ML4CO-Bench-101: Benchmark Machine Learning for Classic Combinatorial Problems on Graphs

Jiale Ma[1,2], Wenzheng Pan[1], Yang Li[1,2], Junchi Yan[1,2]

[1]Shanghai Jiao Tong University [2]Shanghai Innovation Institute

RETHINKLAB SJTU

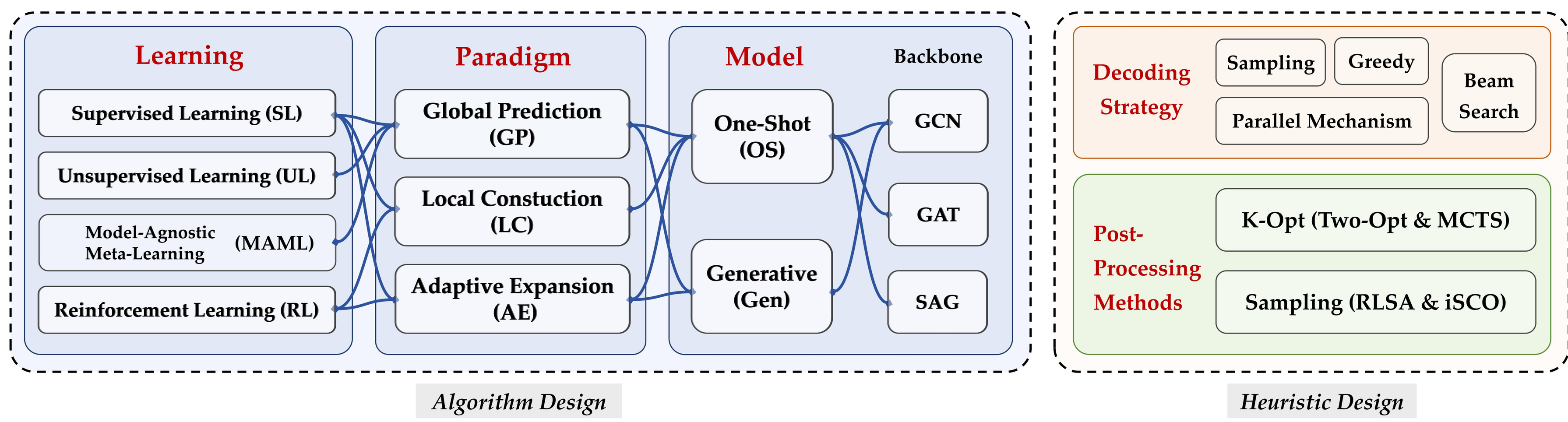
SHANGHAI JIAO TONG UNIVERSITY

SHANGHAI INNOVATION INTITUTE

NEURAL INFORMATION PROCESSING SYSTEMS

Benchmark

We provide a detailed analysis of existing methods of ML4CO from: a) ***algorithm design***, b) ***decoding strategy***, and c) ***post-processing optimization***



Algorithm Design *Heuristic Design*

Algorithm Designs

We compile a collection of representative papers from the past decade that apply machine learning techniques to 7 COPs

Table 1: Categorization of methods: Global Prediction (GP), Local Construction (LC), Adaptive Expansion (AE), Unsupervised/Supervised Learning (U/SL). Model-Agnostic Meta-Learning (MAML).

