# ELEVATION: Exploiting Large Language Models to Improve Efficiency, Scalability, and Flexibility of Heuristics in Solving Optimization Problems

***Submitted to special session 1:***

***AI-Driven Optimization: Transforming Optimization with LLMs***

Lorenzo Saccucci[1,3], Marco Boresta[2,3], Alessandro Pannone[1,3], Antonio Vicinanza[3], and Francesco Romito[3]

[1] Department of Computer, Control, and Management Engineering "Antonio Ruberti" (DIAG), Sapienza University of Rome, V. Ariosto, 25, Rome, 00185, RM, Italy `{lorenzo.saccucci,alessandro.pannone}@uniroma1.it`

[2] Institute of Systems Analysis and Computer Science "Antonio Ruberti", National Research Council (CNR), Via dei Taurini, 19, Rome, 00185, RM, Italy `marco.boresta@iasi.cnr.it`

[3] Spindox S.p.A., Rome, Italy `{francesco.romito,antonio.vicinanza}@spindox.it`

**Abstract.** Heuristics are widely used to address optimization problems. However, manual heuristic design remains a labor-intensive process that requires considerable expertise. ELEVATION introduces an innovative multi-agent framework, inspired by Google's FunSearch, which integrates Large Language Models (LLMs) with automatically generated heuristics to facilitate the automatic design of optimization solutions. Starting from basic heuristics or from scratch, the LLMs generate and refine executable code through a robust self-assessment mechanism that ensures solution quality. Although the Vehicle Routing Problem (VRP) constitutes the primary use case, the framework is inherently adaptable to a broad spectrum of optimization challenges. Empirical evaluations on a real-world case study demonstrate the potential of ELEVATION to automatically discover effective heuristics for complex optimization problems.

**Keywords:** Large Language Models · AI-Driven Optimization · Automated Heuristic Design

## 1 Introduction

Optimization plays a central role in the operation of industrial and service systems, supporting decision-making in domains such as healthcare [1, 7], manufacturing [31, 34], logistics [6, 8], energy management [11], transportation [9, 13], and telecommunications [3, 14].

While combinatorial problems like the VRP are ubiquitous, accurately modeling them and translating those models into robust code remains a significant challenge, especially when working in a real-world context . Because exact methods are often limited to small-scale instances, real-world applications typically rely on heuristics and metaheuristics [18, 30]. However, designing these algorithms is a complex engineering task that requires specialized expertise and extensive empirical validation, both of which are often scarce in practice.

To overcome these limitations, recent research has turned toward automated algorithm design. Recent advances in LLMs offer a promising paradigm to streamline this process by supporting code generation, algorithmic reasoning, and heuristic refinement [24, 33].

In this work, we introduce ELEVATION, an end-to-end framework for the automated evolution of heuristics. By embedding LLMs within a structured evolutionary loop, ELEVATION systematically modifies and evaluates existing heuristic code to achieve significant performance gains over baseline implementations.

This paper makes the following contributions:

- We describe the design and architecture of the ELEVATION framework, a problem-agnostic agent-based evolutionary system for the automated generation and refinement of heuristics.
- We present a real-world case study in which ELEVATION evolves a simple baseline heuristic to address industrial-scale VRP instances.
- We report empirical results showing consistent quantitative improvements achieved through automated heuristic evolution.

The remainder of the paper is organized as follows. Section 2 reviews related work and positions ELEVATION within the growing literature on LLM-based optimization. Section 3 introduces the methodology and architecture of the proposed framework, detailing the problem representation and the evolutionary process. Section 4 presents the experimental evaluation and numerical results obtained on a real-world industrial case study. Finally, Section 5 concludes the paper and outlines directions for future research.

