Learning Collaborative Policies to Solve NP-hard Routing Problems

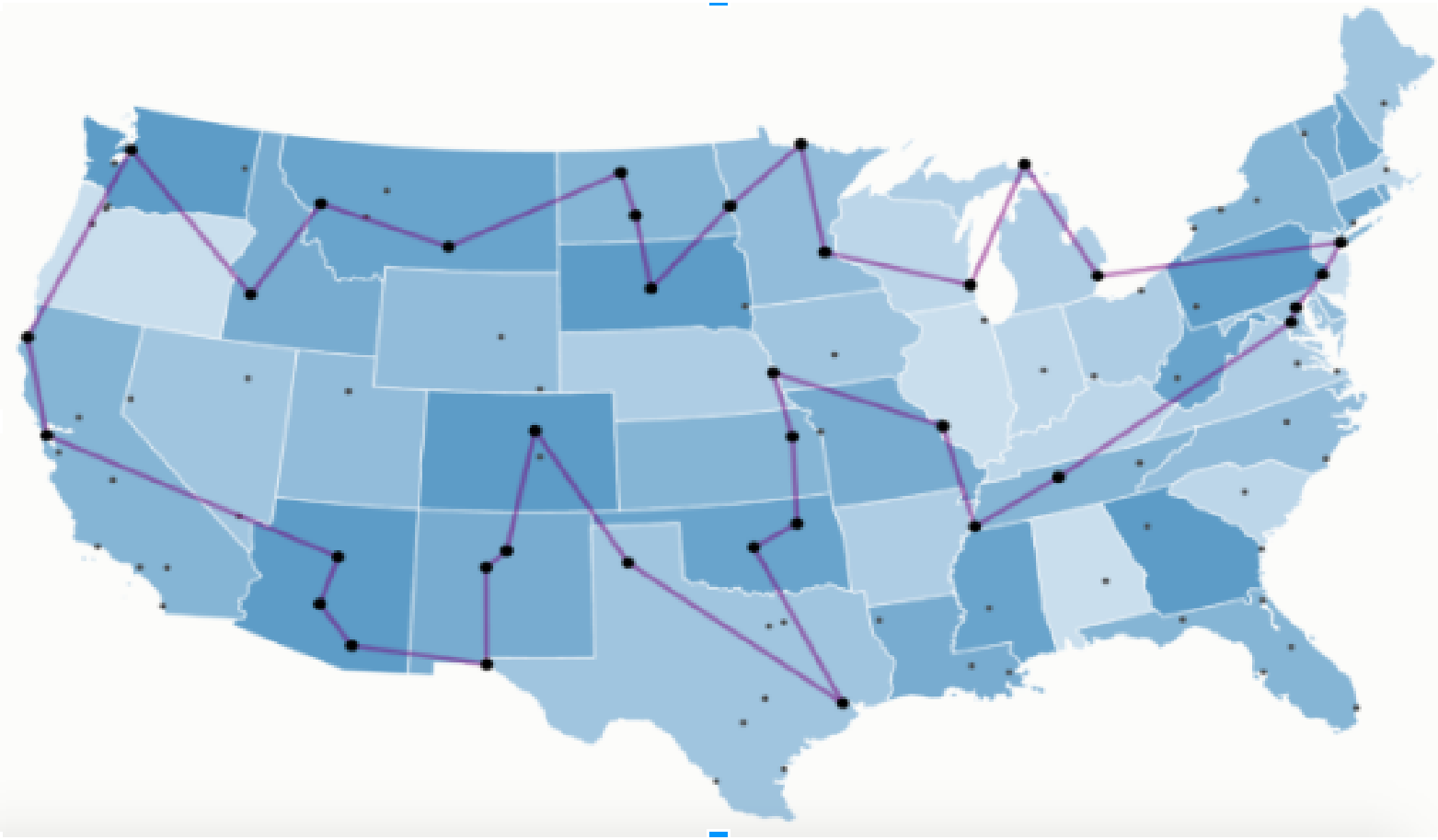
Minsu Kim, Jinkyoo Park and Joungho Kim

Korea Advanced Institute of Science and Technology (KAIST)

November 5, 2021

NP-hard Routing: Travelling Salesman Problem (TSP)

- Task: Visit every city and back to start city.
- Objective: Minimization of Tour Length.

