sample efficiency and probabilistic reasoning, which make them effective in scenarios where LLM-driven models operate under computational or data constraints. Bayesian frameworks are typically used to complement LLM-generated heuristics, enabling faster convergence and better trade-offs in high-cost or black-box environments.

Finally, several studies introduce custom or derivative-free algorithms, often generated entirely by LLMs using structured prompt templates. These novel approaches deviate from conventional metaheuristics and reflect the experimental potential of LLMs as creative agents in algorithm design. As such, they are grouped under the derivative-free and custom optimisation algorithms category, highlighting the ability of LLMs to generate functional yet unconventional search strategies.

Overall, this analysis highlights a rapidly evolving research space where LLMs play diverse and increasingly central roles in metaheuristic development. Whether guiding operators, generating new heuristics, or evolving full algorithms, LLMs offer scalable, adaptive, and generative capabilities that align well with the challenges of modern optimisation.

4.3. Large language models used

The analysis of the LLMs used across the 25 studies shows a strong preference for recent, high-performance models, with a focus on versatility, model size, and adaptability to different tasks. The selected models fall into a few distinct categories, representing a broad spectrum of capabilities from general-purpose models to highly specialised variants.

OpenAI models:

The most frequently used models come from OpenAI, particularly GPT-4 and GPT-3.5. These models are mentioned in various versions, including GPT-4 Turbo, GPT-3.5 Turbo, and GPT-based models with specific configurations such as retrieval-augmented generation for enhanced knowledge retrieval. GPT-4 Vision Preview also appears, indicating interest in multimodal capabilities for more complex tasks. A large portion of the studies included in this review have employed a range of ChatGPT versions from OpenAI for metaheuristic applications, as indicated in several sources [22–24,26,28,30,31,34–36,39–42,44,45].

Table 5
Summary of LLMs used across studies.

Category	LLMs used	Number of studies	Reference(s)
OpenAI Models	GPT 2.0 GPT-4, GPT-3.5, GPT-4 Turbo, GPT-3.5 Turbo, GPT-4 Vision Preview	15	[22–24,26,28,30,31,34–36,39–42,44,45]
Anthropic Models	Claude-3-Opus	1	[28]
Meta's LLaMA Models	LLaMA2, LLaMA3, Code Llama	6	[22,27,29,35,41,44]
Mistral and Qwen Models	Mixtral, Mistral 7B, Qwen1.5-7B	4	[27,28,37,44]
Hybrid and Custom Configurations	Various LLMs (PaLM2, Gemini, UniTS Prompt)	4	[25,29,33,43]
Used LLMs are not specified	–	3	[15,32,38]

Anthropic models:
Claude models, specifically Claude-3-Opus, represent another choice of LLMs. Anthropic's focus on safety and interpretability might explain their inclusion in studies where ethical considerations are paramount. [28] introduces a novel hybrid approach that integrates metaheuristics with LLMS, including Claude-3-Opus, leveraging their pattern recognition capabilities to enhance the performance of metaheuristics in solving combinatorial optimisation problems, such as the Multi-Hop Influence Maximisation problem, demonstrating improved results compared to traditional metaheuristics and other machine learning-based integrations.

Meta's LLaMA models:
Meta's LLaMA series (LLaMA2, LLaMA3) and related models like Code Llama are widely used, often in versions with substantial parameter counts (e.g., 70B). These models are noted for their high performance and accessibility for researchers seeking open-source alternatives to proprietary models. Several studies in this review have used LLaMA models as LLMs to enhance metaheuristics [22,27,29,35,41,44].

Mistral and Qwen models:
Mistral models, such as Mixtral and Mistral 7B, are used across a few studies, often alongside OpenAI and Meta models. Similarly, Qwen models (e.g., Qwen1.5-7B) appear as specialised options for research requiring extensive fine-tuning capabilities. [27,28,37,44] have used these models along with metaheuristics in their experiments.

Hybrid and custom configurations:
Some studies, such as [25,29,33,43], used a mix of models or mentioned generic LLM capabilities without specific model names. This approach indicates the adaptability of LLMs for hybrid or ensemble-based research, leveraging various LLMs in combination to optimise specific outcomes.

Table 5 provides a detailed breakdown of the LLMs used, highlighting the dominance of OpenAI models, the increasing preference for open-source models like LLaMA, and the niche roles of Mistral, Qwen, and other specialised configurations. This structure provides a clear view of how different LLMs are employed across the studies, illustrating trends and preferences within the research landscape.

The analysis of LLMs used across the 25 studies highlights several key trends, reflecting a diverse landscape of model selection based on performance, flexibility, and suitability for specific research needs. Among the models used, OpenAI's offerings, particularly GPT-4 and GPT-3.5, emerged as the most frequently chosen, with different versions such as GPT-4 Turbo, GPT-3.5 Turbo, and GPT-4 Vision Preview being prominently featured. These models' versatility and high-performance capabilities make them ideal for a broad range of tasks, and the presence of GPT-4 Vision Preview further indicates interest in multimodal capabilities as researchers begin exploring applications that extend beyond text-only inputs.

Anthropic Claude models, including Claude-3-Opus, are also represented in the studies. Known for their focus on interpretability and safety, these models are often selected for studies where ethical considerations and transparency are essential. The presence of Claude models demonstrates an effort within the research community to address ethical concerns and to use models that are designed with robust safety mechanisms.

The frequent use of Meta's LLaMA series, particularly LLaMA2 and LLaMA3, and related models like Code Llama, underscores a growing interest in open-source alternatives to proprietary models. The LLaMA models, often available with substantial parameter counts (such as 70B), provide researchers with a high level of control, enabling customisation and fine-tuning for specific applications. This trend towards open-source models aligns with a broader shift in the research community towards transparency and accessibility in AI development, which fosters collaborative advancements in LLM research.

Mistral and Qwen models, including Mistral 7B, Mixtral, and Qwen1.5-7B, are also used across several studies, sometimes in combination with OpenAI or Meta models. These models offer specialised capabilities, with some studies leveraging their fine-tuning potential for niche research applications. Their inclusion in various studies suggests that researchers are exploring beyond mainstream models to find specific tools that best align with metaheuristics.

In addition to these well-known models, a subset of studies utilised hybrid or custom configurations, sometimes without specifying particular model names. These studies refer to general LLM capabilities, which suggest a flexible approach where various models are combined or adapted based on the specific requirements of the task. This adaptability reflects the evolving nature of LLM research, as researchers experiment with hybrid frameworks or custom configurations that integrate multiple models to optimise performance. Foundational transformer models like RoBERTa [47] and GPT-2 remain relevant for lightweight applications, though none of the reviewed studies directly implemented RoBERTa.

The range of LLMs employed in these studies highlights a balanced approach to model selection. Researchers evaluated each large language model based on its capabilities, openness, and suitability to their research objectives and the metaheuristics they employed. The strong presence of both proprietary and open-source models provides flexibility and enables a tailored approach to research in diverse domains. The integration of both cutting-edge and foundational models reveals a research landscape that values both innovation and practical utility as researchers strive to leverage LLMs effectively for a wide array of applications in metaheuristics.

The choice of LLMs across studies reflects a balance between performance, customisability, interpretability, and resource constraints tailored to the optimisation context. Proprietary models like GPT-4 and GPT-3.5 were frequently chosen due to their high accuracy, robust generalisation, and strong in-context learning capabilities, particularly for tasks such as prompt engineering, symbolic regression, and evolutionary strategy refinement. Claude models, known for their alignment with safety and transparency, were employed in scenarios demanding human-aligned decision support and interpretable optimisation flows. Open-source alternatives such as LLaMA2 and LLaMA3 were favoured in studies that required fine-tuning, local deployment, or integration into hybrid frameworks, offering flexibility and transparency over proprietary solutions. Lightweight models like Mistral and Qwen were

