HSEvo: Elevating Automatic Heuristic Design with Diversity-Driven Harmony Search and Genetic Algorithm Using LLMs

Pham Vu Tuan Dat[1], **Long Doan**[2], **Huynh Thi Thanh Binh**[1]

[1]Hanoi University of Science and Technology, Hanoi, Viet Nam
[2]George Mason University, Virginia, United States
dat.pvt210158@sis.hust.edu.vn, ldoan5@gmu.edu, binhht@soict.hust.edu.vn

Abstract

Automatic Heuristic Design (AHD) is an active research area due to its utility in solving complex search and NP-hard combinatorial optimization problems in the real world. The recent advancements in Large Language Models (LLMs) introduce new possibilities by coupling LLMs with evolutionary computation to automatically generate heuristics, known as LLM-based Evolutionary Program Search (LLM-EPS). While previous LLM-EPS studies obtained great performance on various tasks, there is still a gap in understanding the properties of heuristic search spaces and achieving a balance between exploration and exploitation, which is a critical factor in large heuristic search spaces. In this study, we address this gap by proposing two diversity measurement metrics and perform an analysis on previous LLM-EPS approaches, including FunSearch, EoH, and ReEvo. Results on black-box AHD problems reveal that while EoH demonstrates higher diversity than FunSearch and ReEvo, its objective score is unstable. Conversely, ReEvo's reflection mechanism yields good objective scores but fails to optimize diversity effectively. With this finding in mind, we introduce HSEvo, an adaptive LLM-EPS framework that maintains a balance between diversity and convergence with a harmony search algorithm. Through experimentation, we find that HSEvo achieved high diversity indices and good objective scores while remaining cost-effective. These results underscore the importance of balancing exploration and exploitation and understanding heuristic search spaces in designing frameworks in LLM-EPS.

Code — https://github.com/datphamvn/HSEvo
Extended version — https://arxiv.org/pdf/2412.14995

1 Introduction

Heuristics are widely utilized to address complex search and NP-hard combinatorial optimization problems (COPs) in real-world systems. Over the past few decades, significant efforts have been made to develop efficient heuristics, resulting in the creation of many meta-heuristic methods such as simulated annealing, tabu search, and iterated local search. These meticulously crafted methods have been effectively applied across various practical systems.

Nevertheless, varied applications with distinct constraints and goals may necessitate different algorithms or algorithm setups. The manual creation, adjustment, and configuration of a heuristic for a specific problem can be extremely laborious and requires substantial expert knowledge. This is a bottleneck in many application domains. To address this issue, Automatic Heuristic Design (AHD) has been proposed aims to selects, tunes, or constructs effective heuristics for a given problem class automatically (Choong, Wong, and Lim 2018). Various approaches have been used in AHD, including Hyper-Heuristics (HHs) (Pillay and Qu 2018) and Neural Combinatorial Optimization (NCO) (Qu, Kendall, and Pillay 2020). However, HHs are still constrained by heuristic spaces that are predefined by human experts. Additionally, NCO faces limitations related to the necessity for effective inductive bias (Drakulic et al. 2024), and challenges regarding interpretability and generalizability (Liu et al. 2023).

Recently, the rise of Large Language Models (LLMs) has opened up new possibilities for AHD. It is believed that LLMs (Nejjar et al. 2023; Austin et al. 2021) could be a powerful tool for generating new ideas and heuristics. However, standalone LLMs with prompt engineering can be insufficient for producing novel and useful ideas beyond existing knowledge (Mahowald et al. 2024). Some attempts have been made to coupling LLMs with evolutionary computation to automatically generate heuristics, known as LLM-based Evolutionary Program Search (LLM-EPS) (Liu et al. 2024b; Meyerson et al. 2024; Chen, Dohan, and So 2024). Initial works such as FunSearch (Romera-Paredes et al. 2024) and subsequent developments like Evolution of Heuristic (EoH) (Liu et al. 2024a) and Reflective Evolution (ReEvo) (Ye et al. 2024a) have demonstrated significant improvements over previous approaches, generating quality heuristics that often surpass current methods. Even so, ReEvo yields state-of-the-art and competitive results compared with evolutionary algorithms, neural-enhanced meta-heuristics, and neural solvers.

A key difference between LLM-EPS and classic AHD lies in their heuristic search spaces. Classic AHD typically operates in well-defined mathematical spaces such as R^n, while LLM-EPS searches within the space of functions, where each function represents a heuristic as a program. LLM-EPS utilizes LLMs within an evolutionary framework to enhance the quality of generated functions iteratively. Therefore, it is crucial to study and understand the search spaces of heuristics to establish foundational theories and principles for the