ID	Method	Algorithm Design: Paradigm	Model	Learning	Involved Problems: TSP	ATSP	CVRP	MIS	MCl	MVC	MCut
1	Intel [38]	GP	One-Shot	SL	–	–	–	✓	–	–	–
2	GCN [39]	GP	One-Shot	SL	✓	–	–	–	–	–	–
3	Att-GCN [29]	GP	One-Shot	SL	✓	–	–	–	–	–	–
4	GNNGLS [44]	GP	One-Shot	SL	✓	–	–	–	–	–	–
*	GP4CO (Ours)	GP	One-Shot	SL	✓	✓	–	✓	✓	✓	✓
5	DIMES [18]	GP	One-Shot	MAML (RL)	✓	–	–	✓	–	–	–
6	Meta-EGN [20]	GP	One-Shot	MAML (UL)	–	–	–	–	✓	✓	–
*	GP4CO (Ours)	GP	One-Shot	MAML	✓	–	–	✓	✓	✓	✓
7	UTSP [28]	GP	One-Shot	UL	✓	–	–	–	–	–	–
8	VAG-CO [26]	GP	One-Shot	UL	–	–	–	✓	✓	✓	✓
*	GP4CO (Ours)	GP	One-Shot	UL	✓	–	–	✓	✓	✓	✓
9	DIFUSCO [16]	GP	Generative	SL	✓	–	–	✓	–	–	–
10	T2T [36] & Fast-T2T [40]	GP	Generative	SL	✓	–	–	✓	–	–	–
11	UniCO-MatDIFFNet [33]	GP	Generative	SL	✓	✓	–	–	–	✓	–
*	GP4CO (Ours)	GP	Generative	SL	✓	✓	✓	✓	✓	✓	✓
12	DiffUCO [19]	GP	Generative	UL	–	–	–	✓	✓	–	✓
13	SDDS [21]	GP	Generative	UL	–	–	–	✓	✓	–	✓
*	GP4CO (Ours)	GP	Generative	UL	–	–	–	✓	✓	–	✓
14	RL4CO [41, 45, 30, 42]	LC	One-Shot	RL	✓	–	✓	–	–	–	–
15	MatNet [31]	LC	One-Shot	RL	–	✓	–	–	–	–	–
16	UniCO-MatPOENet [33]	LC	One-Shot	RL	✓	✓	–	–	–	–	–
*	LC4CO (Ours)	LC	One-Shot	RL	✓	✓	✓	–	–	–	–
17	BQ-NCO [46]	LC	One-Shot	SL	✓	✓	✓	–	–	–	–
18	GOAL [32]	LC	One-Shot	SL	–	✓	✓	✓	–	✓	–
*	LC4CO (Ours)	LC	One-Shot	SL	✓	✓	✓	✓	✓	✓	–
19	LwD [43]	AE	One-Shot	RL	–	–	–	✓	–	–	–
*	AE4CO (Ours)	AE	One-Shot	RL	–	–	–	✓	–	–	–
20	COExpander [37]	AE	Generative	SL	✓	✓	✓	✓	✓	✓	✓
*	AE4CO (Ours)	AE	Generative	SL	✓	✓	✓	✓	✓	✓	✓

Decoding & Post-Processing

We compile and implement the most representative decoding and post-processing strategies for 7 COPs

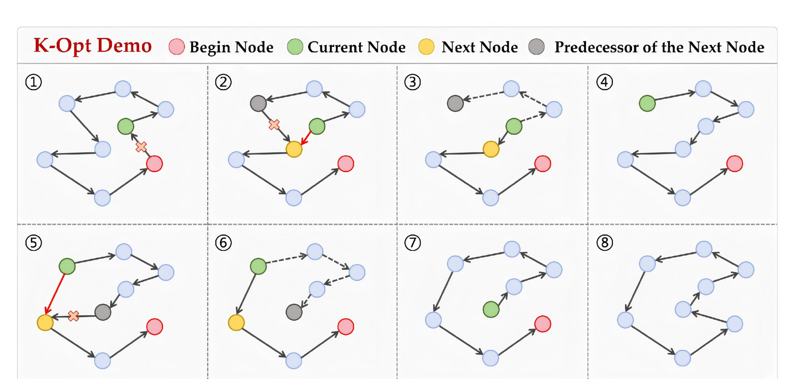


Figure 2: Demonstration of the K-Opt (K=3) post-processing method for TSP.

Algorithm 1 Original RLSA

Input: Graph G, Energy Function $H(\cdot)$, hyperparameter (τ_0, d, k, t).
Parallel k:
$\mathbf{x} \leftarrow \mathrm{Bern}(p=0.5)$
for $s = 1$ to t **do**
$\tau \leftarrow \tau_0(1 - \frac{s}{t})$
$\Delta \leftarrow (2\mathbf{x} - 1) \odot \nabla H(\mathbf{x})$
for $i = 1$ to N **do**
$p \leftarrow \mathrm{Sigmoid}\left((\Delta_i - \Delta_{(d)})/(2\tau)\right)$
$c \leftarrow \mathrm{Bern}(p)$
$\mathbf{x}_i \leftarrow \mathbf{x}_i \cdot (1 - c) + (1 - \mathbf{x}_i) \cdot c$
end for
if $H(\mathbf{x}) < H(\mathbf{x}^*)$ **then**
$\mathbf{x}^* \leftarrow \mathbf{x}$
end if
end for
Output: Solution $\mathbf{x}$

