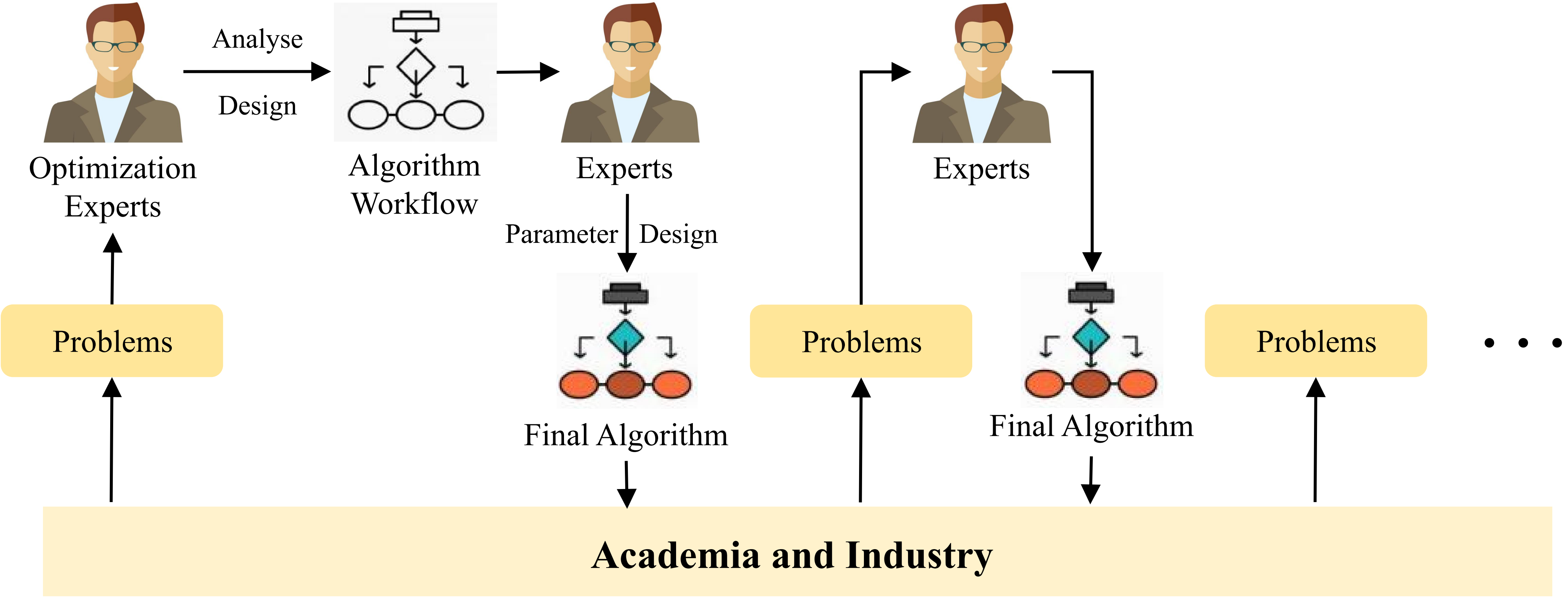
South China University of Technology

Massachusetts Institute of Technology

DesignX: Human-Competitive Algorithm Designer for Black-Box Optimization

Hongshu Guo, Zeyuan Ma, Yining Ma, Xinglin Zhang, Wei-Neng Chen, Yue-Jiao Gong

