TRACE-EA: ADAPTIVE TRAJECTORY OBSERVATION FOR AUTOMATIC OPERATOR DESIGN IN MULTI-OBJECTIVE COMBINATORIAL OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have enabled automatic heuristic design (AHD) for multi-objective combinatorial optimization problems (MOCOPs), but execution feedback used to guide heuristic revision remains centered on aggregate performance measures. Such summaries provide limited information about how progress and stagnation are distributed across trade-off regions or how previously attained coverage is lost during search. We present TRACE-EA, an LLM-driven AHD method that couples heuristic evolution with selective trajectory observation and observation-policy adaptation. An LLM observer selects tools and query arguments to examine trajectories recorded during candidate evaluation within a fixed multi-objective evolutionary solver. The retrieved evidence supports a diagnosis that guides the selection and generation of heuristic variations. A second feedback loop uses the history of observations, variation actions, and candidate outcomes to adapt the observation policy's temporal window, query-round allowance, and tool priors. Policy updates are retained or reverted based on subsequent candidate outcomes. Experiments across 18 multi-objective settings show that TRACE-EA achieves competitive performance, with advantages that carry over to unseen instances and larger problem scales.

1 INTRODUCTION

Search operators determine how evolutionary and metaheuristic algorithms explore the solution space. Their effectiveness depends on problem-specific design and coordination within the solver (Bezerra et al., 2015). Designing these operators is particularly challenging for multi-objective combinatorial optimization problems (MOCOPs), such as vehicle routing and production scheduling (Zhang et al., 2023), where search must advance toward the Pareto front while maintaining diverse trade-offs. Developing effective heuristics for these roles requires domain expertise and repeated empirical evaluation, making manual design costly. This burden has motivated sustained research on automated heuristic and algorithm design (Ma et al., 2025; Liu et al., 2026b).

Large Language Models (LLMs) offer a flexible route to automatic heuristic design (AHD): they can translate algorithmic ideas into executable code and revise candidate heuristics using empirical feedback (Liu et al., 2024; Ye et al., 2024). Recent methods have expanded this process from evolving individual heuristics to designing complementary heuristic sets (Liu et al., 2026a) and meta-optimizing the generation process (Shi et al., 2026). Other work uses search dynamics to guide heuristic revision and online hyperparameter adaptation (Wang et al., 2026; Xu et al., 2026). These developments motivate closer examination of what execution evidence can guide heuristic design for multi-objective search.

For MOCOPs, MPaGE uses Pareto-grid guidance to promote diversity within the heuristic population (Hieu et al., 2026), while E2OC searches for combinations of interdependent operators (Qiu et al., 2026). However, the execution feedback used to guide their revision remains centered on aggregate performance measures, such as hypervolume and runtime, providing limited information about how the approximation set evolves during a solver run.

In multi-objective search, progress involves advancing the approximation set while maintaining coverage across trade-off regions (Shang et al., 2020). Hypervolume can increase even as previously

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

attained coverage is lost (López-Ibáñez et al., 2011), and improvements in some directions can coexist with stagnation in others. Distinguishing these behaviors can inform which aspects of a candidate heuristic to revise. The relevant evidence spans different time intervals, trade-off regions, and operator behaviors, and the focus of diagnosis may change as the heuristic population evolves. This motivates selective observation of multi-objective solver trajectories, with the scope and depth of observation adapted to support successive heuristic revisions.

In this paper, we present TRACE-EA, an LLM-driven AHD method for MOCOPs that couples heuristic evolution with adaptive trajectory observation. The first feedback loop uses trajectories recorded during candidate evaluation within a fixed multi-objective evolutionary algorithm (MOEA) solver. An LLM observer selectively queries these trajectories and synthesizes the retrieved evidence into a diagnosis that guides both the choice of variation action and the generation of the next candidate heuristic. The second loop uses the history of observations, variation actions, and candidate outcomes to adapt the observation policy's temporal window, query-round allowance, and tool priors. Observation-policy updates are assessed using subsequent candidate outcomes, which determine whether each update is retained or reverted. Our contributions are as follows:

- **Selective trajectory observation.** We introduce TRACE-EA, an LLM-driven AHD method for MOCOPs. An observer selects tools and query arguments to examine recorded solver trajectories and produces a diagnosis supported by the retrieved evidence to guide heuristic variation.
- **Observation-policy adaptation.** We adapt the observation policy using the history of heuristic evolution, adjusting the temporal window, query-round allowance, and tool priors. Subsequent candidate outcomes determine whether policy updates are retained or reverted.
- **Performance and cross-scale transfer.** Across 18 multi-objective settings on TSP, CVRP and FJSP problems with NSGA-II and SEMO solvers, TRACE-EA achieves competitive performance, with advantages that carry over to unseen instances and larger scales.

2 Related Work

LLM-driven Automatic Heuristic Design in Multi-Objective Optimization. Leveraging the reasoning and code generation abilities of pre-trained models, LLMs are increasingly used in AHD (Romera-Paredes et al., 2024; Liu et al., 2026b). Recently, this paradigm has been extended from single-objective problems to multi-objective problems. MEoH formulates heuristic search as a multi-objective optimization problem and manages the heuristic population through a dominance-dissimilarity mechanism, while its experiments focus on single-objective optimization tasks (Yao et al., 2025). Later, MPaGE and E2OC extended the problem settings to MOCOP (Hieu et al., 2026; Qiu et al., 2026). MPaGE extends reflection-based search using Pareto Front Grid to promote diversity and reduce redundancy in program population (Hieu et al., 2026). E2OC formulates multi-operator optimization as a Markov decision process and uses Monte Carlo Tree Search to search combinations of operator design strategies (Qiu et al., 2026). In these MOCOP AHD methods, execution feedback remains centered on aggregate performance measures, leaving detailed anytime behavior of the approximation set unexploited in heuristic generation.

Trajectory Observation and Performance Assessment. A parallel line of work asks what can be measured from the optimization process itself. Since no single indicator fully characterizes an approximation set (Zitzler et al., 2003), process-level descriptions have been developed: anytime behavior has been treated as a design objective for MOEAs (Radulescu et al., 2013; Vermetten et al., 2025), empirical attainment functions describe outcome distributions over the objective space (Fonseca & Fleming, 1996), and search trajectory networks represent a run as a graph over visited states (Ochoa et al., 2021; Lavinas et al., 2022; Ochoa et al., 2023), which has been used to attribute the behavior of automatically configured MOEAs to individual algorithmic components (Lavinas et al., 2024). This idea has recently reached LLM-driven AHD in single-objective settings: AutoEP computes online landscape features from the ongoing run and uses LLM agents to adjust hyperparameters during that run (Xu et al., 2026), while DASH uses changes in solution quality over time to guide the co-evolution of search mechanisms and runtime schedules (Wang et al., 2026). These studies demonstrate the value of search-process information for algorithm design. However, how to select and use such observations for MOCOP heuristic design remains underexplored.

3 PRELIMINARIES

3.1 MULTI-OBJECTIVE COMBINATORIAL OPTIMIZATION

Given an instance $d \in \mathcal{D}$, a MOCOP aims to optimize multiple, often conflicting, objectives over a feasible solution space. In this work, we consider the minimization form

$$\min_{x \in \mathcal{X}_d} f_d(x) = \big(f_{d,1}(x), \ldots, f_{d,M}(x)\big), \tag{1}$$

where $\mathcal{X}_d$ denotes the feasible solution space associated with instance d, M is the number of objectives, and $f_d : \mathcal{X}_d \to \mathbb{R}^M$ maps a feasible solution to its objective vector.

Since multiple objectives are optimized simultaneously, solutions are compared using the notion of *Pareto dominance*. For two feasible solutions $x_a, x_b \in \mathcal{X}_d$, x_a is said to dominate x_b, denoted by $x_a \prec x_b$, if $f_{d,i}(x_a) \leq f_{d,i}(x_b), \forall i = 1, \ldots, M$, and the inequality is strict for at least one objective. A feasible solution $x^\star$ is *Pareto optimal* if there exists no other feasible solution $x \in \mathcal{X}_d$ such that $x \prec x^\star$. The set of all Pareto-optimal solutions is called the *Pareto set*, defined as

$$\mathcal{P}_d = \{x \in \mathcal{X}_d \mid \nexists\, x' \in \mathcal{X}_d \text{ such that } x' \prec x\}\,. \tag{2}$$

Its image in the objective space is called the *Pareto front* $\mathcal{F}_d$.

Computing the complete Pareto set of a MOCOP is often infeasible because the number of Pareto-optimal solutions can grow exponentially with instance size. Thus, practical solvers return a finite approximation set $\mathcal{A} \subseteq \mathcal{X}_d$; its nondominated solutions approximate the Pareto set, and their objective vectors $f_d(\mathcal{A})$ approximate the Pareto front. We evaluate this approximation using the metrics in Appendix A.4.

3.2 INNER MOEA SOLVERS

Evolutionary algorithms are widely used to construct approximations to the Pareto front $\mathcal{F}_d$, and we consider two multi-objective search frameworks: Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002) and Simple Evolutionary Multiobjective Optimizer (SEMO) (Laumanns et al., 2004). NSGA-II maintains a fixed-size population and generates offspring through crossover and mutation. In its population update process, selection prioritizes lower nondomination ranks and uses larger crowding distances to break ties. SEMO, as described by Hieu et al. (2026), maintains an archive of solutions. In its basic form, each iteration selects an archive member, generates one neighbor, and updates the archive according to Pareto dominance. Unlike NSGA-II's fixed-size population, SEMO does not impose a fixed archive size.

4 METHODOLOGY

4.1 OVERVIEW

TRACE-EA evolves a population of heuristics that serve as operators within a MOEA solver, such as NSGA-II or SEMO. As shown in Figure 1, heuristic evolution (blue) selects parents, generates candidate heuristics with an LLM, and evaluates them within the solver, retaining heuristics by quality and efficiency. This evolution is guided by two coupled feedback loops. In the first loop (green), an observer queries the solver trajectories recorded during evaluation through tools and synthesizes the retrieved evidence into a diagnosis that guides subsequent heuristic generation. In the second loop (orange), the history of heuristic evolution is used to adapt the observation policy, adjusting which trajectory evidence the observer queries and how closely it examines that evidence.