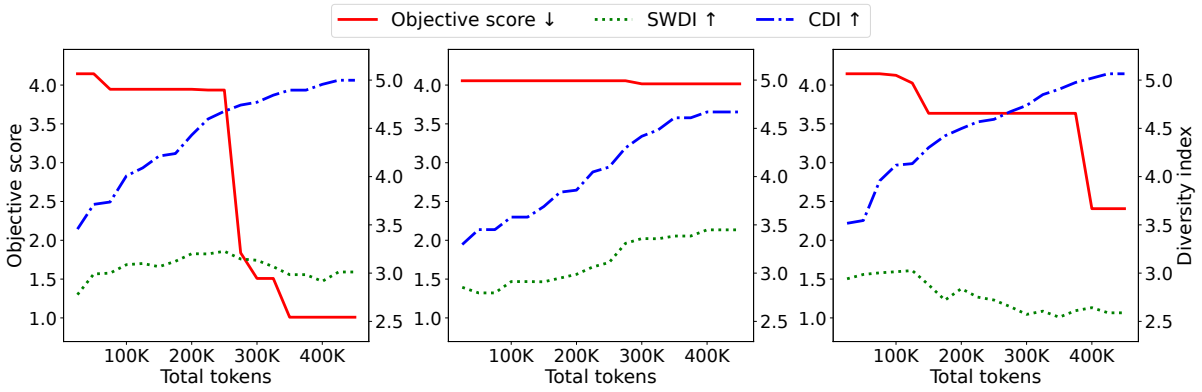

Figure 1: Diversity indices and objective scores of ReEvo framework on BPO problem through different runs.

automatic design of algorithms. This aspect has been largely unconsidered in previous LLM-EPS frameworks.

In this paper, we introduce two diversity metrics inspired by the Shannon entropy to monitor population diversity. We also explore relationships between our proposed diversity metrics with objective performance of LLM-EPS frameworks on different optimization problems. Through our analysis, we gain an understanding on the characteristics and properties of heuristic search spaces in LLM-EPS frameworks. Finally, building on these foundational principles, we propose a new framework called **HSEvo**, which incorporates a new components based on harmony search algorithm (Shi, Han, and Si 2012) and enhances other components, including initialization, crossover, and mutation, to optimize objective scores and diversity metrics.

In summary, our contributions are as follows:

- Two diversity measurement metrics, the Shannon–Wiener Diversity Index and the Cumulative Diversity Index, are used to evaluate the evolutionary progress of populations within the LLM-EPS framework.
- A novel framework, HSEvo, that aims to balance between the diversity and objective performance to improve the optimization process.

2 Background and Related Works

2.1 LLM-Based Evolutionary Program Search

Recent advances in LLM-EPS have shown promising results in AHD. Evolutionary methods have been adopted in both code generation (Nejjar et al. 2023; Ma et al. 2023; Hemberg, Moskal, and O'Reilly 2024) and text generation (Guo et al. 2023; Yuksekgonul et al. 2024). Notable among these methods are FunSearch, EoH, and ReEvo. FunSearch employs an island-based evolution strategy, leveraging LLMs like Codey and StarCoder to evolve heuristics for mathematical and COPs, outperforming traditional methods on tasks such as the cap set and admissible set problems. EoH utilizes genetic algorithm with Chain of Thought prompt engineering, consistently achieving superior performance in the traveling salesman and online bin packing problems. ReEvo introduces a reflective component to the evolution process, employing two instances of GPT-3.5 to generate and refine heuristics, which has demonstrated effectiveness across various optimization tasks. These methods highlight the potential of integrating LLMs with evolutionary strategies to enhance the efficiency and effectiveness of AHD solutions.

2.2 Diversity in Evolutionary Computation

Diversity plays a pivotal role in enhancing the efficiency of algorithms in multi-objective optimization within the domain of evolutionary computation (Solteiro Pires, Tenreiro Machado, and de Moura Oliveira 2014). Numerous studies have investigated various methods to measure and maintain diversity within populations, as it critically impacts the convergence and overall performance of these algorithms (Pires, Tenreiro Machado, and Moura Oliveira 2019; Wang and Chen 2012). Among these, the application of Shannon entropy has been particularly prominent in quantifying diversity and predicting the behavior of genetic algorithms. However, to the best of our knowledge, no existing studies have thoroughly explored diversity within the context of LLM-EPS. This gap in the literature motivates our in-depth exploration of diversity in LLM-EPS frameworks.

3 Diversity Measurement Metrics

In this section, we introduce a method to encode LLM-EPS population and propose two diversity metrics, Shannon-Wiener diversity index (SWDI) and cummulative diversity index (CDI). We also conduct a diversity analysis on previous LLM-EPS frameworks, FunSearch, EoH and ReEvo.