Background: Existing Problem Solving Process

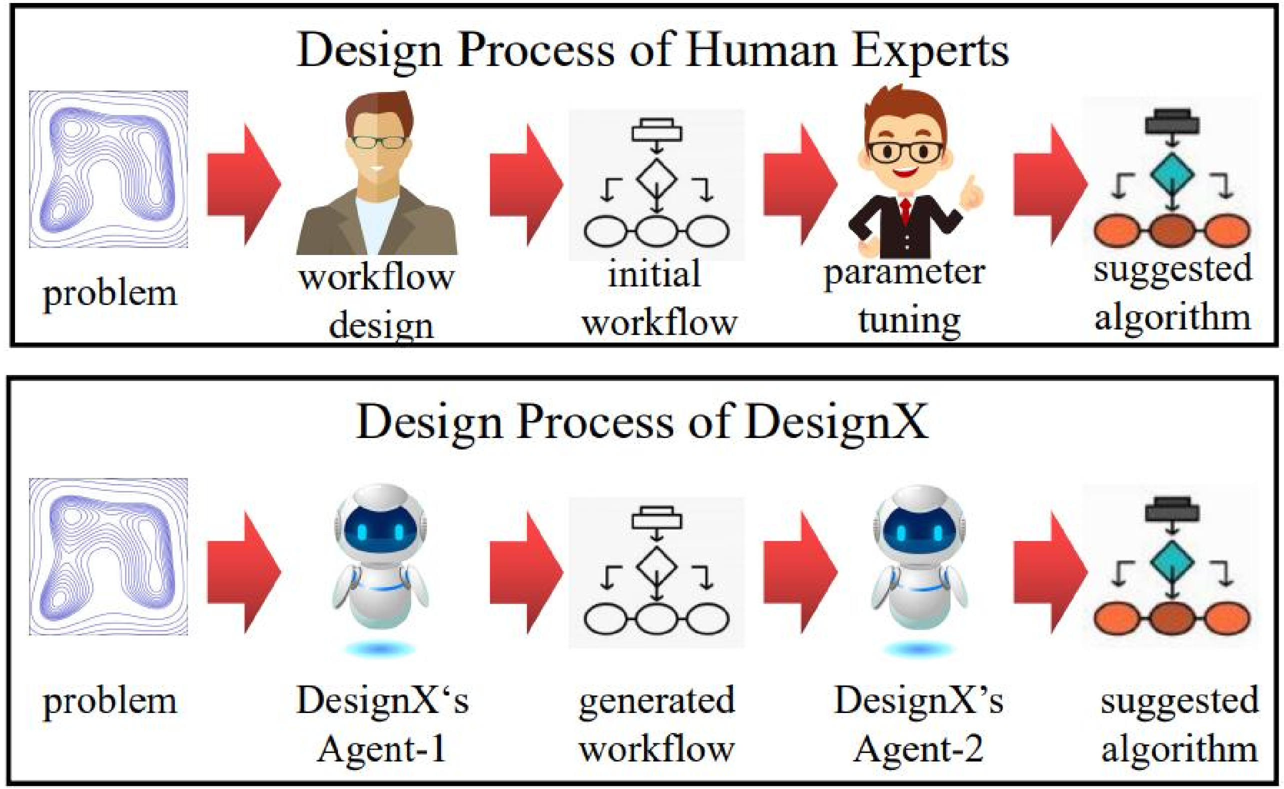


- ➢ The huge number of optimization problems in the world require a huge number of experts to handle;
- ➢ We need an automatic generalizable algorithm design and configuration framework!!!

