Position Paper: Suggestions from Large Language Models about Experimental Settings for Benchmarking Evolutionary Multi-objective Optimization Algorithms

Lie Meng Pang
Department of Computer Science and Engineering, Southern University of Science and Technology,
Shenzhen 518055, China
panglm@sustech.edu.cn

Hisao Ishibuchi
Department of Computer Science and Engineering, Southern University of Science and Technology,
Shenzhen 518055, China
hisao@sustech.edu.cn

***Abstract*—Benchmarking is essential for the evaluation of evolutionary multi-objective optimization (EMO) algorithms, yet benchmarking results are sensitive to experimental settings such as test problems, compared algorithms, performance indicators, and their parameter specifications. Recently, large language models (LLMs) have been increasingly used in EMO research and academic activities (such as assisting with paper reviews). As a result, LLMs may influence how benchmarking studies are conducted, especially for newcomers to the EMO field. This position paper examines the reliability of benchmarking suggestions generated by two LLMs by comparing their responses with those reported in earlier studies at IEEE CEC 2025. Using the same prompts and experimental settings, we analyze similarities and differences in the suggested algorithms, test problems, and performance indicators for benchmarking EMO algorithms between the two LLMs and also between the two years. In addition, we report benchmarking suggestions from the LLMs for constrained multi-objective optimization. The observations indicate that, while many LLM-generated suggestions remain consistent with widely used benchmarking practices, variations and inaccuracies are also observed. These findings highlight the need for careful interpretation of LLM-generated suggestions and emphasize the importance of benchmarking guidelines when evaluating EMO algorithms.**

***Index Terms*—Benchmarking; evolutionary multi-objective optimization; experimental settings; large language models**

I. Introduction

In the last three decades, a large number of evolutionary multi-objective optimization (EMO) algorithms have been proposed in the literature [1]–[3]. When a new EMO algorithm is introduced, its performance is typically assessed through empirical comparisons with state-of-the-art EMO algorithms on multi-objective test problems. However, the reliable evaluation of EMO algorithms remains challenging [4]. This is because we need to evaluate solution sets for multi-objective test problems, which is inherently more difficult than evaluating individual solutions for single-objective test problems. In addition, the evaluation results depend on various factors, such as the choice of test problems, compared algorithms, performance indicators, and the parameter specifications of both the indicators and the algorithms. That is, the performance comparison results of EMO algorithms depend on the experimental settings used for benchmarking [4]. In recent years, increasing attention has been paid to benchmarking methodologies for EMO algorithms (see, for example, [5], [6]). The objective of these studies is to establish more rigorous and reliable evaluation procedures than those commonly used in existing empirical studies.

Recently, the use of large language models (LLMs) has become very popular, and they have been used in various application fields [7], [8]. In the EMO field, the use of LLMs has been explored in various directions, including offspring solution creation, variation operator design, use as optimizers, and EMO algorithm design [9]–[13]. In these studies, benchmarking EMO algorithms is explicitly or implicitly needed or assumed to evaluate generated offspring solutions, designed variation operators, and designed EMO algorithms. One controversial issue with respect to the use of LLMs is AI review. It seems that LLMs have already been directly and indirectly used for paper review. Many reviewers may use LLMs as their helpers for paper reviews at various levels, from checking the language quality of their own review comments to searching for state-of-the-art results in the literature and checking the mathematical correctness of proofs. One well-known example is AAAI 2026, which received more than 30,000 submissions and piloted an AI-assisted review system [14]. Another example is ICML 2026 [15], which adopted a two-policy framework for LLM use in reviewing: a conservative policy, where LLM use in reviewing is strictly prohibited, and a permissive policy, where LLMs may be used to help reviewers understand the paper but not to generate review suggestions. Each reviewer is asked to declare which policy they are willing to follow, and the authors of each paper can indicate which policy they require for the review of

This work was supported by National Natural Science Foundation of China (Grant No. 62376115), and Guangdong Provincial Key Laboratory (Grant No. 2020B121201001) (*Corresponding author: Hisao Ishibuchi*)

their paper. These examples indicate that the use of LLMs in the review process is becoming increasingly common. Another possible avenue is the following. When a newcomer (such as a student or a young researcher who has just joined the EMO field) enters the field, it is natural nowadays for them to turn to LLMs to obtain a quick guide on how to construct reliable experimental setups. One concern is the reliability of those suggestions about benchmarking, which is the main topic to be discussed in this position paper.

At IEEE CEC 2025, Pang and Ishibuchi [16] discussed the reliability of suggestions from LLMs on benchmarking EMO algorithms with respect to various issues, including the choice of EMO algorithms to be compared, parameter specifications for each EMO algorithm (e.g., population size), performance indicators, parameter specifications for each indicator (e.g., the HV reference point and the IGD reference point sets), and test problems. Their paper [16] concluded that the suggestions from LLMs were mainly based on historically common benchmarking settings. Although these settings had been used in many studies in the literature, they were not necessarily appropriate. That is, the overall conclusion in [16] was that LLMs' suggestions were not satisfactory since some important findings in recent studies on benchmarking EMO algorithms were not incorporated. For example, LLMs suggested the performance evaluation of EMO algorithms on ZDT, DTLZ and WFG using HV and IGD, but did not suggest more realistic test problems. LLMs also did not suggest the need for careful specification of the HV reference point and the IGD reference point set. It is therefore worth examining the progress of LLMs over the last year, since the CEC 2025 paper [16] concluded that LLMs' suggestions were not satisfactory, while LLMs have increasingly been utilized for paper reviews [14].