3.1 Population Encoding

One particular problem of measuring diversity in LLM-EPS is how to encode the population. While each individual in traditional evolutionary algorithm is encoded as a vector, in LLM-EPS they are usually presented as a string of code snippet/program. This poses a challenge in applying previous diversity metrics on population of LLM-EPS frameworks. To tackle this issue, we suggest an encoding approach consists of three steps: (i) removing comments and docstrings using abstract-syntax tree, (ii) standardizing code

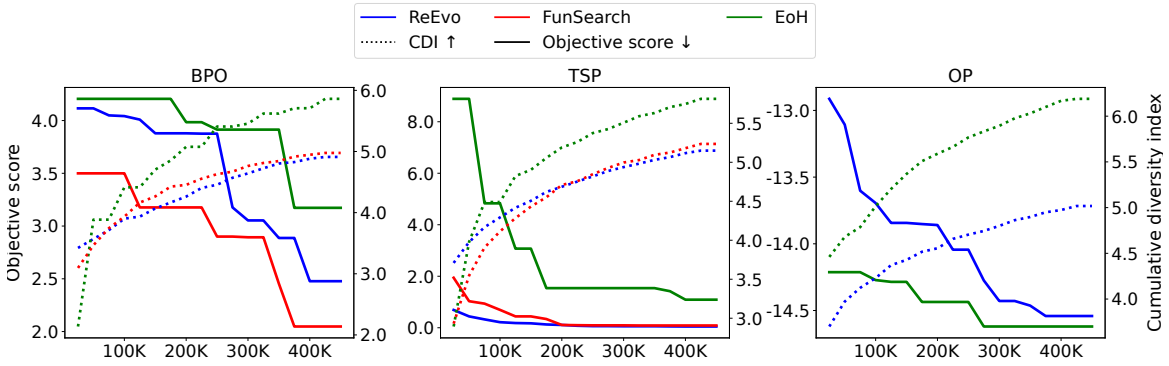

Figure 2: CDI and objective scores of previous LLM-EPS on different AHD problems.

snippets into a common coding style (e.g., PEP8[1]), (iii) convert code snippets to vector representations using a code embedding model.

3.2 Shannon–Wiener Diversity Index

Inspired by ecological studies, SWDI (Nolan and Callahan 2006) provides a quantitative measure of species diversity within a community. In the context of search algorithms, this index aims to quantify the diversity of the population at any given time step based on clusters of individuals. To compute SWDI for a given set of individuals within a community, or archive, first, we need to determine the probability distribution of individuals across clusters. This is represented as:

$$p_i = \frac{|C_i|}{M} \tag{1}$$

where C_i is a cluster of individuals and M represents total number of individuals across all clusters $\{C_1, \ldots, C_N\}$. The SWDI, $H(X)$, is then calculated using Shannon entropy:

$$H(X) = -\sum_{i=1}^{N} p_i \log(p_i) \tag{2}$$

This index serves as a crucial metric in maintaining the balance between exploration and exploitation within heuristic search spaces. A higher index score suggests a more uniform distribution of individuals across the search space, thus promoting exploration. Conversely, a lower index score indicates concentration around specific regions, which may enhance exploitation but also increase the risk of premature convergence.

To obtain SWDI score, at each time step t, we add all individuals $R_i = \{r_1, \ldots, r_n\}$ generated at time step t to an archive. We encode the archive using our proposed population encoding in Section 3.1 to obtain their vector representations $\mathcal{V} = \{v_1, \ldots, v_n\}$. After that, we compute the similarity between two embedding vectors v_i and v_j using cosine similarity:

$$\text{similarity}(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\| \|v_j\|}$$

[1]https://peps.python.org/pep-0008

To find clusters in the archive, we consider each embedding vector v_i. We assign v_i to a cluster C_i if the similarity between v_i and all members in C_i is greater than a threshold α. Mathematically, v_i is assigned to C_i if

$$\text{similarity}(v_i, v_k) \geq \alpha \quad \forall k \in \{1, \ldots, |C_i|\}$$

If no cluster $C_i \in \{C_1, \ldots, C_N\}$ can satisfy the above condition, we create a new cluster C_{N+1} and assign v_i to C_{N+1}. Finally, we compute SWDI score by computing Eq. (1), (2) across found clusters.

3.3 Cumulative Diversity Index

While SWDI focuses on diversity of different groups of individuals, the Cumulative Diversity Index (CDI) plays a crucial role in understanding the spread and distribution of the whole population within the search space (Jost 2006). In the context of heuristic search, the CDI measures how well a system's energy, or diversity, is distributed from a centralized state to a more dispersed configuration.

To calculate the CDI, we also consider all individuals within an archive, represented by their respective embeddings. We construct a minimum spanning tree (MST) that connects all individuals within the archive A, where each connection between individuals is calculated using Euclidean distance. This MST provides a structure to assess the diversity of the population. Let d_i represent the distance of an edge within the MST, where $i \in \{1, 2, \ldots, \#A-1\}$. The probability distribution of these distances is given by:

$$p_i = \frac{d_i}{\sum_{j=1}^{\#A-1} d_j}$$

The cumulative diversity is then calculated using the Shannon entropy:

$$H(X) = -\sum_{i=1}^{\#A-1} p_i \log(p_i)$$

This approach allows us to capture the overall diversity of the search space, providing insights into the spread of solutions. Higher CDI values indicate a more distributed and diverse population, which is essential for maintaining a robust search process.

