Pareto-Grid-Guided Large Language Models for Fast and High-Quality Heuristics Design in Multi-Objective Combinatorial Optimization

Ha Minh Hieu[1*], Hung Phan[1*], Tung Duy Doan[1], Tung Dao[1], Cong Dao Tran[2], Huynh Thi Thanh Binh[1]

[1]Hanoi University of Science and Technology, Vietnam
[2]FPT Software AI Center, Vietnam
{hieu.hm220026, hung.pd252073m, tung.dd224906, tung.dv242050m}@sis.hust.edu.vn, daotc2@fpt.com, binhht@soict.hust.edu.vn

Abstract

Multi-objective combinatorial optimization problems (MOCOP) frequently arise in practical applications that require the simultaneous optimization of conflicting objectives. Although traditional evolutionary algorithms can be effective, they typically depend on domain knowledge and repeated parameter tuning, limiting flexibility when applied to unseen MOCOP instances. Recently, integration of Large Language Models (LLMs) into evolutionary computation has opened new avenues for automatic heuristic generation, using their advanced language understanding and code synthesis capabilities. Nevertheless, most existing approaches predominantly focus on single-objective tasks, often neglecting key considerations such as runtime efficiency and heuristic diversity in multi-objective settings. To bridge this gap, we introduce **M**ulti-heuristics for MOCOP via **Pa**reto-**G**rid-guided **E**volution of LLMs (MPaGE), a novel enhancement of the Simple Evolutionary Multiobjective Optimization (SEMO) framework that leverages LLMs and Pareto Front Grid (PFG) technique. By partitioning the objective space into grids and retaining top-performing candidates to guide heuristic generation, MPaGE utilizes LLMs to prioritize heuristics with semantically distinct logical structures during variation, thus promoting diversity and mitigating redundancy within the population. Through extensive evaluations, MPaGE demonstrates superior performance over existing LLM-based frameworks, and achieves competitive results to traditional Multi-objective evolutionary algorithms (MOEAs), with significantly faster runtime.

Code — https://github.com/langkhachhoha/MPaGE
Extended version — https://arxiv.org/pdf/2507.20923

1 Introduction

Multi-objective combinatorial optimization problems (MOCOP) commonly arise in real-world applications such as vehicle routing, production planning, where multiple conflicting objectives must be optimized simultaneously over a large, discrete solution space (Türkyılmaz et al. 2020; Liu et al. 2020; Phan Duc et al. 2025). Unlike single-objective optimization, MOCOP aims to approximate the Pareto front, which captures trade-offs among non-dominated solutions.

Due to the NP-hard nature of these problems, exact algorithms are often impractical, resulting in the widespread use of heuristic and metaheuristic methods such as NSGA-II, MOEA/D, and MOEA/D-DE (Deb et al. 2002; Zhang and Li 2007a; Li and Zhang 2008). Despite their success, these methods typically rely on domain-specific knowledge and extensive iterative search. Several neural approaches have been proposed for MOCOP, aiming to automatically learn heuristics and improve adaptability (Zhang et al. 2022; Hieu et al. 2024; Fan et al. 2024). However, these methods often require retraining for different problem sizes, demand substantial computational resources, and struggle to generalize to problems with unseen input formats.

Recently, Large Language Models (LLMs) have shown strong capabilities in automatic heuristic design, offering a new paradigm for optimization algorithm development (Wu et al. 2024; Liu et al. 2025; Novikov et al. 2025; Tran et al. 2025). By leveraging their language understanding and code generation abilities, LLMs can produce heuristics and the corresponding implementations with minimal human intervention. Recent approaches integrate LLMs with evolutionary strategies to iteratively evolve effective problem-solving programs (Liu et al. 2024; Ye et al. 2024; Romera-Paredes et al. 2024a; van Stein and Bäck 2024). While achieving competitive performance and often surpassing traditional methods, most existing LLM-based evolution frameworks primarily target single-objective problems, with limited exploration of their applicability to multi-objective settings. The inherent complexity and trade-off structure of MOCOP introduce significant challenges that require tailored and robust design strategies. To address this, Huang et al. (Huang, Zhang, and Liu 2025) propose an LLM-based framework for discovering and refining evolutionary operators suited to diverse MOCOP. However, runtime efficiency, which is crucial for the real-world deployment of MOCOP solvers, remains an underexplored aspect of LLM-driven heuristic design. MEoH (Yao, Chen, and Wang 2025) considers both optimality and efficiency within a multi-objective evolutionary framework. Additionally, prior LLM-based methods tend to produce populations of algorithms with similar operational

*These authors contributed equally.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

logic and slight representation difference, limiting both population diversity and the creativity of LLMs.

To address the aforementioned issues, we propose MPaGE, a novel framework designed to solve MOCOP while simultaneously discovering a pareto front of LLM-generated heuristics that jointly optimize solution quality and runtime efficiency, with an explicit emphasis on promoting heuristic diversity. Our approach curates heuristic algorithms for the Simple Evolutionary Multiobjective Optimization (SEMO) paradigm, leveraging Pareto Front Grid (PFG) to guide the design of LLM-based variation heuristics by partitioning the objective space into grids, retaining leading individuals from promising regions, and enhancing both solution quality and search efficiency. From these regions, MPaGE builds a pool of elitist candidates and employs LLMs to assess their semantic structures, clustering them into groups of similar logic. Variation is then performed with respect to these clusters, promoting semantic diversity and mitigating redundancy within the heuristic population. To the best of our knowledge, this is the first comprehensive evaluation of LLM-generated heuristics on standard MOCOP, addressing both solution quality, computational efficiency and semantic diversity. Our main contributions are as follows:

- We introduce MPaGE, the first framework to systematically combine LLMs with the SEMO paradigm and PFG, aiming to solve MOCOP by jointly optimizing runtime, solution quality, and maintaining semantic diversity.
- We leverage LLMs to verify the logical structure of heuristics and perform cross-cluster recombination, thereby enhancing diversity and reducing redundancy through logically dissimilar variations.
- We conduct extensive experiments on standard MOCOP benchmarks, demonstrating consistent improvements in runtime efficiency, solution quality, and semantic diversity over LLM-based baselines and traditional MOEAs.