Algorithm 2 RLSA as Local Search

Input: Graph G, Energy Function $H(\cdot)$, hyperparameter $(\tau_0, d, k, t, \alpha_1, \alpha_2)$, Initial Solution $\mathbf{x}_0$.
Parallel k:
$\mathbf{x} \leftarrow \mathrm{Bern}((1 - \alpha_1) \cdot \mathbf{x}_0 + \alpha_2 \cdot \mathrm{Bern}(p=0.5))$
for $s = 1$ to t **do**
$\tau \leftarrow \tau_0(1 - \frac{s}{t})$
$\Delta \leftarrow (2\mathbf{x} - 1) \odot \nabla H(\mathbf{x})$
for $i = 1$ to N **do**
$p \leftarrow \mathrm{Sigmoid}\left((\Delta_i - \Delta_{(d)})/(2\tau)\right)$
$c \leftarrow \mathrm{Bern}(p)$
$\mathbf{x}_i \leftarrow \mathbf{x}_i \cdot (1 - c) + (1 - \mathbf{x}_i) \cdot c$
end for
if $H(\mathbf{x}) < H(\mathbf{x}^*)$ **then**
$\mathbf{x}^* \leftarrow \mathbf{x}$
end if
end for
Output: Solution $\mathbf{x}$

Dataset

We provide the open-source datasets: including 24 train datasets and 65 test datasets

Table 11: Overview of our benchmark datasets for the seven COPs. The "MODEL" column indicates the dataset on which the model is trained. I.e., datasets with the same "TYPE" and "MODEL" is suggested for i.i.d. training-testing, whereas datasets with different "TYPE" and "MODEL" (with "testing" part only) are designed for evaluating the o.o.d. generalization performance of models trained on the corresponding (smaller-scaled, differently distributed, etc.) data. The parentheses indicate the solver parameters used: the maximum solution time for Gurobi, KaMIS and HGS; the maximum trials for LKH. Blue: Cross-distribution Generalization Dataset.