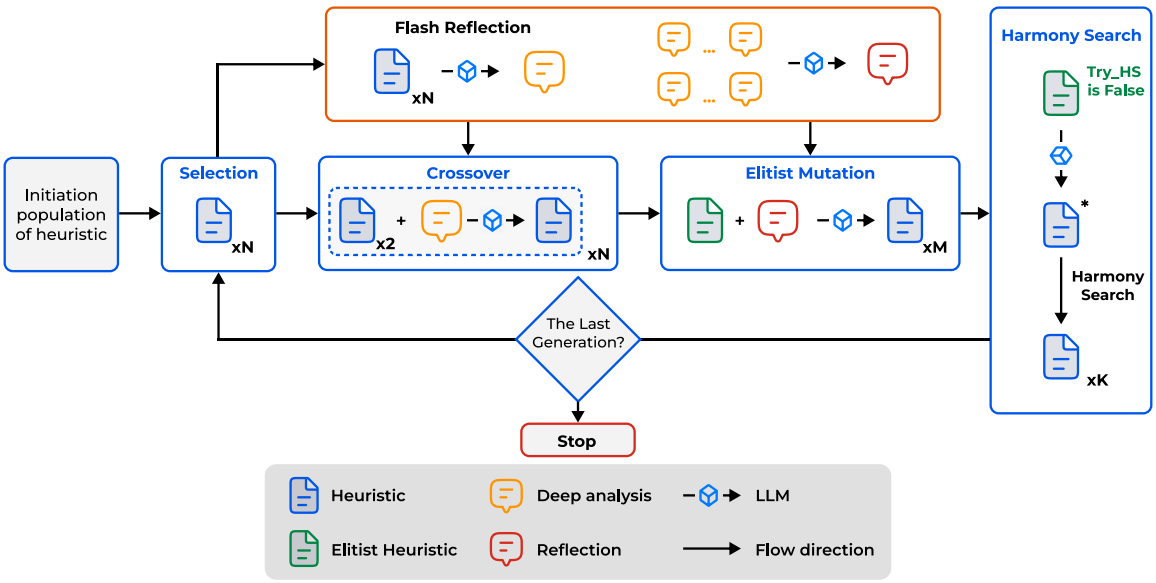

Figure 3: Overview of the HSEvo framework.

An interesting point in Shannon diversity index (SDI) theory is that normalizing the SDI with the natural log of richness is equivalent to the evenness value $\frac{H'}{\ln(S)}$ where H' represents the richness of SDI and S is the total number of possible categories (Heip et al. 1998). The significance of evenness is to demonstrate how the proportions of p_i are distributed. Now, consider normalizing with the two current diversity indices. First, with SWID, if we have N clusters and the number of individuals in each cluster from C_1 to C_N is equal, then evenness will always be 1, regardless of the value of N. This is not the desired behavior since we consider the number of clusters to be a component of diversity. Similarly, normalizing CDI will also ignore the impact of the number of candidate heuristics, which correspond to the nodes in the MST. This leads to the proposal not to normalize the SWID and CDI metrics in order to achieve a $[0, 1]$ bound.

3.4 Exploring the Correlation Between Objective Score and Diversity Measurement Metrics

To examine the correlation between the two diversity measurement metrics and objective score, we conducted three experimental runs using ReEvo on bin-packing online problem (Seiden 2002). The experiment details can be found in the Extended version - Appendix. The objective scores and the two diversity metrics from these runs are presented in Fig. 1. Additional results for other tasks are also available in the Extended version - Appendix.

From the figure, there are a few observations that we can drawn. First, the two diversity measurement metrics have a noticeable correlation on the objective scores of the problem. When the objective score is converged into a local optima, the framework either try to focus on the exploration by increasing the diversity of the population, through the increase of SWDI in first and second run, or try to focus on the exploitation of current population, thus decrease the SWDI in the third run. We can also see that focusing on exploration can lead to significant improvement on the objective score, while focusing on exploitation can make the population stuck in local optima for a longer time. However, if the whole population is not diverse enough, as can be seen with the low CDI of the second run, the population might not be able to escape the local optima.

3.5 Diversity Analysis on Previous LLM-EPS Framework

To analyse the overall diversity of previous LLM-EPS frameworks, we conduct experiments on three different LLM-EPS frameworks, including FunSearch (Romera-Paredes et al. 2024), ReEvo (Ye et al. 2024a), and EoH (Liu et al. 2024a), on three distinct AHD problems, bin-packing online (BPO), traveling salesmen problem with guided local search solver (TSP) (Voudouris and Tsang 1999), and orienteering problem with ACO solver (OP) (Ye et al. 2024b). The details of each experiment are presented in the Extended version - Appendix. Note that, as highlighted in the previous section, SWDI focuses on understanding the diversity of the population during a single run. As such, it may not be helpful for quantifying the diversity across multiple experiment runs. Fig. 2 presents the experiment results.