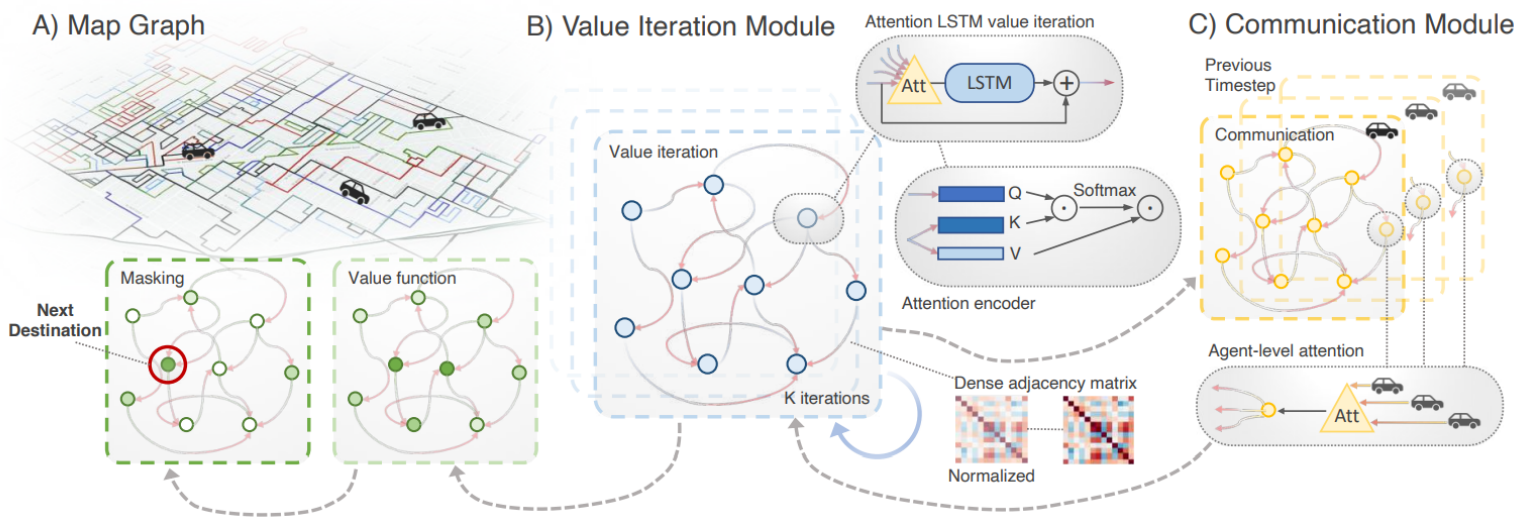


Motivation of Deep Reinforcement Learning (DRL) for Combinatorial Optimization (CO)

- Can DRL method reach a state-of-the-art (SOTA) in TSP task? NO!
  - The Concorde is much faster and accurate than SOTA DRL solver.
- Can DRL method applied to REAL WORLD TASKS ? Yes!

2020 Hardware Routing
2020 Molecular Design
2020 Uber-dispatching
2020 System-on-chip (SoC) scheduling
2021 Hardware Chip Placement

Figure by Sykora et al., "Multi-agent Routing Value Iteration Network", ICML 2020



Background: Constructive DRL heuristic

- Action: Constructing solution from partial solution.
- State: Partial solution.
  - Initial state: Empty solution.
  - Terminal state: Complete solution.
- Reward: Cost for the terminal state or transition between state,

2015 Vinyards et al., Bello et al.: Seq-to-seq scheme (**PointerNet**)
2017 Dai et al.: Graph Neural Network (**S2V-DQN**)
2019 Kool et al.: Transformer-style PointerNet (**AM**)
2020 Kwan et al., Xin et al.: AM-variants (**POMO; MDAM**)

Benefit 1

Construction scheme is expanding to other similar problems without problem-specific knowledge.