ID	Problem	Type	Model	Training: Data Size	Solver	Storage	Testing: Data Size	Solver	Obj.
1	TSP	Uniform-50	Uniform-50	1,280,000	Concorde	2.48GB	1280	Concorde	5.688
2	TSP	Uniform-100	Uniform-100	1,280,000	Concorde	4.95GB	1280	Concorde	7.756
3	TSP	Uniform-200	Uniform-200	128,000	Concorde	1.05GB	128	Concorde	10.719
4	TSP	Uniform-500	Uniform-500	64,000	Concorde	1.26GB	128	Concorde	16.546
5	TSP	Uniform-1K	Uniform-1K	64,000	LKH(1000)	2.53GB	128	Concorde	23.118
6	TSP	Uniform-10K	Uniform-10K	6,400	LKH(500)	2.59GB	16	LKH(500)	71.755
7	TSP	TSPLIB	Mixed	–	–	–	49	Concorde	8.062
8	TSP	Cluster-50	Uniform-50	–	–	–	1280	Concorde	3.730
9	TSP	Cluster-100	Uniform-100	–	–	–	1280	Concorde	5.526
10	TSP	Cluster-200	Uniform-200	–	–	–	128	Concorde	6.912
11	TSP	Cluster-500	Uniform-500	–	–	–	128	Concorde	10.723
12	TSP	Gaussian-50	Uniform-50	–	–	–	1280	Concorde	23.840
13	TSP	Gaussian-100	Uniform-100	–	–	–	1280	Concorde	34.031
14	TSP	Gaussian-200	Uniform-200	–	–	–	128	Concorde	48.127
15	TSP	Gaussian-500	Uniform-500	–	–	–	128	Concorde	77.521
16	ATSP	Uniform-50	Uniform-50	640,000	LKH(500)	14.72GB	2500	LKH(1000)	1.5545
17	ATSP	Uniform-100	Uniform-100	128,000	LKH(500)	11.78GB	2500	LKH(1000)	1.5660
18	ATSP	Uniform-200	Uniform-200	32,000	LKH(1000)	11.76GB	100	LKH(1000)	1.5647
19	ATSP	Uniform-500	Uniform-500	6,400	LKH(1000)	14.70GB	100	LKH(1000)	1.5734
20	ATSP	HCP-50	Uniform-50	–	–	–	2500	–	0.0000
21	ATSP	HCP-100	Uniform-100	–	–	–	2500	–	0.0000
22	ATSP	HCP-200	Uniform-200	–	–	–	100	–	0.0000
23	ATSP	HCP-500	Uniform-500	–	–	–	100	–	0.0000
24	ATSP	SAT-54	Uniform-50	–	–	–	2500	–	0.0000
25	ATSP	SAT-102	Uniform-100	–	–	–	2500	–	0.0000
26	ATSP	SAT-200	Uniform-200	–	–	–	100	–	0.0000
27	ATSP	SAY-507	Uniform-500	–	–	–	100	–	0.0000
28	CVRP	Uniform-50	Uniform-50	1,280,000	HGS(1s)	2.83GB	10,000	HGS(1s)	10.366
29	CVRP	Uniform-100	Uniform-100	640,000	HGS(20s)	1.11GB	10,000	HGS(20s)	15.563
30	CVRP	Uniform-200	Uniform-200	128,000	HGS(60s)	1.11GB	100	HGS(60s)	19.630
31	CVRP	Uniform-500	Uniform-500	12,800	HGS(360s)	285MB	100	HGS(360s)	37.154