This paper has two goals. The first is to examine the progress of LLMs (i.e., changes in LLM suggestions) since CEC 2025 with respect to the experimental settings used for benchmarking EMO algorithms. The second is to examine suggestions from LLMs about the experimental settings for benchmarking constrained multi-objective optimization.

The organization of this paper is as follows. Section II reports the progress of LLMs over the past year. Section III examines the sensitivity of the wording in the prompts to the responses generated by the LLMs. Section IV reports benchmarking suggestions by the LLMs for constrained multi-objective optimization. Section V concludes the paper.

II. One year progress of LLMs

In order to examine the progress of knowledge embedded in LLMs over the past year, in this section, we follow the same experimental settings as in [16]. That is, we use the same prompts as in [16] to examine the responses of two LLMs provided by OpenAI and DeepSeek, which were the two LLMs used in [16]. OpenAI has released its newest flagship model, called GPT-5.2 [17]. Meanwhile, DeepSeek also released its newest model in December 2025, called DeepSeek-V3.2 [18]. Thus, we use these two models in this study and compare their responses with those reported in [16] for the two previous models, GPT-4o [19] and DeepSeek-V3 [20]. Part of the prompts is shown in Fig. 1. Each prompt is repeated 10 times to account for the randomness of the responses given by the LLMs.

A. General Suggestions

The first prompt requests the two LLMs to provide general suggestions for benchmarking EMO algorithms. We compare the responses provided by the two latest LLMs (i.e., GPT-5.2 and DeepSeek-V3.2) with the responses reported for their predecessors (i.e., GPT-4o and DeepSeek-V3) in [16]. The general suggestions about benchmarking EMO algorithms from the two LLMs appear professional and reasonable, and are similar to those reported in [16]. The responses generated by the LLMs are summarized as follows: the LLMs suggested (i) using diverse test problems by combining synthetic benchmarks with problem-specific or real-world problems; (ii) using standard performance indicators and avoiding reliance on a single indicator; (iii) evaluating performance across increasing numbers of objectives and decision variables; (iv) performing multiple independent runs and applying non-parametric statistical tests (e.g., Wilcoxon, Friedman with post-hoc analysis); (v) reporting reproducibility details by publishing code, random seeds, parameter settings, and reference Pareto fronts where applicable; (vi) including Pareto front plots and convergence curves to aid interpretation; (vii) ensuring fair experimental settings by comparing algorithms under equal numbers of function evaluations rather than generations; and (viii) reporting runtime and robustness, including wall-clock time, anytime behavior, and sensitivity to parameters.

B. Suggested EMO Algorithms

The second prompt asks the LLMs to suggest the number of algorithms that should be included in a benchmarking experiment. GPT-5.2 suggested no more than 10 algorithms, with 5 to 10 algorithms being a reasonable choice. DeepSeek-V3.2 provided similar responses, with most responses suggesting no more than 10 algorithms and some indicating a maximum of 15 if a comprehensive comparison is needed. Both LLMs explained that including too many algorithms often reduces clarity and complicates analysis without providing additional insight. The responses also suggested including representative algorithms from the three major EMO algorithm paradigms: decomposition-based, indicator-based, and Pareto-based algorithms. In general, the suggested number of EMO algorithms is similar to that reported in [16].

Next, the LLMs were asked to suggest five EMO algorithms for benchmarking. Table I shows the percentage of times each algorithm was suggested by the two LLMs. The numbers in parentheses show the increase (+) or decrease (-) in percentage relative to [16]. For example, SPEA2 [21] was suggested 7 times among the 10 responses provided by GPT-5.2. Since the corresponding percentage in [16] was 100%, the decrease is 30%, and thus 70 (-30) is assigned to SPEA2 for GPT-5.2.

Prompt:
You are an expert in evolutionary multi-objective optimization (EMO). What are your suggestions for benchmarking EMO algorithms? Keep your response short and concise.

Prompt:
You are an expert in evolutionary multi-objective optimization (EMO). How many EMO algorithms should be included in a benchmarking experiment? Keep your response short and concise.

Prompt:
You are an expert in evolutionary multi-objective optimization (EMO). Please suggest five EMO algorithms for benchmarking. Keep your response short and concise.

Prompt:
You are an expert in evolutionary multi-objective optimization (EMO). Please suggest five state-of-the-art EMO algorithms for benchmarking. Provide a short and concise response.

Prompt:
You are an expert in evolutionary multi-objective optimization (EMO). Please suggest test problems for benchmarking EMO algorithms. Keep your response short and concise.

Fig. 1. Part of the prompts used in [16]

TABLE I
PERCENTAGE (%) OF TIMES EACH ALGORITHM WAS SUGGESTED BY GPT-5.2 AND DEEPSEEK-V3.2 IN RESPONSE TO THE REQUEST FOR FIVE EMO ALGORITHMS (BASED ON 10 RESPONSES PER MODEL).

Algorithm (Year)	GPT-5.2 (%)	DeepSeek-V3.2 (%)
NSGA-II (2002)	100 (0)	100 (0)
MOEA/D (2007)	100 (0)	100 (0)
NSGA-III (2014)	100 (0)	100 (0)
SPEA2 (2001)	70 (-30)	70 (-30)
SMS-EMOA (2007)	60 (+60)	100 (+100)
IBEA (2004)	40 (+40)	0 (-40)
RVEA (2016)	30 (-30)	30 (+30)
HypE (2011)	0 (-20)	10 (-50)

Note: The percentages for DeepSeek-V3.2 do not sum to 500% because one response suggested more than five algorithms using "or." That is, one response listed: (1) NSGA-II, (2) MOEA/D, (3) NSGA-III, (4) SMS-EMOA, and (5) SPEA2 or HypE. In this case, both SPEA2 and HypE are counted.