Benefit 2

Construction scheme is usually fast; the number of neural net inferences is proportional to problem size.

Background: Improvement DRL heuristic

- Action: Improve complete solution (local-search)
- State: Complete solution.
  - Initial state: Initial feasible solution.
  - Terminal state: Improved solution.
- Reward: Improved cost between consecutive states.

2020 Hottung et al.: Neural Large Neigborhood Search (**NLNS**)

2020 Costa et al.: DRL-based 2opt (**DRL-2opt**)

Benefit 1

Improvement heuristic can reduce optimal gap (than constructive heuristic), when enough iterations are provided.

Negative 1

Improvement heuristic is slower than constructive heuristics.

Background: Hybrid ML method with classic OR tools

- Supervised Learning by Labeled data of classical solvers + Search Method
  - 2019 Joshi et al.: GNN with beam search
  - 2021 Fu et al.: GNN with MCTS
  - 2021 Kool et al, GNN with Dynamic Programming
- Controls classical solvers with DNN
  - 2020 Lu et al.: **AM**-controller + Classic Solver
  - 2020 Song et al.: GNN-controller + Classic Solver
  - 2021 Sonnerat et al.: GNN-controller + Classic Solver

Benefit 1

Promising performances on target tasks.

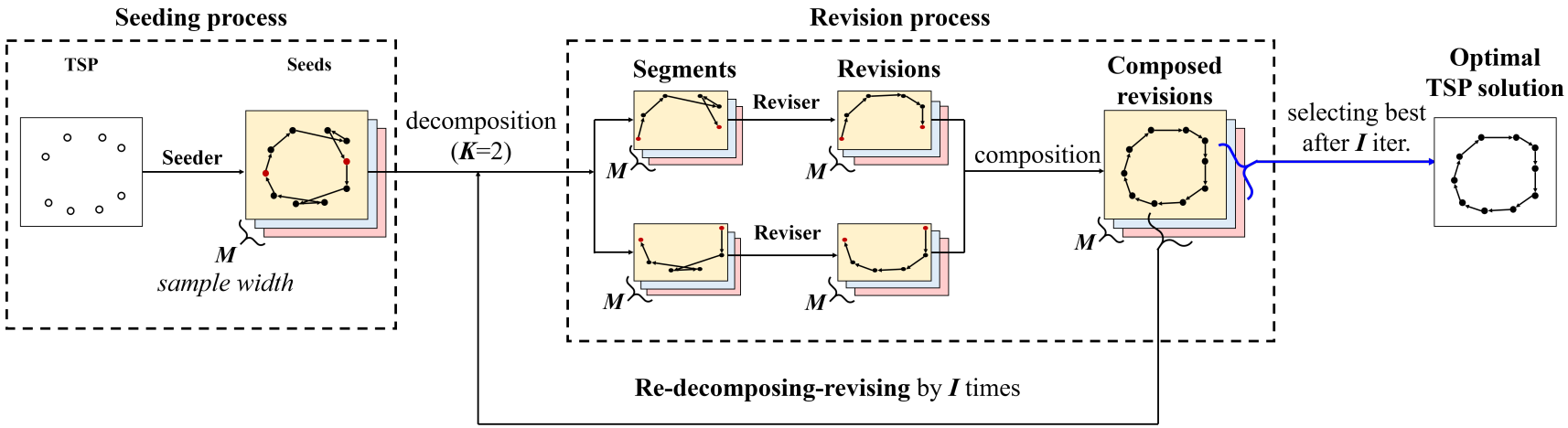
Negative 1

Not fully DRL method; No advantages on task scalability that DRL researches usually pursue.