32	CVRP	CVRPLIB	Mixed	–	–	–	70	–	45.183
33	MIS	RB-SMALL	RB-SMALL	64,000	KaMIS(10s)	3.52GB	500	KaMIS(60s)	20.090
34	MIS	RB-LARGE	RB-LARGE	6,400	KaMIS(60s)	4.74GB	500	KaMIS(60s)	43.004
35	MIS	RB-GIANT	RB-LARGE	–	–	–	50	KaMIS(60s)	49.260
36	MIS	ER-700-800	ER-700-800	12,800	KaMIS(60s)	7.83GB	128	KaMIS(60s)	44.969
37	MIS	ER-1400-1600	ER-700-800	–	–	–	128	KaMIS(60s)	50.938
38	MIS	SATLIB	SATLIB	39,500	KaMIS(60s)	3.75GB	500	KaMIS(60s)	425.954
39	MIS	HK-SMALL	ER-700-800	–	–	–	500	KaMIS(60s)	79.372
40	MIS	HK-LARGE	ER-700-800	–	–	–	500	KaMIS(60s)	330.946
41	MIS	WS-SMALL	ER-700-800	–	–	–	500	KaMIS(60s)	76.904
42	MIS	WS-LARGE	ER-700-800	–	–	–	500	KaMIS(60s)	262.570
43	MCl	RB-SMALL	RB-SMALL	64,000	Gurobi(60s)	3.42GB	500	Gurobi(60s)	19.082
44	MCl	Twitter	RB-SMALL	–	–	–	195	Gurobi(60s)	14.210
45	MCl	COLLAB	RB-SMALL	–	–	–	1000	Gurobi(60s)	42.113
46	MCl	RB-LARGE	RB-LARGE	6,400	Gurobi(300s)	4.74GB	500	Gurobi(300s)	40.182
47	MCl	RB-GIANT	RB-LARGE	–	–	–	50	Gurobi(3600s)	81.520
48	MCl	HK-SMALL	RB-SMALL	–	–	–	500	Gurobi(60s)	6.792
49	MCl	HK-LARGE	RB-LARGE	–	–	–	500	Gurobi(300s)	6.774
50	MCl	WS-SMALL	RB-SMALL	–	–	–	500	Gurobi(60s)	7.164
51	MCl	WS-LARGE	RB-LARGE	–	–	–	500	Gurobi(300s)	5.978
52	MVC	RB-SMALL	RB-SMALL	128,000	Gurobi(60s)	7.01GB	500	Gurobi(60s)	205.764
53	MVC	Twitter	RB-SMALL	–	–	–	195	Gurobi(60s)	85.251
54	MVC	COLLAB	RB-SMALL	–	–	–	1000	Gurobi(60s)	65.086
55	MVC	RB-LARGE	RB-LARGE	6,400	Gurobi(300s)	4.74GB	500	Gurobi(300s)	968.228
56	MVC	RB-GIANT	RB-LARGE	–	–	–	50	Gurobi(300s)	2398.480
57	MVC	HK-SMALL	RB-SMALL	–	–	–	500	Gurobi(60s)	142.506
58	MVC	WS-SMALL	RB-SMALL	–	–	–	500	Gurobi(60s)	154.618
59	MCut	BA-SMALL	BA-SMALL	128,000	Gurobi(60s)	1.78GB	500	Gurobi(60s)	727.844
60	MCut	BA-LARGE	BA-LARGE	128,000	Gurobi(300s)	8.08GB	500	Gurobi(300s)	2936.886
61	MCut	BA-GIANT	BA-LARGE	–	–	–	50	Gurobi(300s)	7217.900
62	MCut	HK-SMALL	BA-SMALL	–	–	–	500	Gurobi(60s)	1540.608
63	MCut	HK-LARGE	BA-LARGE	–	–	–	500	Gurobi(300s)	6401.320
64	MCut	WS-SMALL	BA-SMALL	–	–	–	500	Gurobi(60s)	872.116
65	MCut	WS-LARGE	BA-LARGE	–	–	–	500	Gurobi(300s)	3454.176