2 Related Works

2.1 Multi-objective Optimization Algorithms

Solving MOCOP often involves extending metaheuristics to handle multiple objectives in discrete domains. NSGA-II (Deb et al. 2002) and MOEA/D (Zhang and Li 2007a) remain popular for effectively maintaining Pareto front diversity. Local search techniques like Pareto Local Search (PLS) (Paquete and Stützle 2004), SEMO (Laumanns, Thiele, and Zitzler 2004) improve exploitation by iteratively exploring non-dominated neighbors. More recent neural approaches aim to learn the entire Pareto set, as in PMOCO (Lin, Yang, and Zhang 2022), or boost diversity through dual mechanisms, as in NHDE (Chen et al. 2023b). Despite their success in solving MOCOP, these methods still rely on handcrafted components, struggle to generalize across diverse problem settings, and incur substantial computational costs.

2.2 LLMs for Heuristic Design

LLMs are increasingly used to automate heuristic design in combinatorial optimization. Core approaches like EoH, AEL, and FunSearch employ evolutionary frameworks to generate, combine, and refine heuristics in natural language or code, outperforming traditional methods on TSP, bin packing, and scheduling tasks (Liu et al. 2024, 2023; Romera-Paredes et al. 2024b). Similarly, HSEvo incorporates harmony search principles into the LLM-driven generation process to promote diversity and adaptability in solutions (Dat, Doan, and Binh 2025). Other work combines LLMs with synthesis techniques like MCTS for guided search (Zheng et al. 2025). While promising, most of these methods primarily focus on single-objective tasks, with limited attention to MOCOP, neglects other practical criteria like efficiency and complexity (Gutjahr 2012).

2.3 Multi-Objective Optimization with LLMs

Recent LLM-based heuristics have been applied to metaheuristic design. Huang, Zhang, and Liu (2025) used LLMs to generate crossover and mutation operators for multi-objective optimization, later extending this to evolutionary multitasking with LLM-designed knowledge transfer models (Huang et al. 2024). Nevertheless, these methods overlook the computational cost of generating and evaluating heuristics, a key challenge in MOEAs. MEoH, REMoH (Yao, Chen, and Wang 2025; Forniés-Tabuenca et al. 2025) present a multi-objective evolutionary framework that optimizes multiple performance metrics, such as optimality and efficiency, to discover trade-off algorithms in a single run. However, they struggle to distinguish heuristics with similar logic but different implementations, reducing diversity on the pareto front and hindering multi-objective exploration.

3 Preliminary

3.1 Multi-Objective Combinatorial Optimization

Generally, we consider MOCOP defined as: $\min_{x\in\mathcal{X}} \quad \mathbf{f}(x) = (f_1(x), \ldots, f_M(x))$, where $\mathcal{X}$ is a finite or discrete feasible set, and $\mathbf{f}\colon \mathcal{X} \to \mathbb{R}^M$ maps each solution to M objectives to be minimized. Trade-offs among objectives are characterized by Pareto optimality, as defined below, and solution quality is commonly assessed using the hypervolume (HV) indicator (Zitzler and Thiele 1999).

Definition 1 (Pareto dominance). For solutions $x^a, x^b \in \mathcal{X}$, x^a dominates x^b, denoted $x^a \prec x^b$, if $f_i(x^a) \le f_i(x^b)$ for all $i \in \{1, \ldots, M\}$, and there exists some j such that $f_j(x^a) < f_j(x^b)$.

Definition 2 (Pareto optimality). A solution $x^* \in \mathcal{X}$ is Pareto optimal if no $x \in \mathcal{X}$ satisfies $x \prec x^*$. The set of all such solutions forms the Pareto set (PS), and Pareto front $\mathrm{PF} = \{\mathbf{f}(x) \mid x \in \mathrm{PS}\}$.

3.2 Simple Evolutionary Multiobjective Optimization

Simple Evolutionary Multiobjective Optimization (SEMO) is a minimalist yet effective MOEAs, widely used in theoretical analyses of runtime performance (Laumanns, Thiele, and Zitzler 2004). Similar to Pareto Local Search (PLS) (Paquete and Stützle 2004), SEMO maintains an archive of non-dominated solutions. Each iteration selects a random solution from the archive, explores one of its neighbors, and

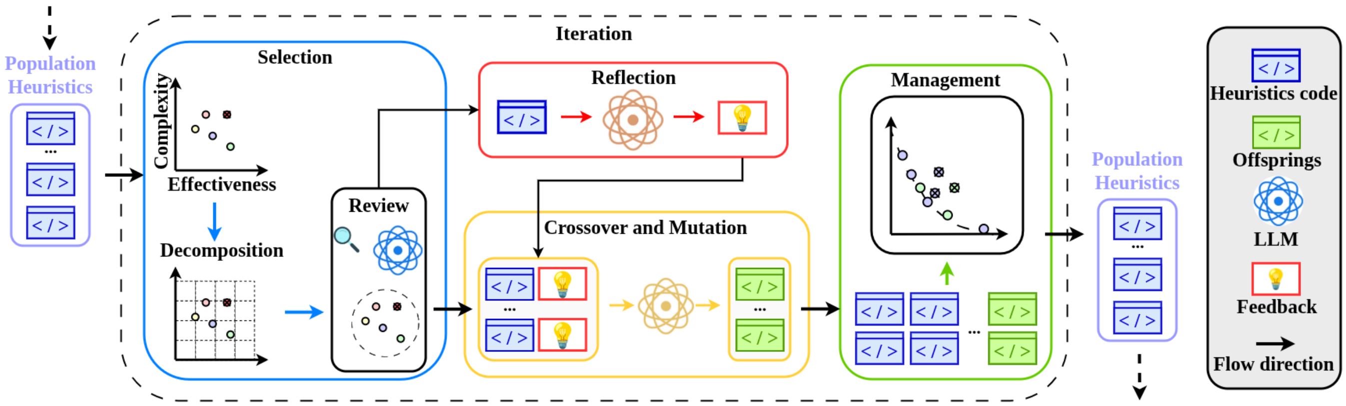


Figure 1: Overview of the proposed method. The heuristics are first represented in a feature space, and then partitioned into grid cells based on their objective values. Potential parent heuristics are then chosen with the assistance of LLM-based review from grids. Crossover and mutation operations are applied, guided by informative feedback from LLM-based reflection to enhance population diversity. Finally, the non-dominated individuals within each cell are retained to form the next generation for the subsequent iteration.

updates the archive if a new nondominated solution is discovered. Unlike PLS, which evaluates all neighbors, SEMO samples only one neighbor per iteration, which may limit its anytime performance. See appendix B for detailed insights compared to other frameworks.