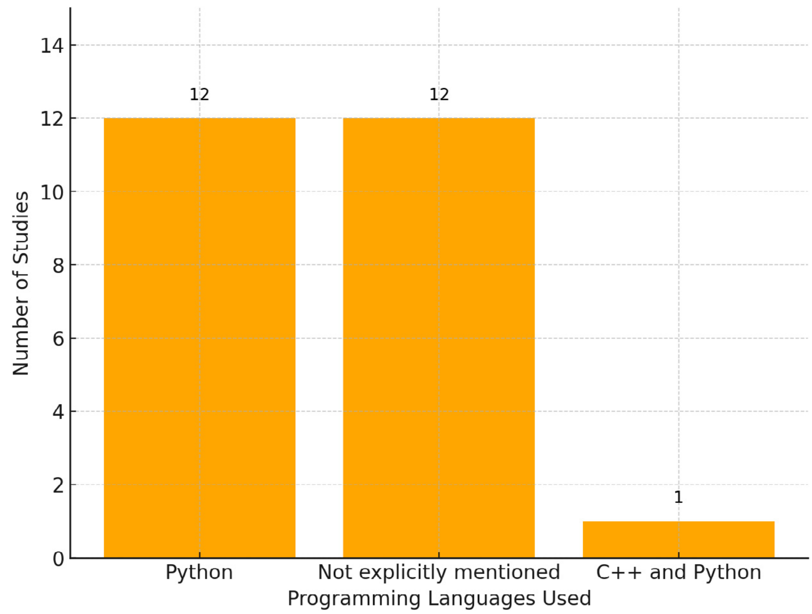

Fig. 10. Programming Languages Used In the Studies.

selected in computationally constrained environments or real-time systems, where efficiency took precedence over scale. This diverse usage suggests that model selection is often not solely performance-driven but also dictated by strategic alignment with optimisation objectives, infrastructure capabilities, and integration needs. Future studies are likely to benefit from systematic benchmarking across these models to better guide selection in varying application domains.

4.4. Programming languages used for the implementation of algorithms

In analysing the programming languages utilised in the studies, in order to implement the metaheuristics algorithms, Python emerged as the dominant choice, with 12 out of the 25 studies explicitly mentioning its use. This trend aligns with Python's reputation for its simplicity, extensive libraries, and strong community support in the fields of machine learning and optimisation. Python's versatility and ease of integration with various optimisation libraries make it a favourable choice for researchers implementing metaheuristics.

In contrast, 12 studies did not explicitly mention the programming language used, which may indicate a diversity of languages or environments depending on the study's requirements or the availability of implementation details. Additionally, one study, [28] reported using both C++ and Python (C++ for the metaheuristics and Python for LLMs), showcasing a hybrid approach that leverages C++'s performance efficiency alongside Python's flexibility for prototyping and integration with AI models. Fig. 10 summarises the distribution of programming languages used across the studies. This figure highlights the dominance of Python and the diversity in programming language reporting in the reviewed literature.

4.5. Application areas of the studied metaheuristics

The reviewed literature addresses a wide range of application areas, each leveraging optimisation and computational methods to tackle specific challenges. Engineering optimisation and Black-box optimisation emerge as the most frequently explored areas. Engineering optimisation focuses on developing optimal solutions to complex design and operational issues in engineering, while black-box optimisation addresses problems where internal system details are inaccessible, relying on adaptive algorithms to navigate input-output relationships effectively. The prevalence of these areas highlights a foundational reliance on optimisation techniques across diverse industries.

Another prominent focus is the travelling salesman problem, a classic combinatorial optimisation problem essential to logistics, transportation, and supply chain management. Closely related topics, such as vehicle routing problems and capacitated vehicle routing problems, illustrate the literature's attention to routing and scheduling issues, which are critical for optimising resource allocation and minimising operational costs. Additionally, explainable AI and neural architecture search reflect the integration of machine learning into optimisation research, addressing the need for interpretability in AI models and optimising neural network structures. Further application areas, including portfolio management and numerical optimisation, show the versatility of optimisation techniques, extending their utility beyond traditional engineering applications. Fig. 11 provides a detailed breakdown of these application areas, illustrating the frequency of each topic within the reviewed studies.

5. Content summary and thematic analysis

The current literature review shows that the idea of using LLMs in metaheuristics mainly began in 2023. As recent surveys such as [13,14] have highlighted, the integration of LLMs into metaheuristics is an emerging trend, with applications spanning algorithmic design, parameter tuning, and heuristic generation. Our review builds upon this work by conducting a structured thematic analysis of LLM contributions to metaheuristics, offering a more systematic categorisation of their roles and impacts. While these surveys provide an overview, our review takes a more detailed, role-based approach to understanding how LLMs influence each phase of the optimisation process.

To systematically classify the role of LLMs in metaheuristics, we introduce a taxonomy categorising their contributions. Fig. 12 visually represents the primary ways LLMs enhance metaheuristic optimisation, structured into four major roles with specific subcategories. As shown in Fig. 12, LLMs contribute significantly across multiple stages of the optimisation process. Algorithm design benefits from heuristic generation, parameter tuning is enhanced with automation, hybridisation improves metaheuristic fusion, and problem-solving sees advancements in constraint-handling and adaptive search strategies. This classification provides a structured understanding of how LLMs influence metaheuristics and guides further research in this domain.

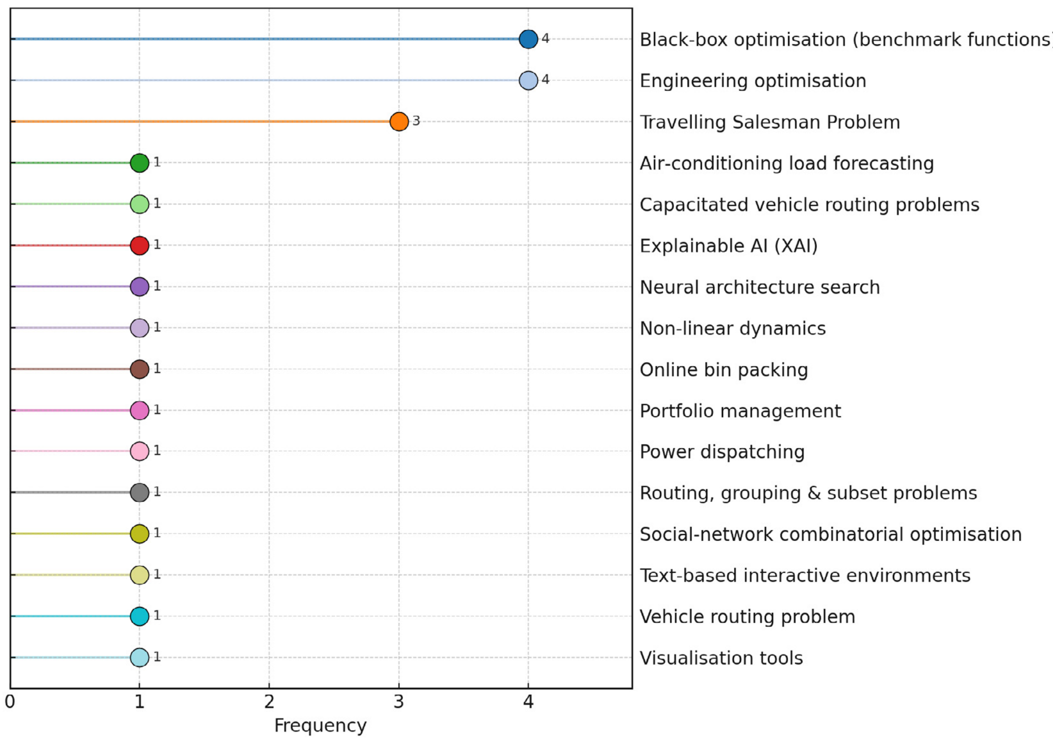

Fig. 11. Frequency of Application Areas for Metaheuristics in Reviewed Studies.