Experiments

We conduct the following experiments:

1. Without any post-processing, we use the simplest ***greedy decoding strategy*** to directly compare the performance and efficiency of different algorithmic designs
2. We analyze the performance and efficiency of various decoding and post-processing strategies, and—taking both into account—present a comprehensive comparison between SOTA ML4CO and traditional solvers on 65 datasets across 7 COPs

Table 8: ML4CO-101 vs baseline solvers: a summarized comparaison. Rd denotes random.

Problem	Dataset	Learning-free Baseline: Method	Obj.	Time↓	Learning-based Methods: Method Type	Settings	Obj.	Drop↓	Time↓
TSP	TSP-50	Concorde (67)	5.688	0.059s	GP-OS-SL	Rd-16 + MCTS (0.05s)	5.688	0.001%	0.031s
TSP	TSP-100	Concorde (67)	7.756	0.238s	GP-OS-SL	Rd-16 + MCTS (0.05s)	7.756	0.005%	0.119s
TSP	TSP-500	Concorde (67)	16.546	18.672s	AE-Gen-SL	4×Greedy + Two-Opt	16.588	0.257%	0.695s
TSP	TSP-1K	Concorde (67)	23.118	84.413s	AE-Gen-SL	4×Greedy + Two-Opt	23.271	0.662%	2.609s
TSP	TSP-10K	LKH(500) (6)	71.755	332.758s	AE-Gen-SL	Greedy + Two-Opt	72.832	1.450%	33.485s
TSP	TSPLIB	Concorde (67)	8.062	–	–	–	8.095	0.356%	–
ATSP	ATSP-50	LKH (1K) (6)	1.554	0.097s	AE-Gen-SL	4×Greedy + Two-Opt	1.557	0.171%	0.103s
ATSP	ATSP-100	LKH (1K) (6)	1.566	0.238s	AE-Gen-SL	4×Greedy + Two-Opt	1.581	0.946%	0.516s
ATSP	ATSP-200	LKH (1K) (6)	1.565	0.724s	AE-Gen-SL	4×Greedy + Two-Opt	1.588	1.501%	1.097s
ATSP	ATSP-500	LKH (1K) (6)	1.573	4.376s	AE-Gen-SL	4×Greedy + Two-Opt	1.598	1.568%	6.597s
CVRP	CVRP-50	HGS (69)	10.366	1.005s	LC-OS-RL	N×Sampling + Classic-LS	10.489	1.179%	0.154s
CVRP	CVRP-100	HGS (69)	15.563	20.027s	LC-OS-RL	N×Sampling + Classic-LS	15.822	1.663%	0.202s
CVRP	CVRP-200	HGS (69)	19.630	60.024s	LC-OS-RL	N×Sampling + Classic-LS	20.091	2.359%	0.439s
CVRP	CVRP-500	HGS (69)	37.154	360.376s	LC-OS-SL	Beam-16 + Classic-LS	37.901	2.031%	64.589s
CVRP	CVRPLIB	HGS (69)	45.183	–	LC-OS-RL	N×Sampling + Classic-LS	48.263	5.469%	–
MIS	RB-SMALL	Gurobi (5)	20.090	0.538s	AE-Gen-SL	Greedy + RLSA	20.070	0.093%	0.471s
MIS	RB-LARGE	KaMIS (70)	43.004	56.974s	AE-Gen-SL	Greedy + RLSA	42.400	1.366%	1.816s
MIS	ER-700-800	KaMIS (70)	44.969	60.753s	AE-Gen-SL	Greedy + RLSA	**44.984**	**-0.041%**	**1.390s**
MIS	SATLIB	KaMIS (70)	425.954	24.368s	AE-Gen-SL	Greedy + RLSA	425.316	0.151%	1.775s
MIS	ER-1400-1600	KaMIS (70)	50.938	60.824s	AE-Gen-SL	Greedy + RLSA	50.719	0.418%	4.102s
MIS	RB-GIANT	KaMIS (70)	49.260	60.960s	AE-Gen-SL	Greedy + RLSA	47.880	2.741%	9.490s
MCl	RB-SMALL	Gurobi (5)	19.082	0.900s	GP-OS-SL	Beam-16 + RLSA	**19.082**	**0.000%**	**0.041s**
MCl	RB-LARGE	Gurobi (5)	40.182	276.657s	GP-OS-SL	Beam-16 + RLSA	**40.256**	**-0.275%**	**0.171s**
MCl	TWITTER	Gurobi (5)	14.210	0.276s	GP-OS-SL	Beam-16 + RLSA	**14.210**	**0.000%**	**0.044s**
MCl	COLLAB	Gurobi (5)	42.113	0.063s	GP-OS-SL	Beam-16 + RLSA	**42.113**	**0.000%**	**0.024s**
MCl	RB-GIANT	Gurobi (5)	81.520	3606.201s	GP-OS-SL	Beam-16 + RLSA	**85.380**	**-7.912%**	**4.342s**
MVC	RB-SMALL	Gurobi (5)	205.764	3.340s	AE-Gen-SL	Greedy + RLSA	205.772	0.004%	0.612s
MVC	RB-LARGE	Gurobi (5)	968.228	290.227s	AE-Gen-SL	Greedy + RLSA	968.398	0.018%	1.592s
MVC	TWITTER	Gurobi (5)	85.251	0.133s	AE-Gen-SL	Greedy + RLSA	**85.251**	**0.000%**	**0.115s**
MVC	COLLAB	Gurobi (5)	65.086	0.058s	AE-Gen-SL	Greedy + RLSA	65.086	0.000%	0.158s
MVC	RB-GIANT	Gurobi (5)	2396.780	60.612s	AE-Gen-SL	Greedy + RLSA	2397.360	0.026%	8.590s
MCut	BA-SMALL	Gurobi (5)	727.844	60.612s	AE-Gen-SL	Greedy + RLSA	**729.706**	**-0.240%**	**0.727s**
MCut	BA-LARGE	Gurobi (5)	2936.886	300.214s	GP-Gen-SL	Greedy + RLSA	**2994.118**	**-1.932%**	**0.999s**
MCut	BA-GIANT	Gurobi (5)	7217.900	3601.342s	GP-Gen-SL	Greedy + RLSA	**7389.300**	**-2.383%**	**2.228s**

Table 9: Generalization Study on Cross-distribution Datasets.