South China University of Technology

Massachusetts Institute of Technology

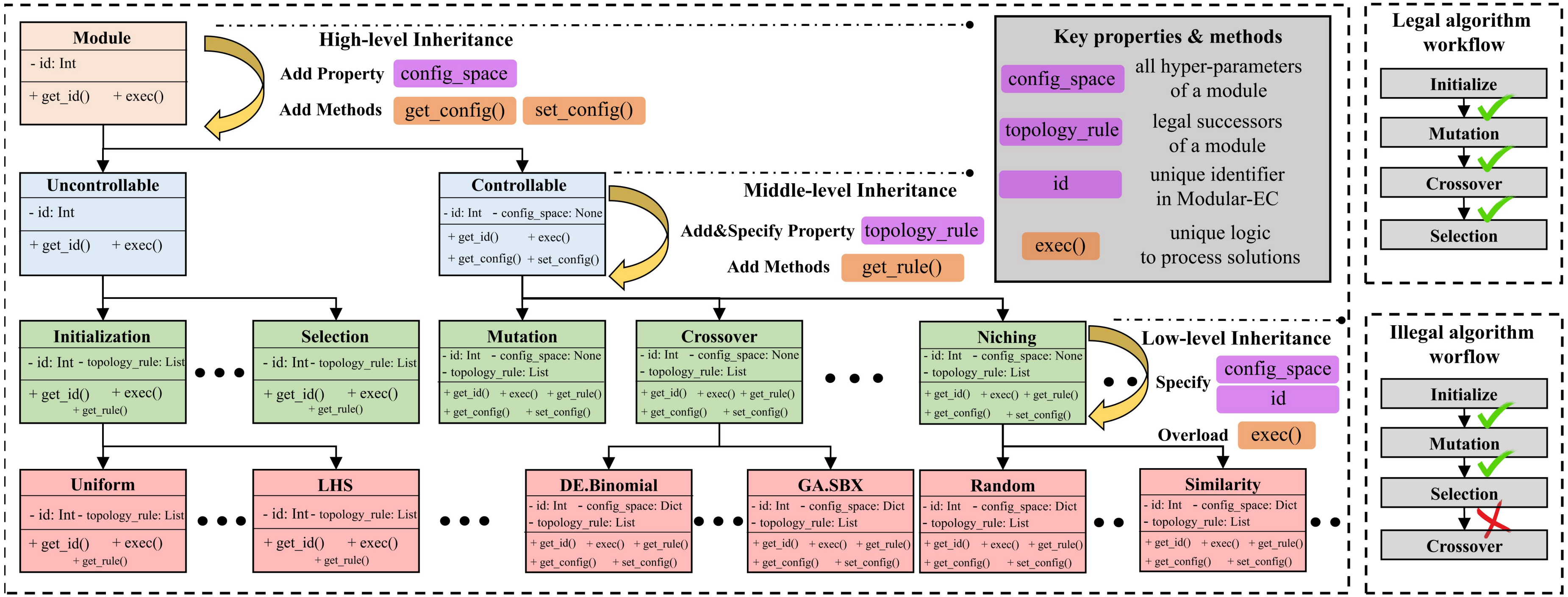
Motivation



- By integrating algorithm modularization system and Deep RL, Agent-1 achieves automatic algorithm design;
- By using Deep RL-based generalizale algorithm configuration framework Agent-2, algorithms with various different structures can be uniformly controlled.

South China University of Technology
Massachusetts Institute of Technology

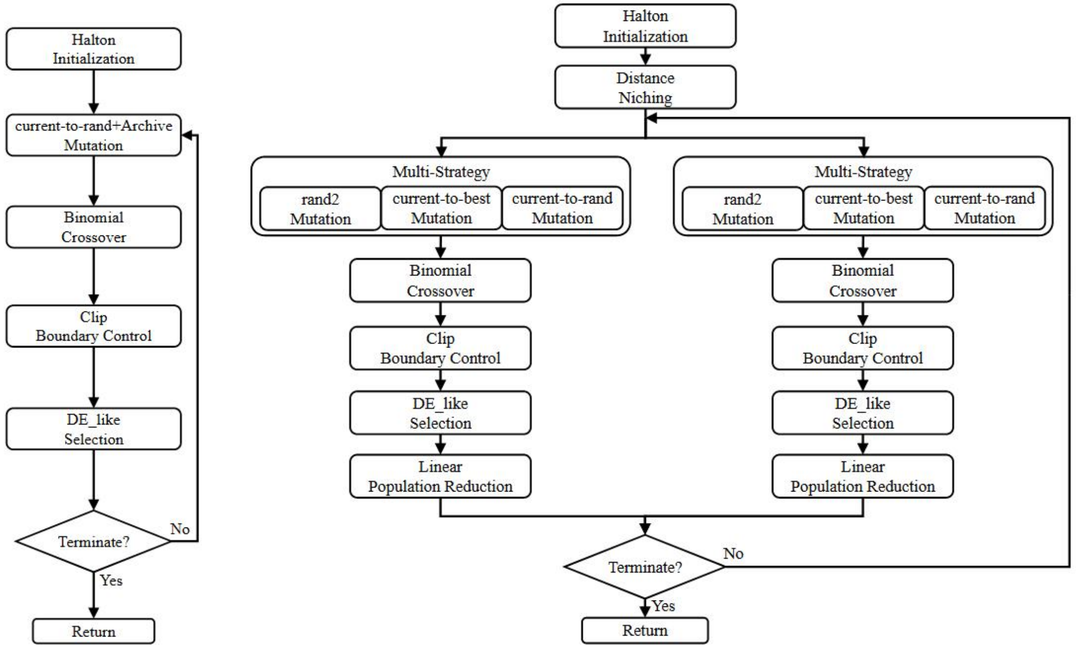
Modular-EC



- We propose Modular-EC, a Polymorphism system via multiple levels of Python inheritance;
- Module ids are assigned to identify different modules, uniform action space is achieved by config_space attribute;
- Topological rules between low-level classes are designed to ensure legal algorithm workflow.