Algorithm 1: Simple Evolutionary Multiobjective Optimization

$s \leftarrow rand\_generation()$
$A \leftarrow \{s\}$
repeat
  $s \leftarrow$ ***selection***(A) ; // Random s from A
1 $s' \leftarrow$ ***neighborhood_exploration***(s) ; // Select a neighbor of s randomly
2 **if** $\nexists a \in A : a \prec s'$ **then** // Check dominance
3    $A \leftarrow A \cup \{s'\} \setminus \{a \in A \mid s' \prec a\}$
until $stop\_condition()$;
return A

4 Methodology

4.1 Problem Formulation

We formalize heuristics design for MOCOP as a Language Multi-Criteria Heuristic Design (LMHD) problem, aiming to discover heuristics that balance evaluation criteria reflecting various aspects of heuristic behavior, such as effectiveness, efficiency, or generalization across problems. The key components are introduced as follows:

Multi-Criteria Evaluation Function. We define $E : \mathcal{H} \to \mathbb{R}^M$ to evaluate each heuristic $h \in \mathcal{H}$ over M criteria, capturing its expected performance and key behavioral aspects on a given MOCOP.

Language Multi-Criteria Heuristic Design. Language Multi-Criteria Heuristic Design (LMHD) is a variant of hyper-heuristic that leverages LLMs to generate a diverse set of heuristics. The goal is to discover heuristics that balance multiple criteria simultaneously. Given the heuristic space $\mathcal{H}$, the evaluation function $E(h) = (e_1(h), e_2(h), \ldots, e_M(h))$ assigns to each heuristic h a vector of M criteria to be minimized. The aim is to approximate the Pareto front of nondominated heuristics in $\mathcal{H}$.

Our goal is to design MOCOP-specific heuristics for the ***selection*** and ***neighborhood exploration*** steps. , corresponding to lines 4 and 5 in Algorithm 1.

The proposed LMHD takes problem specifications as input and generates heuristics optimized over two practical and complementary criteria: solution quality and computational efficiency. Specifically, we define:

$$E : \mathcal{H} \to \mathbb{R}^2, \quad E(h) = (e_1(h), e_2(h)), \tag{1}$$

where e_1 is the average negative hypervolume across a set of instances, measuring solution quality, and e_2 is the total running time. These two criteria reflect critical trade-offs in real-world applications, where high performance must be balanced with efficiency. Each heuristic $h \in \mathcal{H}$ is a complete algorithm that, given the current population A, returns a new candidate solution s' by internally performing both selection and neighborhood generation:

$$h : 2^{|\mathcal{S}|} \to \mathcal{S}, \quad s' = h(A), \tag{2}$$

where $\mathcal{S}$ denotes the space of feasible solutions.

4.2 Overview

MPaGE integrates LLMs into a multi-phase evolutionary framework to design diverse and effective heuristics, as illustrated in Figure 1. The process begins by initializing a population of heuristics tailored for SEMO paradigm. Each heuristic is represented in a feature space defined by solution quality and running time, and the population is progressively refined until the stopping criterion is met, yielding a set of non-dominated heuristics. At each iteration, PFG generates grids that partition the objective space, thereby grouping heuristics into distinct regions (Section 4.3). A pool of elitism heuristics is selected from these grids, and an

LLM-based review analyzes the logical semantics of candidate heuristics to cluster them into behaviorally similar groups (Section 4.4). Subsequently, search operators such as crossover and mutation are applied to these clusters, guided by informative feedback from LLM-based reflection, to generate new offspring. These offspring are added to the population, and non-dominated individuals are selected to form the next generation for the subsequent iteration (Section 4.5).

4.3 Pareto Front Grid Mechanism

Dominance-based approaches like NSGA-II (Deb et al. 2002) and decomposition-based methods such as MOEA/D (Zhang and Li 2007b) often struggle to balance convergence and diversity in tracking the true Pareto Fronts, particularly in heuristic design scenarios where objectives like runtime and solution quality exhibit irregular or non-uniform distributions. To address these limitations, we adopt the Pareto Front Grid (PFG) approach (Xu et al. 2023a) to achieve a better balance between convergence and diversity. Guiding the search using leading solutions in PFG helps focus on promising regions, improving solution quality and efficiency while reducing redundancy.

PFG Generation: Motivated by (Xu et al. 2023a), let the objective space be denoted by $\mathcal{Z} \subset \mathbb{R}^2$. At generation t, let $\mathcal{P}^{(t)} = \{E(h_i) \in \mathbb{R}^2 \mid h_i \in \mathcal{H}^{(t)}\}$ be the set of objective vectors of the current population, where each h_i is a heuristic algorithm and $\mathcal{H}^{(t)}$ denotes the heuristic space. We define the *ideal* and *nadir* points for each objective $j \in \{1, 2\}$ as $z_j^* = \min_{h \in \mathcal{P}^{(t)}} e_j(h)$ and $z_j^n = \max_{h \in \mathcal{P}^{(t)}} e_j(h)$, where $e_j(h)$ denotes the j-th objective value of heuristic h. The objective space is scaled via min-max normalization:

$$\tilde{E} = \left(\frac{e_1 - z_1^*}{z_1^n - z_1^*}, \frac{e_2 - z_2^*}{z_2^n - z_2^*} \right)^\top . \tag{3}$$

To structure the population in the normalized objective space, we partition $[0, 1]^2$ into a grid of cells of side length $\delta_1 > 0$, $\delta_2 > 0$ along each objective axis, respectively, depending on the number of segments along each dimension. For each solution h_i, assign its objective vector $\tilde{E}(h_i)$ to a grid cell $G(h_i) \in \mathbb{N}^2$, and define the PFG mapping as:

$$G(h_i) = \left(\left\lfloor \frac{e_1(h_i)}{\delta_1} \right\rfloor, \left\lfloor \frac{e_2(h_i)}{\delta_2} \right\rfloor \right) \in \mathbb{N}^2. \tag{4}$$