Learning Collaborative Policies (LCP): Research Objective

- We focus on building **a reusable scheme** based on policy collaboration for **accelerating RL-based constructive heuristics without modifying neural network**.
  - We target **AM**-style constructive heuristics: **PointerNet** and **AM**.
  - We solve TSP-variants: TSP, Prize Collecting TSP (PCTSP) and Capacitated Vehicle Routing Problem (CVRP).
- Keeping advantages of constructive heuristics and increase performances with a simple hierarchical solving strategy.

Learning Collaborative Policies (LCP): Method Outline



- This research proposed a two-policies collaboration system with **Seeder** and **Reviser**.
- Seeder generates multiple candidate solutions, trained to **explorer** various near-optimal solutions.
- Reviser **exploits** the policy to solves multiple candidate solutions in a parallel manner, iteratively with restricted solution spaces.
- The seeder and reviser is parameterized by existing constructive model (**AM** or **PointerNet**)

Formulation of Routing Problem (EX. TSP)

- **Routing tasks**: Routing problems are a type of NP-hard combinatorial optimization where a sequential order of input arguments strongly affects the quality of the solutions.
- **Problem**: The TSP graph can be represented as a sequence of N nodes in 2D Euclidean space, $\boldsymbol{s} = \{x_i\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^2$.
- **Solution**: represented as the permutation $\boldsymbol{\pi}$ of input sequences:

$$\boldsymbol{\pi} = \bigcup_{t=1}^{t=N} \{\pi_t\}, \quad \pi_t \in \{1, ..., N\}, \quad \pi_{t_1} \neq \pi_{t_2} \quad \text{if} \quad t_1 \neq t_2$$

Markov Decision Process (MDP) of TSP

State. State of MDP is represented as a partial solution of TSP or a sequence of previously selected actions: $\boldsymbol{\pi}_{1:t-1}$.
Action. Action is defined as selecting one of un-served tasks. Therefore, action is represented as π_t where the $\pi_t \in \{\{1, ..., N\} \setminus \{\boldsymbol{\pi}_{1:t-1}\}\}$.
Cumulative Reward. We define cumulative reward for solution (a sequence of assignments) from problem instance $\boldsymbol{s}$ as negative of tourlength: $-L(\boldsymbol{\pi}|\boldsymbol{s})$.
Constructive Policy. Finally we define constructive policy $p(\boldsymbol{\pi}|\boldsymbol{s})$ that generates a solution $\boldsymbol{\pi}$ from TSP graph $\boldsymbol{s}$. The constructive policy $p(\boldsymbol{\pi}|\boldsymbol{s})$ is decomposed as:

$$p(\boldsymbol{\pi}|\boldsymbol{s}) = \prod_{t=1}^{t=N} p_\theta(\pi_t|\boldsymbol{\pi}_{1:t-1}, \boldsymbol{s})$$

Where $p_\theta(\pi_t|\boldsymbol{\pi}_{1:t-1}, \boldsymbol{s})$ is a single-step assignment policy parameterized by parameter θ.

Seeding Process

Solution space. Solution space of seeder is a set of full trajectory solutions : $\{\boldsymbol{\pi}^{(1)}, ..., \boldsymbol{\pi}^{(M)}\}$.
Policy structure. Seeder is a constructive policy:

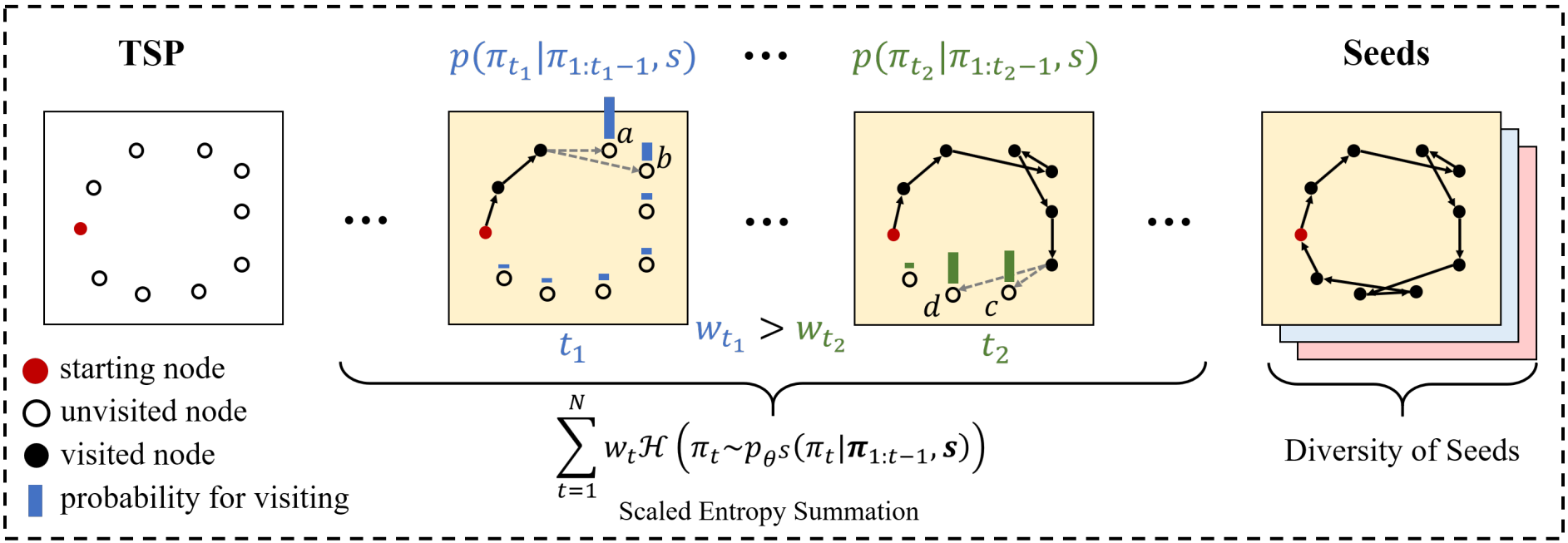
$$p^S(\boldsymbol{\pi}|\boldsymbol{s}) = \prod_{t=1}^{t=N} p_{\theta^S}(\pi_t|\boldsymbol{\pi}_{1:t-1}, \boldsymbol{s})$$

The segment policy $p_{\theta^S}(\pi_t|\boldsymbol{\pi}_{1:t-1}, \boldsymbol{s})$, parameterized by θ^S , is derived form **AM**.
Entropy Reward. To force the seeder policy p^S to sample **diverse solutions**, we trained p^S such that the entropy $\mathcal{H}$ of p^S to be maximized.

Scaled Entropy Maximization by *Minsu Kim et al.* (2021)

$$R^S = \mathcal{H}\left(\boldsymbol{\pi} \sim \prod_{t=1}^{t=N} p_{\theta^S}(\pi_t|\boldsymbol{\pi}_{1:t-1}, \boldsymbol{s})\right) \approx \sum_{t=1}^{N} w_t \mathcal{H}\left(\pi_t \sim p_{\theta^S}(\pi_t|\boldsymbol{\pi}_{1:t-1}, \boldsymbol{s})\right)$$

Weighted Entropy Maximization Scheme



- We use a linear scheduler (time-varying weights) $w_t = \frac{N-t}{N_w}$ to boost exploration at the earlier stage of composing a solution.
- Higher randomness imposed by the higher weight w_t at the early stage tends to generate more diversified full trajectories later.

Training scheme for Seeder.