Modular-EC

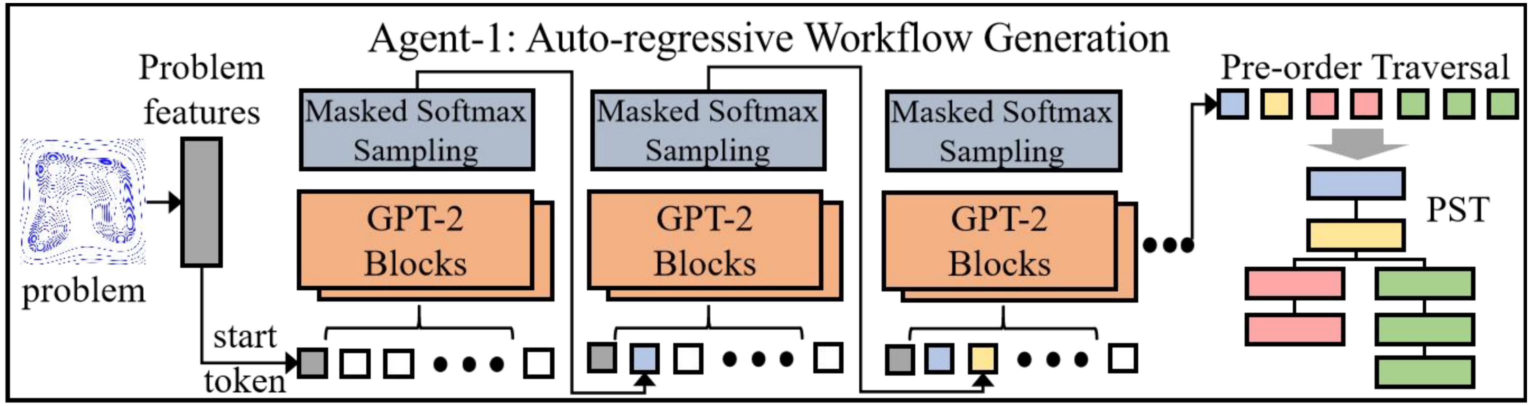
- Modular-EC maintains10 module types: INITIALIZATION, MUTATION, CROSSOVER, Other_UPDATE, BOUNDARY_CONTROL, SELECTION, NICHING, INFORMATION_SHARING, RESTRT_STRATEGY, POPULATION_REUCTION;
- Over 100 module variants covering commonly used techniques in existing EC literature;
- This results in millions of possible algorithm workflows, significantly enhancing the expressiveness of Modular-EC.

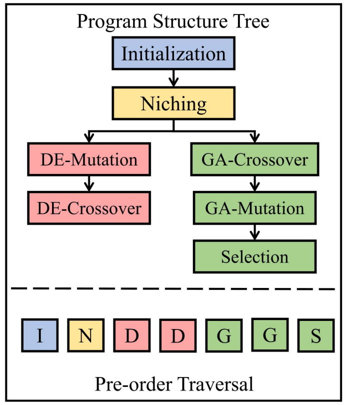


South China University of Technology

Massachusetts Institute of Technology

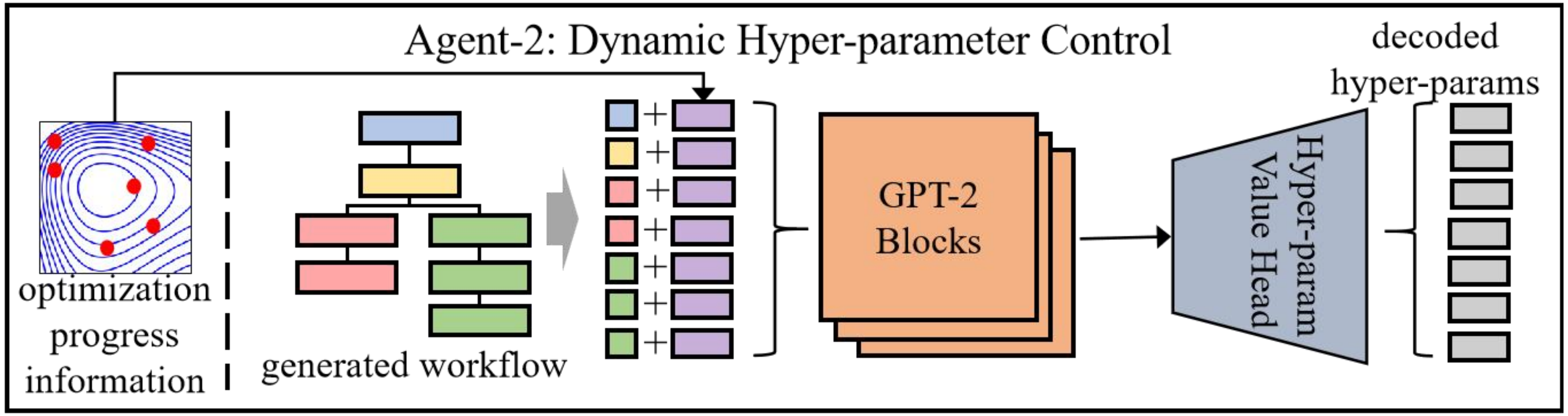
Dual-agent Algorithm Design System: Agent-1