This discretization forms a structured grid over the objective space, grouping solutions into distinct regions. We define the following mapping:

$$\mathcal{G} : \mathbb{N}^2 \to 2^{|\mathcal{H}|}, \quad \mathcal{G}(g) = \left\{ h_i \in \mathcal{H}^{(t)} \mid G(h_i) = g \right\}. \tag{5}$$

where each cell g contains all solutions whose objective vectors fall within that grid region. For each non-empty cell $g \in \mathrm{dom}(\mathcal{G})$, retain a representative subset $\mathcal{R}_g \subset \mathcal{G}(g)$, selected using non-dominated sorting within the cell. The union of all representatives yields the elite set:

$$\mathcal{E}^{(t)} = \bigcup_g \mathcal{R}_g. \tag{6}$$

PFG Selection: To facilitate reproduction, the population is organized into a grid, where heuristics exhibiting similar running time or solution quality are placed in neighboring cells. This spatial organization encourages crossover between heuristics with related characteristics, promoting the inheritance and refinement of useful traits.

Selection for mating is performed over a pool $\mathcal{P}$ formed using the grid structure of the objective space. With probability ϵ, a group of elitism candidates is selected from a set of adjacent grid cells. Specifically, a grid cell g is randomly sampled, and the pool is formed as the union of $\mathcal{G}(g)$ and its neighboring cells $\mathcal{G}(g')$, where g' denotes the cells adjacent to g along both objective axes, as shown in Figure 2. Otherwise, the parents are selected from entire $\mathcal{E}^{(t)}$. Formally:

$$\mathcal{P} = \begin{cases} \mathcal{G}(g) \cup \mathcal{G}(g'), & \text{if } \mathcal{U}[0, 1] < \epsilon, \\ \mathcal{E}^{(t)}, & \text{otherwise}, \end{cases} \tag{7}$$

This hybrid strategy balances local exploitation and global exploration, promoting diversity while guiding search toward the Pareto front. See Appendix C for details.

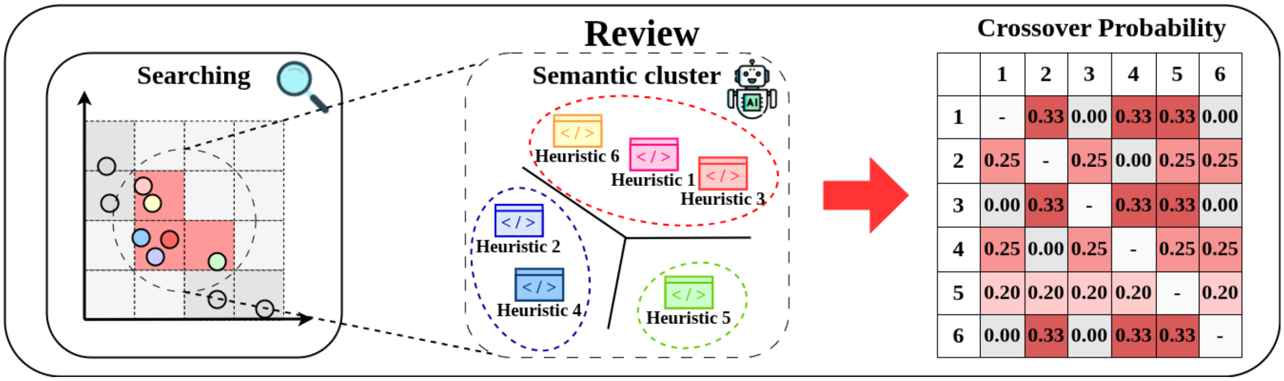
4.4 Semantic Clustering

Previous methods struggle to maintain effective diversity in multi-objective heuristic design, especially for complex MOCOP. The standard MOEAs rely on Pareto dominance in the objective space $\mathbf{f}(x)$, which is unreliable due to the stochastic nature of the heuristic performance. MEoH addresses this by introducing a dominance-dissimilarity measure based on Abstract Syntax Trees (ASTs) (Yao, Chen, and Wang 2025; Neamtiu, Foster, and Hicks 2005). However, ASTs fail to capture semantic similarity between heuristics with similar logic but different implementations, resulting in redundant individuals and reduced diversity. To address these challenges, instead of relying solely on syntactic features, we query LLMs to assess the semantic and behavioral similarity among elite heuristics $P = \{h_1, h_2, \ldots, h_n\}$, and group them into coherent clusters as depicted in Figure 2:

$$\begin{aligned} &\text{SemClust}(P) = \{C_1, C_2, \ldots, C_m\} \\ &\text{s.t.} \quad \bigcup_{i=1}^{m} C_i = P, \\ &\qquad C_i \cap C_j = \emptyset \quad (i \neq j). \end{aligned} \tag{8}$$

Each cluster C_i contains heuristics whose code segments implement similar underlying logic, despite syntactic or structural differences. This semantic clustering mitigates redundancy and enhances behavioral diversity at a higher abstraction level. During variation, we apply mutation within clusters to explore local variations, and perform crossover across clusters to combine diverse behaviors. Specifically, given a randomly selected cluster C_i and a heuristic $h \in C_i$, the offspring heuristic o is generated as follows:

$$o \leftarrow \begin{cases} \text{Mutate}(h), & \text{if } \mathcal{U}[0, 1] < \gamma, \\ \text{Crossover}(h, h'), & \text{otherwise}, \quad h' \sim \mathcal{U}\left(\bigcup_{k \neq i} C_k \right), \end{cases} \tag{9}$$



	1	2	3	4	5	6
1	-	0.33	0.00	0.33	0.33	0.00
2	0.25	-	0.25	0.00	0.25	0.25
3	0.00	0.33	-	0.33	0.33	0.00
4	0.25	0.00	0.25	-	0.25	0.25
5	0.20	0.20	0.20	0.20	-	0.20
6	0.00	0.33	0.00	0.33	0.33	-

Figure 2: LLM-based review clustering applies elitism heuristics based on semantic similarity and employs a probability matrix to guide variation among clusters.