4.2 HEURISTIC EVOLUTION

We initialize the outer search with a population of feasible heuristics and evolve it by generating candidates from selected parents. Parent selection balances quality and diversity, allowing the search to build on strong heuristics while exploring alternative designs. To guide parent variation, the observer examines solver trajectories recorded during parent evaluation, and provides a diagnosis

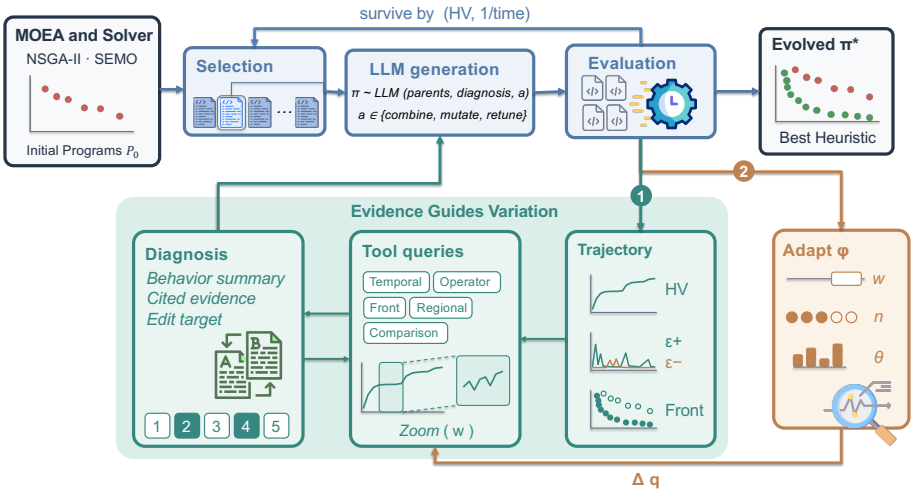


Figure 1: Overview of TRACE-EA: heuristic evolution (blue), trajectory observation (green), and observation-policy adaptation (orange).

(Section 4.3). The LLM then uses this diagnosis in one of four variation actions: combining ideas from multiple parents, changing a design choice shared by the parents, or revising the structure or the parameters of a single parent (Appendix B). An action policy chooses the type of variation based on recent search history, quality changes associated with each action, and the previous diagnosis.

To select the heuristic population, we evaluate each candidate π within the MOEA solver on the instances $\mathcal{D}$. We repeat each evaluation N times and compute its mean quality and runtime:

$$q(\pi) = \frac{1}{|\mathcal{D}|N} \sum_{d\in\mathcal{D}} \sum_{n=1}^{N} \mathrm{HV}'_s(F_{d,n}(\pi)), \qquad \bar{t}(\pi) = \frac{1}{|\mathcal{D}|N} \sum_{d\in\mathcal{D}} \sum_{n=1}^{N} t_{d,n}(\pi). \tag{3}$$

Here, $F_{d,n}(\pi)$ is the final nondominated front, and $t_{d,n}(\pi)$ is the corresponding runtime. We refer to $q(\pi)$ as the design quality. These measurements define two objectives for heuristic population selection: higher quality $q(\pi)$ and greater efficiency $1/\bar{t}(\pi)$. We rank heuristics by nondominated sorting and use crowding-based truncation to retain diverse quality–efficiency trade-offs. At the end of the search, we output the heuristic with the highest design quality.

4.3 Selective Trajectory Observation

For a fixed candidate heuristic $\pi \in \Pi$, the MOEA solver $\mathcal{S}$ runs on instance d with random seed ω, producing an approximation set $\mathcal{A} \subseteq \mathcal{X}_d$ with a recorded trajectory,

$$(\mathcal{A}, \boldsymbol{\tau}) = \mathcal{S}(d \mid \pi, \omega), \qquad \boldsymbol{\tau} = (\tau^{(1)}, \ldots, \tau^{(T)}), \tag{4}$$

where $\tau^{(t)}$ is a recorded snapshot containing front coordinates and search statistics at the t-th of T recorded points during the run.

Characterizing search dynamics. Each snapshot records trajectory features that describe how the front changes between recorded points, beyond aggregate hypervolume. Let F_t denote the nondominated front at recorded point t; front comparisons use objectives scaled by the fixed reference point $\boldsymbol{r}_s$ of equation 13. We compare consecutive fronts with the binary additive ε-indicator (Zitzler et al., 2003) in both directions; its positive parts measure coverage gained and lost, and both can be positive even when hypervolume increases. Weighted Tchebycheff utilities (Hansen & Jaszkiewicz, 1998) over a fixed set of weight vectors indicate which trade-off directions improve and which stagnate. We further record the fraction of the previous front that is retained, the concentration of exclusive hypervolume contributions for NSGA-II, hypervolume, population diversity, front geometry, the count of new archive points, and solver-specific operator statistics. Appendix C.1 gives the full definitions and recording protocol.

Evidence across runs. To support diagnosis beyond aggregate performance, we retain the final front and recorded trajectory from each evaluation. For a candidate heuristic π, these records form the evaluation bundle

$$\mathcal{B}(\pi) = \left\{ \left(F_{d,n}(\pi),\, \boldsymbol{\tau}_{d,n}(\pi) \right) \;\middle|\; d \in \mathcal{D},\; n = 1, \ldots, N \right\}. \tag{5}$$

While the aggregate criteria in equation 3 guide heuristic population selection, the bundle provides run-level evidence for diagnosis, allowing the observer to examine variation in outcomes and recurring search behaviors across instances and repeats.

Evidence-guided diagnosis. Evidence for guiding heuristic variation depends on the selected parents' search behavior. Aggregate quality changes alone do not reveal when the behavior emerges or which trade-off directions are affected. We therefore let the observer choose which aspects of the recorded trajectories to inspect and refine its queries using the evidence already retrieved.

The observer accesses the selected parents' evaluation bundles through eleven tools. These tools provide complementary views of search behavior: overall progress and temporal changes, front geometry and direction-specific progress, operator statistics, and comparisons across runs or candidate heuristics. Tool arguments specify which parent heuristic to inspect and, where applicable, the time window, metrics, or front points of interest. This interface allows the observer to investigate a particular aspect of search behavior without receiving the full trajectory in its prompt. Appendix C.2 lists the tools and their arguments.

Starting from summaries of the selected parents, the observer selects tools and query arguments through function calling to investigate their search behavior. Each tool use provides evidence that informs subsequent queries, allowing the observer to focus on relevant time intervals, trade-off regions, and aspects of heuristic behavior. The observer synthesizes the retrieved evidence into a diagnosis of the observed search behavior, supported by explicit citations. This diagnosis and any suggested modification target then guide the generation of the next candidate heuristic.

4.4 Adapting the observation policy

The usefulness of trajectory evidence may vary as the heuristic population evolves. Within the first feedback loop, the observer autonomously selects tools and query arguments under a given observation policy. The policy itself, however, is not updated within this loop. The second feedback loop adapts this policy using the history of observations, variation actions, and resulting candidate quality. We define the observation policy as

$$\varphi = \left(w,\, n_{\text{rounds}},\, \boldsymbol{\theta} \right), \tag{6}$$

where w is the default temporal window, n_{rounds} is the maximum number of query rounds, and $\boldsymbol{\theta}$ contains tool priors. Tool priors guide the LLM observer's tool selection, which remains conditioned on the available evidence.

The evolution history records observations, variation actions, and candidate evaluation outcomes. For each step i with an active observation and a scored candidate, we measure the quality change relative to its best parent:

$$\Delta q_i = q(\pi_i') - \max_{\pi \in \Pi_{\text{pa},i}} q(\pi), \tag{7}$$

where π_i' is the generated candidate heuristic and $\Pi_{\text{pa},i}$ is its parent set. This feedback uses the design quality q defined in equation 3.

The action policy can also select an adaptation action, at most once per generation. A bounded deterministic rule then uses this history to adjust at most one policy component. Temporal-window updates favor scopes associated with larger mean quality changes. Query-round updates use the fraction of retrieved evidence cited in the diagnosis and the extent to which previous observations used the available rounds: low citation rates favor fewer rounds, whereas high citation rates combined with frequent exhaustion of the allowance favor more rounds. Tool-prior updates reweight tools whose evidence was cited in the diagnosis: their weights increase after a positive quality change and decrease after a nonpositive change. Uncited tools retain their weights.

Each policy change is assessed using subsequent candidate outcomes before another change is proposed. Because variation actions can differ in their typical quality changes, we compare outcomes

using action-adjusted residuals:

$$r_i = \Delta q_i - \bar{\Delta} q_{a_i}, \tag{8}$$

where a_i is the variation action at step i and $\bar{\Delta} q_{a_i}$ is its mean quality change before the policy update. Once enough subsequent outcomes are available, we compare their mean residual with that of recent pre-update steps, considering only actions observed before the update. We revert the policy change if the mean residual decreases and retain it otherwise.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTINGS

Benchmarks. We evaluate TRACE-EA on benchmarks of 18 settings covering three multi-objective combinatorial problems: the Traveling Salesman Problem (TSP), the Capacitated Vehicle Routing Problem (CVRP), and Flexible Job-Shop Scheduling (FJSP), each with bi- and tri-objective variants at three scales. TSP and CVRP instances are generated following the multi-objective formulation of (Chen et al., 2023), FJSP follows the procedure of (Song et al., 2022). For CVRP, we fix the fleet to the minimum feasible size and add a route-balance objective as the third objective. Heuristics are evolved on three medium-scale instances (seen instances) and evaluated on these instances and on unseen instances at all three scales. All instances are generated in advance and kept fixed throughout the LLM-driven heuristic searching and the final evaluation, as detailed in Appendix A.

Experiment Setup. Experiments are conducted on a Mac Studio with an Apple M4 Max chip (16-core CPU, 64 GB unified memory). All LLM-driven methods use `gpt-5.6-terra` with temperature 1.0, with independent search seeds and a budget of 200 candidate heuristics per task per seed. The main results compare the approximation sets obtained after an equal wall-clock time of 30 s; results at 60 s and under fixed function-evaluation (FE) budgets are reported in Appendix D.

Baselines. We compare TRACE-EA with two classical MOEAs, NSGA-II (Deb et al., 2002) and MOEA/D (Zhang & Li, 2007), and four LLM-driven AHD methods, EoH (Liu et al., 2024), MEoH (Yao et al., 2025), MPaGE (Hieu et al., 2026) and E2OC (Qiu et al., 2026). EoH is adapted to evolve the same NSGA-II as E2OC, MEoH is adapted to evolve the same SEMO as MPaGE. Implementations and hyperparameters are detailed in Appendix B.

Performance metric. We evaluate the performance of algorithms using Hypervolume (HV) and Inverted Generational Distance (IGD). A higher HV indicates better overall performance in terms of convergence and diversity, while a lower IGD indicates better proximity and distribution relative to a reference Pareto front, formed as the nondominated union of the fronts obtained by all compared methods. Each LLM-driven method is run independently with multiple search seeds, producing one heuristic per seed. Every heuristic is then executed five times on each of the three evaluation instances, and every traditional MOEA baseline is likewise executed five times per instance. Reported HV and IGD are means over all such executions. Details of reference points and reference Pareto front are elaborated in Appendix A.4.