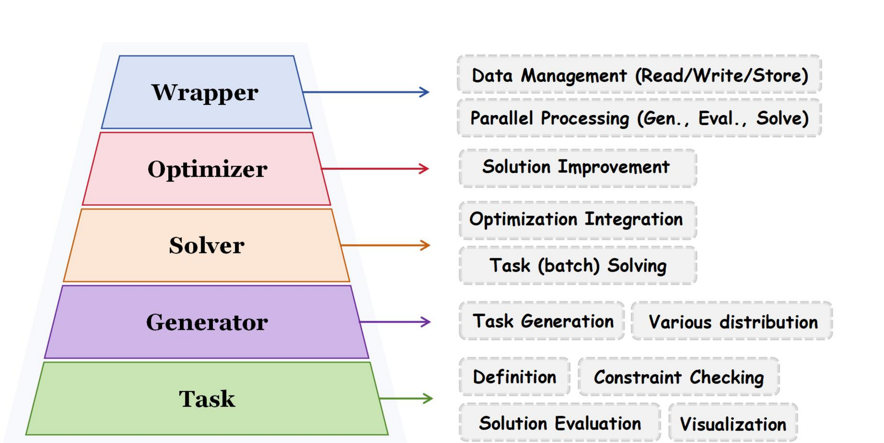
Problem	Dataset	Learning-free Baseline: Method	Obj.	Time↓	Learning-based Methods: Method Type	Settings	Obj.	Drop↓	Time↓
TSP	Cluster-50	Concorde (67)	3.730	0.140s	GP-OS-SL	Rd-16 + MCTS (0.05s)	3.730	0.001%	0.098s
TSP	Cluster-100	Concorde (67)	5.526	0.290s	GP-OS-SL	Rd-16 + MCTS (0.05s)	5.527	0.011%	0.604s
TSP	Cluster-500	Concorde (67)	10.723	5.073s	AE-Gen-SL	4×Greedy + Two-Opt	10.937	1.998%	1.250s
TSP	Gaussian-50	Concorde (67)	23.840	0.166s	GP-OS-SL	Rd-16 + MCTS (0.05s)	23.841	0.005%	0.256s
TSP	Gaussian-100	Concorde (67)	34.031	0.438s	GP-OS-SL	Rd-16 + MCTS (0.05s)	34.044	0.037%	0.604s
TSP	Gaussian-500	Concorde (67)	77.521	19.952s	AE-Gen-SL	4×Greedy + Two-Opt	78.122	0.774%	1.169s
ATSP	HCP-50	LKH (1K) (6)	0.000	0.107s	AE-Gen-SL	Greedy + Two-Opt	1.468	–	0.075s
ATSP	HCP-100	LKH (1K) (6)	0.000	0.211s	AE-Gen-SL	Greedy + Two-Opt	1.380	–	0.109s
ATSP	HCP-200	LKH (1K) (6)	0.000	0.355s	AE-Gen-SL	Greedy + Two-Opt	1.050	–	2.508s
ATSP	HCP-500	LKH (1K) (6)	0.000	1.410s	AE-Gen-SL	Greedy + Two-Opt	0.880	–	3.219s
ATSP	SAT-50	LKH (1K) (6)	0.151	0.079s	AE-Gen-SL	Greedy + Two-Opt	5.018	–	0.076s
ATSP	SAT-100	LKH (1K) (6)	0.079	0.128s	AE-Gen-SL	Greedy + Two-Opt	12.717	–	0.107s
ATSP	SAT-200	LKH (1K) (6)	0.130	0.192s	AE-Gen-SL	Greedy + Two-Opt	21.470	–	1.768s
ATSP	SAT-500	LKH (1K) (6)	0.430	0.781s	AE-Gen-SL	Greedy + Two-Opt	2.340	–	3.222s
MIS	HK-SMALL	KaMIS (70)	79.372	54.174s	AE-Gen-SL	Greedy + RLSA	79.358	0.017%	0.353s
MIS	HK-LARGE	KaMIS (70)	330.946	67.272s	AE-Gen-SL	Greedy + RLSA	330.422	0.154%	1.600s
MIS	WS-SMALL	KaMIS (70)	76.904	51.490s	AE-Gen-SL	Greedy + RLSA	76.894	0.013%	0.357s
MIS	WS-LARGE	KaMIS (70)	262.570	37.792s	AE-Gen-SL	Greedy + RLSA	260.114	0.926%	1.502s
MCl	HK-SMALL	Gurobi (5)	6.792	1.838s	GP-OS-SL	Beam-16 + RLSA	**6.792**	**0.000%**	**0.039s**
MCl	HK-LARGE	Gurobi (5)	6.774	46.502s	GP-OS-SL	Beam-16 + RLSA	**6.774**	**0.000%**	**0.169s**
MCl	WS-SMALL	Gurobi (5)	7.164	1.589s	GP-OS-SL	Beam-16 + RLSA	**7.164**	**0.000%**	**0.040s**
MCl	WS-LARGE	Gurobi (5)	5.978	27.051s	GP-OS-SL	Beam-16 + RLSA	5.976	0.033%	0.172s
MVC	HK-SMALL	Gurobi (5)	142.506	0.382s	AE-Gen-SL	Greedy + RLSA	**142.506**	**0.000%**	**0.353s**
MVC	WS-SMALL	Gurobi (5)	154.618	1.596s	AE-Gen-SL	Greedy + RLSA	**154.610**	**-0.005%**	**0.317s**
MCut	HK-SMALL	Gurobi (5)	1540.608	60.089s	AE-Gen-SL	Greedy + RLSA	**1542.810**	**-0.141%**	**0.382s**
MCut	HK-LARGE	Gurobi (5)	6401.320	300.357s	GP-Gen-SL	Greedy + RLSA	**6454.100**	**-0.766%**	**0.991s**
MCut	WS-SMALL	Gurobi (5)	872.116	60.357s	AE-Gen-SL	Greedy + RLSA	**874.372**	**-0.250%**	**0.338s**
MCut	WS-LARGE	Gurobi (5)	3454.176	300.247s	GP-Gen-SL	Greedy + RLSA	**3489.304**	**-1.007%**	**0.668s**