4.5 Automatic Heuristic Generation

In this section, we propose the MPaGE framework for automatic heuristic design. Examples of the prompts used at each stage are provided in the Appendix E.

Individual Representation. Following previous works (Liu et al. 2024; Ye et al. 2024), MPaGE encodes each individual as a natural language description and its corresponding code implementation in Python, both generated by LLMs, together with an associated fitness score. The fitness is evaluated over a set of problem-specific instances by measuring both performance and running time.

Population initialization. MPaGE initializes the heuristic population by querying LLMs with prompts that describe the problem and specify the signature of the heuristic function to be discovered. Specifically, each heuristic is tailored for selection mechanisms and neighborhood exploration within the SEMO paradigm, then evaluated on benchmark instances and assigned a fitness score.

Selection. MPaGE partitions the objective space into grid cells to construct an elitism pool (Section 4.3), then clusters the candidates and selects parent pairs based on dissimilarity as detailed in Section 4.4, or sample from the entire population as Equation 7.

Feedback reflection. Feedback reflection enhances heuristic design by providing LLMs with clear, interpretable signals that guide improvement. For each pair of heuristic parents, an LLM-based reflection module analyzes their respective strengths and weaknesses, returning a textual suggestion on how to improve or effectively combine them.

Crossover and Mutation. MPaGE prompts the generator LLM to create an offspring heuristic by recombining or modifying parent elements, guided by reflective feedback. The prompt includes: (i) task specifications, (ii) parent heuristics in code and natural language, (iii) reflection guidance, and (iv) instructions to generate the new heuristic.

Population Management. Offspring are merged into the population, with non-dominated individuals forming the next generation, as defined by Equation 6. The process repeats until the maximum number of iterations is reached.

5 Experiments

5.1 Experimental Setup

Benchmarks. We evaluate the proposed MPaGE framework on four widely recognized MOCOP that are extensively investigated in the literature: Bi-TSP, Tri-TSP, Bi-CVRP, and Bi-KP. Comprehensive descriptions of these problems are provided in Appendix A.

Experiment settings. The generated heuristics are evaluated on 10 instances for each problem, with sizes of 20, 20, 50, and 50 for Bi-TSP, Tri-TSP, Bi-CVRP and Bi-KP, following the setup in (Chen et al. 2023a). The experimental parameters are configured as follows: the number of generations is set to 20, and the population size is 10 for all problems. Each crossover operator selects two parent heuristics to generate offspring heuristics. All heuristics are designed and evaluated under the SEMO search paradigm, with a runtime constraint of 2000 iterations and a time limit of 60 seconds. The probabilities $\epsilon = 0.9$, $\gamma = 0.3$, and the number of PFG segments is set to 4. Experiments were conducted using an Intel Core i7 (11th Gen) processor. `GPT-4o-mini` (temperature 0.7) was used as the pre-trained LLM, while `GPT-4o` was used for assessing and clustering task. We refer to Appendix J for the hyperparameter study.

Performance Metric.

Objectives To evaluate a heuristic, we consider two objectives, both averaged across instances and minimized simultaneously: 1) *Negative Hypervolume (NHV)*: Minimizing the negative hypervolume guides the search toward high-quality solutions. 2) *Running Time*: Measures the execution time of the heuristic to encourage efficiency.

Metrics Solution quality is evaluated using the hypervolume (HV) and Inverted Generational Distance (IGD). HV indicates how well the Pareto front is approximated (higher is better), while IGD measures proximity and distribution relative to a reference front. Specifically, when comparing LLM-based methods, HV captures overall performance, as each heuristic is evaluated by NHV and execution time; for MOEAs, it reflects final solution quality. In term of diversity measurement, we use the Shannon-Wiener Diversity Index

(SWDI) and Cumulative Diversity Index (CDI) (Dat, Doan, and Binh 2025). Details are provided in Appendix D.

Baseline Methods For LLM-based automated heuristic design, we compare our method against representative approaches: EoH, MEoH, ReEvo and HSEvo (Liu et al. 2024; Yao, Chen, and Wang 2025; Ye et al. 2024; Dat, Doan, and Binh 2025) and additionally against widely adopted MOEAs, including NSGA-II, MOEA/D, SEMO and PFG-MOEA (Deb et al. 2002; Li and Zhang 2008; Xu et al. 2023b; Laumanns, Thiele, and Zitzler 2004). See Appendix B for the full experimental setup and descriptions.

5.2 Experimental Results

Pareto Fronts and Convergence Analysis As illustrated in Table 1, MPaGE consistently outperforms other LLM-based baselines in both convergence and Pareto front quality across all benchmarks. On Bi-TSP20 and Tri-TSP20, MPaGE achieves the highest HV scores of 0.911 and 0.936, and the lowest IGD scores of 0.010 and 0.050, respectively, indicating faster convergence and superior solution quality. Similar trends are observed on Bi-KP50 and Bi-CVRP50, where MPaGE maintains strong performance.

Method	Bi-TSP20		Tri-TSP20		Bi-CVRP50		Bi-KP50	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
EoH	0.756	0.117	0.755	0.148	0.957	0.538	0.602	0.363
ReEvo	0.541	0.435	0.694	0.547	0.658	0.462	**0.996**	0.138
HSEvo	0.557	0.329	0.715	0.308	0.626	0.450	0.730	0.182
MEoH	0.724	0.067	0.884	0.114	0.322	0.286	0.748	0.248
MPaGE (Ours)	**0.911**	**0.010**	**0.936**	**0.050**	**0.980**	**0.007**	0.932	**0.035**

Table 1: Results of MPaGE compared to other baselines regarding HV and IGD on four benchmark problems.

MPaGE consistently yields a broader and more diverse set of Pareto-optimal solutions, spanning wider regions of the objective space, converges faster and clearly outperforms other baselines in terms of HV and IGD. In contrast, EoH shows slower HV growth and higher IGD due to its narrow focus on minimizing performance gaps alone. MEoH, while enhancing diversity over EoH, still trails MPaGE in convergence and final quality, likely due to the complexity of MOCOP tasks and the limited impact of its diversity mechanism.