5.2 MAIN RESULTS

Performance on seen instances. Table 1 compares performance on the instances used during heuristic search under a 30 s wall-clock time budget. TRACE-EA is competitive within both solver groups, with consistent advantages on TSP and CVRP. Performance on FJSP is more task-dependent, with existing methods retaining advantages on some tasks.

Generalization and cross-scale transfer. Table 3 evaluates the same heuristics on unseen instances at three scales. The results show that advantages on TSP and CVRP extend to unseen instances, including larger instances, whereas transfer to smaller instances is less consistent. Generalization remains task-dependent, with no method leading uniformly across tasks and scales.

Anytime behavior. Figure 2(a) traces HV over the 30 s run on Bi-TSP50. On both seen and unseen instances, EoH leads within the first second while TRACE-EA starts lowest, but TRACE-EA attains

Table 1: Equal wall-clock time results at 30 s on seen instances at the medium scale.

Method	Bi-TSP50		Tri-TSP50		Bi-CVRP50		Tri-CVRP50		Bi-FJSP10j5m		Tri-FJSP10j5m	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
Traditional methods												
NSGA-II	0.6095	0.0249	0.2972	0.1381	0.2615	0.3232	0.2082	0.3654	0.3797	0.0204	0.1374	0.0073
MOEA/D	0.5792	0.0425	0.3002	0.1141	0.2835	0.2943	0.2348	0.3080	0.3725	0.0282	0.1357	0.0122
NSGA-II-based methods												
E2OC	0.5517	0.0654	0.2565	0.1709	0.2790	0.3011	0.2411	0.3155	**0.3860**	**0.0147**	0.1367	0.0090
EoH	0.5215	0.0911	0.3413	0.0931	0.3068	0.2632	0.2555	0.2987	0.3803	0.0205	**0.1383**	**0.0063**
TRACE-EA (ours)	**0.6366**	**0.0080**	**0.3832**	**0.0792**	**0.5094**	**0.0382**	**0.4541**	**0.0844**	0.3844	0.0167	0.1379	0.0082
SEMO-based methods												
MEoH	0.6150	0.0212	0.3729	0.0725	0.5355	0.0121	0.4996	0.0503	**0.3725**	**0.0315**	0.1323	0.0241
MPaGE	0.6059	0.0280	0.3658	0.0724	0.5361	0.0106	0.5105	0.0350	0.3619	0.0454	0.1320	0.0241
TRACE-EA (ours)	**0.6371**	**0.0052**	**0.4431**	**0.0191**	**0.5413**	**0.0081**	**0.5256**	**0.0094**	0.3658	0.0374	**0.1359**	**0.0114**

HV ↑; IGD ↓. Bold: best within each solver group.

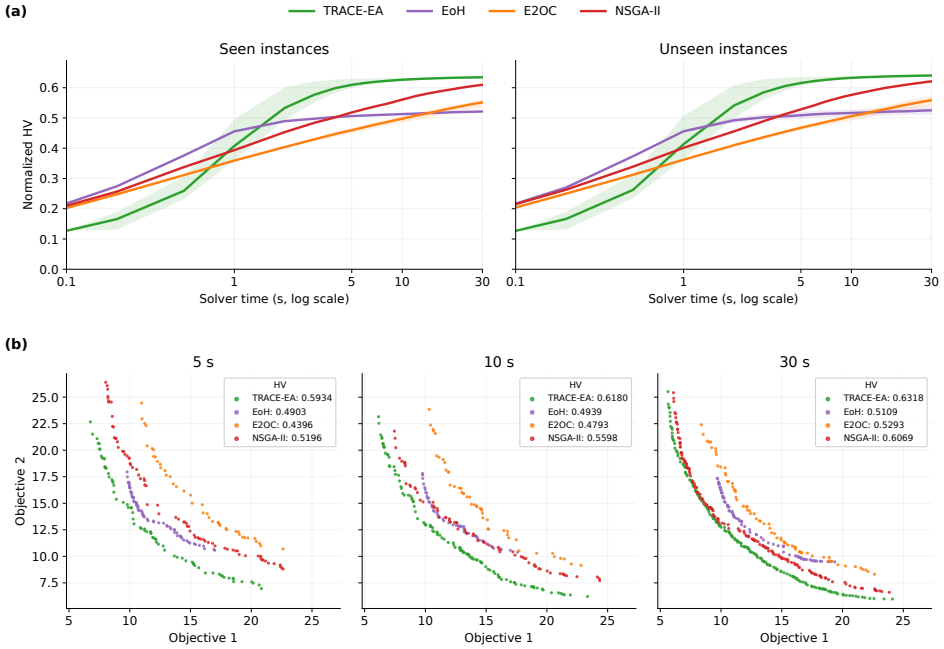


Figure 2: Anytime behavior on Bi-TSP50 with the NSGA-II solver. (a) Normalized HV over solver time; for AHD methods, shading spans the search-seed means. (b) Approximation sets at 5, 10, and 30 s on one seen instance; both objectives are minimized.

the highest HV from 2 s onward. After 5 s, EoH nearly stalls, whereas E2OC and NSGA-II keep improving and overtake EoH without reaching TRACE-EA. In objective space (Figure 2(b)), the TRACE-EA front attains the lowest values of both objectives at all three times, and by 30 s NSGA-II approaches it only at low Objective 1 values.

Observer tool use. Figure 3 summarizes 78,074 tool calls over 6,813 observed steps, 38,000 of which were cited in the returned diagnoses. The observer tailors the queried evidence through tool arguments: calls to window-based tools always set their time window explicitly, and metric-level inspection requested over 100 distinct metric combinations per solver. Observation typically proceeds from broad to targeted evidence. Over 80% of steps begin with the trajectory overview; together with the failure-mode scan, these broad tools account for 37% of calls but are cited less often. Tools that inspect specific time windows, trade-off directions, or front properties are called

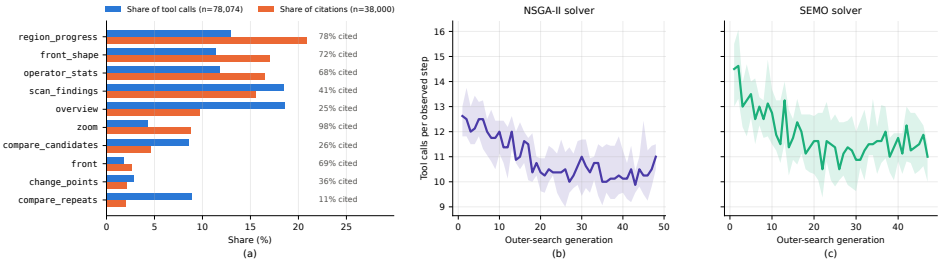


Figure 3: Observer tool use in the 36 TRACE-EA searches behind the main results. (a) Share of tool calls and of cited calls per tool, annotated with each tool's citation rate; `correlate` is omitted (34 calls, <0.1%). (b, c) Tool calls per observed step with NSGA-II and SEMO; lines show the median over the 18 searches per solver and shading the interquartile range.

later and cited more often, accounting for most citations. As the search proceeds, the observer uses fewer calls per step, dropping from 12–14 in early generations to 10–12 later.

5.3 Ablations

We ablate the two feedback loops of TRACE-EA on Bi-TSP50 with the NSGA-II solver. Table 2 reports the best design quality found during search (search HV) and the solver HV at 2, 5, and 30 s. The first loop is varied through three columns. *Trajectory Feedback* specifies the trajectory information available for diagnosis: *full* includes the feature families in Section 4.3, whereas a dash removes trajectory feedback from heuristic generation while retaining candidate evaluation and population selection. *Tool Selection* specifies whether the observer's tools are chosen by the LLM or at random, and *Variation Action* specifies whether variation actions follow a fixed round-robin order or are selected by the LLM or at random. For the second loop, *Policy Adaptation* indicates whether the observation policy is updated during heuristic evolution.

Table 2: Ablation on Bi-TSP50 with the NSGA-II solver, seen instances.

Variant	Configuration				Search	Solver HV		
	Trajectory Feedback	Tool Selection	Variation Action	Policy Adaptation	Best HV ↑	@2 s ↑	@5 s ↑	@30 s ↑
No feedback	–	–	fixed	–	0.5903	0.4047	0.5493	0.6116
Active observation	full	LLM	fixed	–	0.6038	0.5455	0.6145	0.6307
+ LLM action selection	full	LLM	LLM	–	0.6080	0.4763	0.5965	0.6297
Random tool selection	full	random	fixed	–	0.5907	0.4616	0.5617	0.6198
Random action selection	full	LLM	random	–	0.5954	0.4954	0.6042	0.6335
TRACE-EA (ours)	full	LLM	LLM	✓	**0.6282**	**0.5930**	**0.6317**	**0.6366**

HV ↑. Search: best design quality q after 200 candidates. @t s: HV of the selected heuristic at solver time t. Means over three search seeds, each averaging three instances and five solver seeds. Bold: best mean across variants.

Results. Both loops contribute to performance on Bi-TSP50. In the first loop, trajectory feedback with LLM-selected tools improves every metric over the no-feedback variant and outperforms random tool selection on all metrics, whereas random selection leaves search HV nearly unchanged from removing feedback. Action selection alone yields no consistent improvement over the fixed order. The second loop further improves every metric over the same configuration without adaptation, with the largest gains in search HV and early solver HV, and the resulting full configuration achieves the best results across all metrics.

Table 3: Equal wall-clock time results at 30 s on unseen instances across three scales.