In BPO and TSP, EoH obtain the highest CDI but got the worst objective score. This implies that EoH does not focus enough on exploitation to optimize the population better. In contrast, while ReEvo and FunSearch obtain lower CDI than EoH on BPO and TSP, they achieve a better objective performance on all three problems. The experiments on BPO and TSP problems highlight the inherent trade-off between diversity and objective performance of LLM-EPS frameworks. However, on OP problem, we can see that a high CDI is required to obtain a better objective score, which align with our findings in Section 3.4.

```python
import numpy as np

def heuristics_v2(prize: np.ndarray,
distance: np.ndarray, maxlen: float) -> np.
    ndarray:
    reward_distance_ratio = prize / distance
    cost_penalty = np.exp(-distance)
    heuristics = (prize * prize[:, np.
        newaxis])
    heuristics[distance > maxlen] = 0

    return heuristics
```

(a) Origin Code

```python
import numpy as np

def heuristics_v2(prize: np.ndarray,
distance: np.ndarray, maxlen: float,
reward_threshold: float = 0,
distance_threshold: float = 0,
cost_penalty_weight: float = 1) -> np.ndarray:
    reward_distance_ratio = prize / distance
    cost_penalty = np.exp(-distance)
    heuristics = (prize * prize[:, np.newaxis]) / (distance *
        distance) * cost_penalty
    heuristics[(distance > maxlen) | (reward_distance_ratio <
        reward_threshold) | (distance < distance_threshold)] = 0

    return heuristics

parameter_ranges = {
    'reward_threshold': (0, 1),
    'distance_threshold': (0, 100),
    'cost_penalty_weight': (0, 2)
}
```

(b) Modified Code using LLMs

Figure 4: An example of how Harmony search component works in HSEvo.

Description of Setting	**Value**
LLM (generator and reflector)	gpt-4o-mini-2024-07-18
LLM temperature (generator and reflector)	1
Maximum budget tokens	425K tokens
Population size (for EoH, ReEvo and HSEvo)	30 (initial stage), 10 (other stages)
Mutation rate (for ReEvo and HSEvo)	0.5
# of islands, # of samples per prompt (for FunSearch)	10, 4
Number of independent runs per experiment	3
HS size, HMCR, PAR, bandwidth, max iterations (for Harmony Search)	5, 0.7, 0.5, 0.2, 5
Maximum evaluation time for each heuristic	100 sec (TSP-GLS), 50 sec (Other)

Table 1: Summary of parameters settings.

4 Automatic Heuristic Design with HSEvo

In this section, we propose a novel LLM-EPS framework called **H**armony **S**earch **Evo**lution (HSEvo). HSEvo aim to promote the diversity of the population while still achieve better optimization and alleviate the trade-off between diversity and optimization performance with a individual tuning process based on harmony search. HSEvo also aim to reduce the cost incurred from LLM with an efficient flash reflection component. Figure 3 illustrates the pipeline of our HSEvo framework. Examples of prompts used in each stage can be found in the Extended version - Appendix.

Individual encoding. Follow previous works in LLM-EPS (EoH (Liu et al. 2024a), ReEvo (Ye et al. 2024a)), HSEvo encodes each individual as a string of code snippet generated by LLMs (Fig. 4a). This encoding method allow the flexibility of each individual, i.e., not constrained by any predefined encoding format, and also make it easier to evaluate on the AHD problem.

Initialization. HSEvo initializes a heuristic population by prompting the generator LLM with task specifications that describe the problem and detail the signature of the heuristic function to be searched. Additionally, to leverage existing knowledge, the framework can be initialized with a seed heuristic function and/or external domain knowledge. We promote diversity by implementing in the prompts with various role instructions (e.g., "You are an expert in the domain of optimization heuristics...", "You are Albert Einstein, developer of relativity theory..."). This approach aims to enrich the heuristic generation process by incorporating diverse perspectives and expertise.

Selection. HSEvo randomly selects parent pairs from the population, aiming to maintain the balance between exploration and exploitation during our optimization process. The random selection may also counteract premature convergence, which is observed through the SWID trajectory outlined in Section 3.4.

Flash reflection. Reflections can provide LLMs with reinforcement learning reward signals in a verbal format for code generation tasks, as discussed by (Shinn et al. 2024). Later, (Ye et al. 2024a) also proposed integrating Reflections into LLM-EPS in ReEvo as an approach analogous to providing "verbal gradient" information within heuristic spaces. However, we argue that reflecting on each pair of heuristic

Method	BPO		TSP		OP	
	CDI (↑)	Obj. (↓)	CDI (↑)	Obj. (↓)	CDI (↑)	Obj. (↓)
FunSearch	4.97 ± 0.24	2.05 ± 2.01	5.24 ± 0.14	0.09 ± 0.06	-	-
EoH	$\mathbf{5.86 \pm 0.49}$	3.17 ± 2.97	$\mathbf{5.81 \pm 0.23}$	1.09 ± 3.11	$\mathbf{6.17 \pm 0.42}$	-14.62 ± 0.22
ReEvo	4.91 ± 0.53	2.48 ± 3.74	5.15 ± 0.19	0.05 ± 0.06	5.02 ± 0.13	-14.54 ± 0.21
HSEvo (ours)	5.68 ± 0.35	$\mathbf{1.07 \pm 1.11}$	5.41 ± 0.21	$\mathbf{0.02 \pm 0.03}$	5.67 ± 0.41	$\mathbf{-14.62 \pm 0.12}$

Table 2: Experiment results on different AHD problems.