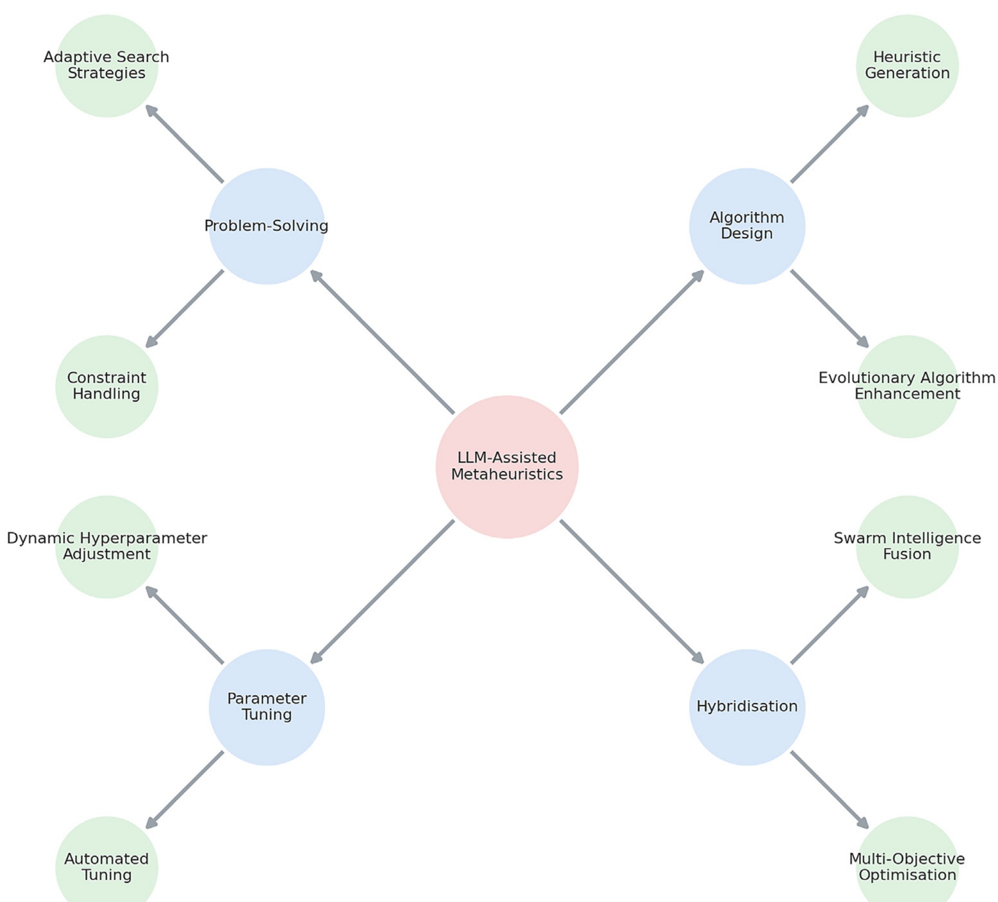

Fig. 12. Role-Based Taxonomy of LLM-Assisted Metaheuristics. This taxonomy categorises LLM contributions into four key roles: Algorithm Design, Parameter Tuning, Hybridisation, and Problem-Solving, each with specific subcategories representing how LLMs enhance metaheuristic optimisation.

Fig. 13. A word cloud made based on the abstract of 25 papers that we extracted data from.

Table 6
Summary of LLM contributions in metaheuristic optimisation.

Contribution area	Key contributions
Algorithm Design	LLMs assist in generating new heuristic strategies, designing novel metaheuristic frameworks, and automating mutation and crossover mechanisms in evolutionary algorithms.
Parameter Tuning	LLMs dynamically adjust hyperparameters, automate tuning through reinforcement learning techniques, and fine-tune metaheuristics to adapt to different problem instances.
Hybridisation	LLMs enable intelligent hybridisation of multiple metaheuristics, improving search efficiency by combining swarm intelligence with evolutionary algorithms and Bayesian optimisation.
Problem-solving	LLMs improve constraint-handling, adaptive search strategies, and multi-objective optimisation, offering explainability and assisting in balancing exploration-exploitation trade-offs.

Authors in [23,30] and [45] published the very first studies in this area. In [23], GPT-4 was used to create an improved mutation strategy for the differential evolution algorithm by defining a dynamic switching mutation strategy. Their results showed that the new mutation strategy was effective. This study highlights GPT-4's potential to innovate optimisation algorithms, although it may be challenging to replicate for more advanced differential evolution variants. In the other study [30], GPT-4 was used to create hybrid algorithms by combining six existing swarm intelligence algorithms. This led to two new metaheuristics, the enhanced swarm exploration and exploitation optimiser and the limited evaluation swarm optimiser, which balanced exploration and exploitation. Although promising for continuous optimisation, the study noted some issues with the correctness and efficiency of the generated code and algorithms. The other work in 2023 was done by Liu et al. [45], in which LLMs were integrated into an evolutionary framework to create, modify, and combine new algorithms for optimisation tasks. This proposed framework performed better than traditional greedy heuristics and domain-based models on different sizes of the travelling salesman problem, which showed improved scalability and generalisation brought to the algorithm by LLMs. Although the study showed positive results, it focused only on the travelling salesman problem, suggesting that more evaluation of other optimisation problems could strengthen the findings. Following these three foundational studies, our literature review reveals a steadily increasing interest in this area of research in 2024.

We have divided all the studies into five themes, which are discussed in the following sections. These themes demonstrate the integration of LLMs with metaheuristics as explored in the studies. A word cloud of the abstracts of all papers we reviewed in this systematic literature review is also given in Fig. 13.

To provide a structured overview of the contributions of LLMs in metaheuristics, we summarise their key roles in Table 6. This table highlights their impact across different optimisation tasks, including algorithm design, parameter tuning, hybridisation, and problem-solving. As illustrated in Table 6, LLMs significantly improve adaptability, automation, and hybridisation in metaheuristic frameworks. The following subsections will explore these contributions in greater detail, focusing on their specific applications and challenges.

5.1. Theme 1: Using LLMs to generate or improve metaheuristic algorithms

In the first theme, LLMs are used to improve or generate metaheuristics. For instance, Zhong [31] used ChatGPT3.5 to create a new optimisation algorithm called zoological search optimisation. The proposed algorithm outperformed others, but the author highlighted the challenges of prompt engineering and human oversight when using LLMs. In similar studies, [24,30] conducted two research works in which LLMs were employed to progressively improve existing metaheuristics. For instance, they took the self-organising migrating algorithm through 30 cycles of improvements and then found out that only one version showed significant improvement. In their second study, the differential evolution algorithm went through the same process. Their experimental results showed that LLM can improve the performance initially, but as we progressively do this by using an improved algorithm from the previous cycle, complexity increases without added benefits. Their findings demonstrate the potential of LLMs in algorithm design. However, LLMs introduced inconsistencies despite using the same query and human intervention was needed.

Another work in this category was done by Stein et al. [26], where an automated system named LLaMEA was developed, enabling LLMs to select, mutate, and generate algorithms based on a feedback mechanism. Some of the algorithms generated by LLaMEA managed to outperform existing metaheuristics. However, they only focused on continuous optimisation problems, and the overall process is computationally very expensive and highly dependent on prompt quality. In another similar work, Cai et al. [38] explored how LLMs can enhance evolutionary algorithms by designing the population and improving strategies, which led to the same conclusion on the drawbacks mentioned above. Our review of the literature also identified studies that employ LLMs to optimise the tuning of metaheuristics. In [36], for instance, an in-the-loop hyperparameter optimisation was proposed for LLM-based automated design of heuristics. In other words, an LLM was used to generate algorithmic structures, while evolutionary algorithms handled the selection and mutation. Hyper-parameter tuning was offloaded to an optimisation tool (SMAC). Their proposed approach also suffers from a high computational cost and faces scaling challenges.

Collectively, the studies in this theme demonstrate that LLMs have the potential to generate and improve metaheuristics. This process brings innovative solutions and tends to outperform non-LLM-based algorithms. As discussed above, however, common limitations are inconsistent performance improvements, high computational costs, reliance on well-crafted prompts, and the necessity of human oversight to guide the AI effectively. The real-world relevance of these LLM-assisted algorithm design approaches is evident across several application domains. In the reviewed studies, LLMs were used to support symbolic regression for discovering governing equations in physical systems, which has implications in scientific modelling and engineering simulations. Other practical scenarios include route optimisation for delivery logistics using variants of the travelling salesman problem, adaptive hybrid design of swarm algorithms for smart energy systems, and iterative improvement of optimisation logic in automated planning. These examples reflect the growing potential of LLMs to aid in developing new metaheuristics tailored to real-world constraints, including computational cost, solution quality, and domain-specific feasibility.

5.2. Theme 2: Enhancing metaheuristics with LLM guidance in optimisation processes

In the second theme, LLMs are paired with traditional metaheuristics to enhance or guide the metaheuristic processes in optimisation tasks. What makes these different from theme one is that LLMs do not generate/improve metaheuristics or heuristics but are a part of the optimisation and problem-solving process for specific applications shoulder to shoulder with metaheuristics. In other words, this theme includes studies that use LLMs as assistive tools in the optimisation lifecycle before or after metaheuristics in the optimisation pipeline.