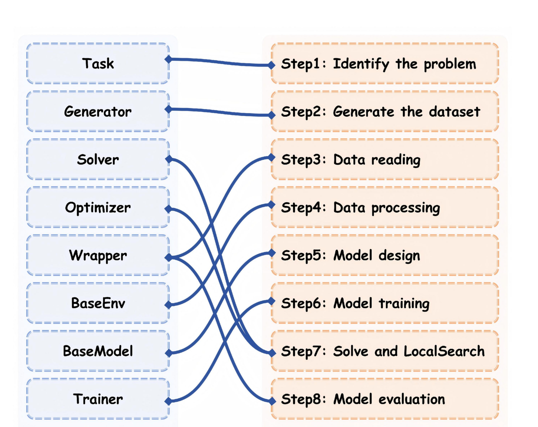
ML4CO-Kit

We have developed the open-source toolkit *ML4CO-Kit*, which aims to provide: decoding or post-processing algorithms, traditional solvers, evaluators, data generators, visualization tools, and read and write functions for various COPs

ML4CO-KiT

pip install ml4co-kit





Task	Definition	Check Constraint	Evaluation	Render	Special R/O
Routing Tasks					
Asymmetric TSP (ATSP)	✓	✓	✓	📅	`tsplib`
Capacitated Vehicle Routing Problem (CVRP)	✓	✓	✓	✓	`vrplib`
Orienteering Problem (OP)	✓	✓	✓	📅	
Prize Collection TSP (PCTSP)	✓	✓	✓	📅	
Stochastic PCTSP (SPCTSP)	✓	✓	✓	📅	
Traveling Salesman Problem (TSP)	✓	✓	✓	✓	`tsplib`
Graph Tasks					
Maximum Clique (MCl)	✓	✓	✓	✓	`gpickle`, `adj_matrix`, `networkx`, `csr`
Maximum Cut (MCut)	✓	✓	✓	✓	`gpickle`, `adj_matrix`, `networkx`, `csr`
Maximum Independent Set (MIS)	✓	✓	✓	✓	`gpickle`, `adj_matrix`, `networkx`, `csr`
Minimum Vertex Cover (MVC)	✓	✓	✓	✓	`gpickle`, `adj_matrix`, `networkx`, `csr`
Portfolio Tasks					
Maximum Return Portfolio Optimization (MaxRetPO)	✓	✓	✓	📅	
Minimum Variance Portfolio Optimization (MinVarPO)	✓	✓	✓	📅	
Multi-Objective Portfolio Optimization (MOPO)	✓	✓	✓	📅	

ML4CO-Bench-101 is based on ML4CO-Kit *Under active development, Contributions welcome!*

Can't make it to the conference in person. Please feel free to reach out to me via github or email: heatingma@sjtu.edu.cn