- Use Exploratory Landscape Analysis (ELA) to calculate the problem features from sampling;
- Take the ELA feature embeddings as start token, Agent-2 selects the token of the next module (action) and constructs the preorder traversal module sequence auto-regressively using GPT-2 based attention network. Turn the sequence into a program structure tree;
- Masks are generated based on the topological rules of the modules to ensure that only legal subsequent modules will be sampled.

Dual-agent Algorithm Design System: Agent-2

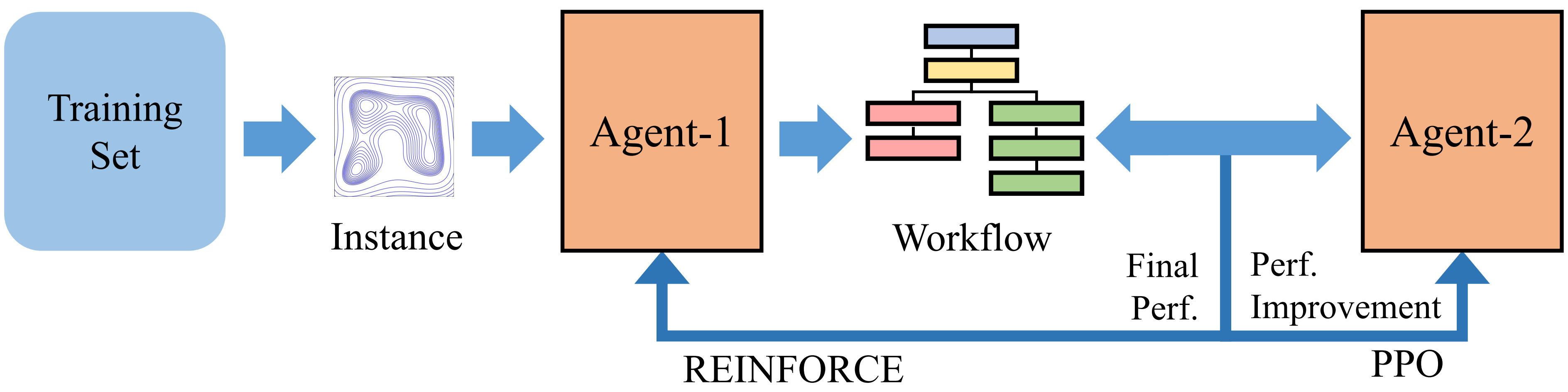


- Based on the algorithm workflow generated by Agent-1;
- Fitness Landscape Analysis (FLA) is adopted for optimization progress information. Combining the information, module id and positional encoding, Agent-2 uses GPT-2 based self-attention to generate proper hyper-parameters for each controllable modules (action);
- The hyper-parameters are then assigned to the algorithm modules and boost the performance.

South China University of Technology

Massachusetts Institute of Technology

Dual-agent Algorithm Design System: Training



- Model the algorithm construction process and algorithm configuration during optimization as MDPs;
- Workflows are generated by Agent-1 for each instance sampled from the set, then Agent-2 controls the workflows during optimization;
- Rewards are calculated based on the optimization process and the found optimal solution, we use REINFORCE to train Agent-1 and PPO for Agent-2.

$$\mathcal{J}(\phi, \theta) = \mathbb{E}_{p\sim\mathcal{D}_{train}}[\sum_{t=1}^{T} r_t] = \frac{1}{|\mathcal{D}_{train}|}\sum_{i=1}^{|\mathcal{D}_{train}|}\sum_{t=1}^{T} r_t$$

Experiment

➢ Experiment Setup:

Training dataset: A large-scale training set with 9600 instances.

In-Dsitribution Test Set: A large-scale testing set with 3200 instances.

Out-Of-Distribution Test Sets: Three realistic optimization benchmarks: Protein-Docking, HPO-B and UAV Path Planning.