Method	OP	
	CDI (↑)	Obj. (↓)
ReEvo	5.02 ± 0.13	-14.54 ± 0.21
ReEvo + HS	5.11 ± 0.27	-14.58 ± 0.38
HSEvo (ours)	$\mathbf{5.67 \pm 0.41}$	$\mathbf{-14.62 \pm 0.12}$

Table 3: Ablation results on harmony search.

Method	OP	
	CDI (↑)	Obj. (↓)
ReEvo	4.43 ± 0.23	-13.84 ± 1.13
ReEvo + F.R.	4.63 ± 0.37	$\mathbf{-14.36 \pm 0.19}$
HSEvo (ours)	$\mathbf{4.77 \pm 0.46}$	-14.07 ± 0.38

Table 4: Ablation results on flash reflection.

parents individually is not generalized and wastes resources. To address these issues flash reflection proposed to alternative Reflection method for LLM-EPS. Flash reflection includes two steps as follows:

- At time step t, we perform grouping on all individuals that are selected from the selection stage and remove all duplication. After that, we use LLM to perform a deep analysis on the ranking on parent pairs. This take inputs as mixed ranking pairs (i.e., one good performance parent and one worse performance), good ranking pairs (i.e., both good performance), and worse ranking pairs (i.e., both worse performance), then return a comprehensive description on how to improve as a text string.
- From the current analysis at time step t, we use LLM to compare with previous analysis at time step $t-1$ and output a guide information, which can be used in subsequent stages.

Crossover. In this stage, new offspring algorithms are generated by combining elements of the parent algorithms. The goal is to blend successful attributes from two parents via guide result guide information part of flash reflections. Through this step, HSEvo hopes to produce offspring that inherit the strengths of both, which can potentially lead to better-performing heuristics. The prompt includes task specifications, a pair of parent heuristics, guide information part of flash reflection, and generation instructions.

Elitist mutation. HSEvo uses an elitist mutation strategy, where the generator LLM is tasked with mutation an elite individual—representing the best-performing heuristic—by incorporating insights derived from LLM-analyzed flash reflections. Each mutation prompt includes detailed task specifications, the elite heuristic, deep analysis part of flash reflections, and specific generation instructions. This approach leverages accumulated knowledge to enhance the heuristic, ensuring continuous improvement in performance while preserving the quality of the top solutions.

Harmony Search. From the analysis on Section 3.4 and 3.5, we hypothesize that if the population is too diverse, each individuals inside will more likely to be not optimized, which may cause harm to the optimization process. We employ the Harmony Search algorithm to alleviate this issue by optimizing the parameters (e.g., thresholds, weights, etc.) of best individuals in the population. The process is as follows:

- First, we use LLM to extract parameters from the best individual (i.e., code snippets, programs) of the population and define a range for each parameters (Fig. 4).
- Following the work (Shi, Han, and Si 2012), we optimize the above parameters with harmony search algorithm.
- After parameter optimization, we mark this individual and add it back to the population. All marked individuals will not be optimized again in the future time steps.

Fine-tuning these parameters makes the individual more optimized, therefore allowing us to encourage diversity during the prompting while avoid the trade-off between objective performance and diversity, as can be seen with EoH.

5 Experiments

5.1 Experimental Setup

Benchmarks: To assess the diversity and objective scores of HSEvo compared to previous LLM-EPS frameworks, we adopt the same benchmarks as Section 3.4 and conduct experiments on three different AHD problems, BPO (Seiden 2002), TSP (Hoffman et al. 2013) and OP (Ye et al. 2024b).

- BPO: packing items into bins of fixed capacity in real-time without prior knowledge of future items. In this benchmark, LLM-EPS frameworks need to design deterministic constructive heuristics to solve.
- TSP: find the shortest possible route that visits each city exactly once and returns to the starting point. In this setting, LLM-EPS frameworks need to design heuristics to enhance the perturbation phase for the Guided Local Search solver.
- OP: find the most efficient path through a set of locations, maximizing the total score collected within a limited time or distance. This setting requires LLM-EPS frameworks to design herustics used by the ACO solver.