- To train the seeder, we use the REINFORCE algorithm with rollout baseline b introduced by Kool et al.
- Then the gradient of each objective function is expressed as follows:

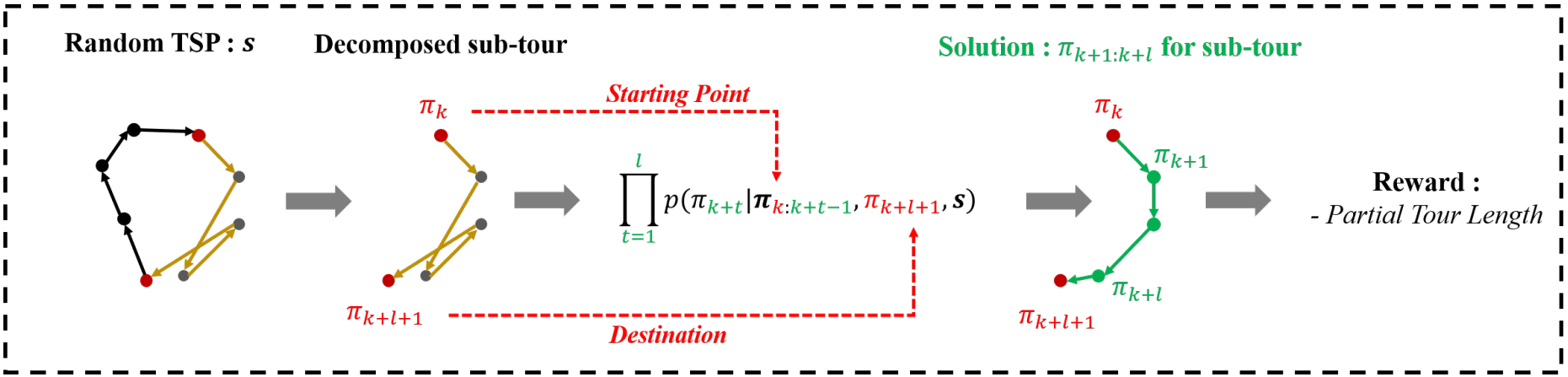
$$\nabla J(\theta^S|\boldsymbol{s}) = E_{\pi\sim p^S}[(L(\boldsymbol{\pi}|\boldsymbol{s}) - \alpha R^S(p^S_{1:N}, \boldsymbol{\pi}) - b(\boldsymbol{s}))\nabla log(p^S)]$$

$$\nabla J(\theta^R|\boldsymbol{s}) = E_{\pi\sim p^R}[(L(\boldsymbol{\pi}|\boldsymbol{s}) - b(\boldsymbol{s}))\nabla log(p^R)]$$

$$\nabla L^{IL}(\theta|\boldsymbol{x}) = E_{\boldsymbol{a}^*\sim p_{ex}}[-\nabla log(p(\boldsymbol{a}^*))]$$

$$\nabla L^{RL}(\theta|\boldsymbol{x}) = E_{\boldsymbol{a}\sim p}[(-R(\boldsymbol{a}|\boldsymbol{x}) - b(\boldsymbol{x}))\nabla log(p(\boldsymbol{a}))]$$

Revision Process



Solution space. Solution space of reviser is a partial segment of full trajectory solution represented as $\pi_{k+1:k+l}$. .

Policy structure. Reviser is a constructive policy as follows:

$$p^R(\pi_{k+1:k+l}|s) = \prod_{t=1}^{t=l} p_{\theta^R}(\pi_{k+t}|\pi_{k:k+t-1}, \pi_{k+l+1}, s)$$

Revision Reward: negative of partial tour length:

$$L^R(\pi_{k+1:k+l}|s) = \sum_{1}^{l+1} ||x_{\pi_{k+t}} - x_{\pi_{k+t-1}}||_2.$$

Experimental Setup and Baseline

- Experimental Results of LCP is presented with three main perspectives.
  1. Improvement from baseline of LCP (constructive model): i.e. **AM** or **PointerNet**.
  2. Comparison with the SOTA DRL-based improvement heuristics: e.g. **NLNS, DRL-2opt.**
  3. Comparison with problem-specific heuristic or MILP optimizer for proving near-optimality: e.g. **Concorde, LKH, ILS, Gurobi, OR-tools.**

Results (TSP)