## 2 Related Works

The integration of LLMs into combinatorial optimization is a rapidly emerging frontier. LLMs have demonstrated the ability to discover novel algorithms by generating executable code rather than just point solutions, effectively shifting the role of the model from a simple solver to an automated designer. Frameworks like FunSearch [28] and Evolution of Heuristics (EoH) [24] pioneered this by embedding LLMs in evolutionary loops to achieve iterative, reflective improvement. This paradigm has since been extended to Neural Architecture Search [27, 12], multi-objective optimization [32], and hyper-heuristic strategies like ReEvo [33], which leverage historical performance data to guide the generation of increasingly sophisticated strategies.

The Vehicle Routing Problem remains a primary benchmark due to its industrial complexity and the vast array of constraints encountered in practical scenarios. Recent frameworks like ARS [22] and AutoDH [25] leverage LLM agents and reinforcement learning to generate constraint-aware heuristics or perform adaptive algorithm selection. Beyond text-based modeling, multimodal approaches have shown that incorporating visual representations of VRP instances can better capture spatial relationships [19], while LLM-driven Large Neighborhood Search has demonstrated success in generating adaptive local search operators. Furthermore, as transparency is vital for industrial adoption, tools like Route-Explainer [21] have begun addressing the interpretability of automated routing decisions.

The convergence of these directions suggests that LLMs, when integrated with traditional optimization and evolutionary algorithms, offer a powerful paradigm for automated heuristic design. ELEVATION builds upon these foundations by introducing a self-assessment mechanism that ensures the robustness and quality of evolved heuristics, balancing the generative flexibility of LLMs with the reliability required for production-level industrial applications.

## 3 Methodology

ELEVATION is a general-purpose framework for the automated discovery and refinement of heuristics for combinatorial optimization problems. It combines principles from evolutionary computation with the generative and reasoning capabilities of LLMs, enabling a systematic exploration of heuristic design spaces that domain experts traditionally handcraft.

The framework operates through an iterative analyze–generate–evaluate–evolve loop, where candidate heuristics are represented as executable programs. Starting from basic heuristics or from scratch, algorithms are progressively refined through performance-driven feedback. ELEVATION is designed to be both problem-agnostic and language-agnostic, allowing it to be applied across different optimization domains and execution environments.

The framework is built above a precise problem representation, described in section 3.1, which allows us to define the objectives, the heuristic interface, and the evaluation dataset.

The framework adopts an agent-centric architecture, in which each agent is assigned a clearly defined task and is capable of utilizing external tools such as web search, mathematical solvers, and other computational resources. The framework is model-agnostic, as it does not rely on any specific large language model and is compatible with a wide range of models. The general architecture of ELEVATION, illustrated in Figure 1, highlights the two main steps that allow initializing and then evolving the heuristics population.

### 3.1 Problem Representation

Each optimization task is defined through a compact set of artifacts that specify what must be optimized, how heuristics are expressed, and how they are evaluated. In particular, the framework requires the following items:

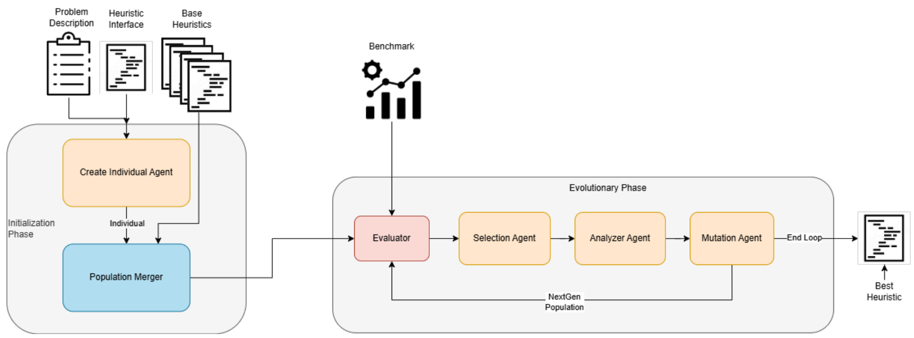


**Fig. 1.** High-level architecture of ELEVATION. The framework operates in two stages: (i) an *Initialization Phase* that builds the initial population of heuristics by generating and/or loading individuals, and (ii) an *Evolution Phase* that iteratively evaluates, selects, analyzes, and transforms heuristics until termination, returning the best-performing solution and an evolution report.