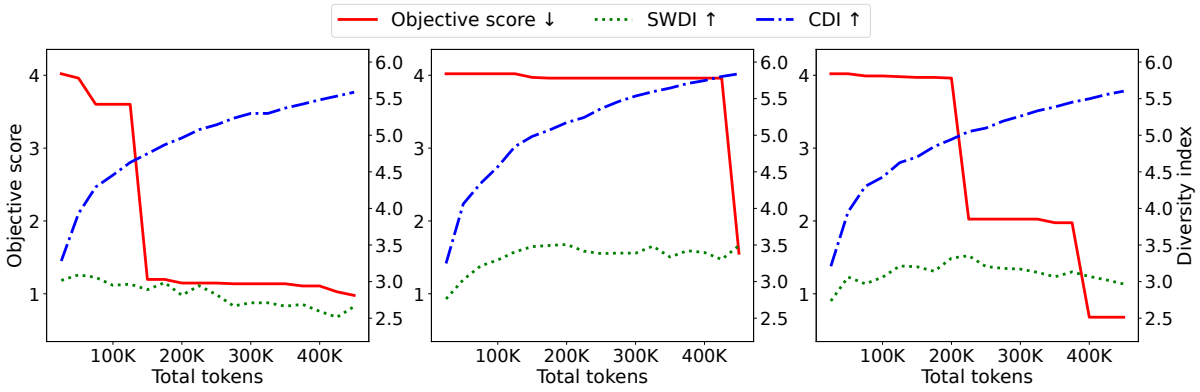

Figure 5: Diversity indices and objective scores of HSEvo framework on BPO problem through different runs.

Experiment settings: All frameworks were executed under identical environmental settings listed in Table 1. The heuristic search processes across the LLM-EPS frameworks and the evaluations of heuristics were conducted using a single core of an Xeon Processors CPU.

5.2 Experiment Results

Table 2 presents our experiment results on all three AHD problems[2]. From the table, we observe that while our framework still haven't obtained better CDI than EoH, HSEvo is able to achieve the best objective score on all tasks. On BPO, HSEvo outperforms both FunSearch and ReEvo by a huge margin. This highlights the important of our findings from the analysis in Section 3.5, where it is crucial to improve diversity in order to optimize the population better.

To investigate the impact of our framework on the diversity of the population, we plot the diversity metrics and objective score of HSEvo through different runs on BPO problem in Fig. 5. We can draw a similar observation with findings in Section 3.4, where with a high SWDI and CDI, the objective score can be optimized significantly. One thing to note here is that in HSEvo first run and ReEvo third run in Fig. 1, both have the SWDI decreasing overtime. However, in HSEvo, the SWDI is at around 3.0 when the objective score improve significantly, then only decrease marginally after that. In ReEvo, the objective score improves when SWDI at around 3.0 and 2.7, and the magnitude of the improvement is not as large as HSEvo, which implies the important of diversity in the optimization of the problem and also the impact of our proposed harmony search.

5.3 Ablation Study

To gain a better understanding on our novel framework, we conduct ablation studies on our proposed components, the harmony search and flash reflection.

Harmony search analysis As harmony search is a vital component in our HSEvo framework, instead of removing it from HSEvo, we conduct an experiment where we add harmony search into ReEvo framework. Table 3 presents our experiment results on OP problem. From the results, we can see that HSEvo outperforms both ReEvo and ReEvo with harmony search variant on both objective score and CDI. Here, notice that harmony search can only marginally improve ReEvo. This can be explained that ReEvo does not have any mechanism to promote diversity, therefore it is not benefitted from the advantage of harmony search process.

Flash reflection analysis We also conduct another experiment where we replace the reflection used in ReEvo with our flash reflection component. As our flash reflection is more cost efficient than original reflection, we reduce the number of tokens used for optimization to 150K. Table 4 presents our experiment results on OP problem. The results show that given a smaller number of timestep, ReEvo with flash reflection mechanism can outperform HSEvo in optimization performance while obtain comparable results on CDI. However, when running with a larger number of tokens, ReEvo with flash reflection cannot improve on both objective score and CDI, while HSEvo improve both metrics to 5.67 and -14.62, respectively. This implies that without diversity-promoting mechanism, flash reflection is not enough to improve the optimization process of LLM-EPS.

6 Conclusion

In this paper, we highlight the importance of population diversity in LLM-EPS for AHD problems. We propose two diversity measure metrics, SWDI and CDI, and conduct an analysis on the diversity of previous LLM-EPS approaches. We find that previous approaches either lack focus on the diversity of the population or suffered heavily from the diversity and optimization performance trade-off. We also introduce HSEvo, a novel LLM-EPS framework with diversity-driver harmony search and genetic algorithm for AHD problems. Our experiment results show that our framework can maintain a good balance between diversity and optimization performance trade-off. We also perform additional ablation studies to verify the effectiveness of components of our proposed framework. We hope our work can benefit future research in LLM-EPS community.

[2]Extending FunSearch to solve the OP problem with an ACO solver caused conflicts we could not resolve with reasonable effort.

Acknowledgments

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

References

Austin, J.; Odena, A.; Nye, M.; Bosma, M.; Michalewski, H.; Dohan, D.; Jiang, E.; Cai, C.; Terry, M.; Le, Q.; et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Chen, A.; Dohan, D.; and So, D. 2024. EvoPrompting: language models for code-level neural architecture search. *Advances in Neural Information Processing Systems*, 36.

Choong, S. S.; Wong, L.-P.; and Lim, C. P. 2018. Automatic design of hyper-heuristic based on reinforcement learning. *Information Sciences*, 436: 89–107.

Drakulic, D.; Michel, S.; Mai, F.; Sors, A.; and Andreoli, J.-M. 2024. Bq-nco: Bisimulation quotienting for efficient neural combinatorial optimization. *Advances in Neural Information Processing Systems*, 36.

