

ReEvo: Large Language Models as Hyper-Heuristics with Reflective Evolution

Haoran Ye¹, Jiarui Wang², Zhiguang Cao³, Federico Berto⁴, Chuanbo Hua⁴, Haeyeon Kim⁴, Jinkyoo Park⁴, Guojie Song¹



Project website

¹PKU ²SEU ³SMU ⁴KAIST *AI4CO

Background

- **Heuristics** are problem-solving methods that use practical shortcuts to produce good solutions quickly when exact solutions are too costly to compute.
- NP-hard **combinatorial optimization problems (COPs)** traditionally required manual heuristic design through trial and error.
- **Hyper-heuristics (HHs)** automate heuristic design using techniques like genetic programming.
- The rise of **large language models (LLMs)** opens up new possibilities for hyper-heuristics.

Language Hyper-Heuristics (LHHs)

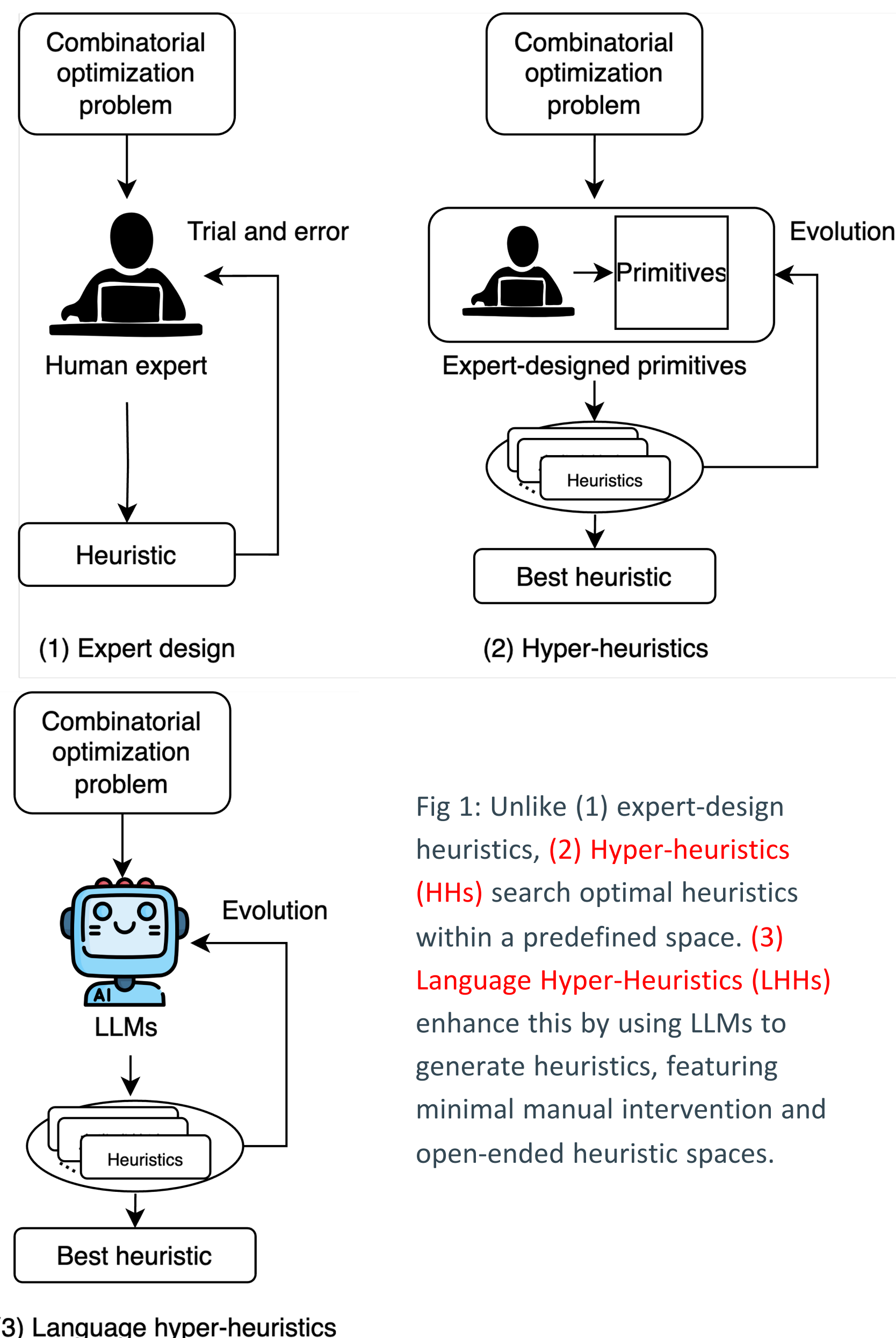


Fig 1: Unlike (1) expert-design heuristics, (2) Hyper-heuristics (HHs) search optimal heuristics within a predefined space. (3) Language Hyper-Heuristics (LHHs) enhance this by using LLMs to generate heuristics, featuring minimal manual intervention and open-ended heuristic spaces.