- **Problem Description**, expressed in natural language, describing objectives, constraints, and domain assumptions;
- **Heuristic Interface**, defining the required function signature and implicitly the target programming language;
- **Benchmark Dataset**, composed of representative instances and reference solutions used to compute task-specific KPIs;
- **Base Heuristics**, which can be used to seed the initial population. It is optional.

All domain-specific knowledge is encapsulated in these artifacts, allowing the core evolutionary process to remain independent of the specific optimization problem.

### 3.2 Initialization Phase

During the initialization phase, the framework constructs the initial population of candidate heuristics. Each individual corresponds to a complete heuristic implemented as executable source code that conforms to the specified heuristic interface.

The population is constructed through two complementary strategies. First, an agent generates a diverse set of heuristics based on the problem description and interface, thereby promoting the exploration of heterogeneous algorithmic approaches. Second, when available, pre-existing expert-designed heuristics can be injected into the population, providing strong baseline solutions and preserving domain-specific knowledge. The resulting population integrates both sources into a unified set of individuals.

### 3.3 Evolutionary Phase

Starting from the initialized population, ELEVATION executes an iterative evolutionary loop, as depicted in Figure 1. The loop is repeated for a fixed number of epochs or until a termination condition is met. At each epoch, every individual undergoes the following sequence of steps:

1. *Evaluation*: This core agent evaluates heuristics by executing them on the benchmark under uniform resource and time constraints. In addition, it performs syntactic correction of the code when necessary. Upon completion of the evaluation, task-specific key performance indicators (KPIs)—such as feasibility, solution quality, and runtime—are computed to characterize the performance of each individual.
2. *Selection*: Individuals are ranked according to a prioritized KPI hierarchy, where feasibility is typically enforced as a hard constraint. A Top-$K$ strategy retains the best-performing heuristics, forming an elite subset.
3. *Analysis*: The agent analyzes selected heuristics. It inspects the source code and KPI outcomes in the context of the problem description. It can use a web-search tool to increase its knowledge and find new hints for enhancing the source code. The agents identify weaknesses and propose structured, performance-oriented improvement suggestions.
4. *Mutation*: The agent applies the analyzer's recommendations to each heuristic with the objective of translating the collected ideas into executable code. To this end, the agent may leverage external tools, such as web search, to identify effective implementation strategies. The resulting mutations are semantic and performance-aware, yielding refined variants rather than random syntactic alterations.

This evolutionary loop progressively improves the population over successive epochs. At termination, ELEVATION returns the best-performing heuristic identified during the search, together with a structured evolution log that records KPI trajectories and generated code for reproducibility and analysis.

## 4 Numerical Experiments

### 4.1 Problem Description

We evaluate ELEVATION on a real-world Pickup and Delivery Problem with Time Windows arising from the EU Horizon 2020 E2COMATION project, which focuses on optimizing energy consumption in manufacturing and logistics processes. The problem concerns the distribution of a fast-moving consumer goods (FMCG) supply chain, where a heterogeneous fleet of vehicles stationed at a central depot (warehouse) must deliver goods to a set of supermarkets. Each problem instance corresponds to a single operational day.

Each vehicle must execute a set of *jobs*, where a job consists of (1) loading a specified quantity of goods at the depot and (2) delivering the goods to a designated supermarket.

Both loading and unloading operations require service times that depend on the quantity handled plus a fixed component; however, since quantities are predetermined, each activity has a known, fixed service time. Vehicles may perform *multiple trips* [10]: after completing a set of deliveries, a vehicle can return to the depot, reload, and depart for another delivery round, with a constraint enforcing that vehicles must be completely unloaded before reloading; i.e., mixing loads from different trips is not allowed.