Guo, Q.; Wang, R.; Guo, J.; Li, B.; Song, K.; Tan, X.; Liu, G.; Bian, J.; and Yang, Y. 2023. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*.

Heip, C. H.; Herman, P. M.; Soetaert, K.; et al. 1998. Indices of diversity and evenness. *Oceanis*, 24(4): 61–88.

Hemberg, E.; Moskal, S.; and O'Reilly, U.-M. 2024. Evolving code with a large language model. *Genetic Programming and Evolvable Machines*, 25(2): 21.

Hoffman, K. L.; Padberg, M.; Rinaldi, G.; et al. 2013. Traveling salesman problem. *Encyclopedia of operations research and management science*, 1: 1573–1578.

Jost, L. 2006. Entropy and diversity. *Oikos*, 113(2): 363–375.

Liu, F.; Xialiang, T.; Yuan, M.; Lin, X.; Luo, F.; Wang, Z.; Lu, Z.; and Zhang, Q. 2024a. Evolution of Heuristics: Towards Efficient Automatic Algorithm Design Using Large Language Model. In *Forty-first International Conference on Machine Learning*.

Liu, S.; Chen, C.; Qu, X.; Tang, K.; and Ong, Y.-S. 2024b. Large language models as evolutionary optimizers. In *2024 IEEE Congress on Evolutionary Computation (CEC)*, 1–8. IEEE.

Liu, S.; Zhang, Y.; Tang, K.; and Yao, X. 2023. How good is neural combinatorial optimization? A systematic evaluation on the traveling salesman problem. *IEEE Computational Intelligence Magazine*, 18(3): 14–28.

Ma, Y. J.; Liang, W.; Wang, G.; Huang, D.-A.; Bastani, O.; Jayaraman, D.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2023. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*.

Mahowald, K.; Ivanova, A. A.; Blank, I. A.; Kanwisher, N.; Tenenbaum, J. B.; and Fedorenko, E. 2024. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*.

Meyerson, E.; Nelson, M. J.; Bradley, H.; Gaier, A.; Moradi, A.; Hoover, A. K.; and Lehman, J. 2024. Language model crossover: Variation through few-shot prompting. *ACM Transactions on Evolutionary Learning*, 4(4): 1–40.

Nejjar, M.; Zacharias, L.; Stiehle, F.; and Weber, I. 2023. Llms for science: Usage for code generation and data analysis. *Journal of Software: Evolution and Process*, e2723.

Nolan, K.; and Callahan, J. 2006. Beachcomber Biology: The Shannon-Weiner Species Diversity Index. *Proc. Workshop ABLE*, 27.

Pillay, N.; and Qu, R. 2018. *Hyper-heuristics: theory and applications*. Springer.

Pires, E.; Tenreiro Machado, J.; and Moura Oliveira, P. 2019. Dynamic Shannon Performance in a Multiobjective Particle Swarm Optimization. *Entropy*, 21: 827.

Qu, R.; Kendall, G.; and Pillay, N. 2020. The general combinatorial optimization problem: Towards automated algorithm design. *IEEE Computational Intelligence Magazine*, 15(2): 14–23.

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

Seiden, S. S. 2002. On the online bin packing problem. *Journal of the ACM (JACM)*, 49(5): 640–671.

Shi, W. W.; Han, W.; and Si, W. C. 2012. *A Hybrid Genetic Algorithm Based on Harmony Search and its Improving*, 101–109. Springer London. ISBN 9781447148029.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Solteiro Pires, E. J.; Tenreiro Machado, J. A.; and de Moura Oliveira, P. B. 2014. Diversity study of multiobjective genetic algorithm based on Shannon entropy. In *2014 Sixth World Congress on Nature and Biologically Inspired Computing (NaBIC 2014)*, 17–22.

Voudouris, C.; and Tsang, E. 1999. Guided local search and its application to the traveling salesman problem. *European journal of operational research*, 113(2): 469–499.

Wang, L.; and Chen, Y. 2012. Diversity Based on Entropy: A Novel Evaluation Criterion in Multi-objective Optimization Algorithm. *International Journal of Intelligent Systems and Applications*, 4(10): 113–124.

Ye, H.; Wang, J.; Cao, Z.; Berto, F.; Hua, C.; Kim, H.; Park, J.; and Song, G. 2024a. ReEvo: Large Language Models as Hyper-Heuristics with Reflective Evolution. In *Advances in Neural Information Processing Systems*. https://github.com/ai4co/reevo.

Ye, H.; Wang, J.; Cao, Z.; Liang, H.; and Li, Y. 2024b. DeepACO: neural-enhanced ant systems for combinatorial optimization. *Advances in Neural Information Processing Systems*, 36.

Yuksekgonul, M.; Bianchi, F.; Boen, J.; Liu, S.; Huang, Z.; Guestrin, C.; and Zou, J. 2024. TextGrad: Automatic" Differentiation" via Text. *arXiv preprint arXiv:2406.07496*.