TABLE II
PERCENTAGE (%) OF TIMES EACH ALGORITHM WAS SUGGESTED BY GPT-5.2 AND DEEPSEEK-V3.2 IN RESPONSE TO THE REQUEST FOR FIVE STATE-OF-THE-ART EMO ALGORITHMS (BASED ON 10 RESPONSES PER MODEL).

Algorithm (Year)	GPT-5.2 (%)	DeepSeek-V3.2 (%)
NSGA-III (2014)	100 (+20)	100 (0)
RVEA (2016)	100 (0)	100 (+60)
HypE (2011)	100 (+20)	0 (-100)
MOEA/D-DE (2009)	70 (+50)	30 (-40)
SMS-EMOA (2007)	60 (+30)	80 (+50)
MOEA/D (2007)	40 (+40)	10 (+10)
θ-DEA (2016)	30 (+30)	0 (0)
MOEA/DD (2015)	0 (0)	80 (+50)
LMEA (2018)	0 (0)	40 (0)
IM-MOEA (2015)	0 (0)	40 (+40)
AR-MOEA (2018)	0 (0)	20 (-70)
AGE-MOEA (2019)	0 (0)	20 (+20)

Note: The percentages for DeepSeek-V3.2 do not sum to 500% because some responses suggested more than five algorithms using "or."

Table I shows that NSGA-II [22], MOEA/D [23], and NSGA-III [24] are always included in the suggestions of the two LLMs, and that these three algorithms were also always included in the responses reported in [16]. Meanwhile, SMS-EMOA [25] (which was not included in the suggestions reported in [16]) is always suggested by DeepSeek-V3.2. It is also suggested by GPT-5.2 six times out of the 10 responses. Table I also shows that, other than NSGA-II, MOEA/D, and NSGA-III, the percentages of suggestions have changed from those reported in [16] (e.g., decreases for SPEA2 [21] and HypE [26], and increases for SMS-EMOA). This may indicate that the underlying EMO-related knowledge of the two LLMs has been updated over the past year through the new training data used for these models. However, the knowledge update in the two LLMs is not always consistent as shown by the percentage changes of suggestions for IBEA [27] and RVEA [28] in Table I.

Table II reports the percentage of times each algorithm was included in the responses provided by the LLMs when they were asked to suggest five state-of-the-art EMO algorithms for benchmarking. The variety of algorithms suggested by GPT-5.2 as state-of-the-art is smaller than that suggested by DeepSeek-V3.2. For the algorithms suggested by DeepSeek-V3.2, we sometimes observed incorrect explanations for some algorithms. For example, IM-MOEA [29], which is an inverse model-based algorithm, was described as an indicator-based algorithm that uses IGD/IGD$^+$ approximation for selection by DeepSeek-V3.2. It can also be observed in Table II that there are large differences (i.e., increases and decreases in percentages) from the results reported in [16]. This may indicate that the knowledge about the state-of-the-art EMO algorithms in each LLM has been largely updated.

C. Suggested Test Suites

Table III shows the percentage of times each problem suite was suggested by the two LLMs for benchmarking EMO algorithms. In general, the suggestions from GPT-5.2 are largely similar to the results reported in [16], except that the MaF [30] and CF [31] test suites are included more frequently. For DeepSeek-V3.2, the real-world problem suite (i.e., RE [32]) was suggested in 70% of the 10 responses. In addition, some recently proposed specialized test suites, such as LSMOP [33] (i.e., a large-scale problem suite) and IMOP [34] (i.e., problems with irregular Pareto fronts), are included in the responses. DeepSeek-V3.2 also suggested constrained benchmark problems, such as C-DTLZ [35] and CTP [36], in

TABLE III
PERCENTAGE (%) OF TIMES EACH PROBLEM SUITE WAS SUGGESTED BY GPT-5.2 AND DEEPSEEK-V3.2 IN RESPONSE TO THE REQUEST FOR SUGGESTING TEST SUITES FOR BENCHMARKING (BASED ON 10 RESPONSES PER MODEL).

Test suite (Year)	GPT-5.2 (%)	DeepSeek-V3.2 (%)
DTLZ (2002)	100 (0)	100 (0)
WFG (2006)	100 (0)	100 (0)
MaF (2017)	100 (+70)	100 (+100)
ZDT (2000)	100 (0)	20 (-80)
UF (2009)	100 (0)	0 (-100)
CF (2009)	100 (+50)	0 (0)
LSMOP (2017)	0 (0)	100 (+100)
CEC competitions	0 (-70)	60 (+60)
RE (2020)	0 (0)	70 (+70)
IMOP (2019)	0 (0)	40 (+40)
C-DTLZ (2014)	0 (0)	30 (+30)
CTP (2001)	0 (0)	30 (+30)

TABLE IV
PERCENTAGE (%) OF TIMES EACH PERFORMANCE INDICATOR WAS SUGGESTED BY GPT-5.2 AND DEEPSEEK-V3.2 IN RESPONSE TO THE REQUEST FOR SUGGESTING PERFORMANCE INDICATORS FOR BENCHMARKING (BASED ON 10 RESPONSES PER MODEL).