Method	Bi-TSP20		Bi-TSP50		Bi-TSP100		Tri-TSP20		Tri-TSP50		Tri-TSP100	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
Traditional methods												
NSGA-II	0.6198	0.0008	0.6210	0.0216	0.5611	0.1120	0.4607	0.0106	0.2968	0.1430	0.1866	0.3096
MOEA/D	0.6185	0.0024	0.5877	0.0435	0.5202	0.1478	0.4419	0.0155	0.2980	0.1205	0.2084	0.2387
NSGA-II-based methods												
E2OC	0.6192	0.0020	0.5590	0.0655	0.4766	0.1726	0.4445	0.0203	0.2564	0.1744	0.1622	0.3383
EoH	0.5885	0.0217	0.5253	0.0923	0.4655	0.2090	**0.4604**	**0.0066**	0.3446	0.0964	0.2269	0.2469
TRACE-EA (ours)	**0.6194**	**0.0011**	**0.6403**	**0.0089**	**0.6852**	**0.0392**	0.4426	0.0230	**0.3876**	**0.0847**	**0.3798**	**0.1267**
SEMO-based methods												
MEoH	0.6172	**0.0032**	0.6222	0.0217	0.6303	0.0837	0.4415	0.0186	0.3740	0.0787	0.3757	0.1236
MPaGE	0.6170	0.0039	0.6097	0.0308	0.6105	0.0940	0.4588	**0.0111**	0.3702	0.0746	0.3150	0.1591
TRACE-EA (ours)	**0.6193**	0.0045	**0.6438**	**0.0055**	**0.7061**	**0.0055**	**0.4596**	0.0164	**0.4485**	**0.0194**	**0.4770**	**0.0473**

Method	Bi-CVRP20		Bi-CVRP50		Bi-CVRP100		Tri-CVRP20		Tri-CVRP50		Tri-CVRP100	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
Traditional methods												
NSGA-II	0.5788	0.0108	0.3069	0.3024	0.1220	0.6292	0.5261	0.0225	0.2492	0.3379	0.0781	0.6944
MOEA/D	0.5802	0.0093	0.3194	0.2839	0.1253	0.6253	0.5311	0.0174	0.2688	0.2954	0.0785	0.6929
NSGA-II-based methods												
E2OC	0.5783	0.0109	0.3192	0.2857	0.1299	0.6175	0.5271	0.0211	0.2812	0.2974	0.0841	0.6775
EoH	0.5821	0.0077	0.3406	0.2574	0.1301	0.6152	0.5334	0.0139	0.2859	0.2907	0.0847	0.6794
TRACE-EA (ours)	**0.5869**	**0.0044**	**0.5263**	**0.0584**	**0.1798**	**0.5252**	**0.5391**	**0.0074**	**0.4586**	**0.1096**	**0.1373**	**0.5572**
SEMO-based methods												
MEoH	0.5831	0.0069	0.5694	0.0170	0.6051	0.0206	0.5319	0.0170	0.5198	0.0582	0.5635	0.0652
MPaGE	0.5832	**0.0054**	0.5723	**0.0143**	0.6067	0.0201	0.5338	0.0138	0.5409	0.0345	0.5769	0.0599
TRACE-EA (ours)	**0.5837**	0.0064	**0.5757**	0.0145	**0.6185**	**0.0110**	**0.5373**	**0.0124**	**0.5554**	**0.0180**	**0.5881**	**0.0392**

Method	Bi-FJSP5j5m		Bi-FJSP10j5m		Bi-FJSP25j5m		Tri-FJSP5j5m		Tri-FJSP10j5m		Tri-FJSP25j5m	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
Traditional methods												
NSGA-II	0.5091	0.0108	0.3076	0.0255	0.2068	0.0256	0.1726	0.0045	0.0797	0.0076	0.0554	0.0042
MOEA/D	0.4979	0.0210	0.2969	0.0408	0.1995	0.0372	0.1701	0.0108	0.0775	0.0160	0.0532	0.0117
NSGA-II-based methods												
E2OC	0.5045	0.0174	**0.3153**	**0.0190**	**0.2131**	**0.0179**	0.1712	0.0090	0.0785	0.0123	0.0543	0.0077
EoH	0.5062	0.0141	0.3072	0.0284	0.2074	0.0246	**0.1729**	**0.0041**	**0.0800**	**0.0074**	**0.0554**	**0.0047**
TRACE-EA (ours)	**0.5127**	**0.0096**	0.3130	0.0231	0.1998	0.0393	0.1720	0.0070	0.0797	0.0097	0.0442	0.0469
SEMO-based methods												
MEoH	**0.5046**	**0.0178**	**0.3042**	**0.0352**	**0.2025**	**0.0331**	0.1677	0.0182	0.0754	0.0240	0.0523	0.0156
MPaGE	0.4961	0.0257	0.2897	0.0559	0.1955	0.0456	0.1683	0.0162	0.0753	0.0239	0.0502	0.0244
TRACE-EA (ours)	0.4978	0.0239	0.2939	0.0457	0.1973	0.0402	**0.1687**	**0.0146**	**0.0768**	**0.0145**	**0.0546**	**0.0064**

HV ↑; IGD ↓. Bold: best within each solver group.

6 Conclusion

We presented TRACE-EA, an LLM-driven method for automatic heuristic design in multi-objective combinatorial optimization that couples heuristic evolution with adaptive trajectory observation. An observer selectively queries recorded solver trajectories and synthesizes retrieved evidence into diagnoses that guide heuristic variation. A second feedback loop adapts the observation policy using the history of heuristic evolution, retaining or reverting updates based on subsequent candidate outcomes. Experiments on TSP, CVRP, and FJSP with NSGA-II and SEMO backbones demonstrate competitive performance, with advantages extending to unseen instances and larger scales. Overall, our findings support selective trajectory observation as a promising way to inform heuristic evolution and motivate further study of how evidence selection and heuristic evolution can be jointly adapted.

AI USE STATEMENT

Large Language Models are a component of the method studied in this paper: TRACE-EA and the LLM-driven baselines use `gpt-5.6-terra` to generate candidate heuristics, observe solver trajectories, and select variation actions, as described in Sections 4 and 5. All reported results are computed by executing the generated heuristics in the fixed solvers and scoring them with the metrics of Appendix A.4; no result relies on LLM judgement. Separately, we used generative AI tools to assist with the literature survey, with implementing and debugging the experimental code, and with drafting and polishing this manuscript. All benchmark instances are generated procedurally following the cited procedures and do not involve generative AI. All AI-assisted code was reviewed and tested by the authors, and all AI-assisted text was reviewed and verified against the cited sources and the experimental records. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, personal data, or crowdsourced annotation. All benchmark instances are synthetic and generated procedurally from fixed random seeds (Appendix A), so no data are collected from individuals or organizations. The method automates the design of search heuristics for routing and scheduling benchmarks, and we are not aware of harmful applications beyond those of existing optimization software. Heuristics generated by the LLM are executed only within the fixed solvers on these synthetic instances. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

Section 5 specifies the benchmark settings, the compared methods, the LLM and its sampling temperature, the candidate budget, the hardware, and the evaluation metrics. Appendix A describes instance generation and the evaluation metrics with their reference points; Appendix B gives the solver configurations, baseline adaptations, candidate-search budgets, final evaluation protocols, and result aggregation; Appendix C defines the recorded trajectory features, the tool interface, and the observation-policy adaptation rules. Because heuristic generation samples an API-served LLM at temperature 1.0, repeated searches may yield different heuristics; we therefore report results over independent search seeds and will release the source code, including the instance generators, together with the evolved heuristics upon publication, so that the final evaluation can be rerun directly. Since the main comparisons use wall-clock budgets that depend on hardware, we also report results under fixed evaluation budgets in Appendix D.2.

REFERENCES

Leonardo CT Bezerra, Manuel López-Ibáñez, and Thomas Stützle. Automatic component-wise design of multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 20(3):403–417, 2015.

Jinbiao Chen, Zizhen Zhang, Zhiguang Cao, Yaoxin Wu, Yining Ma, Te Ye, and Jiahai Wang. Neural multi-objective combinatorial optimization with diversity enhancement. *Advances in Neural Information Processing Systems*, 36:39176–39188, 2023.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2): 182–197, 2002.

Carlos M Fonseca and Peter J Fleming. On the performance assessment and comparison of stochastic multiobjective optimizers. In *International conference on parallel problem solving from nature*, pp. 584–593. Springer, 1996.

Michael Pilegaard Hansen and Andrzej Jaszkiewicz. Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7, Institute of Mathematical Modelling, Technical University of Denmark, 1998.

Ha Minh Hieu, Hung Phan, Tung Duy Doan, Tung Dao, Cong Dao Tran, and Huynh Thi Thanh Binh. Pareto-grid-guided large language models for fast and high-quality heuristics design in multi-objective combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 36964–36972, 2026.

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

Yuri Lavinas, Claus Aranha, and Gabriela Ochoa. Search trajectories networks of multiobjective evolutionary algorithms. In *International conference on the applications of evolutionary computation (part of EvoStar)*, pp. 223–238. Springer, 2022.

Yuri Lavinas, Marcelo Ladeira, Gabriela Ochoa, and Claus Aranha. Multiobjective evolutionary component effect on algorithm behaviour. *ACM Transactions on Evolutionary Learning and Optimization*, 4(2):1–24, 2024.

Fei Liu, Tong Xialiang, Mingxuan Yuan, Xi Lin, Fu Luo, Zhenkun Wang, Zhichao Lu, and Qingfu Zhang. Evolution of heuristics: Towards efficient automatic algorithm design using large language model. In *Forty-first International Conference on Machine Learning*, 2024.

Fei Liu, Yilu Liu, Qingfu Zhang, Tong Xialiang, and Mingxuan Yuan. Eoh-s: Evolution of heuristic set using llms for automated heuristic design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37090–37098, 2026a.

Fei Liu, Yiming Yao, Ping Guo, Zhiyuan Yang, Xi Lin, Zhe Zhao, Xialiang Tong, Kun Mao, Zhichao Lu, Zhenkun Wang, et al. A systematic survey on large language models for algorithm design. *ACM Computing Surveys*, 58(8):1–32, 2026b.

Manuel López-Ibáñez, Joshua Knowles, and Marco Laumanns. On sequential online archiving of objective vectors. In *International conference on evolutionary multi-criterion optimization*, pp. 46–60. Springer, 2011.

Zeyuan Ma, Hongshu Guo, Yue-Jiao Gong, Jun Zhang, and Kay Chen Tan. Toward automated algorithm design: A survey and practical guide to meta-black-box-optimization. *IEEE Transactions on Evolutionary Computation*, 30(2):667–687, 2025.

Gabriela Ochoa, Katherine M Malan, and Christian Blum. Search trajectory networks: A tool for analysing and visualising the behaviour of metaheuristics. *Applied Soft Computing*, 109:107492, 2021.

Gabriela Ochoa, Arnaud Liefooghe, Yuri Lavinas, and Claus Aranha. Decision/objective space trajectory networks for multi-objective combinatorial optimisation. In *European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar)*, pp. 211–226. Springer, 2023.

Junhao Qiu, Xin Chen, Liang Ge, Liyong Lin, Zhichao Lu, and Qingfu Zhang. Evolving interdependent operators with large language models for multi-objective combinatorial optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

Andreea Radulescu, Manuel López-Ibáñez, and Thomas Stützle. Automatically improving the anytime behaviour of multiobjective evolutionary algorithms. In *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 825–840. Springer, 2013.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Ke Shang, Hisao Ishibuchi, Linjun He, and Lie Meng Pang. A survey on the hypervolume indicator in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 25(1):1–20, 2020.