Training Settings: Agent-1: REINFORCE, Agent-2: PPO; both 1e-4 learning rate in 100 epochs

Baselines:

- **before 00:** GA, PSO, DE
- **00-10:** CMAES, FIPSO, SaDE, CLPSO, JADE
- **10-20:** CoDE, IPSO, SHADE, LM-CMA-ES, GLPSO
- **after 20:** MadDE, jDE21, MMES, NL-SHADE-LBC
- **MetaBBO:** GLHF, DEDQN, GLEET

Experiment

- **In-Dsitribution Test:**

Table 1: The in-distribution generalization performance in terms of absolute optimization performance results on $\mathcal{D}_{test}$. The best is labeled in green and the second best is labeled in red.

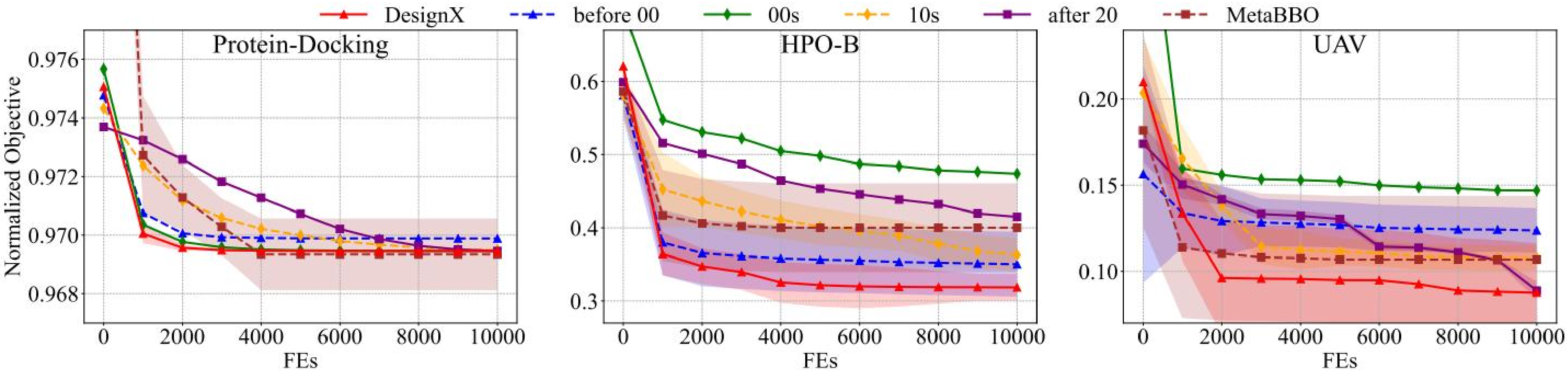
	before 00	00s	10s	after 20	MetaBBO	DesignX
F1 MAH, 50D, 30000 FEs	6.60E+00 ±3.74E+00 +	1.64E+00 ±1.64E+00 +	1.27E+00 ±4.41E-01 +	5.32E+00 ±3.70E+00 +	2.80E+00 ±0.00E+00 +	**2.89E-01 ±3.93E-01**
F79 UAH, 5D, 50000 FEs	2.98E+00 ±9.95E-01 +	3.70E+00 ±1.71E+00 +	5.38E+00 ±4.05E-01 +	1.81E+00 ±1.83E-01 +	9.95E-01 ±0.00E+00 +	**5.68E-02 ±1.17E+00**
F125 UAH, 10D, 40000 FEs	1.39E-03 ±1.38E-03 +	3.50E-06 ±3.50E-06 +	1.48E-04 ±1.33E-04 +	1.69E-05 ±7.99E-06 +	1.08E-04 ±0.00E+00 +	**4.81E-07 ±2.66E-07**
F154 UAH, 50D, 10000 FEs	1.35E+03 ±2.26E+02 +	1.44E+03 ±3.45E+02 +	1.38E+03 ±2.40E+02 +	1.46E+03 ±6.17E+02 +	**5.47E+02 ±0.00E+00** −	6.99E+02 ±7.45E+01
F211 MAH, 5D, 40000 FEs	6.55E-01 ±2.92E-01 +	8.04E-01 ±6.99E-01 +	2.64E-01 ±9.96E-02 +	1.28E-01 ±2.43E-02 +	1.59E-01 ±0.00E+00 +	**7.28E-02 ±6.56E-02**
F240 MWL, 20D, 20000 FEs	6.39E+00 ±4.25E+00 +	8.72E+00 ±1.71E+00 +	8.24E+00 ±2.19E+00 +	3.97E+00 ±3.31E+00 +	2.05E+00 ±0.00E+00 +	**1.27E-01 ±2.99E+00**
F326 UAL, 10D, 40000 FEs	1.10E+00 ±1.22E-01 +	1.15E+00 ±4.53E-01 +	2.47E+00 ±4.64E-01 +	7.66E-01 ±5.10E-02 +	1.22E+00 ±0.00E+00 +	**5.84E-01 ±1.66E+00**
F411 UAL, 10D, 50000 FEs	2.50E-01 ±9.51E-02 +	4.07E-01 ±9.27E-02 +	2.68E-01 ±7.41E-02 +	1.28E-01 ±1.11E-02 +	1.87E-01 ±0.00E+00 +	**7.91E-02 ±4.83E-02**
F545 UWL, 5D, 40000 FEs	2.98E+00 ±9.94E-01 +	1.49E+00 ±4.97E-01 +	3.36E+00 ±6.23E-01 +	8.41E-01 ±1.54E-01 +	1.99E+00 ±0.00E+00 +	**2.61E-08 ±6.48E-01**