Performance indicator	GPT-5.2 (%)	DeepSeek-V3.2 (%)
HV	100 (0)	100 (0)
IGD	100 (0)	0 (-100)
IGD$^+$	100 (+100)	100 (+100)
R2	100 (+90)	0 (0)
ϵ-indicator	100 (+10)	0 (-100)
Spread	100 (+70)	100 (0)
Spacing	80 (0)	50 (+10)
Runtime	0 (-50)	30 (+30)
Number of FE	0 (0)	30 (+30)
GD	0 (-30)	10 (-50)

some of the responses. For GPT-5.2, the constrained test suite, i.e., CF, is also always included in the suggestions, whereas it was suggested by GPT-4o in [16] in only 50% of the responses. This may be because constrained multi-objective optimization has recently become a popular research topic in the EMO field.

The two LLMs were then asked to suggest only one test suite for benchmarking EMO algorithms. GPT-5.2 always suggested the DTLZ test suite [37] (consistent with the results reported in [16]), while DeepSeek-V3.2 always suggested the MaF test suite, which differs from DeepSeek-V3, for which the DTLZ test suite was always suggested in [16]. From this observation (and some other observations in this paper), it seems that the knowledge of DeepSeek-V3.2 is based on more recent publications than that of GPT-5.2.

D. Suggested Performance Indicators

Table IV shows the percentage of times each performance indicator was suggested by GPT-5.2 and DeepSeek-V3.2. It can be observed that HV [38] is always suggested by both LLMs, which is consistent with the results reported in [16]. For GPT-5.2, IGD [39], IGD$^+$ [40], R2 [41], ϵ-indicator [42], and Spread [22] are always included in the responses, whereas DeepSeek-V3.2 suggests a smaller set of indicators. In particular, IGD and ϵ-indicator are not suggested by DeepSeek-V3.2 (i.e., 0% in Table IV), whereas they were included in all responses reported in [16] (i.e., 100%). Both LLMs always suggest IGD$^+$, showing a 100% increase in frequency compared with [16]. Both LLMs also always suggest the use of multiple indicators. In particular, DeepSeek-V3.2 always suggests the use of HV and IGD$^+$ as an essential pair to assess the performance of EMO algorithms.

The two LLMs were then asked to suggest only a single performance indicator for benchmarking EMO algorithms. Both LLMs consistently suggested HV as the most appropriate single performance indicator. Since a reference point is required for calculating the HV value, the two LLMs were asked to explain how to specify a reference point for HV calculation. For GPT-5.2, there were inconsistent explanations of how to specify the reference point. In some responses, it suggested using a fixed, problem-dependent reference point. However, in some other responses, it suggested using a fixed, problem-independent reference point. In any case, GPT-5.2 typically suggested a point slightly worse than the nadir point of the true (or approximated) Pareto front. For DeepSeek-V3.2, the suggestion was to use a point slightly worse than the observed nadir point across all algorithm runs on a problem, although a slightly different procedure was explained in each of the 10 responses.

Next, the two LLMs were asked to suggest how to specify the reference point for HV calculation in a normalized objective space for two-, three-, five-, and ten-objective problems. For GPT-5.2, it explained that the general rule is to specify the reference point $\mathbf{r} = (1+\epsilon, 1+\epsilon, \ldots, 1+\epsilon)$, where $\epsilon \in [0.05, 0.2]$. It also noted that the population size does not affect the choice of the reference point. The recommended reference point was always $(1.1, \ldots, 1.1)$ for three-objective, five-objective, and ten-objective problems. For two-objective problems, GPT-5.2 recommended $(1.1, 1.1)$ in three responses and suggested $\mathbf{r} = (1+\epsilon, \ldots, 1+\epsilon)$ with $\epsilon \in [0.05, 0.2]$ in the remaining responses, with common choices including $(1.05, 1.05)$, $(1.1, 1.1)$, and $(1.2, 1.2)$.

For DeepSeek-V3.2, it always suggested $(1.1, \ldots, 1.1)$ as the reference point for two-objective and three-objective problems. For five-objective problems, $(1.1, \ldots, 1.1)$ was suggested in seven responses, while $(1.2, \ldots, 1.2)$ was suggested in three responses. As the number of objectives increases, DeepSeek-V3.2 explained that the reference point must be far enough to overcome the "curse of dimensionality" and ensure that the HV indicator remains sensitive to improvements in both convergence and spread. Accordingly, for ten-objective problems, $(1.5, \ldots, 1.5)$ was suggested in two responses and $(2.0, \ldots, 2.0)$ was suggested in eight responses. As explained in [43], an appropriate specification of the reference point mainly depends on the population size and the number of objectives. The suggested specification of the reference point $\mathbf{r} = (r, r, \ldots, r)$ for the typical population size settings is as follows: $r = 1.01$ ($m = 2$), 1.08 ($m = 3$), 1.25 ($m = 5$), 1.33 ($m = 10$). It seems that DeepSeek-V3.2 understands the necessity of using a larger (i.e., worse) reference point in

a higher-dimensional objective space (whereas GPT-5.2 does not).

III. "EMO Algorithms" or "MOEAs"

In the literature, both "EMO algorithms" and "multi-objective evolutionary algorithms (MOEAs)" have been used to refer to evolutionary algorithms for multi-objective optimization. In general, these two terms are used interchangeably, and it is easy to understand that these two terms refer to the same concept. On Google Scholar, searching for "EMO algorithms" returns far fewer results than searching for "MOEAs." Since the output of LLMs usually depends on the prompt quality, one question is whether changing "EMO algorithms" in the prompts used in [16] to "multi-objective evolutionary algorithms (MOEAs)" would have any effect on the benchmarking suggestions provided by the LLMs.