Reflective Evolution (ReEvo)

- ReEvo aims to elicit the power of LHH.
- ReEvo couples evolutionary search for efficiently exploring the heuristic space, and LLM reflections to provide verbal gradients (guiding information) within the space.
- ReEvo emulates human experts by reflecting on the relative performance of two heuristics and gathering insights across iterations.

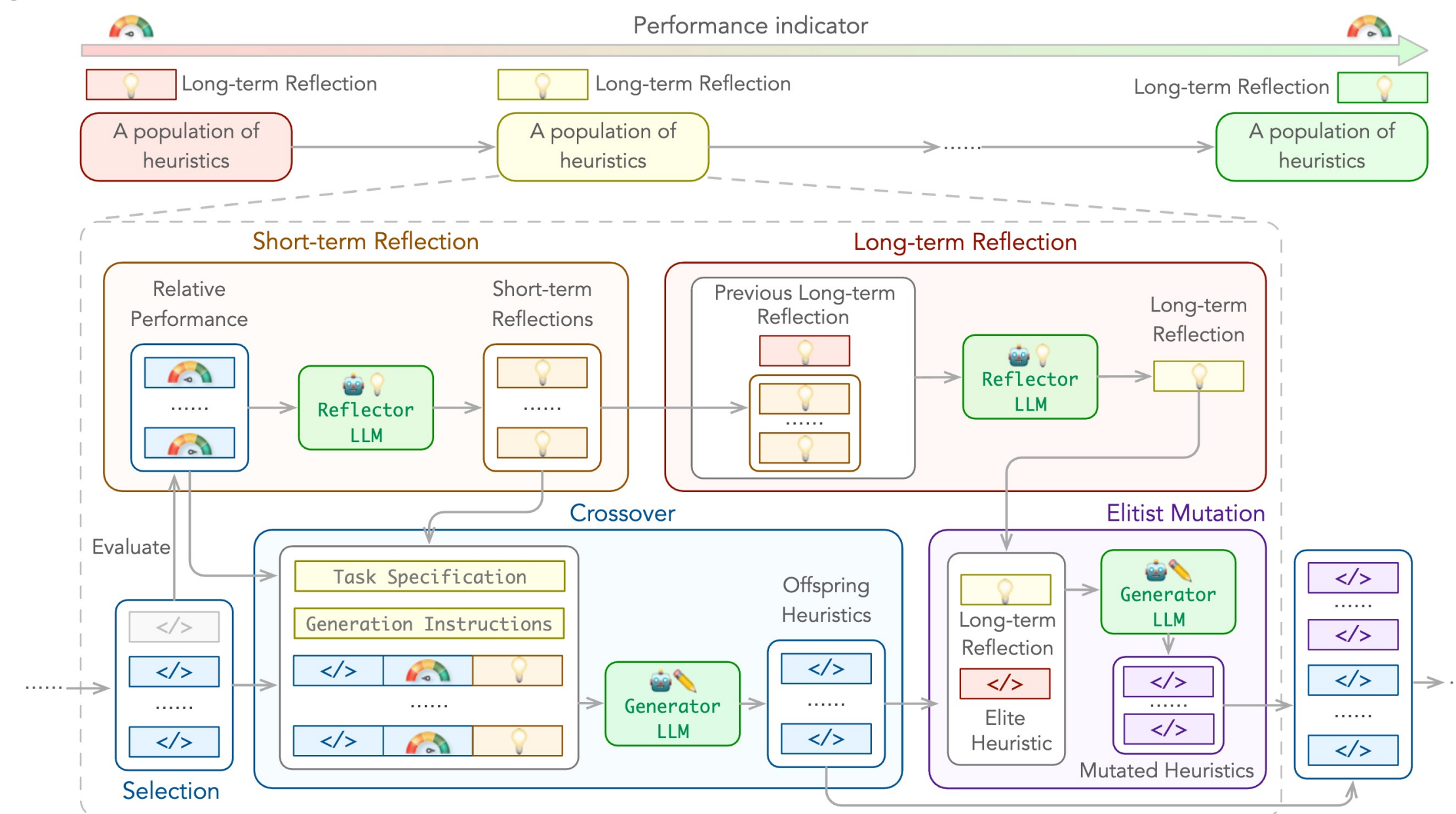


Fig 2: **ReEvo pipeline**. **Top**: ReEvo evolves a population of heuristics. Insights and knowledge are verbalized as long-term reflections and accumulated throughout iterations. **Bottom**: A ReEvo iteration contains five sequential steps: selection, short-term reflection, crossover, long-term reflection, and elitist mutation.