The problem incorporates several classes of constraints commonly found in rich vehicle routing formulations [30]:

- **Time windows**: each supermarket has an associated time window during which service must occur [15, 29];
- **Vehicle capacity**: the total load on each vehicle cannot exceed its capacity at any point during a trip [4];
- **Maximum driving time**: labor regulations impose a maximum driving time per vehicle per day [16, 2], which is mandatory in real world vehicle routing problems.

Given the real-world nature of the problem, feasible solutions may not serve all jobs. Unassigned jobs do not make a solution infeasible but incur a high penalty in the objective function. The optimization criterion is a weighted multi-objective function [20, 17] that minimizes: (1) the total distance traveled; (2) the total $CO_2$ emissions [5, 23, 26], where each vehicle has a specific emission factor (kg $CO_2$/km).

### 4.2 Computational Results

Table 2 presents a comprehensive comparison between the heuristic generated by the proposed ELEVATION framework and a Greedy heuristic baseline (Algorithm 1). The Greedy baseline was selected due to its simplicity and representativeness of approaches that practitioners without domain expertise might naturally employ when addressing this problem. The evaluation encompasses 20 instances organized into four distinct groups based on problem characteristics and scale. The BASE group comprises five real-world instances with problem sizes ranging from 300 to 450 customers. The remaining instances are synthetically generated from the real-world scenarios by aggregating customers according to their retail outlet typology. These instances are then partitioned into three groups based on problem size: Group 1 includes five instances with approximately 50 customers each, Group 2 consists of five instances with approximately 100 customers, and Group 3 comprises five instances with approximately 250 customers. Performance is assessed across three key metrics: objective function value, total distance traveled, and total emissions generated. Testing experiments were conducted on a server equipped with dual Intel Xeon Gold 5218 processors (12 cores total, 2.30GHz), 251 GB RAM. Table 1 summarizes the main parameters of the experimental setup adopted for the evaluation of the ELEVATION framework. In this evaluation, the Top-$K$ strategy was instantiated with $K = 1$, retaining only the best-performing heuristic at each evolutionary epoch.

| Parameter | Value |
|---|---|
| Language Model (LLM) | Claude Sonnet 4 |
| Evolutionary Epochs | 10 |
| Elite Population Size ($K$) | 1 |
| Max Iterations per Epoch | 1000 |

**Table 1.** Experimental configuration for ELEVATION framework.

**Algorithm 1** Greedy heuristic baseline.

```
Initialize empty routes for all vehicles
for each job j ∈ J do
    best_cost ← ∞, best_insertion ← null
    for each vehicle v compatible with j do
        for each feasible insertion position p in v's route do
            cost ← insertion_cost(v, j, p)
            if cost < best_cost then
                best_cost ← cost, best_insertion ← (v, p)
            end if
        end for
    end for
    if best_insertion ≠ null then
        Insert j at best_insertion
    else
        Mark j as unassigned
    end if
end for
```

The results indicate that the heuristic automatically generated by ELEVATION consistently outperforms the Greedy baseline across all tested instances. In terms of objective function value, the ELEVATION-generated solution yields better results in all 20 instances, with an average improvement of 23.87% over Greedy (range: 7.71%–38.01%). Performance gains are observed across all instance groups, indicating that the framework maintains effectiveness over different problem scales. Similar improvements are observed for secondary performance indicators. On average, total distance traveled and total emissions are reduced by 30.12% and 32.17%, respectively. The reductions are more pronounced for the BASE and larger synthetic instances, suggesting potential benefits for real-world application scenarios.

Notably, the ELEVATION-generated heuristic achieves improved performance across all three metrics simultaneously for all tested instances. This consistency across multiple evaluation criteria is encouraging and suggests that the framework can discover optimization strategies that address multiple objectives concurrently. However, it should be noted that these results are obtained against a single baseline heuristic, and comparison with more sophisticated optimization methods would provide additional insight into the framework's competitive performance.