To answer this question, we modify the prompt by changing "EMO algorithms" to "multi-objective evolutionary algorithms (MOEAs)," while all other parts of the prompt remain the same as those shown in Fig. 1. Due to space limitations, the tables in this section use "EMOAs" to denote EMO algorithms.

Table V and Table VI show the percentage of times each algorithm was suggested by GPT-5.2 and DeepSeek-V3.2 in response to the requests for "five state-of-the-art EMO algorithms" and "five state-of-the-art MOEAs," respectively. For GPT-5.2 (see Table V), it seems that with this slight change in the prompt, the responses obtained are quite similar. For example, NSGA-III [24] and RVEA [28] are always included in the suggestions regardless of the choice between "EMOAs" and "MOEAs". We also observed some minor variations in the responses generated by GPT-5.2, such as θ-DEA [44] being suggested as a state-of-the-art algorithm more frequently when "MOEAs" is used in the prompt (30% as EMOAs vs. 60% as MOEAs), whereas MOEA/D-DE [45] is suggested less frequently (70% as EMOAs vs. 40% as MOEAs).

TABLE V
PERCENTAGE (%) OF TIMES EACH ALGORITHM WAS SUGGESTED BY GPT-5.2 IN RESPONSE TO THE REQUEST FOR "FIVE-STATE-OF-THE-ART EMO ALGORITHMS" AND "FIVE STATE-OF-THE-ART MOEAS" (BASED ON 10 RESPONSES PER MODEL).

Algorithm (Year)	EMOAs (%)	MOEAs (%)
NSGA-III (2014)	100	100
RVEA (2016)	100	100
HyPE (2011)	100	90
θ-DEA (2016)	30	60
MOEA/D (2007)	40	50
MOEA/D-DE (2009)	70	40
SMS-EMOA (2007)	60	30
SPEA2+SDE (2014)	0	20
MOEA/D-STM (2014)	0	10

For DeepSeek-V3.2 (see Table VI), we observed a larger variation in the generated responses when "MOEAs" was used in the prompt. The generated responses included more than 20 algorithms across the 10 responses. Table VI presents only the algorithms that appeared more frequently in the responses. From Table VI, we can see that none of the algorithms were always included in the responses when "MOEAs" was used in the prompt, which is different from the responses obtained when using "EMO algorithms" in the prompt. For example, NSGA-III was always suggested by DeepSeek-V3.2 when "EMO algorithms" was used in the prompt, while it was included in only 60% of the responses when "MOEAs" was used. Similarly, RVEA was suggested only once among the 10 responses when "MOEAs" was used, whereas it was always suggested when "EMO algorithms" was used in the prompt. The results in Table VI show the sensitivity of DeepSeek-V3.2 to the wording used in the prompt for benchmarking EMO algorithms.

TABLE VI
PERCENTAGE (%) OF TIMES EACH ALGORITHM WAS SUGGESTED BY DEEPSEEK-V3.2 IN RESPONSE TO THE REQUEST FOR "FIVE-STATE-OF-THE-ART EMO ALGORITHMS" AND "FIVE STATE-OF-THE-ART MOEAS" (BASED ON 10 RESPONSES PER MODEL).

Algorithm (Year)	EMOAs (%)	MOEAs (%)
NSGA-III (2014)	100	60
RVEA (2016)	100	10
HypE (2011)	0	20
MOEA/DD (2015)	80	70
SMS-EMOA (2007)	80	0
LMEA (2018)	40	60
IM-MOEA (2015)	40	30
AR-MOEA (2018)	20	60
AGE-MOEA (2019)	20	20
DGEA (2022)	0	30

TABLE VII
PERCENTAGE (%) OF TIMES EACH PROBLEM SUITE WAS SUGGESTED WAS SUGGESTED BY GPT-5.2 IN RESPONSE TO THE REQUEST FOR SUGGESTING TEST SUITES FOR BENCHMARKING "EMO ALGORITHMS" AND "MOEAS" (BASED ON 10 RESPONSES PER MODEL).

Test suite (Year)	EMOAs (%)	MOEAs (%)
ZDT (2000)	100	100
DTLZ (2002)	100	100
WFG (2006)	100	100
MaF (2017)	100	100
UF (2009)	100	0
CF (2009)	100	0
Real-world benchmarks	0	60
CEC MO test suites	0	40

The sensitivity to the wording is also observed in Table VII

TABLE VIII
PERCENTAGE (%) OF TIMES EACH PROBLEM SUITE WAS SUGGESTED WAS SUGGESTED BY DEEPSEEK-V3.2 IN RESPONSE TO THE REQUEST FOR SUGGESTING TEST SUITES FOR BENCHMARKING "EMO ALGORITHMS" AND "MOEAS" (BASED ON 10 RESPONSES PER MODEL).

Test suite (Year)	EMOAs (%)	MOEAs (%)
DTLZ (2002)	100	100
WFG (2006)	100	100
MaF (2017)	100	70
LSMOP (2017)	100	0
REAL / RE (2020)	70	10
CEC competitions	60	100
IMOP (2019)	40	80
ZDT (2000)	20	60
LZ (2009)	0	30

Prompt:
You are an expert in constrained multi-objective optimization. Please suggest five representative constrained multi-objective evolutionary algorithms (CMOEAs) for benchmarking. Keep your response short and concise.