Yiding Shi, Jianan Zhou, Wen Song, Jieyi Bi, Yaoxin Wu, Zhiguang Cao, and Jie Zhang. Generalizable heuristic generation through llms with meta-optimization. In *International Conference on Learning Representations*, volume 2026, pp. 119314–119349, 2026.

Wen Song, Xinyang Chen, Qiqiang Li, and Zhiguang Cao. Flexible job-shop scheduling via graph neural network and deep reinforcement learning. *IEEE Transactions on Industrial Informatics*, 19(2):1600–1610, 2022.

Diederick Vermetten, Jeroen Rook, Oliver L Preuß, Jacob de Nobel, Carola Doerr, Manuel López-Ibañez, Heike Trautmann, and Thomas Bäck. Mo-iohinspector: Anytime benchmarking of multi-objective algorithms using iohprofiler. In *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 242–256. Springer, 2025.

Rongzheng Wang, Yihong Huang, Muquan Li, Jiakai Li, Di Liang, Bob Simons, Pei Ke, Shuang Liang, and Ke Qin. Rethinking llm-driven heuristic design: Generating efficient and specialized solvers via dynamics-aware optimization. *arXiv preprint arXiv:2601.20868*, 2026.

Zhenxing Xu, Yizhe Zhang, Weidong Bao, Hao Wang, Ming Chen, Haoran Ye, Wenzheng Jiang, Hui Yan, and Ji Wang. Autoep: Llms-driven automation of hyperparameter evolution for metaheuristic algorithms. In *International Conference on Learning Representations*, volume 2026, pp. 139927–139945, 2026.

Shunyu Yao, Fei Liu, Xi Lin, Zhichao Lu, Zhenkun Wang, and Qingfu Zhang. Multi-objective evolution of heuristic using large language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27144–27152, 2025.

Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, and Guojie Song. Reevo: Large language models as hyper-heuristics with reflective evolution. *Advances in neural information processing systems*, 37:43571–43608, 2024.

Fangfang Zhang, Yi Mei, Su Nguyen, and Mengjie Zhang. Survey on genetic programming and machine learning techniques for heuristic design in job shop scheduling. *IEEE Transactions on Evolutionary Computation*, 28(1):147–167, 2023.

Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.

Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132, 2003.

APPENDIX

A Benchmark and Problem Setup **14**

A.1 Multi-objective TSP 14

A.2 Multi-objective CVRP 14

A.3 Multi-objective FJSP 15

A.4 Evaluation metrics and reference points 15

A.4.1 Hypervolume 15

A.4.2 Reference Pareto front and IGD 15

B Solver Configuration **16**

B.1 Candidate-search configuration 16

B.2 Common final artifact evaluation 17

B.3 Solver-specific evaluator semantics 18

B.4 Aggregation and reporting 18

C Trajectory Observation and Policy Adaptation **18**

C.1 Recorded trajectory features 18

C.2 Tool-calling interface 19

C.3 Observer and observation-policy adaptation 20

D Additional Experiment Results **20**

D.1 Equal wall-clock evaluation at 60 seconds 20

D.2 Fixed evaluation/work budget 20

A BENCHMARK AND PROBLEM SETUP

The experiments use a fixed subset of the generated instances. Heuristic search uses three seen instances at the medium scale. Evaluation on unseen instances uses three further medium-scale instances and three instances at each of the smaller and larger scales.

The subset therefore contains 36 distinct physical problem instances: six medium-scale instances and three instances at each of the smaller and larger scales for each of the three problem families. Each physical instance defines paired bi-objective and tri-objective tasks, giving 72 task instances in total. Table 4 lists the instances used in each reported evaluation.

Table 4: Instance splits and problem scales.

Evaluation	Instances	TSP/CVRP size	FJSP size
Seen instances	3	50	10 jobs, 5 machines
Unseen medium-scale instances	3	50	10 jobs, 5 machines
Unseen smaller-scale instances	3	20	5 jobs, 5 machines
Unseen larger-scale instances	3	100	25 jobs, 5 machines

All physical instances are generated in advance from fixed random seeds. Bi-objective and tri-objective tasks share the same physical instance; they differ only in how many of its stored objectives are evaluated.

A.1 MULTI-OBJECTIVE TSP

The TSP instances contain $n \in \{20, 50, 100\}$ cities. For each physical instance, three objective-specific coordinate sets $\{x_j^{(i)}\}_{j=1}^{n}$, $i \in \{1, 2, 3\}$, are sampled independently from $\mathcal{U}([0, 1]^2)$. The bi-objective task uses the first two sets, and the tri-objective task uses all three. A feasible solution is a permutation of all cities, which is decoded as a closed tour. For a tour π, its length under objective i is

$$L_i(\pi) = \sum_{k=1}^{n} \left\| x_{\pi_k}^{(i)} - x_{\pi_{k+1}}^{(i)} \right\|_2, \qquad \pi_{n+1} = \pi_1. \tag{9}$$

The objective vectors are

$$\boldsymbol{f}^{\mathrm{bi}}(\pi) = (L_1(\pi), L_2(\pi)), \qquad \boldsymbol{f}^{\mathrm{tri}}(\pi) = (L_1(\pi), L_2(\pi), L_3(\pi)). \tag{10}$$

All objectives are minimized.

A.2 MULTI-OBJECTIVE CVRP

The CVRP instances contain $n \in \{20, 50, 100\}$ customers and one depot. Customer and depot coordinates are sampled independently from $\mathcal{U}([0, 1]^2)$, customer demands are sampled from the discrete uniform distribution on $\{1, \ldots, 9\}$, and the vehicle capacities for the three scales are 30, 40, and 50. Each problem instance has a fixed fleet size k, defined as the minimum feasible number of vehicles, computed once for each physical instance by a mixed-integer program. The same k is used by its bi-objective and tri-objective tasks.

Table 5 reports the fleet sizes of the instances used in the paper.

Table 5: CVRP capacities and fleet sizes.

Customers	Capacity	Instances	Fleet sizes k
20	30	3	$[4, 3, 4]$
50	40	6	$[7, 6, 7, 7, 7, 7]$
100	50	3	$[11, 10, 10]$

A feasible solution visits every customer exactly once, uses exactly k non-empty routes, begins and ends every route at the depot, and respects the capacity of every route. Let $\ell_1, \ldots, \ell_k$ be the

Euclidean route lengths. The objectives are

$$\boldsymbol{f}^{\mathrm{bi}} = \left(\sum_{j=1}^{k} \ell_j, \max_j \ell_j \right), \qquad \boldsymbol{f}^{\mathrm{tri}} = \left(\sum_{j=1}^{k} \ell_j, \max_j \ell_j, \sigma(\ell_1, \ldots, \ell_k) \right), \tag{11}$$

where σ is the population standard deviation.

A.3 Multi-objective FJSP

The FJSP scales contain 5, 10, or 25 jobs and $m = 5$ machines. Instances are generated with the FJSP-DRL procedure. The number of operations in each job is sampled uniformly from the integers $[\lfloor 0.8m \rfloor, \lfloor 1.2m \rfloor]$. The number of eligible machines for an operation is sampled uniformly from $\{1, \ldots, m\}$; and a nominal processing time is sampled uniformly from $\{1, \ldots, 20\}$ with a 20% deviation band. A schedule must satisfy job precedence, machine eligibility, and machine non-overlap.

Let $C_{\max}$ be makespan and W_m the workload assigned to machine m. The objectives are

$$\boldsymbol{f}^{\mathrm{bi}} = (C_{\max}, \max_m W_m), \qquad \boldsymbol{f}^{\mathrm{tri}} = (C_{\max}, \max_m W_m, \sum_m W_m). \tag{12}$$

These objective vectors match E2OC componentwise.

A.4 Evaluation metrics and reference points

A.4.1 Hypervolume

For each benchmark setting s, we fix an ideal point $\boldsymbol{z}_s$ and a hypervolume reference point $\boldsymbol{r}_s$ before method comparison. The pair is shared by all methods and all evaluated instances in that setting and is never updated from a method's output. Reference points are specified independently across problem families, scales, and objective dimensions.

The ideal point is the natural nonnegative lower bound $\boldsymbol{z}_s = \mathbf{0}$. Each $\boldsymbol{r}_s$ is a manually selected pre-evaluation engineering reference point appropriate to the corresponding generator, scale, objective definition, and objective dimension (Table 6). The TSP reference points match MPaGE at the problem scales used here. The bi-objective CVRP points at 50 and 100 customers also match MPaGE; our 20-customer point is $(30, 10)$ rather than MPaGE's $(30, 8)$. The tri-objective CVRP and generated-FJSP points are extensions defined for our benchmark and are not taken from MPaGE or E2OC.

For an approximation front F, hypervolume is first computed in the original objective units and subsequently divided by the fixed scaling-region volume:

$$\mathrm{HV}'_s(F) = \frac{\mathrm{HV}_{\boldsymbol{r}_s}(F)}{\prod_i (r_{s,i} - z_{s,i})}. \tag{13}$$

A point contributes positive volume only if it is strictly below $\boldsymbol{r}_s$ in every coordinate. For these nonnegative objectives and $\boldsymbol{z}_s = \mathbf{0}$, the scaled hypervolume lies in $[0, 1]$. Hypervolume is computed separately for every run; fronts from different instances or solver seeds are not pooled before aggregation. For SEMO, whose archive may retain dominated solutions from initialization, hypervolume is computed on the nondominated subset of the returned archive.

A.4.2 Reference Pareto front and IGD

For each evaluation subset and task instance, the reference Pareto front R is the nondominated union of fronts produced by the participating methods and seeds. Normalize objective vectors componentwise as $\tilde{\boldsymbol{f}} = (\boldsymbol{f} - \boldsymbol{z}_s)/(\boldsymbol{r}_s - \boldsymbol{z}_s)$. For a nonempty front F,

$$\mathrm{IGD}(F, R) = \frac{1}{|R|} \sum_{\boldsymbol{y} \in \tilde{R}} \min_{\boldsymbol{x} \in \tilde{F}} \|\boldsymbol{y} - \boldsymbol{x}\|_2. \tag{14}$$

For wall-clock evaluation, a separate reference union is constructed at each of 30 and 60 s; no other-time or fixed-budget fronts enter that union.

Table 6: Ideal and reference points for hypervolume in the 18 benchmark settings.