For instance, Sartori et al. [28] used LLM as a pattern recognition tool within a variant of genetic algorithm to solve combinatorial problems. LLM was used to generate patterns to be assigned as probabilities to graph nodes, which led to outperforming state-of-the-art algorithms. However, the study faced limitations such as high computational costs and difficulties scaling to larger graphs due to token limits in LLMs. Zhong et al. [43] employed LLM to design neural networks, which is considered to be a combinatorial optimisation task. This allowed them to create an LLM-assisted optimiser for adversarial robustness neural architecture search. Despite the merits of their work, the study lacked details on computational costs, human involvement, and the potential benefits of using different LLMs or metaheuristics.

Chen et al. [32] proposed an LLM-guided framework for dynamic power dispatch in space-air-ground vehicular networks. The authors integrated LLMs with a dual deep deterministic policy gradient framework and the whale optimisation algorithm. LLMs were used to guide the power dispatching models, and the whale optimisation algorithm was used to optimise the generated candidate strategies. This method outperformed existing dispatch strategies, but the study suffers from limitations such as reliance on a single LLM and metaheuristic. This questions the generalisability of the proposed model across different scenarios.

In [33], the problem of air conditioning load forecasting was tackled by combining a time series prediction model based on prompts and LLMs with the grey wolf optimisation algorithm for hyperparameter tuning. The integration led to a slight improvement in prediction accuracy. However, the study might have been limited by small or region-specific datasets and the high computational cost associated with running the optimisation algorithm multiple times to ensure optimal solutions.

Collectively, the studies in this theme showed once again the potential of integrating LLMs with metaheuristics to enhance the optimisation process. LLM was not engaged in tuning or enhancing the metaheuristics but rather worked along them. This showed the success of this synergy across domains such as combinatorial optimisation, neural network design, power dispatch, and load forecasting. The coming challenges were computational resource demands, scalability issues, reliance on specific LLMs or metaheuristics, and the need for further validation in larger, real-world applications.

The integration of LLMs into metaheuristics has demonstrated measurable improvements in optimisation performance across multiple domains. The findings of the current review show that LLM-assisted metaheuristics consistently outperform traditional methods in terms of convergence speed, solution quality, and computational efficiency. The improved convergence speed reflects LLMs' ability to enhance search heuristics, while the superior solution quality and computational efficiency indicate their potential to optimise problem-solving strategies more effectively. The reviewed studies demonstrate that LLM-based parameter tuning methods have strong potential in real-world scenarios where continuous adaptation is crucial. For example, tuning strategies informed by LLMs have been applied in dynamic optimisation environments such as intelligent traffic signal control, where real-time adjustments improve traffic flow under changing conditions. Other examples include hyperparameter tuning in deep learning pipelines for medical diagnostics and resource allocation in edge computing, where task priorities and constraints vary over time. These cases highlight the role of LLMs in reducing manual intervention, accelerating convergence, and enhancing the responsiveness of optimisation algorithms in time-sensitive, data-driven contexts.

5.3. Theme 3: Using LLMs as primary solvers for optimisation tasks

In this theme, LLMs are directly applied to specific optimisation or symbolic regression tasks. In other words, the LLM itself acts as a primary solver or aid in generating solutions or models for complex systems. For instance, in [41], Du et al. proposed a framework using LLMs for the automatic discovery of governing equations in nonlinear dynamical systems. By generating and refining candidate equations, the LLM demonstrated better generalisation and physical interpretability than traditional symbolic regression methods. However, the method's performance under sparse or noisy data conditions requires further exploration. In addition, high computational costs are a significant limitation, especially with advanced LLMs like GPT-4.

In [29], Lange et al. investigated the use of LLMs within evolutionary optimisation algorithms for black-box optimisation problems. Their method, called EvoLLM, integrated an LLM as a recombination operator. Their comparison showed that EvoLLM shows better results than baseline algorithms on synthetic functions and small neuroevolution tasks. However, its performance may depend on the pre-training and fine-tuning of the LLM. The scalability is also limited by the context length and processing costs associated with LLMs.

in [39], Huang et al. enhanced the optimisation performance of LLMs using a multimodal approach for solving the capacitated vehicle routing problem. By incorporating both textual and visual inputs, the multimodal LLM outperformed the text-only version. This allowed it to capture problem structures more effectively and show improved routing solutions. Limitations include challenges in handling larger and more challenging capacitated vehicle routing problem instances, which were not explored by them. Also, there are difficulties in accurately remembering detailed information when only presented in text.

Collectively, the studies in this theme demonstrate the potential of LLMs in specialised optimisation tasks as primary solvers to provide innovative solutions and improved performance over traditional methods. Once again, a common challenge was high computational costs. Other limitations include handling of complex or unstructured data, scalability issues, and the need for further validation under various data conditions. The hybrid strategies outlined in the reviewed studies demonstrate considerable practical value across diverse domains. For instance, combining LLMs with swarm intelligence and evolutionary algorithms has been applied in real-world scenarios such as power dispatch optimisation in smart grids, where systems must respond adaptively to load fluctuations and user preferences. In the context of social network analysis, LLM-guided hybrids have been used to improve influence maximisation strategies for marketing and information dissemination. Hybrid models have also supported robust optimisation in dynamic scheduling, logistics planning, and energy forecasting, where multiple heuristic approaches must be integrated to balance competing objectives, uncertainty, and evolving constraints. These examples underscore the versatility and operational value of LLM-driven hybrid optimisation frameworks.

5.4. Theme 4: enhancing explainability using LLMs

In this theme's studies, LLMs have been leveraged to enhance the explainability of optimisation algorithms and their results. In [44], for instance, the authors integrated GPT-4 into the STNWeb tool to automate the interpretation of optimisation algorithms' behaviour. By generating natural language reports and basic graphics, the LLM made results more accessible to users unfamiliar with complex visualisations. Limitations of this work, though, included potential struggles with complex mathematical reasoning, dependence on basic prompt structures, and reproducibility issues due to reliance on proprietary models.

In [27], Baumann et al. made evolution strategies more explainable by integrating LLMs to generate user-friendly summaries of the optimisation process. The approach effectively provided clear and coherent explanations, which helped users to better and conveniently understand key aspects such as convergence behaviour and encounters with local optima. However, the first limitation is imposed by the LLM's context length, which may hinder processing longer log files. Another limitation is related to the challenges in extracting exact numerical values due to the stochastic nature of the LLM.

These studies illustrate that integrating LLMs can significantly enhance the explainability and accessibility of optimisation algorithms, aiding users in understanding complex processes and results. Challenges remain in handling complex mathematical reasoning, ensuring reproducibility without reliance on proprietary models, and managing limitations related to context length and data extraction. The integration of LLMs into constraint handling and multi-objective optimisation has promising real-world applicability. In energy systems, LLMs have supported grid balancing by navigating trade-offs between cost, load stability, and emission reduction. In healthcare resource planning, constraint-aware LLMs have been used to ensure treatment schedules meet regulatory and staffing constraints while optimising patient flow. In transportation planning, multi-objective frameworks enhanced by LLMs have enabled route design that balances travel time, fuel consumption, and traffic regulations. These examples illustrate how LLMs can enhance adaptability and decision support in scenarios where multiple, often conflicting, constraints must be satisfied simultaneously.

5.5. Theme 5: Using metaheuristics to optimise LLMs

In the fifth theme, the role of LLMs and metaheuristics swapped. Other than studies included in this review, during the paper inclusion and exclusion phase, we identified some studies in the literature that used metaheuristics and optimisation techniques to optimise LLMs or for prompt engineering. As these studies fell outside the scope of this systematic literature review, we excluded them.

For Instance, [48] proposes Plum as a paradigm for prompt learning using metaheuristic techniques, including hill climbing, simulated annealing, genetic algorithms (with and without crossover), tabu search, and harmony search, to optimise prompts for LLMs efficiently across black-box and white-box scenarios. [49] presented an end-to-end framework for enhancing customer service in unmanned retail stores, integrating visual recognition with fine-tuned LLMs to interpret customer needs and offer recommendations. Metaheuristic optimisation techniques, specifically the genetic algorithm and shuffled frog leaping algorithm, are employed to efficiently fine-tune the model parameters in a low-dimensional space, improving computational efficiency with limited training data. [50] explores Bayesian optimisation for tuning prompts in LLMs, demonstrating that it enhances task performance and reduces computational costs in black-box settings. [51] presents EvoPrompt, a metaheuristic approach that applies evolutionary optimisation techniques, including crossover and mutation, to iteratively refine prompts for improved post-ASR error correction using LLMs.