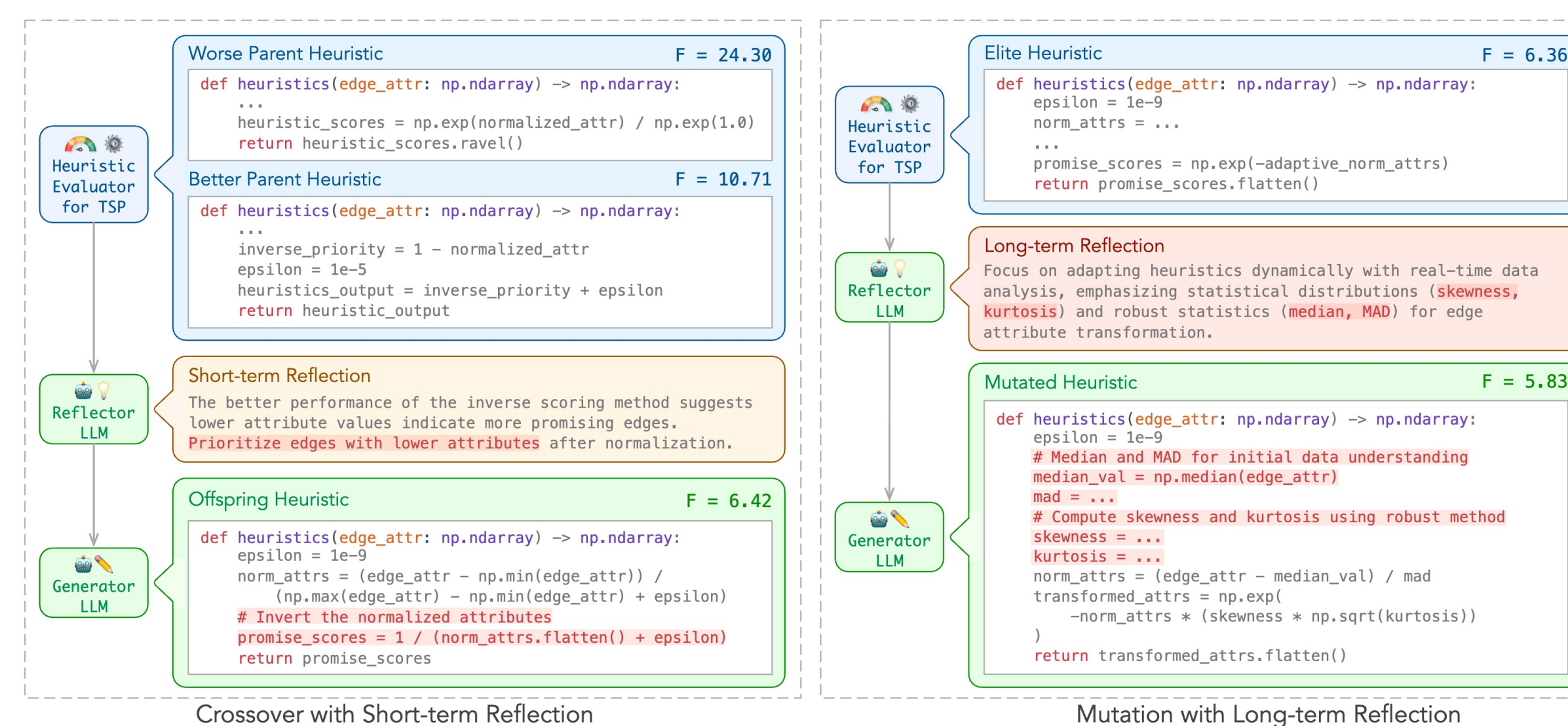


Fig 3: Examples of reflections for **black-box TSP**. Heuristics are designed for Ant Colony Optimization. **Left**: Given a pair parent heuristics, ReEvo correctly infers the TSP objective and generates a better offspring accordingly. **Right**: Given the elite heuristic and accumulated long-term reflections, ReEvo incorporates the suggested statistics and yields a better mutated heuristic.