Prompt:
You are an expert in constrained multi-objective optimization. Please suggest test suites for benchmarking constrained multi-objective evolutionary algorithms (CMOEAs). Keep your response short and concise.

Prompt:
You are an expert in constrained multi-objective optimization. Please suggest only one test suite for benchmarking constrained multi-objective evolutionary algorithms (CMOEAs). Keep your response short and concise.

Prompt:
You are an expert in constrained multi-objective optimization. Please suggest performance indicators for benchmarking constrained multi-objective evolutionary algorithms (CMOEAs). Keep your response short and concise.

Prompt:
You are an expert in constrained multi-objective optimization. I want to use a single performance indicator for benchmarking constrained multi-objective evolutionary algorithms (CMOEAs). Which one would you suggest? Keep your response short and concise.

Fig. 2. Prompts used to ask the LLMs for benchmarking suggestions in constrained multi-objective optimization

and Table VIII regarding test suite suggestions for benchmarking. For example, in Table VII, GPT-5.2 suggests "Real-world benchmarks" only for MOEAs (60%). Interestingly, in Table VIII, DeepSeek-V3.2 suggests "REAL / RE" for MOEAs with only 10% frequency, whereas it is suggested for EMOAs more frequently (70%). That is, the necessity of benchmarking based on real-world problems is suggested by GPT-5.2 only for MOEAs and by DeepSeek-V3.2 only for EMOAs. In Table VII, "UF" and "CF" are suggested only when "EMOAs" is used (100%). In Table VIII, "LSMOP" is suggested only when "EMOAs" is used (100%). Overall, these results further suggest that LLM-generated benchmarking suggestions are sensitive to prompt wording, even when the underlying meaning of the prompt remains largely unchanged.

IV. Suggestions by LLMs for Constrained Multi-objective Optimization

Constrained multi-objective optimization has recently become a popular topic in the EMO field, and it has been noted that some studies, such as [10], [13], have used LLMs for constrained multi-objective optimization. In this section, we further examine the implicit knowledge of the two LLMs about constrained multi-objective optimization. We use the five prompts shown in Fig. 2 to ask the LLMs for suggestions on representative constrained multi-objective evolutionary algorithms (CMOEAs), test suites, and performance indicators for benchmarking CMOEAs. We use "constrained multi-objective evolutionary algorithms (CMOEAs)" instead of "constrained EMO algorithms" in the prompts because "CMOEAs" appears to be more commonly used in the field, according to Google Scholar search results ("CMOEAs": 554 results, "Constrained EMO algorithms": 13 results; accessed on April 14, 2026).

Table IX shows the percentage of times each algorithm was suggested by the two LLMs across the 10 responses. GPT-5.2 suggested NSGA-II-CDP [22] (i.e., NSGA-II with the constraint-domination principle) in 100% of the responses, while DeepSeek-V3.2 suggested it in 70% of the responses. It should be noted that NSGA-II-CDP is sometimes referred to as C-NSGA-II in some of the responses. In general, it can be seen that GPT-5.2 and DeepSeek-V3.2 have clearly different knowledge regarding the suggested CMOEAs. For example, C-TAEA [46] is suggested in 90% of the responses by DeepSeek-V3.2, whereas it is suggested in only 10% of the responses by GPT-5.2.

TABLE IX
Percentage (%) of times each algorithm was suggested by GPT-5.2 and DeepSeek-V3.2 in response to the request for five representative CMOEAs algorithms (based on 10 responses per model).

Algorithm (Year)	GPT-5.2 (%)	DeepSeek-V3.2 (%)
NSGA-II-CDP (2002)	100	70
CCMO (2021)	60	0
MOEA/D-IEpsilon (2016)	60	0
C-NSGA-III (2014)	60	30
MOEA/D-CDP	50	10
C-MOEA/D	50	80
CMOEA-MS (2022)	40	40
ϵ-MOEA	40	0
ToP (2019)	20	70
C-TAEA (2018)	10	90
C-MOEA/DD (2015)	0	30
PPS (2019)	0	30

Note: In the responses generated by GPT-5.2, ϵ-MOEA refers to MOEAs that use ϵ-constraint handling to balance feasibility and optimality. Since the specific algorithm is unclear, no proposed year is included. Similarly, C-MOEA/D denotes MOEA/D with constraint handling, and MOEA/D-CDP denotes MOEA/D with the Constraint Dominance Principle; because the specific variants are unspecified, no proposed years are included.

Table X shows the percentage of times each test suite was suggested by the two LLMs across the 10 responses. Both GPT-5.2 and DeepSeek-V3.2 suggested the CF and C-DTLZ test suites in 100% of their responses. DeepSeek-V3.2 always suggested DOC [47], whereas GPT-5.2 did not suggest it. CTP [36] and DAS-CMOP [48] were suggested in 70% and 60% of the responses by GPT-5.2 whereas they were not suggested by DeepSeek-V3.2. When the two LLMs were asked to suggest only one test suite, the CF test suite was suggested in 90% of the responses by GPT-5.2, while the DAS-CMOP test suite [48] was suggested in 10% of the responses. For DeepSeek-V3.2, the CF test suite was suggested in 70% of the responses, and the MW test suite was suggested in 30% of the responses. These results show clear variation in the suggested test suites between the two LLMs.

TABLE X
PERCENTAGE (%) OF TIMES EACH PROBLEM SUITE WAS SUGGESTED BY GPT-5.2 AND DEEPSEEK-V3.2 IN RESPONSE TO THE REQUEST FOR SUGGESTING TEST SUITES FOR BENCHMARKING CMOEAS (BASED ON 10 RESPONSES PER MODEL).