The studies discussed in this theme showed that metaheuristics can effectively optimise LLMs. This allows LLMs to show improved performance without increasing the parameters and expand their capabilities. As discussed, metaheuristics have been used in prompt engineering as well. However, the approaches are often constrained by high computational demands, scalability issues, and data requirements that should be addressed. Metaheuristics and optimisation processes have significant potential to innovate and enhance the use of LLMs. On the positive side, LLMs have proven effective in generating new algorithms, improving existing ones, assisting in specific optimisation tasks, and even enhancing explainability. They offer fresh approaches to complex problems, improve accuracy and adaptability, and provide user-friendly interpretations of optimisation processes. However, we observed common limitations as well. High computational costs as the use of LLMs often requires substantial processing resources, which makes scalability a challenge. Another common issue is that the effectiveness of LLMs heavily relies on prompt quality and can show inconsistent performance. These require human oversight to ensure reliability, which slows down achieving full autonomy in optimisation. Finally, studies often highlighted difficulties in applying LLMs to more complex or unstructured data, which raises concerns about generalisation and applicability across diverse tasks. These pros and cons underline the exciting potential and ongoing challenges in integrating LLMs with metaheuristics and optimisation tasks.

The reviewed studies also highlight real-world potential in using metaheuristics to enhance LLMs. For instance, metaheuristic-driven prompt optimisation has practical value in domains such as personalised education, where prompts must adapt to learner profiles in real time. In customer-facing AI systems, combining visual recognition with metaheuristically fine-tuned LLMs has enabled more effective recommendations and dynamic interaction in unmanned retail and service kiosks. Moreover, the use of search-based techniques to refine LLM parameters has been explored in constrained environments such as embedded systems, where computational efficiency and accuracy are critical. These cases illustrate how metaheuristics can meaningfully support the practical deployment of LLMs across industries.

Among the 25 papers identified, only one was categorised as a survey study. This paper, [15], explores how optimisation algorithms and LLMs enhance decision-making in dynamic environments by blending artificial intelligence with traditional methods, providing novel strategies to tackle complex computational problems and practical applications of language models. The study reviewed emerging frameworks that integrate LLMs into genetic programming, where LLMs generate, mutate, and recombine programmatic solutions using techniques such as language model crossover and reflective operator design.

6. Discussion and future direction

6.1. A new taxonomy-based on LLM roles in metaheuristics

According to the current literature review, the existing and future works in this area can be divided into four categories based on the role of LLM in metaheuristics: Advisor, Refiner, Enhancer, and Innovator.

While prior studies such as [13]. [14], and [15] have provided valuable insights into how LLMs can assist optimisation tasks, their classification approaches tend to be either broad thematic groupings (e.g., reinforcement learning, symbolic regression, or hyperparameter tuning) or application-oriented taxonomies that categorise research by use cases (e.g., scheduling, forecasting, or robotics). These models often overlook the functional depth and operational roles that LLMs play in metaheuristic workflows. In contrast, the proposed taxonomy in the current study introduces a novel role-based structure that classifies LLMs as Advisors, Refiners, Enhancers, or Innovators. This categorisation is grounded in how LLMs interact with core optimisation stages, whether by offering strategic guidance, modifying algorithmic logic, augmenting solution components, or autonomously generating complete algorithmic frameworks. This functional perspective enables clearer mapping between LLM capabilities and optimisation needs. Moreover, this taxonomy is generalisable across different classes of metaheuristics (e.g., swarm, evolutionary, hybrid), and offers practical relevance by guiding developers and researchers in aligning LLM capabilities with task-specific goals, system constraints, and desired levels of automation. Unlike previous taxonomies, it serves not only as a descriptive tool but also as a prescriptive framework for method selection and system design.

In the first class, *advisors*, advisor LLMs assist metaheuristics by providing guidance based on historical data, problem descriptions, algorithm code, and performance. This means that instead of changing the algorithm directly, they suggest potential directions and approaches that can lead to performance improvement and better results. Among the vast number of options available to a designer for modifying and adjusting metaheuristics, an advisor LLM can narrow down the options and prioritise more promising ones. The final intervention is made by humans, but the insights are provided by LLMs, leveraging their remarkable capability learned from training on historical data. In this case, it is recommended to use proper information in the context window so that advisor LLMs provide more helpful insights.

In the second class, *refiners*, LLMs focus on fine-tuning and optimising an existing aspect of a metaheuristic (e.g., controlling parameters). Just like humans, an LLM goes through an iterative process over different settings for those parameters to evaluate their impact and find the best configuration. It can be considered a trial-and-error approach to identify optimal parameters that improve algorithmic performance. The benefit is that the LLM does not make structural changes to the algorithm, which could introduce errors and instability. Instead, it works within the existing code and performs tuning and refining on selected elements to bring out the best out of the algorithm. This approach ensures the algorithm operates at a higher efficiency without compromising its foundational structure. As an advantage, the integrity of the original algorithm is maintained, so the rest of the errors and instability are reduced. Since refiners work only within the existing algorithm, they may miss opportunities for broader improvements that require structural changes.

In the third class, *enhancer*, LLMs engage in modifying and improving the optimisation process. In this case, an enhancer LLM dynamically interacts with metaheuristics and makes real-time adjustments. Utilising their context window, LLMs can suggest modifications to improve convergence, introduce new operators, or adjust components of the code to address challenges faced by metaheuristics during the optimisation process. In other words, this role behaves as a co-optimiser alongside metaheuristics to reach better results. The advantage of such a technique is the ability to adapt to challenges on the fly and be flexible co-optimisers in complex tasks. However, real-time adjustments require high computational power and may slow down the overall optimisation process. This added complexity can also make the optimisation process harder to interpret and debug, so explainability might be an issue.

In the last class, *innovator*, LLMs go beyond the previous roles. This is achieved when they design and propose new metaheuristics. Using their generative capability, they can either combine/hybridise existing metaheuristics or create a completely new algorithm. An innovator LLM brings fresh ideas and techniques into the design and development of metaheuristics. This expands the possibilities that we can achieve and create a unique algorithm for optimisation problems. Innovator LLMs bring high creativity to the table and reduce the need for manual input in designing new algorithms, which eventually leads to saving time. However, the stochastic nature of LLMs makes the outcome unpredictable, and we have to deal with some level of instability. This requires rigorous testing to ensure quality. It is the most computationally expensive role out of the above-mentioned four.

Fig. 14 shows these four roles of LLMs in metaheuristic optimisation positioned along two axes: Difficulty & Complexity and Value & Autonomy. As we move up the curve from advisor to innovator, each role requires higher complexity and autonomy, but greater value is achieved in the optimisation process. Advisor LLMs provide guidance with minimal complexity because they just suggest possible directions without directly altering the algorithm. Refiners take a more active role by fine-tuning parameters. Enhancers are more dynamic and make real-time changes to the optimisation process. This obviously requires more complexity and autonomy, which justifies its position in this figure. Finally, Innovators have the highest level of autonomy and complexity because they can create new metaheuristics. Techniques in this category offer maximum value by bringing unique solutions to an optimisation problem.

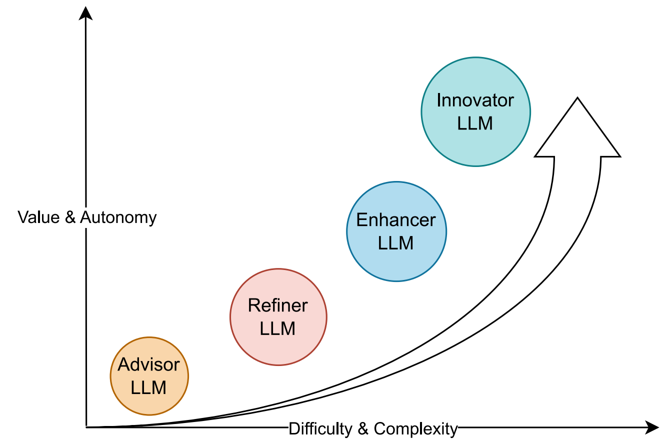

Fig. 14. Visualisation of LLM Roles in metaheuristic optimisation where Advisor, Refiner, Enhancer, and Innovator roles each contribute uniquely to improving algorithm performance and adaptability.