Experiments

(1) ReEvo for SotA heuristics of COP

ReEvo improves the following types of algorithms:

- Neural Combinatorial Optimization (NCO)
 - Genetic Algorithm (GA)
 - Ant Colony Optimization (ACO)
 - Guided Local Search (GLS)
 - Constructive Heuristics
- on the following problems:
- Traveling Salesman Problem (TSP)
 - Capacitated Vehicle Routing Problem (CVRP)
 - Orienteering Problem (OP)
 - Multiple Knapsack Problems (MKP)
 - Bin Packing Problem (BPP)
 - Decap Placement Problem (DPP)

using both black-box and white-box prompts.

E.g.

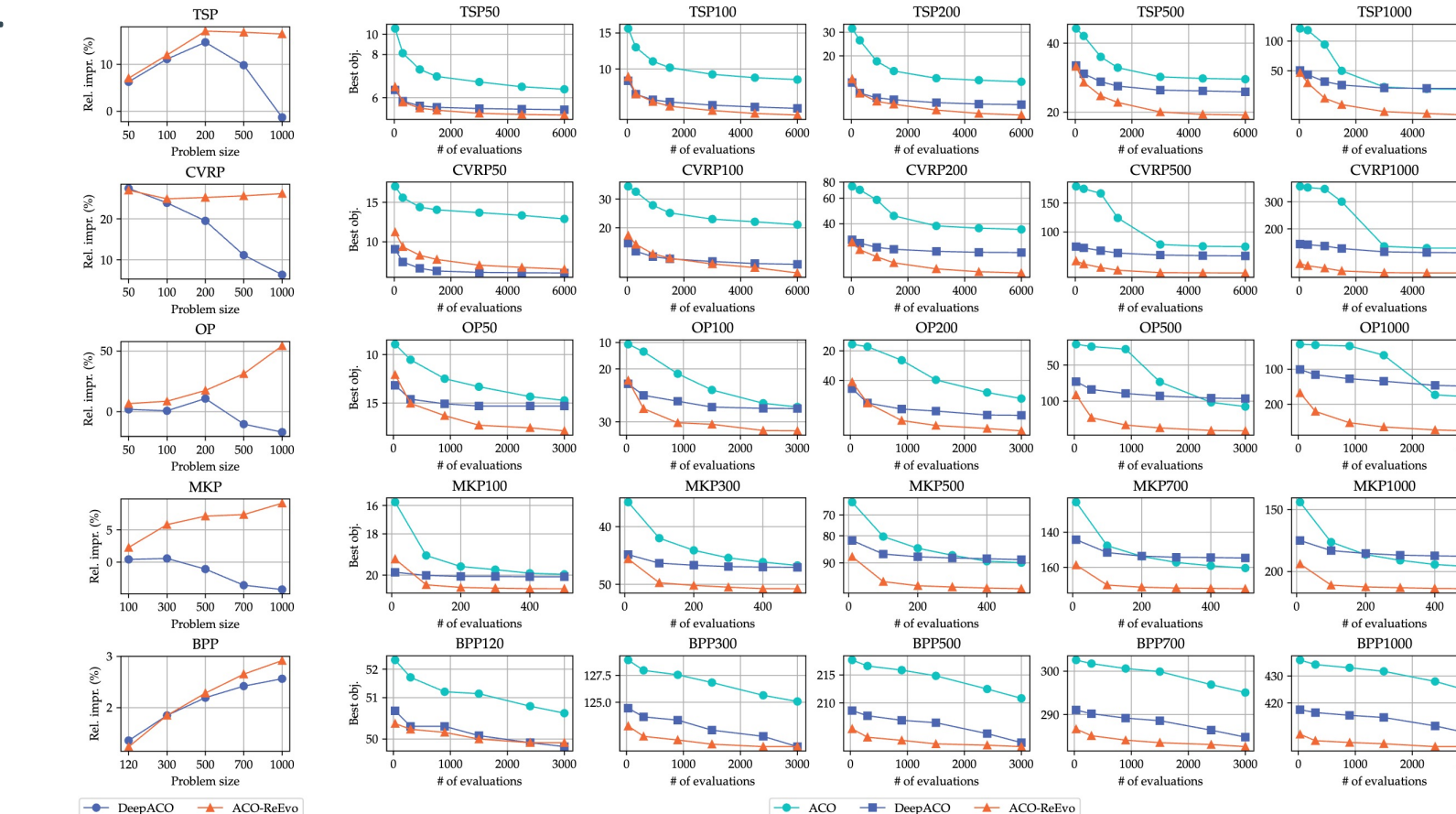


Fig 4: ReEvo-enhanced ACO beats DeepACO [1] and expert designs.