**Table 2.** Performance comparison of the algorithm introduced by ELEVATION against Greedy heuristic

| Instance | Objective Function | | Total Distance (km) | | Total Emissions (kg $CO_2$) | |
|---|---|---|---|---|---|---|
| | **ELEV.** | **Greedy** | **ELEV.** | **Greedy** | **ELEV.** | **Greedy** |
| *Group: BASE* | | | | | | |
| 01 | **17,454** | 21,321 | **11,052** | 14,854 | **36,024** | 48,422 |
| 02 | **18,118** | 23,011 | **11,354** | 16,021 | **36,888** | 52,648 |
| 03 | **11,698** | 14,683 | **7,370** | 10,095 | **23,782** | 33,133 |
| 04 | **15,479** | 20,911 | **9,748** | 14,524 | **31,311** | 47,627 |
| 05 | **18,320** | 21,480 | **12,012** | 14,960 | **38,585** | 49,192 |
| *Group: Group 1* | | | | | | |
| 01 | **2,664** | 2,869 | **1,724** | 1,845 | **5,397** | 5,989 |
| 02 | **2,922** | 3,541 | **1,750** | 2,219 | **5,468** | 6,961 |
| 03 | **1,280** | 1,472 | **820** | 912 | **2,596** | 2,855 |
| 04 | **1,979** | 2,524 | **1,213** | 1,565 | **3,904** | 5,095 |
| 05 | **2,192** | 2,402 | **1,391** | 1,523 | **4,507** | 4,785 |
| *Group: Group 2* | | | | | | |
| 01 | **1,410** | 1,702 | **891** | 1,108 | **2,695** | 3,436 |
| 02 | **1,391** | 1,763 | **833** | 1,131 | **2,582** | 3,571 |
| 03 | **595** | 811 | **343** | 478 | **1,019** | 1,578 |
| 04 | **1,031** | 1,389 | **597** | 881 | **1,840** | 2,834 |
| 05 | **1,336** | 1,518 | **848** | 983 | **2,625** | 3,096 |
| *Group: Group 3* | | | | | | |
| 01 | **22,223** | 27,086 | **15,303** | 19,392 | **50,201** | 63,690 |
| 02 | **22,825** | 28,862 | **15,561** | 20,716 | **51,135** | 67,960 |
| 03 | **10,905** | 13,449 | **7,174** | 9,478 | **23,303** | 30,209 |
| 04 | **19,335** | 26,685 | **13,195** | 19,131 | **43,142** | 63,284 |
| 05 | **22,475** | 28,686 | **15,502** | 20,616 | **50,481** | 67,452 |

## 5 Conclusions

This paper introduced ELEVATION, an evolutionary framework based on LLM agents for the automated generation and refinement of heuristics for combinatorial optimization problems. The framework integrates large language models within a structured evolutionary loop, where candidate heuristics are represented as executable code and iteratively analyzed, evaluated, and improved based on performance-driven feedback. By decoupling domain-specific problem representations from the core evolutionary process, ELEVATION is designed to be problem-agnostic and adaptable to a wide range of optimization domains.

The experimental results obtained on a real-world vehicle routing case study indicate that the heuristics automatically generated by ELEVATION consistently outperform a simple greedy baseline across all tested instances and evaluation metrics. The observed improvements, which exceed 20% on average for objective value, total distance, and emissions, suggest that the framework is capable of discovering effective and robust heuristic strategies without requiring extensive manual heuristic engineering. While the comparison is currently limited to a single baseline, the results provide encouraging evidence of the framework's potential to support automated heuristic design in practical settings.