It is hard to measure the growth rate from advisor to innovator, but it is very likely going to be exponential. A possible justification for this is that each role requires an increase in skills and adds much more value than the last roles. Advisor LLMs only guide the process, which is easy to do. Refiners adjust small details, which need a bit more skill. Enhancers make live changes, which is much harder. Innovators create completely new algorithms/solutions, which are the hardest and bring the most value. With each step, the LLM's tasks become significantly more complex. These large leaps in difficulty and impact suggest exponential growth, where each new role demands much more than the last.

6.2. Future directions

This systematic literature review provides a comprehensive overview of seminal works employing LLMs in metaheuristics and evolutionary computation. It identifies significant unexplored areas worthy of pursuit. Based on this review and the author's insights, the following research directions are recommended. We regard these as "dimensions" that should be considered and explored to advance in this area, as illustrated in Fig. 15. By addressing these gaps, researchers can better harness the potential of LLMs in designing, optimising, and enhancing metaheuristics. The identified directions provide a roadmap for future advancements, which further enable and facilitate deeper integration of LLMs into the field of optimisation.

While the proposed taxonomy in this study is grounded in thematic synthesis, its empirical validation remains an important direction for future research. A practical next step would involve embedding this taxonomy into controlled experimental designs, where each role, advisor, refiner, enhancer, and innovator is explicitly operationalised and evaluated using consistent performance metrics such as convergence quality, generalisability, or runtime efficiency. Benchmarking tools could be developed to compare how effectively different LLMs support each role across various optimisation tasks. Additionally, the taxonomy could be used to guide the design of industry-facing platforms that systematically incorporate LLMs into existing optimisation workflows. These empirical and applied pathways would help establish the taxonomy's robustness and practical relevance.

Most existing studies lack a standard framework or employ generic ones (e.g., CRISPE). Prompt engineering is critically important when using LLMs, as it can significantly impact the design and coding of algorithms. Since no framework specifically tailored for LLMs in metaheuristics currently exists, developing one is a valuable research direction. A dedicated framework would allow consistent and efficient application of LLMs in the metaheuristic domain. This would lead to a

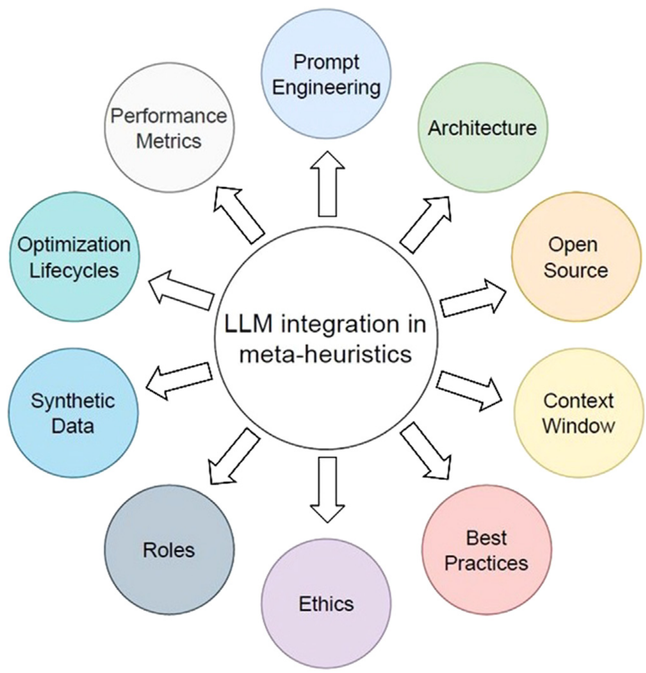

Fig. 15. Suggested future research areas for using LLMs in metaheuristics. Each "dimension" represents a different focus area, such as creating specific frameworks, setting ethical guidelines, using synthetic data, and promoting open-source tools. The diagram highlights different paths researchers can take to advance research in this area.

more standardised assessment of their role and contribution. Also, such a framework could support interoperability and comparability across studies, which accelerates innovation and refinement of metaheuristics.

Another area involves ethical frameworks for using LLMs in metaheuristics. Generating algorithms with LLMs may raise ethical concerns, such as potential biases in the generated solutions, a lack of transparency in decision-making processes, and issues surrounding intellectual property. There are ongoing debates within the community about the ethical guidelines most suitable for using LLMs in metaheuristics. How to establish these guidelines is an important open research question. Developing an ethical framework will help to mitigate these risks and make LLM-based metaheuristics more trustworthy and reliable.

In addition to ethical frameworks, best practices in this field are needed. Due to limited research, compiling a concrete list of best practices is challenging, yet any effort to create guidelines for designing, developing, testing, benchmarking, validating, refining, enhancing, and innovating in this area would be invaluable. A systematic set of best practices will provide practical guidance to researchers and practitioners and help them to make LLM-assisted metaheuristics more cohesive and reproducible. It will also encourage better documentation and sharing of methodologies for better collaboration in the optimisation community and beyond.

Another recommendation is to design a custom problem-solving process for LLMs, creating a new optimisation lifecycle that maximises accuracy and automation while reducing instability and human intervention. This would include involving LLMs in requirements analysis and problem definition, as they could enhance understanding and improve the overall process during stakeholder engagement, which often involves qualitative data. Although this study focused on algorithm design and development, there is a broader gap in frameworks guiding LLM use across the entire optimisation lifecycle. Such a tailored lifecycle has the potential to promote a shift towards greater autonomy in problem-solving.

Currently, there is no specific performance metric for evaluating LLM-based metaheuristics. Existing works use conventional performance metrics like convergence speed, accuracy, and statistical tests to measure the effectiveness of LLM-generated or improved algorithms. However, these do not quantify the LLM's impact on the optimisation process. For example, if an algorithm alters the source code of a metaheuristic, what effect does this have on the code itself? Is it logically more complex? These are challenging questions to answer. Thus, proposing new metrics and methods to effectively evaluate LLMs' impact on optimisation algorithms is recommended. Such new metrics help researchers to better understand the unique advantages and limitations of LLM-driven optimisation.

Another area of interest is investigating the impact of the context window on LLM performance in metaheuristics. Currently, there are no guidelines on what to include in the context window. While the need for a framework for prompt engineering has been previously highlighted, identifying the specific content to include in prompts remains essential, as it guides the content generation process of LLMs. This could include exploring various methods, such as zero-shot, few-shot, or chain-of-thought prompting. Developing a deeper understanding of context window configurations could significantly enhance the accuracy and efficiency of LLMs in solving optimisation tasks. Fine-tuning these settings would allow LLMs to perform more consistently, regardless of task complexity or variability.

The architecture of LLMs significantly impacts their integration with metaheuristics, affecting efficiency, adaptability, and scalability. Enhancements such as efficient token handling, adaptive attention mechanisms, and hybrid architectures can improve performance while reducing computational costs. Future research should explore architectural modifications that align LLMs more closely with the computational demands of metaheuristic frameworks.

Studies on GPT-o1 indicate that synthetic data generation can bridge data gaps and can be used to restrain or fine-tune LLM to achieve better performance. Identifying gaps in datasets used to train LLMs for metaheuristics and addressing these gaps is an exciting area for future research. We strongly encourage researchers in the metaheuristics community to explore this, as it could significantly increase LLM accuracy across various metaheuristic-related applications. Synthetic data could also facilitate the development of LLMs that are less biased and more robust.

Another research area is to propose architectures that allow different roles, as outlined in this work, to collaborate in automating the entire metaheuristics development process. Possible architectures might include multi-role systems, hierarchy-based models, or swarm-based approaches. Such architectures could maximise the potential of roles that LLMs can fulfil or introduce new roles, such as moderators, evaluators, or auditors, facilitating automation throughout the optimisation lifecycle.

The other area we suggest for future research is open-source initiatives to improve the accessibility and inclusivity of LLMs in metaheuristics. Open-source LLMs and tools allow the community to democratise the design, development, and testing of metaheuristics. They also promote transparency, which is a fundamental dimension of AI ethics. Open-source projects would further encourage collaboration and community engagement, too. This also leads to mitigation bias due to a diverse set of perspectives.

Given the novelty of the field, no comprehensive comparisons exist between different LLMs regarding their strengths and limitations in metaheuristics. Our analysis shows that GPT has been the most widely used, yet we believe other LLMs also hold potential, particularly for specialised tasks like coding. Further exploration in this area is encouraged. Such comparative studies provide insights into the adaptability, precision, and computational efficiency of different LLMs, which eventually help researchers to choose the most suitable models for specific metaheuristic tasks.