Family	Scale	Objectives	$\boldsymbol{z}_s$	$\boldsymbol{r}_s$
TSP	20 cities	2	$(0, 0)$	$(20, 20)$
TSP	20 cities	3	$(0, 0, 0)$	$(20, 20, 20)$
TSP	50 cities	2	$(0, 0)$	$(35, 35)$
TSP	50 cities	3	$(0, 0, 0)$	$(35, 35, 35)$
TSP	100 cities	2	$(0, 0)$	$(65, 65)$
TSP	100 cities	3	$(0, 0, 0)$	$(65, 65, 65)$
CVRP	20 customers	2	$(0, 0)$	$(30, 10)$
CVRP	20 customers	3	$(0, 0, 0)$	$(30, 8, 1)$
CVRP	50 customers	2	$(0, 0)$	$(45, 8)$
CVRP	50 customers	3	$(0, 0, 0)$	$(45, 8, 1)$
CVRP	100 customers	2	$(0, 0)$	$(80, 8)$
CVRP	100 customers	3	$(0, 0, 0)$	$(80, 8, 1)$
FJSP	5 jobs, 5 machines	2	$(0, 0)$	$(180, 180)$
FJSP	5 jobs, 5 machines	3	$(0, 0, 0)$	$(180, 180, 320)$
FJSP	10 jobs, 5 machines	2	$(0, 0)$	$(260, 250)$
FJSP	10 jobs, 5 machines	3	$(0, 0, 0)$	$(260, 250, 700)$
FJSP	25 jobs, 5 machines	2	$(0, 0)$	$(450, 450)$
FJSP	25 jobs, 5 machines	3	$(0, 0, 0)$	$(450, 450, 1500)$

B Solver Configuration

Candidate-search evaluation and final artifact evaluation serve different purposes and are configured separately. During automatic design, each method retains its own published outer-search organization. After a search seed produces its final heuristic, all heuristics attached to the same solver are evaluated under the same solver configuration, instances, solver seeds, stopping rule, and metric. Thus solver-level differences are explicit, while comparisons within one solver keep the inner solver unchanged.

Accordingly, the primary solver-controlled comparisons are TRACE-EA–SEMO against MPaGE and adapted MEoH, and TRACE-EA–NSGA-II against E2OC and adapted EoH. Classical NSGA-II and MOEA/D serve as performance anchors.

B.1 Candidate-search configuration

For TRACE-EA, NSGA-II evolves a batch operator, whereas SEMO evolves a neighbor-selection function. Parent selection combines quality-based sampling with sampling from grid cells along individual objective axes. For actions with multiple parents, grouping heuristics by their design logic can further diversify the selection.

The four variation actions differ in the changes they request. *Parent-guided Synthesis* combines ideas from several parents into a new structure. *Backbone Diversification* changes a design choice shared by the parents. *Structural Mutation* revises one parent's structure. *Parameter Retuning* adjusts its numerical or Boolean parameters.

Every LLM-driven method has a candidate-program budget of at most 200 for each task and search seed. The program-population sizes are 10 for TRACE-EA, MPaGE, adapted MEoH, and adapted EoH. MPaGE uses at most 20 outer generations; adapted MEoH uses at most 21 because of its different initial-promotion schedule. E2OC retains its nested search and remains subject to the same 200-candidate cap. Independent search seeds are used; their aggregation is described in Appendix B.4.

Every model response from which a program is successfully extracted counts toward the candidate budget, including initialization candidates and candidates that later fail evaluation.

Table 7: Inner-solver settings during program search.

Method	Solver	Search-time inner configuration
TRACE-EA (NSGA-II)	NSGA-II	population 100; 30 generations; three solver repeats per seen instance; every generation recorded; each candidate evaluated on the three seen instances
TRACE-EA (SEMO)	SEMO	2,000 neighbor calls; initialization size 100/10/20 for TSP/CVRP/FJSP; two solver repeats per seen instance; snapshots every 100 calls, including initial and final states; 360-second candidate wall limit
MPaGE and adapted MEoH	SEMO	2,000 neighbor calls on each of the three seen instances; identical SEMO task evaluator and neighbor-selection interface; 120-second evaluator wall limit
Adapted EoH	NSGA-II	population 100; 30 generations; three solver repeats per seen instance; evolves the mutation operator; 180-second run limit
E2OC	NSGA-II	population 100; 30 generations; three stochastic evaluations per seen instance; its nested operator-design search is capped by the common candidate budget

B.2 Common final artifact evaluation

Each final heuristic is evaluated with five solver seeds per instance under two protocols. For equal wall-clock evaluation, each run lasts 180 s, and the results at 30 and 60 s are read from the last completed generation or iteration at that time. Timing starts after data loading and solver setup, and metrics are computed outside the timed loop.

For fixed-budget evaluation, a run stops at the objective-evaluation or iteration cap appropriate to its solver, or after 3,600 seconds, whichever occurs first; if the wall limit is reached, the archive available at termination is retained. The caps in Table 8 count submitted objective evaluations for NSGA-II and MOEA/D, and initialization plus attempted neighbor calls for SEMO; the counting rule is shared within each solver.

Table 8: Per-run budgets and population sizes for final evaluation.

Task	Objectives	Run cap	NSGA-II population
TSP	2	40,000	200
TSP	3	20,000	200
CVRP	2	40,000	200
CVRP	3	20,000	200
FJSP	2	3,000	100
FJSP	3	3,000	100

MPaGE, adapted MEoH, and the SEMO variant of TRACE-EA are evaluated in the same SEMO solver. Its run cap counts initialization followed by neighbor calls; the remainder of the evaluation cap is used for neighbor calls. The solver initializes an ordered archive of 100, 10, or 20 feasible solutions for TSP, CVRP, or FJSP, respectively, without filtering dominated or duplicate members. A valid neighbor is appended if no archive member strictly dominates it; members it strictly dominates are removed. Equal-objective and duplicate solutions are retained. Initialization, acceptance, feasibility handling, and stopping are shared across the three methods; only the neighbor-selection function differs.

E2OC, adapted EoH, and the NSGA-II variant of TRACE-EA are evaluated in the same NSGA-II solver. The number of generations is set so that the initial population and all offspring evaluations fit within the evaluation cap. The family-specific representations use order crossover and inversion mutation for TSP, order crossover with swap/cut mutation for CVRP, and precedence-preserving OS/MS crossover with swap/MS mutation for FJSP. Crossover is applied with probability 0.9 and mutation with probability 1.0. The E2OC operator chain and the EoH operator replace the solver's mutation stage but not initialization, crossover, survival, stopping, or evaluation. The

classical NSGA-II baseline uses the same family-specific operators. MOEA/D uses population 150, neighborhood size 20, the same family-specific variation, and the same per-run evaluation caps.

B.3 Solver-specific evaluator semantics

The two solvers use fixed, representation-specific validity handling. In the SEMO solver, an evolved neighbor-selection function must return one valid solution in the solver's native representation. In the NSGA-II solver, generated populations use the family-specific representation, and repair is applied after an edit that does not preserve validity. These rules are fixed across all methods evaluated within the same solver.

For FJSP, the decoder is a deterministic serial schedule-generation scheme with earliest-feasible gap insertion: operations are considered in the order encoded by the operation sequence and placed at the earliest feasible time on the selected eligible machine while respecting job precedence and machine non-overlap.

B.4 Aggregation and reporting

For each AHD method, measurements are first averaged over the three instances and five solver seeds evaluated for one outer search seed. The three outer search-seed means are then averaged with equal weight. Classical baselines are averaged directly over instances and solver seeds. Each benchmark setting is reported separately.

The independent AHD replicate is the outer search seed. Repeated inner runs reduce artifact-evaluation noise and are not treated as independent AHD replicates. Reported table entries are sample means.

Reading the result tables. HV denotes the normalized hypervolume of equation 13 (higher is better). IGD uses Euclidean distances after $(F-z)/(r-z)$ scaling without clipping (lower is better). Boldface identifies the best mean within each learned solver group: E2OC, EoH and TRACE-EA for NSGA-II; MEoH, MPaGE and TRACE-EA for SEMO. Ties at four decimal places are included; traditional baselines are unmarked. TRACE-EA in each group denotes the method evaluated on that solver.

C Trajectory Observation and Policy Adaptation

C.1 Recorded trajectory features

The trajectory features do not require a reference Pareto front. Let F_t denote the nondominated front at recorded point t. We scale its objective vectors as $\tilde{F}_t = F_t \oslash \boldsymbol{r}_s$, using the fixed reference point $\boldsymbol{r}_s$ from equation 13 to maintain a common scale across instances and recorded points.

Gain and loss. We compare consecutive fronts using the binary additive ε-indicator (Zitzler et al., 2003). For normalized fronts A and B, it is defined as $I_\varepsilon(A,B) = \max_{\boldsymbol{b}\in B} \min_{\boldsymbol{a}\in A} \max_k (a_k - b_k)$. It measures the smallest uniform additive shift of B needed for A to weakly dominate it. We apply this indicator in both directions between consecutive fronts and retain its positive part:

$$\varepsilon_t^+ = \max\big(I_\varepsilon(\tilde{F}_{t-1}, \tilde{F}_t),\, 0\big), \qquad \varepsilon_t^- = \max\big(I_\varepsilon(\tilde{F}_t, \tilde{F}_{t-1}),\, 0\big). \tag{15}$$

Positive ε_t^+ indicates that the current front contains points not weakly dominated by the previous front, representing a gain in coverage; positive ε_t^- indicates the reverse, meaning a loss of coverage. Both indicators can be positive even when hypervolume increases, revealing gains and losses that an aggregate score can conceal.

Directional progress. We track changes along trade-off directions using weighted Tchebycheff utilities (Hansen & Jaszkiewicz, 1998). For a finite set Λ of weight vectors on the unit simplex, we compute

$$u_{\boldsymbol{\lambda}}(\tilde{F}) = \min_{\boldsymbol{f}\in\tilde{F}} \max_k \lambda_k f_k, \qquad \boldsymbol{\lambda} \in \Lambda. \tag{16}$$

Each utility measures the best value attained by the front under an objective weighting, with lower values indicating better attainment. Their average gives an R2-type quality indicator, which summarizes front quality across the weight vectors. We retain the individual utilities and track their changes across recorded points to identify which trade-off directions improve and which stagnate.