It is important to note that these are preliminary results that demonstrate the feasibility and potential of the ELEVATION framework. Future research directions include several key enhancements: the introduction of a crossover agent to enable recombination of successful algorithmic components, testing on problems with increased complexity and additional constraints, and comprehensive benchmarking against more specialized state-of-the-art algorithmic solutions. These extensions will further validate the framework's capabilities and explore its limits across a broader range of combinatorial optimization scenarios.

**Acknowledgments.** This work was supported by public funding through the European Union's Horizon 2020 programme under the E2COMATION project (Grant Agreement No. 958410), and by the Italian National Recovery and Resilience Plan (PNRR) within the ELEVATION project (PE00000013, PE1 – FAIR "Future Artificial Intelligence Research", Spoke 5 – High-Quality AI, CUP B53C22003980006). The PhD scholarship of one of the authors, Lorenzo Saccucci, is funded by PNRR (CUP B53C24002120004).

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Abdalkareem, Z.A., Amir, A., Al-Betar, M.A., Ekhan, P., Hammouri, A.I.: Healthcare scheduling in optimization context: a review. Health and Technology **11**(3), 445–469 (2021)
2. Aksen, D., Özyurt, Z., Aras, N.: Open vehicle routing problem with driver nodes and time deadlines. Journal of the Operational Research Society **58**(9), 1223–1234 (2007)

3. Avella, P., Calamita, A., Palagi, L.: A compact formulation for the base station deployment problem in wireless networks. Networks **82**(1), 52–67 (2023)
4. Baldacci, R., Battarra, M., Vigo, D.: Routing a heterogeneous fleet of vehicles. In: The vehicle routing problem: latest advances and new challenges, pp. 3–27. Springer (2008)
5. Bektaş, T., Laporte, G.: The pollution-routing problem. Transportation research part b: methodological **45**(8), 1232–1250 (2011)
6. Boresta, M., Croella, A.L., Gentile, C., Palagi, L., Pinto, D.M., Stecca, G., Ventura, P.: Optimal network design for municipal waste management: Application to the metropolitan city of rome. Logistics **8**(3), 79 (2024)
7. Boresta, M., Giovannelli, T., Roma, M.: Managing low–acuity patients in an emergency department through simulation–based multiobjective optimization using a neural network metamodel. Health Care Management Science **27**(3), 415–435 (2024)
8. Boresta, M., Pinto, D.M., Stecca, G.: Bridging operations research and machine learning for service cost prediction in logistics and service industries. Annals of Operations Research **342**(1), 113–139 (2024)
9. Bussieck, M.R., Winter, T., Zimmermann, U.T.: Discrete optimization in public rail transport. Mathematical programming **79**(1), 415–444 (1997)
10. Cattaruzza, D., Absi, N., Feillet, D.: Vehicle routing problems with multiple trips. 4or **14**(3), 223–259 (2016)
11. Cesaroni, E., Ciccarelli, F., Palagi, L., Ruvio, A.: Optimal sizing of energy storage systems in railway transportation. Journal of Energy Storage **140**, 119135 (2025)
12. Chen, A., Dohan, D., So, D.: Evoprompting: Language models for code-level neural architecture search. Advances in neural information processing systems **36**, 7787–7817 (2023)
13. Ciancimino, A., Inzerillo, G., Lucidi, S., Palagi, L.: A mathematical programming approach for the solution of the railway yield management problem. Transportation science **33**(2), 168–181 (1999)
14. D'Andreagiovanni, F., Mannino, C., Sassano, A.: Gub covers and power-indexed formulations for wireless network design. Management Science **59**(1), 142–156 (2013)
15. Dumas, Y., Desrosiers, J., Soumis, F.: The pickup and delivery problem with time windows. European journal of operational research **54**(1), 7–22 (1991)
16. Erera, A.L., Morales, J.C., Savelsbergh, M.: The vehicle routing problem with stochastic demand and duration constraints. Transportation Science **44**(4), 474–492 (2010)
17. Ferreira, J.C., Steiner, M.T.A., Canciglieri Junior, O.: Multi-objective optimization for the green vehicle routing problem: A systematic literature review and future directions. Cogent Engineering **7**(1), 1807082 (2020)
18. Gendreau, M., Potvin, J.Y., et al.: Handbook of metaheuristics, vol. 2. Springer (2010)
19. Huang, Y., Zhang, W., Feng, L., Wu, X., Tan, K.C.: How multimodal integration boost the performance of llm for optimization: Case study on capacitated vehicle routing problems. In: 2025 IEEE Symposium for Multidisciplinary Computational Intelligence Incubators (MCII). pp. 1–7. IEEE (2025)
20. Jozefowiez, N., Semet, F., Talbi, E.G.: Multi-objective vehicle routing problems. European journal of operational research **189**(2), 293–309 (2008)
21. Kikuta, D., Ikeuchi, H., Tajiri, K., Nakano, Y.: Routeexplainer: an explanation framework for vehicle routing problem. In: Pacific-asia conference on knowledge discovery and data mining. pp. 30–42. Springer (2024)