F1045 MWH, 10D, 40000 FEs	7.53E+02 ±1.68E+00 +	4.20E+02 ±1.47E+02 +	2.29E+02 ±6.06E+01 +	1.71E+02 ±1.41E+01 +	9.21E+02 ±0.00E+00 +	**1.67E+02 ±1.06E+02**
F1139 MAH, 10D, 50000 FEs	1.16E+01 ±1.00E+01 +	3.95E+00 ±1.67E+00 +	8.90E+00 ±1.54E+00 +	2.67E+00 ±1.15E+00 +	1.19E+01 ±0.00E+00 +	**1.39E-04 ±1.47E+00**
F1200 MAL, 50D, 40000 FEs	7.47E+00 ±6.29E+00 +	1.14E+00 ±7.00E-03 +	1.15E+00 ±2.56E-02 +	1.33E+01 ±1.22E+01 +	1.27E+00 ±0.00E+00 +	**1.09E+00 ±2.12E-02**
F1556 MAH, 10D, 40000 FEs	3.99E+02 ±3.75E+02 +	2.03E+02 ±1.76E+02 +	2.73E+01 ±7.63E+00 +	1.48E+01 ±7.13E-01 +	**9.45E+00 ±0.00E+00** −	1.01E+01 ±1.13E+02
F1653 MAH, 20D, 10000 FEs	2.55E+01 ±1.53E+00 +	2.53E+01 ±7.11E-01 +	2.72E+01 ±7.28E-01 +	1.85E+01 ±2.56E+00 +	1.69E+01 ±0.00E+00 +	**1.54E+01 ±3.47E+00**
F1687 MAL, 50D, 40000 FEs	8.98E+00 ±6.60E+00 +	2.07E+01 ±1.07E+01 +	4.49E-01 ±2.38E-01 +	2.94E+01 ±2.81E+01 +	1.94E+00 ±0.00E+00 +	**2.24E-02 ±9.09E+00**
F2068 MWH, 20D, 20000 FEs	3.79E+01 ±6.53E+00 +	2.32E+00 ±1.13E-01 +	1.46E+01 ±1.40E+01 +	1.65E+01 ±1.41E+01 +	3.72E+01 ±0.00E+00 +	**5.16E-01 ±1.06E+01**
F2390 MAL, 10D, 30000 FEs	3.93E+00 ±2.15E+00 +	2.78E+00 ±0.00E+00 +	6.34E+00 ±9.03E-01 +	1.54E+00 ±1.10E+00 +	2.04E+01 ±0.00E+00 +	**1.85E-03 ±2.45E+00**
F2473 MAL, 10D, 20000 FEs	1.10E+00 ±9.06E-02 +	3.98E-01 ±6.88E-02 +	8.72E-01 ±1.80E-02 +	6.69E-01 ±2.53E-01 +	**1.42E-01 ±0.00E+00** −	1.63E-01 ±1.59E-01
F2895 MWL, 10D, 50000 FEs	1.90E+01 ±3.88E+00 +	4.34E+00 ±1.66E+00 +	1.18E+01 ±5.35E+00 +	4.23E+00 ±5.26E-01 +	4.98E+00 ±0.00E+00 +	**1.99E+00 ±3.34E+00**
F2986 MAL, 50D, 10000 FEs	4.37E+02 ±1.71E+02 +	4.93E+02 ±3.74E+02 +	1.60E+02 ±6.45E+01 +	2.51E+03 ±2.42E+03 +	1.01E+02 ±0.00E+00 +	**8.90E+01 ±2.65E+01**
Normalized Averaged Objective	2.94E-01 ±1.01E+00 +	1.96E-01 ±1.62E+00 +	1.54E-01 ±2.61E-01 +	1.46E-01 ±2.35E-01 +	1.32E-01 ±7.36E-01 +	**8.26E-02 ±1.75E-01**

- The human-crafted BBO optimizers achieve progressive advancement through the expert-level designs proposed over the past decades.