Method	Bi-TSP20		Tri-TSP20		Bi-CVRP50		Bi-KP50	
	HV ↑	Time ↓	HV ↑	Time ↓	HV ↑	Time ↓	HV ↑	Time ↓
MEoH (best)	0.603	3.265	0.410	5.790	0.241	3.383	0.344	4.139
MEoH (fast)	0.345	**0.177**	0.403	**5.228**	0.141	0.081	0.349	2.246
MPaGE (best)	**0.629**	4.429	**0.478**	8.112	**0.454**	0.191	**0.359**	2.647
MPaGE (fast)	0.451	1.304	0.411	7.590	0.151	**0.063**	0.354	**1.367**

Table 2: Two top heuristics designed by MEoH and MPaGE.

We evaluate the two best-so-far heuristics across multiple benchmark problems (Table 2), each in two configurations: "best" (highest hypervolume) and "fast" (lowest runtime), measured at the final population. Notably, MPaGE discovers heuristics that outperform MEoH in HV. In several cases, its fast variant is quicker and more effective, highlighting MPaGE's strength in balancing quality and efficiency. We evaluate generalization performance in Appendix H.

Impact of Reflection and Heuristic Diversity We conducted experiments on the Bi-TSP20 benchmark to assess the effectiveness of our LLM Reflection approach. The results, summarized in and Table 4a, demonstrate the superior performance of MPaGE over baseline methods (EoH and MEoH) in both HV and IGD metrics. Remarkably, even without the reflection-based feedback, MPaGE outperforms all baselines, indicating that it inherently benefits from capturing the correlation among objectives within each grid. When enhanced with the reflection feedbacks, MPaGE exhibits further improvements, achieving the highest HV and the lowest IGD. These findings underscore the strength of our approach in promoting both convergence and diversity. To investigate the impact of our approach on population diversity, we evaluate all frameworks using the SWDI and CDI metrics, as depicted in Table 4b. Notably, MPaGE demonstrates significantly better performance compared to the other baselines. Higher SWDI and CDI values indicate more uniform heuristic distribution and greater population diversity, promoting effective exploration.

Evaluation Against Baselines We evaluate the best heuristics generated by MPaGE based on HV performance, against baselines on standard MOCOP benchmarks. As shown in Table 3, MPaGE consistently outperforms existing LLM-based heuristics, achieving highest HV on 9 out of 12 test suites and up to 100× speedup. Compared to traditional MOEAs, it delivers comparable or better HV on over half of the problems while being up to 14.6× faster. Although MEoH also offers strong runtime, MPaGE achieves a more balanced trade-off, maintaining high HV even on large instances. For example, on Bi-TSP100 and Bi-CVRP100, it reaches HV of 0.442 and 0.422 while being 46.6× and 9.5× faster than NSGA-II. Overall, MPaGE offers a robust set of heuristics balancing optimality and efficiency, making it highly suitable for large-scale combinatorial optimization. Comparisons with neural methods for MOCOP are provided in Appendix I.

5.3 Ablation Study

Clustering Method We evaluate our semantic clustering method (MPaGE) against three diverse baselines: SWDI-based embeddings (Dat, Doan, and Binh 2025), performance-centric K-Means, and structural AST similarity (Yao, Chen, and Wang 2025). These baselines span distributional representations, raw performance outcomes, and syntactic structures, enabling a comprehensive assessment of our model's semantic reasoning. Quantitative results are reported in Table 5, with implementation details provided in Appendix F.

Performance Impact of PFG We evaluate the impact of PFG on the optimization process and compare its performance with two well-established MOEAs: NSGA-II (Deb

Method	Bi-TSP20			Bi-TSP50			Bi-TSP100			Tri-TSP20			Tri-TSP50			Tri-TSP100		
	HV ↑	Speedup	Time ↓	HV ↑	Speedup	Time ↓	HV ↑	Speedup	Time ↓	HV ↑	Speedup	Time ↓	HV ↑ ↑	Speedup	Time ↓	HV ↑	Speedup	Time ↓
NSGA-II	0.603	1.0x	731.2	0.495	1.0x	875.5	0.419	1.0x	932.5	0.451	3.9x	252.446	0.279	1.7x	669.9	0.195	1.5x	830.3
MOEA/D	0.597	5.8x	126.644	0.473	5.5x	157.796	0.414	4.0x	240.680	0.365	4.5x	216.485	0.189	3.6x	310.583	0.123	3.1x	398.387
PFG-MOEA	0.624	1.2x	617.734	0.532	1.0x	844.914	0.457	1.0x	952.186	0.465	1.0x	984.180	0.289	1.0x	1127.46	0.222	1.0x	1229.857
SEMO	0.543	149.4x	4.894	0.284	80.5x	10.878	0.178	56.5x	16.846	0.295	143.0x	6.881	0.108	**84.8x**	13.290	0.065	33.2x	36.994
EoH	0.584	52.2x	14.004	0.510	18.5x	47.233	0.503	6.0x	159.156	0.420	108.7x	9.050	0.208	39.7x	28.368	0.107	29.3x	42.038
ReEvo	0.623	107.3x	6.812	0.516	86.2x	10.151	0.413	51.0x	18.653	0.427	95.0x	10.361	0.205	41.1x	27.416	0.103	14.8x	83.213
HSEvo	0.620	112.3x	6.513	0.519	95.3x	9.182	0.422	54.1x	17.590	0.420	115.6x	8.515	0.218	55.2x	20.435	0.115	20.3x	60.659
MEoH (best)	0.603	**224.0x**	3.265	0.480	**126.2x**	6.937	0.395	**79.2x**	12.024	0.410	**170.0x**	5.790	0.205	79.4x	14.197	0.111	**48.8x**	25.213
MPaGE (best)	0.629	165.1x	4.429	0.542	126.0x	6.950	0.442	74.9x	12.719	0.478	121.3x	8.112	0.220	76.5x	14.747	0.109	46.6x	26.394