Front retention. We measure how much of the previous front remains at the next recorded point:

$$s_t = \frac{1}{|F_{t-1}|} \sum_{\boldsymbol{f} \in F_{t-1}} \mathbf{1}\left[\min_{\boldsymbol{g} \in F_t} \|\boldsymbol{f} - \boldsymbol{g}\|_\infty < \eta\right], \tag{17}$$

where η is a fixed matching tolerance in raw objective coordinates. A high retention rate indicates that many previous front points remain, while a low rate indicates substantial replacement. Together with the ε-based gain and loss measures, this feature helps distinguish replacement by better points from loss of previously attained coverage.

Front geometry. Spread and spacing are the mean and standard deviation of nearest-neighbor distances within the front, after rescaling each objective by the front's own range. They therefore describe the arrangement of points within a front rather than progress toward the Pareto front. We also record the front size and the number of new nondominated points at each recorded point.

Contribution concentration. For NSGA-II trajectories, we also measure how exclusive hypervolume contributions are distributed across front points. For each $\boldsymbol{f} \in F_t$, its contribution is

$$c_t(\boldsymbol{f}) = \mathrm{HV}_{\boldsymbol{r}_s}(F_t) - \mathrm{HV}_{\boldsymbol{r}_s}(F_t \setminus \{\boldsymbol{f}\}). \tag{18}$$

We record the Gini coefficient of these contributions and their top-three share

$$h_t = \frac{\sum_{i=1}^{\min(3,|F_t|)} c_{t,(i)}}{\sum_{\boldsymbol{f} \in F_t} c_t(\boldsymbol{f})}, \tag{19}$$

where $c_{t,(i)}$ denotes the i-th largest contribution, and $h_t = 0$ when the denominator is zero. A high share indicates that a few points account for most of the exclusive hypervolume contribution. Exclusive contributions are not additive, so $\sum_{\boldsymbol{f}} c_t(\boldsymbol{f}) \le \mathrm{HV}_{\boldsymbol{r}_s}(F_t)$ in general, and this share is therefore taken relative to that sum rather than to the front's hypervolume.

Population diversity. Population diversity is the normalized mean pairwise Hamming distance between encoded solutions, estimated from at most 200 sampled pairs.

C.2 Tool-calling interface

The observer accesses trajectories only through the eleven read-only tools in Table 9. The candidate argument selects one of the parents supplied to the observer.

Table 9: Trajectory tools, main arguments, and returned evidence.

Tool	Main arguments	Returned evidence
`overview`	candidate	Downsampled representative trajectory with the mean and standard deviation of final HV across runs.
`scan_findings`	candidate	Detected failure modes across runs, with severity, temporal range, and number of supporting runs.
`zoom`	candidate, time window, metrics	Recorded values of the selected metrics within the window.
`change_points`	candidate, metric, k	The k recorded intervals with the largest change in the metric.
`correlate`	candidate, two metrics, maximum lag	Lagged correlations between the two metrics.
`front`	candidate, recorded point, k, selection mode	k selected front points (by HV contribution, extremes, or knee) with their HV contribution shares.
`front_shape`	candidate, recorded point	Front size, spacing, spread, grid occupancy, objective correlation, and, where defined, contribution concentration, convexity, knee, and largest gap.
`region_progress`	candidate, time window	Change in each direction utility, and objective-space grid cells gained and lost.
`operator_stats`	candidate, time window	Per-operator statistics for NSGA-II, or call, feasibility, acceptance, duplicate, and HV-gain statistics of the neighbor-selection function for SEMO.
`compare_repeats`	candidate	Per-run final HV, front size, and diversity, with their mean, standard deviation, and range.
`compare_candidates`	two candidates	Differences in final metrics and failure modes specific to either candidate.

C.3 Observer and observation-policy adaptation

The policy $\varphi = (w, n_{\text{rounds}}, \boldsymbol{\theta})$ of equation 6 is initialized with $w = 10$, $n_{\text{rounds}} = 4$, and unit weight for every tool prior. Two display settings remain fixed: trajectory summaries use eight points, and front queries return six points by default.

An observation returns a diagnosis, a suggested function to modify, the evidence items retrieved by the tools, and the items cited by the diagnosis. Adaptation uses only outer-search steps that contained an observation and produced a scored candidate, with quality change Δq_i from equation 7.

Each adaptation call first verifies the most recent unresolved policy change and otherwise changes at most one component, in the order below.

- *Verification.* The change is assessed once at least four subsequent steps are available. Their mean action-adjusted residual equation 8 is compared with that of the last four pre-change steps, and the change is reverted if the mean decreases. Actions without pre-change occurrences are excluded.
- *Temporal window.* Steps are split by whether their queried window was shorter than w. If both groups contain at least three steps and their mean Δq differ, w moves by two toward the better group, within $[3, 20]$.
- *Query rounds.* Over at least four LLM-observed steps, n_{rounds} decreases by one if fewer than half of the retrieved items are cited on average, and increases by one if at least 90% are cited and at least half of the steps exhaust the allowance, within $[2, 6]$.
- *Tool priors.* Given at least four eligible steps, the prior of each cited tool is multiplied by 1.2 after a positive Δq and by 0.9 otherwise, within $[0.2, 5]$.

Tool priors define the sampling distribution in the random-tool ablation and are presented to the LLM observer as a soft preference; no tool is disabled by adaptation.

D Additional Experiment Results

The main text reports the 30-second checkpoint. We first give the 60-second results from the same wall-clock runs, followed by the separate fixed evaluation/work-budget protocol.

D.1 Equal wall-clock evaluation at 60 seconds

Tables 10 and 11 report the same settings at 60 s. The pattern at 30 s is largely preserved. Within the NSGA-II group, TRACE-EA leads CVRP HV in every evaluation, whereas EoH leads or ties Tri-FJSP. Within the SEMO group, the advantages of TRACE-EA remain concentrated in TSP, CVRP, and Tri-FJSP, with MPaGE leading only the smaller-scale Tri-TSP and MEoH leading Bi-FJSP.

D.2 Fixed evaluation/work budget

Tables 12 and 13 report the protocol of Table 8. The numerical task cap is shared across methods and scales, but its unit is submitted evaluations on NSGA-II/MOEA/D and initialization plus attempted neighbor calls on SEMO. Some SEMO-hosted TSP/CVRP runs reached the 3,600 s safety cap before exhausting that budget. IGD uses the fixed-budget reference fronts.

Under this budget, TRACE-EA leads more broadly than under equal wall-clock time. With NSGA-II, TRACE-EA attains the best HV and IGD within its group in every setting, including all FJSP settings where its wall-clock advantage is selective. With SEMO, TRACE-EA leads most settings and all larger-scale ones; its main limitation is transfer to smaller instances, where MPaGE leads Bi-CVRP, Bi-FJSP, and Tri-TSP. The contrast with the wall-clock results indicates that quality under a fixed evaluation budget and quality under a fixed time budget can rank methods differently.

Table 10: Equal wall-clock time results at 60 s on seen instances.

Seen instances: medium scale												
Method	Bi-TSP50		Tri-TSP50		Bi-CVRP50		Tri-CVRP50		Bi-FJSP10j5m		Tri-FJSP10j5m	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
Traditional methods												
NSGA-II	0.6250	0.0133	0.3167	0.1254	0.3037	0.2683	0.2487	0.3018	0.3802	0.0193	0.1382	0.0066
MOEA/D	0.5941	0.0323	0.3175	0.1019	0.3338	0.2314	0.2863	0.2354	0.3726	0.0265	0.1365	0.0125
NSGA-II-based methods												
E2OC	0.5795	0.0461	0.2721	0.1592	0.3200	0.2508	0.2883	0.2555	**0.3866**	**0.0138**	0.1370	0.0101
EoH	0.5255	0.0885	0.3573	0.0816	0.3663	0.1879	0.3137	0.2253	0.3810	0.0181	**0.1388**	**0.0069**
TRACE-EA (ours)	**0.6360**	**0.0068**	**0.3886**	**0.0767**	**0.5405**	**0.0075**	**0.5042**	**0.0371**	0.3862	**0.0138**	0.1386	0.0074
SEMO-based methods												
MEoH	0.6192	0.0181	0.3798	0.0673	0.5369	0.0103	0.5018	0.0465	**0.3744**	**0.0272**	0.1326	0.0255
MPaGE	0.6132	0.0226	0.3832	0.0593	0.5370	0.0096	0.5137	0.0297	0.3633	0.0415	0.1334	0.0233
TRACE-EA (ours)	**0.6374**	**0.0050**	**0.4458**	**0.0170**	**0.5419**	**0.0077**	**0.5265**	**0.0086**	0.3663	0.0350	**0.1362**	**0.0123**

HV ↑; IGD ↓. Bold: best within each learned solver group. Reading conventions: Appendix B.4.

Table 11: Equal wall-clock time results at 60 s on unseen instances across three scales.

(a) TSP

Method	Bi-TSP20		Bi-TSP50		Bi-TSP100		Tri-TSP20		Tri-TSP50		Tri-TSP100	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
Traditional methods												
NSGA-II	0.6200	0.0005	0.6340	0.0117	0.6044	0.0809	0.4627	0.0085	0.3165	0.1300	0.2050	0.2934
MOEA/D	0.6189	0.0020	0.6023	0.0344	0.5506	0.1276	0.4474	0.0131	0.3175	0.1068	0.2304	0.2168
NSGA-II-based methods												
E2OC	**0.6199**	**0.0009**	0.5867	0.0464	0.5186	0.1414	0.4518	0.0162	0.2728	0.1623	0.1773	0.3224
EoH	0.5892	0.0214	0.5279	0.0903	0.4729	0.2032	**0.4622**	**0.0052**	0.3606	0.0856	0.2525	0.2262
TRACE-EA (ours)	0.6195	0.0010	**0.6425**	**0.0070**	**0.6946**	**0.0284**	0.4447	0.0222	**0.3928**	**0.0825**	**0.3985**	**0.1202**
SEMO-based methods												
MEoH	0.6178	**0.0028**	0.6271	0.0178	0.6447	0.0705	0.4444	0.0170	0.3804	0.0742	0.3858	0.1186
MPaGE	0.6176	0.0034	0.6179	0.0250	0.6302	0.0752	**0.4606**	**0.0095**	0.3874	0.0617	0.3460	0.1356
TRACE-EA (ours)	**0.6194**	0.0042	**0.6442**	**0.0052**	**0.7068**	**0.0051**	0.4597	0.0164	**0.4514**	**0.0170**	**0.4940**	**0.0342**