22. Li, K., Liu, F., Wang, Z., Tong, X., Han, X., Yuan, M., Zhang, Q.: Ars: Automatic routing solver with large language models. arXiv preprint arXiv:2502.15359 (2025)
23. Lin, C., Choy, K.L., Ho, G.T., Chung, S.H., Lam, H.: Survey of green vehicle routing problem: past and future trends. Expert systems with applications **41**(4), 1118–1138 (2014)
24. Liu, F., Tong, X., Yuan, M., Lin, X., Luo, F., Wang, Z., Lu, Z., Zhang, Q.: Evolution of heuristics: Towards efficient automatic algorithm design using large language model. arXiv preprint arXiv:2401.02051 (2024)
25. Ma, L., Hao, X., Yang, R., Hai, Z.: Automatic algorithm design assisted by llms for solving vehicle routing problems. In: 2024 IEEE 17th International Conference on Signal Processing (ICSP). pp. 247–252. IEEE (2024)
26. Macrina, G., Pugliese, L.D.P., Guerriero, F.: The green-vehicle routing problem: a survey. In: Modeling and optimization in green logistics, pp. 1–26. Springer (2020)
27. Nasir, M.U., Earle, S., Togelius, J., James, S., Cleghorn, C.: Llmatic: neural architecture search via large language models and quality diversity optimization. In: proceedings of the Genetic and Evolutionary Computation Conference. pp. 1110–1118 (2024)
28. Romera-Paredes, B., Barekatain, M., Novikov, A., Balog, M., Kumar, M.P., Dupont, E., Ruiz, F.J., Ellenberg, J.S., Wang, P., Fawzi, O., et al.: Mathematical discoveries from program search with large language models. Nature **625**(7995), 468–475 (2024)
29. Sartori, C.S., Buriol, L.S.: A study on the pickup and delivery problem with time windows: Matheuristics and new instances. Computers & Operations Research **124**, 105065 (2020)
30. Vidal, T., Crainic, T.G., Gendreau, M., Prins, C.: Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. European Journal of Operational Research **231**(1), 1–21 (2013)
31. Wang, S., Lu, X., Li, X., Li, W.: A systematic approach of process planning and scheduling optimization for sustainable machining. Journal of Cleaner Production **87**, 914–929 (2015)
32. Yao, S., Liu, F., Lin, X., Lu, Z., Wang, Z., Zhang, Q.: Multi-objective evolution of heuristic using large language model. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 27144–27152 (2025)
33. Ye, H., Wang, J., Cao, Z., Berto, F., Hua, C., Kim, H., Park, J., Song, G.: Reevo: Large language models as hyper-heuristics with reflective evolution. Advances in neural information processing systems **37**, 43571–43608 (2024)
34. Zhang, R., Chiong, R.: Solving the energy-efficient job shop scheduling problem: a multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption. Journal of Cleaner Production **112**, 3361–3375 (2016)