Method	Bi-KP50			Bi-KP100			Bi-KP200			Bi-CVRP20			Bi-CVRP50			Bi-CVRP100		
	HV ↑	Speedup	Time ↓	HV ↑	Speedup	Time ↓	HV ↑	Speedup	Time ↓	HV ↑	Speedup	Time ↓	HV ↑	Speedup	Time ↓	HV ↑	Speedup	Time ↓
NSGA-II	0.357	1.0×	926.406	0.483	1.0×	995.178	0.293	1.0×	1176.915	0.573	1.0×	1051.192	0.416	1.0×	1945.667	0.357	1.0×	3433.284
MOEA/D	0.358	10.7×	86.247	0.484	13.9×	71.674	0.286	13.4×	87.621	0.568	16.2x	64.698	0.420	23.5x	82.801	0.353	35.7x	96.200
PFG-MOEA	0.358	1.9x	488.617	0.484	1.7x	596.989	0.330	2.1x	570.174	0.598	6.6x	159.867	0.435	7.0x	277.464	0.318	9.6x	359.108
SEMO	0.195	136.9x	6.766	0.144	529.4x	1.880	0.188	115.0x	10.234	0.518	8342.8x	0.126	0.205	1616.0x	1.204	0.135	944.8x	3.634
EoH	0.343	4.9x	187.455	0.463	5.1x	195.414	0.320	18.5x	63.555	0.539	6332.5x	0.166	0.443	6176.7x	0.315	0.369	5731.7x	0.599
ReEvo	0.357	**368.1x**	2.517	0.452	432.1x	2.303	0.202	550.2x	2.139	0.511	3822.5x	0.275	0.354	2231.3x	0.872	0.249	2257.3x	1.521
HSEvo	0.357	330.5x	2.803	0.450	318.3x	3.127	0.205	456.5x	2.578	0.515	3185.4x	0.330	0.405	1781.7x	1.092	0.305	916.0x	3.748
MEoH (best)	0.344	223.8x	4.139	0.464	219.1x	4.543	0.321	275.3x	4.275	0.503	2280.2x	0.461	0.241	575.1x	3.383	0.126	279.3x	12.291
MPaGE (best)	0.359	350.0x	2.647	0.486	**535.3x**	1.859	0.197	**555.7x**	2.118	0.568	**14599.9x**	0.072	0.454	**10186.7x**	0.191	0.422	**9563.5x**	0.359

Table 3: Comparison results across all benchmarks against baselines. The HV and time are averaged over 50 instances.

Method	**HV ↑**	**IGD ↓**
EoH	0.688	0.141
MEoH	0.659	0.122
MPaGE w/o Feedback	0.829	0.077
MPaGE	**0.941**	**0.023**

(a) Effects of LLM Reflection (Bi-TSP20)

Problems	**Bi-TSP**		**Bi-CVRP**	
	SWDI ↑	CDI ↑	SWDI ↑	CDI ↑
EoH	0.897	1.944	1.168	2.173
ReEvo	0.647	2.133	0.943	1.908
HSEvo	0.757	1.915	1.102	1.964
MEoH	0.639	2.086	0.143	2.181
MPaGE	**1.029**	**2.152**	**1.172**	**2.213**

(b) Comparison in Diversity Indices (SWDI and CDI)

Table 4: Analysis of LLM Reflection and Diversity Indices.

Method	**Bi-TSP20**		**Tri-TSP20**	
	HV ↑	IGD ↓	HV ↑	IGD ↓
SWDI Cluster	0.911	0.024	0.888	0.098
K-Means	0.876	0.030	0.815	0.184
AST Similarity	0.745	0.184	0.757	0.203
MPaGE (Ours)	**0.921**	**0.013**	**0.892**	0.102

Table 5: Comparison on Bi-TSP20 and Tri-TSP20 instances.

et al. 2002) and MOEA/D (Zhang and Li 2007a). As shown in Table 6, based on experiments conducted on Bi-TSP problems, MPaGE consistently outperforms the baselines. By directing the search toward the most promising regions of the objective space, PFG enhances both the solution quality and overall efficiency. Appendix J offers a more comprehensive examination of PFG performance.

Backbone	**Bi-TSP 20**		**Bi-TSP 50**		**Bi-TSP 100**	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
NSGA-II	0.860	0.052	0.801	0.120	0.757	**0.095**
MOEA/D	0.819	0.119	0.768	0.108	0.560	0.157
PFG (Ours)	**0.913**	**0.024**	**0.836**	**0.075**	**0.844**	0.099

Table 6: Efficiency of our Pareto Front Grid

6 Conclusion

In this paper, we propose MPaGE, an LLM-guided framework for solving MOCOP that discovers a Pareto front of heuristics balancing solution quality, runtime, and semantic diversity. By combining LLMs with the SEMO paradigm and a Pareto Front Grid, the method partitions the objective space and guides heuristic evolution toward promising regions, while semantic clustering preserves meaningful diversity. Empirically, MPaGE outperforms prior LLM-based methods in multi-objective trade-offs, diversity, and efficiency, and matches or surpasses traditional algorithms in solution quality while running more efficiently, highlighting scalability and generality for automated heuristic discovery.

Acknowledgments

This research was funded by Hanoi University of Science and Technology under project code T2024-PC-038.

References

Chen, J.; Wang, J.; Zhang, Z.; Cao, Z.; Ye, T.; and Chen, S. 2023a. Efficient meta neural heuristic for multi-objective combinatorial optimization. *Advances in Neural Information Processing Systems*, 36: 56825–56837.

Chen, J.; Zhang, Z.; Cao, Z.; Wu, Y.; Ma, Y.; Ye, T.; and Wang, J. 2023b. Neural Multi-Objective Combinatorial Optimization with Diversity Enhancement. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Dat, P. V. T.; Doan, L.; and Binh, H. T. T. 2025. HSEvo: Elevating Automatic Heuristic Design with Diversity-Driven Harmony Search and Genetic Algorithm Using LLMs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 26931–26938. AAAI Press.

Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2): 182–197.

Fan, M.; Wu, Y.; Cao, Z.; Song, W.; Sartoretti, G.; Liu, H.; and Wu, G. 2024. Conditional Neural Heuristic for Multi-objective Vehicle Routing Problems. *IEEE transactions on neural networks and learning systems*, PP.

Forniés-Tabuenca, D.; Uribe, A.; Otamendi, U.; Artetxe, A.; Rivera, J. C.; and de Lacalle, O. L. 2025. REMoH: A Reflective Evolution of Multi-objective Heuristics approach via Large Language Models. *arXiv preprint arXiv:2506.07759*.