Test suite (Year)	GPT-5.2 (%)	DeepSeek-V3.2 (%)
CF (2009)	100	100
C-DTLZ (2014)	100	100
DC-DTLZ (2018)	100	20
CTP (2001)	70	0
LIR-CMOP (2019)	60	90
DAS-CMOP (2020)	60	0
Real-World Problems/RE	50	10
MW (2019)	40	100
CEC CMOPs	10	10
DOC (2019)	0	100

TABLE XI
PERCENTAGE (%) OF TIMES EACH PERFORMANCE INDICATOR WAS SUGGESTED BY GPT-5.2 AND DEEPSEEK-V3.2 IN RESPONSE TO THE REQUEST FOR SUGGESTING PERFORMANCE INDICATORS FOR BENCHMARKING CMOEAS (BASED ON 10 RESPONSES PER MODEL).

Performance indicator	GPT-5.2 (%)	DeepSeek-V3.2 (%)
HV	100	100
IGD	100	100
IGD^+	100	30
Feasible ratio (FR)	100	100
Constraint Violation (CV)	100	0
ϵ-indicator	40	0
Spacing	60	10
Spread	60	10
MIGD	0	20

Table XI shows the percentage of times each performance indicator was suggested by the two LLMs across the 10 responses. HV, IGD, and the feasible ratio (FR) were always suggested by both LLMs. One clear difference between the responses generated by GPT-5.2 and DeepSeek-V3.2 is that overall constraint violation (CV) was suggested only by GPT-5.2 (100% vs. 0%). IGD^+ was also suggested more frequently by GPT-5.2 than by DeepSeek-V3.2, whereas it was always suggested by both LLMs for EMO algorithms in Table IV. It is also observed that the MIGD indicator (which is a frequently used indicator for evaluating dynamic MOEAs) was suggested in 20% of the responses by DeepSeek-V3.2. This observation suggests that DeepSeek-V3.2 may not clearly distinguish between performance indicators for dynamic multi-objective optimization and those for constrained multi-objective optimization in some responses. While differences were observed between the responses generated by the two LLMs, both emphasized that performance indicators should be applied to feasible solutions.

Finally, the two LLMs were asked to suggest only a single performance indicator for benchmarking CMOEAs. GPT-5.2 consistently suggested using HV computed only on feasible solutions. DeepSeek-V3.2 suggested using HV or IGD, with the feasibility of the solutions taken into account. The explanations generated by DeepSeek-V3.2 in some responses appear somewhat questionable. For example, the following explanation was generated by DeepSeek-V3.2:

Feasibility-Ratio-Adjusted IGD (IFR-IGD) is the most recommended single indicator. It directly multiplies the Inverted Generational Distance (convergence/diversity) by the Feasibility Ratio, punishing algorithms that fail to satisfy constraints.

V. CONCLUDING REMARKS

This position paper examined the benchmarking suggestions provided by two recent LLMs (GPT-5.2 and DeepSeek-V3.2) for evolutionary multi-objective optimization (EMO) algorithms by using the same prompts and experimental settings as in the CEC 2025 paper [16]. In addition, the suggestions provided by the two LLMs for benchmarking constrained multi-objective evolutionary algorithms (CMOEAs) were also reported. The results showed that many suggestions from the two LLMs are consistent with commonly used benchmarking practices, such as the frequent recommendation of widely used EMO algorithms, test suites, and performance indicators. At the same time, differences in the generated suggestions were observed between the two LLMs and between the two versions of the same LLM, particularly in the selection of state-of-the-art algorithms and test suites.

The results also indicated that LLM-generated benchmarking suggestions are sensitive to the wording used in the prompts, even when only minor terminology changes are introduced (i.e., EMO algorithms and MOEAs). Some inconsistencies and inaccuracies were also observed in the detailed explanations, suggesting potential limitations of LLM-generated guidance. These observations imply that, while LLMs may provide convenient general information for benchmarking EMO algorithms, their suggestions should be interpreted carefully and supplemented with domain knowledge. At the same time, our results also show that the suggestions have clearly improved over the last year (e.g., a 100% increase in IGD^+ in the responses from both LLMs, 70% and 100% increases in MaF in the responses from GPT-5.2 and DeepSeek-V3.2, respectively, and a 70% increase in RE in the responses from DeepSeek-V3.2). This observation suggests that LLMs may become more reliable as more reliable training data become available, since their responses are largely consistent with the most frequently-used practices in the literature. That is, the reliability of LLM-generated benchmarking suggestions may be improved by establishing good benchmarking practices in the EMO community.

REFERENCES

[1] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[2] C. Coello Coello, "Evolutionary multi-objective optimization: a historical view of the field," *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28–36, 2006.

[3] K. Li, "A survey of multi-objective evolutionary algorithm based on decomposition: Past and future," *IEEE Transactions on Evolutionary Computation*, pp. 1–1 (Early Access), 2024.

[4] H. Ishibuchi, L. M. Pang, and K. Shang, "Difficulties in fair performance comparison of multi-objective evolutionary algorithms [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 17, no. 1, pp. 86–101, 2022.

[5] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff, "COCO: a platform for comparing continuous optimizers in a black-box setting," *Optimization Methods and Software*, vol. 36, no. 1, pp. 114–144, 2020.