Another promising research direction is to fine-tune existing LLMs to better serve as advisors, refiners, enhancers, or innovators in metaheuristics. Fine-tuning and customising LLMs for different roles would give us the opportunity to optimise their performance in niche areas.

This specialisation would make LLMs valuable tools in the progressive development of metaheuristics.

Future research should also focus on practical challenges such as integrating LLMs with real-time optimisation systems, where latency, decision confidence, and feedback loops are critical. Another key direction is the development of lightweight, domain-specific LLMs that reduce computational overhead and are suitable for edge or embedded systems. Techniques such as reinforcement learning with human feedback, fine-tuning on problem-specific corpora, and retrieval-augmented generation could improve both the responsiveness and context relevance of LLM-generated suggestions. Additionally, research would benefit from the adoption or creation of standard datasets for benchmarking LLM-driven metaheuristics, such as datasets from logistics, smart grid scheduling, and medical resource allocation. Finally, integrating LLMs with open-source optimisation platforms (e.g., DEAP, Nevergrad, JMetalPy) may promote reproducibility and accelerate innovation in the field

7. Limitations

This systematic literature review offered a detailed examination of the integration of LLMs within metaheuristic optimisation algorithms. However, several limitations affect the scope and generalisability of its findings.

The review's scope was constrained by focusing on nine selected databases. While these cover extensive research, some relevant studies from less prominent sources or additional grey literature may have been missed. Limiting the review to English-language publications also introduces a potential bias, as it excludes research published in other languages, which may limit the diversity of methods and perspectives, especially from non-English-speaking regions. The search strategy captured only studies published up to late 2024, prioritising recent advancements while potentially overlooking earlier foundational research that provides historical context. A high degree of methodological variability was observed across the reviewed studies, particularly in terms of research designs and evaluation metrics, complicating direct comparisons and making generalisations less reliable. Despite an emphasis on high-quality, peer-reviewed sources, limitations in the original studies, such as small sample sizes or limited testing environments, may affect the reliability of some findings.

A number of reviewed studies offer practical strategies for addressing common LLM-related limitations. For example, [31] highlighted issues with prompt engineering during the development of the zoological search optimisation algorithm, noting that success varied depending on the structure and clarity of prompts. Similarly, [24,26] reported that scaling LLM-guided improvements across multiple generations introduced diminishing returns and required human oversight to filter viable algorithm versions. To mitigate computational overhead, studies like [16,29] employed lightweight models or hybrid strategies that offload parameter tuning to external optimisers while allowing the LLM to focus on structural decisions. These examples illustrate how prompt sensitivity, resource constraints, and scalability challenges are actively being managed in practice, informing best practices for future work.

Several reviewed studies proposed practical strategies to mitigate core LLM-related limitations. For instance, Zhong et al. [31] acknowledged that prompt quality significantly impacted the success of LLM-generated algorithms, noting frequent inconsistencies even with identical prompts. They recommended iterative refinement and structured prompt templates to improve reliability. Similarly, Pluhacek et al. [30] found that continuous LLM-guided evolution of metaheuristics introduced diminishing performance gains, with human oversight becoming essential to filter out ineffective versions. This underscores the scalability challenge of LLM-driven optimisation cycles. To reduce computational overhead, van Stein et al. [36] proposed an in-the-loop optimisation framework where the LLM focused on structural guidance while hyperparameter tuning was offloaded to the sequential model-based algorithm configuration, achieving performance benefits without incurring full inference costs. Huang et al. [39] demonstrated that multimodal input (e.g., visual + text) improved task performance while reducing context-related constraints in complex routing problems. Meanwhile, Cai et al. [38] highlighted the challenge of scaling LLM-based designs due to growing context size and memory limits, recommending lightweight models and task-specific fine-tuning. These examples reflect how current research is not only identifying limitations like prompt dependency, scalability, and resource usage, but also actively experimenting with architectural, procedural, and hybrid mitigation strategies to manage them in real-world optimisation scenarios.

8. Conclusion

This systematic literature review has examined the evolving role of large language models in metaheuristic optimisation, presenting a structured taxonomy that categorises their contributions across different stages of the optimisation process. The analysis reveals that LLMs are being leveraged for algorithm generation, parameter tuning, hybridisation, and performance enhancement, demonstrating significant potential for automating and improving metaheuristic techniques. Despite these advancements, several challenges persist, including computational cost, scalability, interpretability, and reliance on prompt engineering. Addressing these limitations is crucial for the broader adoption of LLM-driven optimisation. Additionally, ethical concerns and reproducibility issues must be considered as these models become more integrated into decision-making frameworks. As practical implications, this study provides a roadmap for researchers and practitioners seeking to leverage LLMs in optimisation. It highlights emerging trends, identifies research gaps, and underscores the need for more standardised benchmarking and evaluation methodologies. With regard to future research directions, the study suggests that moving forward, research should focus on developing more efficient LLM-optimisation pipelines, reducing computational overhead, improving interpretability, and exploring adaptive self-improving metaheuristics. Additionally, further empirical studies across diverse problem domains are needed to validate and refine existing approaches. By providing a comprehensive synthesis of current research and a strategic outlook for future advancements, this review serves as a reference for the continued exploration of LLM-driven metaheuristic optimisation.

CRediT authorship contribution statement

Reza Ghanbarzadeh: Conceptualization, Methodology, Extraction, Software, Writing - First draft, Writing. **Seyedali Mirjalili:** Supervision, Validation, Extraction, Review & Editing.

Declaration of competing interest

We declare that there is no conflict of interest in connection with this research and publication. We confirm that this paper has not been submitted elsewhere simultaneously and is not under review at any other journal.

Acknowledgement

The authors would like to acknowledge the use of ChatGPT 4-o for refining the language of this paper.

Data availability

Data will be made available on request.

References

[1] Thomas Hanne, Rolf Dornberger, Computational Intelligence, Springer International Publishing, Cham, 2023.

[2] Timothy P. Lillicrap, Adam Santoro, Luke Marris, Colin J. Akerman, Geoffrey Hinton, Backpropagation and the brain, Nature Rev. Neurosci. 21 (6) (2020) 335–346.

[3] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, vol. 30 (2017).

[5] Abeba Birhane, Atoosa Kasirzadeh, David Leslie, Sandra Wachter, Science in the age of large language models, Nat. Rev. Phys. 5 (5) (2023) 277–280.

[6] Kashif Hussain, Mohd Najib Mohd Salleh, Shi Cheng, Yuhui Shi, Metaheuristic research: a comprehensive survey, Artif. Intell. Rev. 52 (2019) 2191–2233.

[7] Tansel Dokeroglu, Tayfun Kucukyilmaz, El-Ghazali Talbi, Hyper-heuristics: A survey and taxonomy, Comput. Ind. Eng. 187 (2024) 109815.

[8] Abhishek Kumar Dey, Gaurav Prakash Gupta, Satyabrata Pradhan Sahu, A metaheuristic-based ensemble feature selection framework for cyber threat detection in IoT-enabled networks, Decis. Anal. J. 7 (2023) 100206.

[9] M.K. Zuhanda, S.A.R.S. Hasibuan, Y.Y. Napitupulu, An exact and metaheuristic optimization framework for solving vehicle routing problems with shipment consolidation using population-based and swarm intelligence, Decis. Anal. J. 13 (2024) 100517.

[10] M.M. Rahman, R. Nguyen, M. Severson, S. Williams, D. Hartley, A decision support tool for e-waste recycling operations using the hen-and-chicks bio-inspired optimization metaheuristic, Decis. Anal. J. 7 (2023) 100216.

[11] Lotfi A. Zadeh, Is there a need for fuzzy logic? Inform. Sci. 178 (13) (2008) 2751–2779.

[12] H. Yu, J. Liu, Deep insights into automated optimization with large language models and evolutionary algorithms, 2024, arXiv preprint arXiv:2410.20848.

[13] Qi Zhao, Qiqi Duan, Bai Yan, Shi Cheng, Yuhui Shi, Automated design of metaheuristic algorithms: A survey, 2023, ArXiv Preprint arXiv:2303.06532.