(2) ReEvo for better sample efficiency of LHH

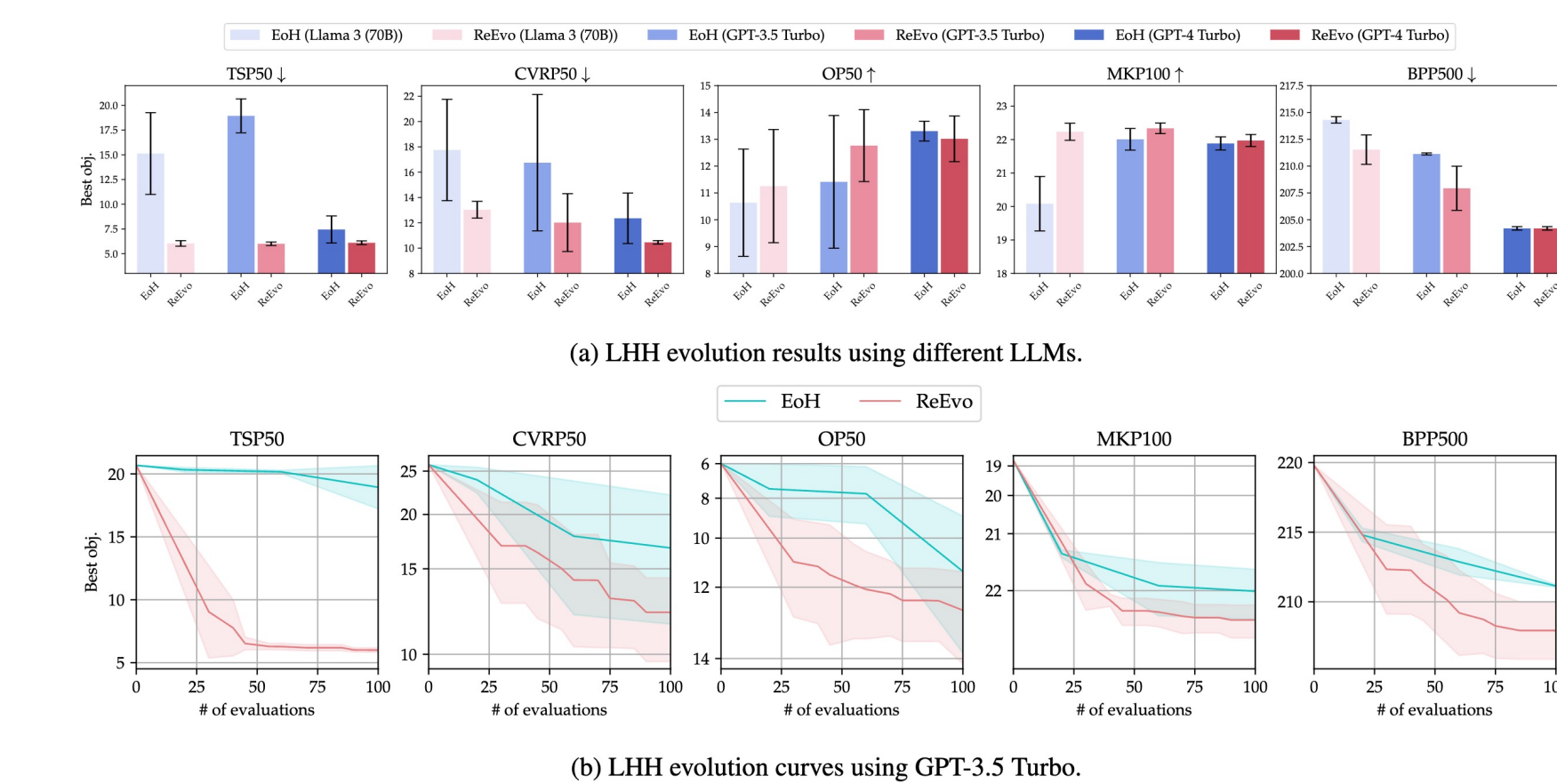


Fig 5: ReEvo shows better sample efficiency than EoH [2].