[6] A. V. Kononova *et al.*, "Benchmarking that matters: Rethinking benchmarking for practical impact," *Arxiv*, 2025. [Online]. Available: https://arxiv.org/abs/2511.12264

[7] M. Raza, Z. Jahangir, M. B. Riaz, M. J. Saeed, and M. A. Sattar, "Industrial applications of large language models," *Scientific Reports*, vol. 15, p. 13755, 2025.

[8] A. F. Pereira and R. Ferreira Mello, "A systematic literature review on large language models applications in computer programming teaching evaluation process," *IEEE Access*, vol. 13, pp. 113 449–113 460, 2025.

[9] N. v. Stein and T. Bäck, "LLaMEA: A large language model evolutionary algorithm for automatically generating metaheuristics," *IEEE Transactions on Evolutionary Computation*, vol. 29, no. 2, pp. 331–345, 2025.

[10] Z. Wang, S. Liu, J. Chen, and K. C. Tan, "Large language model-aided evolutionary search for constrained multiobjective optimization," in *Advanced Intelligent Computing Technology and Applications*, 2024, pp. 218–230.

[11] Y. Huang, S. Wu, W. Zhang, J. Wu, L. Feng, and K. C. Tan, "Autonomous multi-objective optimization using large language model," *IEEE Transactions on Evolutionary Computation*, pp. 1–1 (Early Access), 2025.

[12] F. Liu, X. Lin, S. Yao, Z. Wang, X. Tong, M. Yuan, and Q. Zhang, "Large language model for multiobjective evolutionary optimization," in *Evolutionary Multi-Criterion Optimization*, 2025, pp. 178–191.

[13] S. Chen, H. Tang, J. Zhang, and K. Tang, "A large language model-assisted evolutionary algorithm for expensive constrained multi-objective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, Jul. 2025, pp. 347–350.

[14] M. Bok and A. McCallum, "OpenReview hosts record-breaking AAAI 2026 conference with pioneering AI review system," *Open Review News Article*, Oct. 2025. [Online]. Available: https://openreview.net/forum/bok%7Copenreview_hosts_recordbreaking_aaai_2026_conference_with_pioneering_ai_review_system

[15] [Online]. Available: https://icml.cc/Conferences/2026/LLM-Policy

[16] L. M. Pang and H. Ishibuchi, "Large language model-based benchmarking experiment settings for evolutionary multi-objective optimization," in *2025 IEEE Congress on Evolutionary Computation (CEC)*, Jun. 2025, pp. 1–8.

[17] [Online]. Available: https://openai.com/index/introducing-gpt-5-2/

[18] DeepSeek-AI, "DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models," *Arxiv*, Dec. 2025. [Online]. Available: https://arxiv.org/abs/2512.02556

[19] OpenAI, "GPT-4o System Card," Oct. 2024. [Online]. Available: https://arxiv.org/abs/2410.21276

[20] DeepSeek-AI, "DeepSeek-V3 Technical Report," *Arxiv*, Dec. 2024. [Online]. Available: https://arxiv.org/abs/2412.19437

[21] Zitzler, Eckart, Laumanns, Marco, and Thiele, Lothar, "SPEA2: Improving the strength pareto evolutionary algorithm," Swiss Federal Institute of Technology (ETH) Zurich, Tech. Rep. 103, 2001.

[22] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[23] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[24] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[25] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[26] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[27] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature - PPSN VIII*, 2004, pp. 832–842.

[28] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[29] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 838–856, 2015.

[30] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, "A benchmark test suite for evolutionary many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.

[31] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," Tech. Rep. CES-487, 2009.

[32] R. Tanabe and H. Ishibuchi, "An easy-to-use real-world multi-objective optimization problem suite," *Applied Soft Computing*, vol. 89, p. 106078, 2020.

[33] R. Cheng, Y. Jin, M. Olhofer, and B. sendhoff, "Test problems for large-scale multiobjective and many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4108–4121, 2017.

[34] Y. Tian, R. Cheng, X. Zhang, M. Li, and Y. Jin, "Diversity assessment of multi-objective evolutionary algorithms: Performance metric and benchmark problems [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 14, no. 3, pp. 61–74, 2019.

[35] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[36] K. Deb, A. Pratap, and T. Meyarivan, "Constrained test problems for multi-objective evolutionary optimization," in *Evolutionary Multi-Criterion Optimization*, 2001, pp. 284–298.

[37] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the 2002 Congress on Evolutionary Computation*, May 2002, pp. 825–830.

[38] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms - a comparative case study," in *Parallel Problem Solving from Nature - PPSN V*, 1998, pp. 292–301.

[39] C. A. Coello Coello and M. Reyes Sierra, "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm," in *MICAI 2004: Advances in Artificial Intelligence*, 2004, pp. 688–697.

[40] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *Evolutionary Multi-Criterion Optimization*, 2015, pp. 110–125.

[41] M. P. Hansen and A. Jaszkiewicz, "Evaluating the quality of approximations of the non-dominated set," Technical Univ. of Denmark, Tech. Rep. IMM-REP-1998-7, 1998.

[42] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[43] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "How to specify a reference point in hypervolume calculation for fair performance comparison," *Evolutionary Computation*, vol. 26, no. 3, pp. 411–440, 2018.

[44] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2016.

[45] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[46] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 303–315, 2019.

[47] Z.-Z. Liu and Y. Wang, "Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 870–884, 2019.

[48] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Difficulty adjustable and scalable constrained multiobjective test problem toolkit," *Evolutionary Computation*, vol. 28, no. 3, pp. 339–378, 2020.