Gutjahr, W. J. 2012. Runtime analysis of an evolutionary algorithm for stochastic multi-objective combinatorial optimization. *Evolutionary computation*, 20(3): 395–421.

Hieu, H. M.; Phan, H.; Tran, D. C.; Van, D. C.; Tung, D. V.; and Binh, H. T. T. 2024. Alimentation Deep Multiple Optimal Ant Colony Optimization to solve Vehicle Routing Problem with Time Windows. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '24 Companion, 17–18. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704956.

Huang, L.; Zhang, M.; and Liu, K. 2025. Autonomous Design of Evolutionary Operators for Multi-Objective Optimization Using Large Language Models. *Artificial Intelligence Journal*, 300: 50–65.

Huang, Y.; Lv, X.; Wu, S.; Wu, J.; Feng, L.; and Tan, K. C. 2024. Advancing automated knowledge transfer in evolutionary multitasking via large language models. *arXiv preprint arXiv:2409.04270*.

Laumanns, M.; Thiele, L.; and Zitzler, E. 2004. Running time analysis of multiobjective evolutionary algorithms on pseudo-boolean functions. *IEEE Transactions on Evolutionary Computation*, 8(2): 170–182.

Li, H.; and Zhang, Q. 2008. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE transactions on evolutionary computation*, 13(2): 284–302.

Lin, X.; Yang, Z.; and Zhang, Q. 2022. Pareto Set Learning for Neural Multi-Objective Combinatorial Optimization. In *International Conference on Learning Representations*.

Liu, F.; Lin, X.; Yao, S.; Wang, Z.; Tong, X.; Yuan, M.; and Zhang, Q. 2025. Large language model for multiobjective evolutionary optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, 178–191. Springer.

Liu, F.; Tong, X.; Yuan, M.; Lin, X.; Luo, F.; Wang, Z.; Lu, Z.; and Zhang, Q. 2024. Evolution of Heuristics: Towards Efficient Automatic Algorithm Design Using Large Language Model. In *International Conference on Machine Learning (ICML)*.

Liu, F.; Tong, X.; Yuan, M.; and Zhang, Q. 2023. Algorithm evolution using large language model. *arXiv preprint arXiv:2311.15249*.

Liu, Q.; Li, X.; Liu, H.; and Guo, Z. 2020. Multi-objective metaheuristics for discrete optimization problems: A review of the state-of-the-art. *Applied Soft Computing*, 93: 106382.

Neamtiu, I.; Foster, J. S.; and Hicks, M. 2005. Understanding source code evolution using abstract syntax tree matching. In *Proceedings of the 2005 international workshop on Mining software repositories*, 1–5.

Novikov, A.; Vũ, N.; Eisenberger, M.; Dupont, E.; Huang, P.-S.; Wagner, A. Z.; Shirobokov, S.; Kozlovskii, B.; Ruiz, F. J.; Mehrabian, A.; et al. 2025. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*.

Paquete, L.; and Stützle, T. 2004. Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study. In *Metaheuristics for multiobjective optimisation*, 177–199. Springer.

Phan Duc, H.; Bui Trong, D.; Nguyen Thi, T.; and Huynh Thi Thanh, B. 2025. Pareto Front Grid Guided Multiobjective Optimization In Dynamic Pickup And Delivery Problem Considering Two-Sided Fairness. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '25, 277–285. New York, NY, USA: Association for Computing Machinery. ISBN 9798400714658.

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. 2024a. Mathematical discoveries from program search with large language models. *Nature*, 625(7995): 468–475.

Romera-Paredes, B.; Barekatain, M.; Novikov, A.; Balog, M.; et al. 2024b. Mathematical discoveries from program search with large language models. *Nature*.

Tran, C. D.; Nguyen-Tri, Q.; Binh, H. T. T.; and Thanh-Tung, H. 2025. Large Language Models powered Neural Solvers for Generalized Vehicle Routing Problems. In *Towards Agentic AI for Science: Hypothesis Generation, Comprehension, Quantification, and Validation*.

Türkyılmaz, A.; Şenvar, Ö.; Ünal, I.; and Bulkan, S. 2020. A research survey: heuristic approaches for solving multi objective flexible job shop problems. *Journal of Intelligent Manufacturing*, 31(8): 1949–1983.

van Stein, N.; and Bäck, T. 2024. Llamea: A large language model evolutionary algorithm for automatically generating metaheuristics. *IEEE Transactions on Evolutionary Computation*.

Wu, X.; Wu, S.-h.; Wu, J.; Feng, L.; and Tan, K. C. 2024. Evolutionary computation in the era of large language model: Survey and roadmap. *IEEE Transactions on Evolutionary Computation*.

Xu, Y.; Zhang, H.; Huang, L.; Qu, R.; and Nojima, Y. 2023a. A Pareto Front grid guided multi-objective evolutionary algorithm. *Applied Soft Computing*, 136: 110095.

Xu, Y.; Zhang, H.; Huang, L.; Qu, R.; and Nojima, Y. 2023b. A Pareto Front grid guided multi-objective evolutionary algorithm. *Applied Soft Computing*, 136: 110095.

Yao, M.; Chen, Y.; and Wang, L. 2025. MEoH: Multi-objective Evolution of Heuristics with Large Language Models. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*, 1300–1307. AAAI Press.

Ye, H.; Wang, J.; Cao, Z.; Berto, F.; Hua, C.; Kim, H.; Park, J.; and Song, G. 2024. Reevo: Large language models as hyper-heuristics with reflective evolution. *arXiv preprint arXiv:2402.01145*.

Zhang, Q.; and Li, H. 2007a. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6): 712–731.

Zhang, Q.; and Li, H. 2007b. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6): 712–731.

Zhang, Z.; Wu, Z.; Zhang, H.; and Wang, J. 2022. Meta-learning-based deep reinforcement learning for multiobjective optimization problems. *IEEE Transactions on Neural Networks and Learning Systems*, 34(10): 7978–7991.

Zheng, Z.; Xie, Z.; Wang, Z.; and Hooi, B. 2025. Monte Carlo Tree Search for Comprehensive Exploration in LLM-Based Automatic Heuristic Design. *arXiv preprint arXiv:2501.08603*.

Zitzler, E.; and Thiele, L. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4): 257–271.