(b) CVRP

Method	Bi-CVRP20		Bi-CVRP50		Bi-CVRP100		Tri-CVRP20		Tri-CVRP50		Tri-CVRP100	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
Traditional methods												
NSGA-II	0.5832	0.0068	0.3414	0.2587	0.1405	0.5995	0.5373	0.0097	0.2876	0.2900	0.0939	0.6574
MOEA/D	0.5838	0.0074	0.3631	0.2304	0.1511	0.5775	0.5333	0.0153	0.3082	0.2459	0.0982	0.6378
NSGA-II-based methods												
E2OC	0.5828	0.0080	0.3552	0.2411	0.1515	0.5789	0.5325	0.0153	0.3247	0.2462	0.1089	0.6251
EoH	0.5836	0.0071	0.3950	0.1930	0.1575	0.5669	0.5386	**0.0075**	0.3401	0.2323	0.1097	0.6222
TRACE-EA (ours)	**0.5869**	**0.0044**	**0.5694**	**0.0189**	**0.2611**	**0.4031**	**0.5392**	0.0078	**0.5129**	**0.0579**	**0.2011**	**0.4574**
SEMO-based methods												
MEoH	0.5831	0.0069	0.5715	0.0162	0.6123	0.0159	0.5326	0.0161	0.5248	0.0529	0.5715	0.0647
MPaGE	0.5832	**0.0053**	0.5745	**0.0135**	0.6121	0.0160	0.5340	0.0137	0.5453	0.0306	0.5898	0.0550
TRACE-EA (ours)	**0.5838**	0.0062	**0.5773**	0.0139	**0.6241**	**0.0076**	**0.5373**	**0.0125**	**0.5570**	**0.0177**	**0.6071**	**0.0220**

(c) FJSP

Method	Bi-FJSP5j5m		Bi-FJSP10j5m		Bi-FJSP25j5m		Tri-FJSP5j5m		Tri-FJSP10j5m		Tri-FJSP25j5m	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
Traditional methods												
NSGA-II	0.5094	0.0104	0.3081	0.0262	0.2074	0.0302	0.1729	0.0037	0.0800	0.0071	0.0557	0.0044
MOEA/D	0.4991	0.0200	0.2980	0.0403	0.2000	0.0421	0.1703	0.0102	0.0780	0.0159	0.0541	0.0095
NSGA-II-based methods												
E2OC	0.5045	0.0174	**0.3164**	**0.0178**	**0.2160**	**0.0179**	0.1713	0.0084	0.0788	0.0118	0.0546	0.0081
EoH	0.5069	0.0137	0.3077	0.0281	0.2080	0.0293	**0.1732**	**0.0030**	**0.0803**	**0.0068**	**0.0556**	**0.0047**
TRACE-EA (ours)	**0.5133**	**0.0090**	0.3146	0.0210	0.2079	0.0315	0.1721	0.0063	**0.0803**	0.0075	0.0502	0.0240
SEMO-based methods												
MEoH	**0.5069**	**0.0164**	**0.3061**	**0.0331**	**0.2050**	**0.0343**	0.1682	0.0174	0.0760	0.0225	0.0529	0.0138
MPaGE	0.4984	0.0227	0.2911	0.0548	0.1969	0.0493	0.1687	0.0153	0.0761	0.0210	0.0516	0.0190
TRACE-EA (ours)	0.5002	0.0209	0.2951	0.0449	0.1982	0.0446	**0.1689**	**0.0137**	**0.0769**	**0.0142**	**0.0548**	**0.0064**

HV ↑; IGD ↓. Bold: best within each learned solver group. Reading conventions: Appendix B.4.

Table 12: Fixed evaluation/work-budget results on seen instances.

Seen instances: medium scale												
Method	Bi-TSP50		Tri-TSP50		Bi-CVRP50		Tri-CVRP50		Bi-FJSP10j5m		Tri-FJSP10j5m	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
Traditional methods												
NSGA-II	0.4415	0.1497	0.1725	0.2541	0.4127	0.1386	0.3145	0.2210	0.3522	0.0569	0.1268	0.0416
MOEA/D	0.5340	0.0764	0.2162	0.1885	0.4919	0.0527	0.4258	0.0899	0.3533	0.0506	0.1243	0.0386
NSGA-II-based methods												
E2OC	0.3962	0.1874	0.1494	0.2876	0.4050	0.1495	0.3285	0.2087	0.3465	0.0674	0.1262	0.0459
EoH	0.4848	0.1110	0.2218	0.1924	0.4065	0.1396	0.3601	0.1786	0.3528	0.0574	0.1267	0.0448
TRACE-EA (ours)	**0.6344**	**0.0079**	**0.3747**	**0.0819**	**0.5426**	**0.0057**	**0.5240**	**0.0125**	**0.3845**	**0.0119**	**0.1381**	**0.0069**
SEMO-based methods												
MEoH	0.6301	0.0088	0.3982	0.0506	0.5386	0.0084	0.5109	0.0385	**0.3590**	0.0444	0.1298	0.0290
MPaGE	0.6308	0.0087	0.4247	0.0281	0.5398	0.0075	0.5173	0.0241	0.3537	0.0505	0.1301	0.0279
TRACE-EA (ours)	**0.6386**	**0.0039**	**0.4484**	**0.0158**	**0.5426**	**0.0070**	**0.5281**	**0.0083**	0.3582	**0.0423**	**0.1311**	**0.0228**

HV ↑; IGD ↓. Bold: best within each learned solver group. Reading conventions: Appendix B.4. Budgets use solver-specific units; some runs are time-truncated (Appendix D.2).

Table 13: Fixed evaluation/work-budget results on unseen instances across three scales.

(a) TSP

Method	Bi-TSP20		Bi-TSP50		Bi-TSP100		Tri-TSP20		Tri-TSP50		Tri-TSP100	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
Traditional methods												
NSGA-II	0.6063	0.0127	0.4441	0.1518	0.3349	0.2945	0.3961	0.0453	0.1767	0.2547	0.0956	0.4327
MOEA/D	0.6120	0.0059	0.5402	0.0770	0.4727	0.1990	0.4079	0.0291	0.2176	0.1918	0.1368	0.3364
NSGA-II-based methods												
E2OC	0.5789	0.0321	0.4022	0.1878	0.2995	0.3325	0.3630	0.0685	0.1544	0.2839	0.0853	0.4548
EoH	0.5800	0.0274	0.4879	0.1127	0.3714	0.2618	0.4273	0.0237	0.2206	0.1999	0.1127	0.3937
TRACE-EA (ours)	**0.6194**	**0.0012**	**0.6403**	**0.0089**	**0.6865**	**0.0395**	**0.4409**	**0.0232**	**0.3797**	**0.0890**	**0.3590**	**0.1468**
SEMO-based methods												
MEoH	0.6188	**0.0016**	0.6377	0.0092	0.6814	0.0385	0.4526	0.0112	0.3985	0.0585	0.4207	0.0930
MPaGE	0.6181	0.0026	0.6371	0.0087	0.6862	0.0263	**0.4635**	**0.0051**	0.4295	0.0297	0.4305	0.0754
TRACE-EA (ours)	**0.6196**	0.0035	**0.6454**	**0.0039**	**0.7092**	**0.0041**	0.4597	0.0161	**0.4541**	**0.0155**	**0.5159**	**0.0180**

(b) CVRP

Method	Bi-CVRP20		Bi-CVRP50		Bi-CVRP100		Tri-CVRP20		Tri-CVRP50		Tri-CVRP100	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
Traditional methods												
NSGA-II	0.5803	0.0094	0.4358	0.1513	0.3309	0.3178	0.5084	0.0413	0.3536	0.2174	0.2315	0.4190
MOEA/D	0.5799	0.0086	0.5189	0.0653	0.4860	0.1394	0.5274	0.0199	0.4347	0.1110	0.3573	0.2271
NSGA-II-based methods												
E2OC	0.5785	0.0113	0.4298	0.1584	0.3299	0.3175	0.5114	0.0387	0.3714	0.2002	0.2587	0.3788
EoH	0.5796	0.0104	0.4366	0.1463	0.3429	0.2955	0.5128	0.0389	0.3904	0.1860	0.2534	0.3884
TRACE-EA (ours)	**0.5869**	**0.0043**	**0.5734**	**0.0175**	**0.6274**	**0.0079**	**0.5392**	**0.0084**	**0.5385**	**0.0378**	**0.6016**	**0.0287**
SEMO-based methods												
MEoH	0.5837	0.0064	0.5727	0.0152	0.6230	0.0101	0.5346	0.0146	0.5372	0.0441	0.5914	0.0429
MPaGE	**0.5846**	**0.0053**	**0.5786**	**0.0105**	0.6221	0.0106	0.5337	0.0145	0.5493	0.0298	0.6052	0.0376
TRACE-EA (ours)	0.5842	0.0055	0.5781	0.0137	**0.6286**	**0.0070**	**0.5373**	**0.0131**	**0.5604**	**0.0184**	**0.6199**	**0.0085**

(c) FJSP

Method	Bi-FJSP5j5m		Bi-FJSP10j5m		Bi-FJSP25j5m		Tri-FJSP5j5m		Tri-FJSP10j5m		Tri-FJSP25j5m	
	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓	HV ↑	IGD ↓
Traditional methods												
NSGA-II	0.4912	0.0351	0.2748	0.0733	0.1907	0.0589	0.1672	0.0202	0.0695	0.0521	0.0471	0.0388
MOEA/D	0.4831	0.0354	0.2803	0.0588	0.1913	0.0543	0.1637	0.0221	0.0682	0.0419	0.0440	0.0435
NSGA-II-based methods												
E2OC	0.4880	0.0400	0.2689	0.0847	0.1855	0.0691	0.1659	0.0230	0.0704	0.0485	0.0463	0.0440
EoH	0.4900	0.0361	0.2751	0.0738	0.1885	0.0632	0.1668	0.0209	0.0707	0.0459	0.0464	0.0420
TRACE-EA (ours)	**0.5108**	**0.0114**	**0.3130**	**0.0160**	**0.2131**	**0.0205**	**0.1719**	**0.0072**	**0.0799**	**0.0075**	**0.0546**	**0.0085**
SEMO-based methods												
MEoH	0.4897	0.0330	**0.2827**	0.0573	0.1915	0.0548	0.1664	0.0197	0.0719	0.0351	0.0479	0.0301
MPaGE	**0.4901**	**0.0299**	0.2764	0.0671	0.1878	0.0615	0.1664	**0.0188**	0.0722	0.0372	0.0480	0.0321
TRACE-EA (ours)	0.4863	0.0355	0.2822	**0.0544**	**0.1927**	**0.0515**	**0.1666**	**0.0188**	**0.0744**	**0.0275**	**0.0507**	**0.0203**

HV ↑; IGD ↓. Bold: best within each learned solver group. Reading conventions: Appendix B.4. Budgets use solver-specific units; some runs are time-truncated (Appendix D.2).