[14] Xingyu Wu, Sheng-hao Wu, Jibin Wu, Liang Feng, Kay Chen Tan, Evolutionary computation in the era of large language model: Survey and roadmap, IEEE Trans. Evol. Comput. (2024).

[15] Sen Huang, Kaixiang Yang, Sheng Qi, Rui Wang, When large language model meets optimization, 2024, arXiv preprint arXiv:2405.10098.

[16] V.B. Parthasarathy, A. Zafar, A. Khan, A. Shahid, The ultimate guide to fine-tuning LLMs from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities, 2024, arXiv preprint arXiv:2408.13296.

[17] Seyedali Mirjalili, Jin Song Dong, Andrew Lewis, Nature-inspired optimizers, Stud. Comput. Intell. 811 (2020) 7–20.

[18] David H. Wolpert, William G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.

[19] Chaoning Zhang, Chenshuang Zhang, Sheng Zheng, Yu Qiao, Chenghao Li, Mengchun Zhang, Sumit Kumar Dam, Chu Myaet Thwal, Ye Lin Tun, Le Luang Huy, et al., A complete survey on generative ai (aigc): Is chatgpt from gpt-4 to gpt-5 all you need?, 2023, arXiv preprint arXiv:2303.11717.

[20] Barbara Kitchenham, Procedures for performing systematic reviews, Keele, UK, Keele Univ. 33 (2004) (2004) 1–26.

[21] Matthew J. Page, David Moher, Patrick M. Bossuyt, Isabelle Boutron, Tammy C. Hoffmann, Cynthia D. Mulrow, Larissa Shamseer, Jennifer M. Tetzlaff, Elie A. Akl, Sue E. Brennan, et al., PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews, Bmj 372 (2021).

[22] Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, Guojie Song, Reevo: Large language models as hyper-heuristics with reflective evolution, 2024, arXiv preprint arXiv:2402.01145.

[23] Michal Pluhacek, Anezka Kazikova, Adam Viktorin, Tomas Kadavy, Roman Senkerik, Investigating the potential of AI-driven innovations for enhancing differential evolution in optimization tasks, in: 2023 IEEE International Conference on Systems, Man, and Cybernetics, SMC, IEEE, 2023, pp. 1070–1075.

[24] Michal Pluhacek, Jozef Kovac, Peter Janku, Tomas Kadavy, Roman Senkerik, Adam Viktorin, A critical examination of large language model capabilities in iteratively refining differential evolution algorithm, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2024, pp. 1855–1862.

[25] Rui Zhong, Shilong Zhang, Jun Yu, Masaharu Munetomo, GeminiDE: A novel parameter adaptation scheme in differential evolution, in: 2024 6th International Conference on Data-Driven Optimization of Complex Systems, DOCS, IEEE, 2024, pp. 33–38.

[26] Niki van Stein, Thomas Bäck, Llamea: A large language model evolutionary algorithm for automatically generating metaheuristics, 2024, arXiv preprint arXiv:2405.20132.

[27] Jill Baumann, Oliver Kramer, Towards explainable evolution strategies with large language models, 2024, arXiv preprint arXiv:2407.08331.

[28] Camilo Chacón Sartori, Christian Blum, Filippo Bistaffa, Guillem Rodríguez Corominas, Metaheuristics and large language models join forces: Towards an integrated optimization approach, 2024, ArXiv E-Prints, pages arXiv–2405.

[29] Robert Lange, Yingtao Tian, Yujin Tang, Large language models as evolution strategies, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2024, pp. 579–582.

[30] Michal Pluhacek, Anezka Kazikova, Tomas Kadavy, Adam Viktorin, Roman Senkerik, Leveraging large language models for the generation of novel metaheuristic optimization algorithms, in: Proceedings of the Companion Conference on Genetic and Evolutionary Computation, 2023, pp. 1812–1820.

[31] Rui Zhong, Yuefeng Xu, Chao Zhang, Jun Yu, Leveraging large language model to generate a novel metaheuristic algorithm with crispe framework, Clust. Comput. (2024) 1–35.

[32] Xingchi Chen, Xiaoyang Lu, Qing Li, Dazhou Li, Fa Zhu, Integration of LLM and human-AI coordination for power dispatching with connected electric vehicles under SAGVNs, IEEE Trans. Veh. Technol. (2024).

[33] Xichao Zhao, Zhuofan Tang, Jie Shi, Boya Deng, Xian Yu, Qingrong Zheng, Research on short term prediction technology of regional air conditioning based on time series large model, in: 2024 6th International Conference on Energy Systems and Electrical Power, ICESEP, IEEE, 2024, pp. 591–594.

[34] Michal Pluhacek, Jozef Kovac, Adam Viktorin, Peter Janku, Tomas Kadavy, Roman Senkerik, Using LLM for automatic evolvement of metaheuristics from swarm algorithm SOMA, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2024, pp. 2018–2022.

[35] Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, Guojie Song, Large language models as hyper-heuristics for combinatorial optimization, 2024, arXiv preprint arXiv:2402.01145.

[36] Niki van Stein, Diederick Vermetten, Thomas Bäck, In-the-loop hyper-parameter optimization for LLM-based automated design of heuristics, 2024, arXiv preprint arXiv:2410.16309.

[37] Bingdong Li, Zixiang Di, Yanting Yang, Hong Qian, Peng Yang, Hao Hao, Ke Tang, Aimin Zhou, It's morphing time: Unleashing the potential of multiple LLMs via multi-objective optimization, 2024, arXiv preprint arXiv:2407.00487.

[38] Jinyu Cai, Jinglue Xu, Jialong Li, Takuto Yamauchi, Hitoshi Iba, Kenji Tei, Exploring the improvement of evolutionary computation via large language models, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2024, pp. 83–84.

[39] Yuxiao Huang, Wenjie Zhang, Liang Feng, Xingyu Wu, Kay Chen Tan, How multimodal integration boost the performance of llm for optimization: Case study on capacitated vehicle routing problems, 2024, arXiv preprint arXiv:2403.01757.

[40] Yossathorn Tianrungroj, Hitoshi Iba, ROIL: Rule optimization via large language model for imitation learning, in: 2024 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2024, pp. 1–8.

[41] Mengge Du, Yuntian Chen, Zhongzheng Wang, Longfeng Nie, Dongxiao Zhang, Large language models for automatic equation discovery of nonlinear dynamics, Phys. Fluids 36 (9) (2024).

[42] Fanfan Lin, Xinze Li, Weihao Lei, Juan J. Rodriguez-Andina, Josep M. Guerrero, Changyun Wen, Xin Zhang, Hao Ma, PE-GPT: A new paradigm for power electronics design, IEEE Trans. Ind. Electron. (2024).

[43] Rui Zhong, Yang Cao, Jun Yu, Masaharu Munetomo, Large language model assisted adversarial robustness neural architecture search, 2024, arXiv preprint arXiv:2406.05433.

[44] Camilo Chacón Sartori, Christian Blum, Gabriela Ochoa, Large language models for the automated analysis of optimization algorithms, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2024, pp. 160–168.

[45] Fei Liu, Xialiang Tong, Mingxuan Yuan, Qingfu Zhang, Algorithm evolution using large language model, 2023, arXiv preprint arXiv:2311.15249.

[46] Mariana A. Londe, Luciana S. Pessoa, Carlos E. Andrade, Mauricio G.C. Resende, Biased random-key genetic algorithms: A review, European J. Oper. Res. 321 (1) (2025) 1–22.

[47] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019, CoRRarXiv:1907.11692. URL http://arxiv.org/abs/1907.11692.

[48] Rui Pan, Shuo Xing, Shizhe Diao, Wenhe Sun, Xiang Liu, Kashun Shum, Renjie Pi, Jipeng Zhang, Tong Zhang, Plum: Prompt learning using metaheuristic, 2023, arXiv preprint arXiv:2311.08364.

[49] Wang Wang, Ping Zhang, Changxia Sun, Dengchao Feng, Smart customer service in unmanned retail store enhanced by large language model, Sci. Rep. 14 (1) (2024) 19838.

[50] Antonio Sabbatella, Andrea Ponti, Ilaria Giordani, Antonio Candelieri, Francesco Archetti, Prompt optimization in large language models, Mathematics 12 (6) (2024) 929.

[51] Rithik Sachdev, Zhong-Qiu Wang, Chao-Han Huck Yang, Evolutionary prompt design for LLM-based post-ASR error correction, 2024, arXiv preprint arXiv:2407.16370.