- The optimization performance of DesignX surpasses both MetaBBO and hand-crafted BBO baselines.

Experiment

➢ Out-Of-Distribution Test:

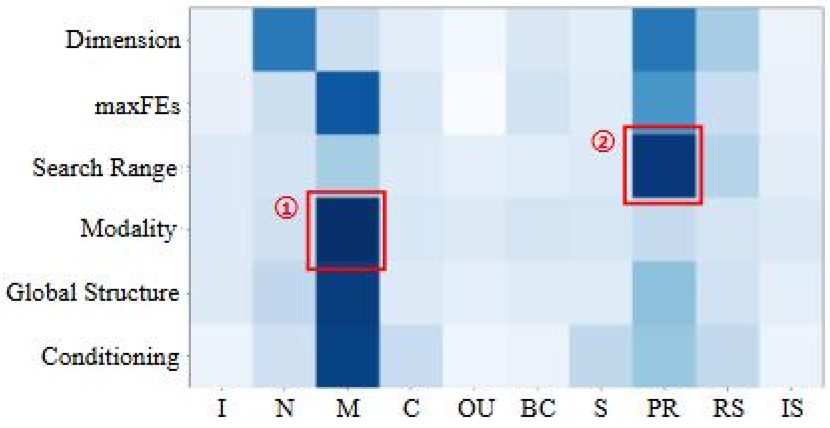


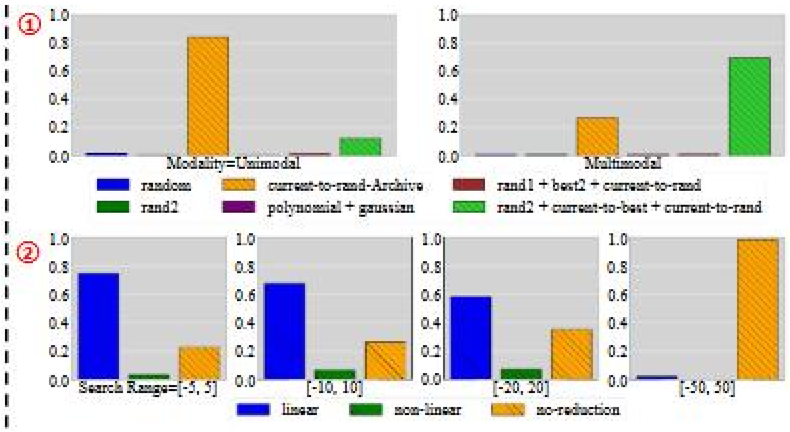
- DesignX generally shows superior optimization behavior to human-crafted optimizers from different decades, designing desirable optimizers robustly for diverse realistic problems it never saw during training

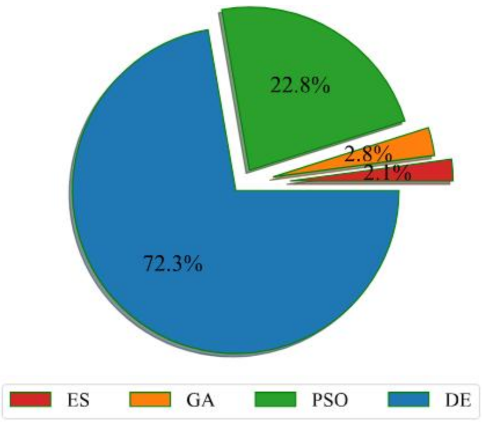
- DesignX consistently outperforms MetaBBO approaches, which demonstrates the novelty of our proposed bi-agent algorithm design system.

Experiment

➢ **In-Depth Analysis:**







➢ DE Mutation and Population_Reduction shows more importance while Initialization strategies has very limited impact on the final performance;

➢ For unimodal problem, it smartly choose greedy-fashion mutation operators to reinforce the exploitation. Population Reduction is an effective mechanism to upgrade an optimizer's local search ability.

➢ The DE-related algorithm sub-modules is primarily considered by DesignX to achieve aforementioned robust optimization performance.

Thanks for Listening!