Method	Type	$N = 20$		$N = 50$		$N = 100$	
		Cost	Gap	Cost	Gap	Cost	Gap
Gurobi	Solver	3.84	0.00%	5.70	0.00%	7.76	0.00%
Concorde	H	3.84	0.00%	5.70	0.00%	7.76	0.00%
S2V-DQN	RL	3.89	1.42%	5.99	5.16%	8.31	7.03%
Drl-2opt	RL, I	3.84	0.00%	5.70	0.12%	7.83	0.87%
AM	RL, S	3.84	0.05%	5.72	0.39%	7.93	2.13%
AM + LCP	RL, S		-	5.70	0.10%	7.85	1.13%
AM + LCP*	RL, S		-	**5.70**	0.02%	**7.81**	0.54%

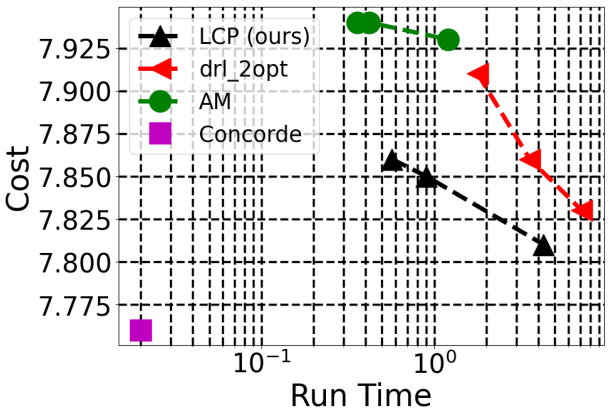
Results (PCTSP)

Method	Type	$N = 20$		$N = 50$		$N = 100$	
		Cost	Gap	Cost	Gap	Cost	Gap
Gurobi	Solver	3.13	0.00%	*OB*		*OB*	
OR Tools	H	3.14	0.05%	4.51	0.70%	6.35	6.21%
ILS (C++)	H	3.16	0.77%	4.50	0.67%	5.98	0.00%
AM	RL, S	3.15	0.41%	4.51	0.72%	6.07	1.57%
AM + LCP	RL, S	3.14	0.17%	4.50	0.51%	6.06	1.42%
AM + LCP*	RL, S	**3.14**	0.08%	**4.49**	0.32%	**6.04**	1.00%

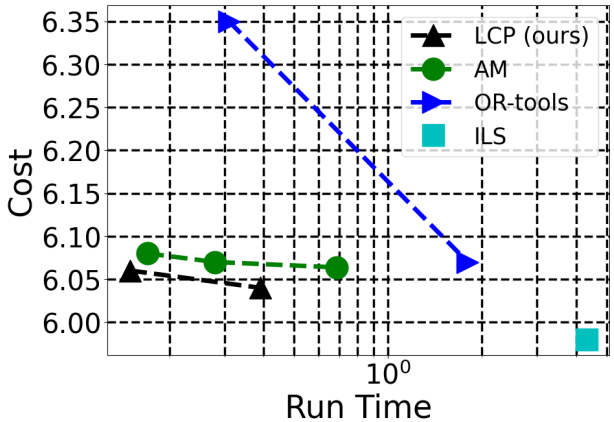
Results (CVRP)

Method	Type	$N = 20$		$N = 50$		$N = 100$	
		Cost	Gap	Cost	Gap	Cost	Gap
Gurobi	Solver	6.10	0.00%	*OB*		*OB*	
OR Tools	H	6.43	5.41%	11.31	9.01%	17.16	9.67%
LKH3	H	6.14	0.58%	10.38	0.00%	15.65	0.00%
RL {10}	RL, S	6.40	4.92%	11.15	7.46%	16.96	8.39%
NLNS	RL, I	6.19	1.47%	10.54	1.54%	16.00	2.17%
AM	RL, S	6.24	2.24%	10.59	2.06%	16.14	3.11%
AM+LCP	RL, S	**6.15**	0.84%	**10.52**	1.38%	16.00	2.24%
AM+LCP*	RL, S	-		-		**15.98**	2.11%

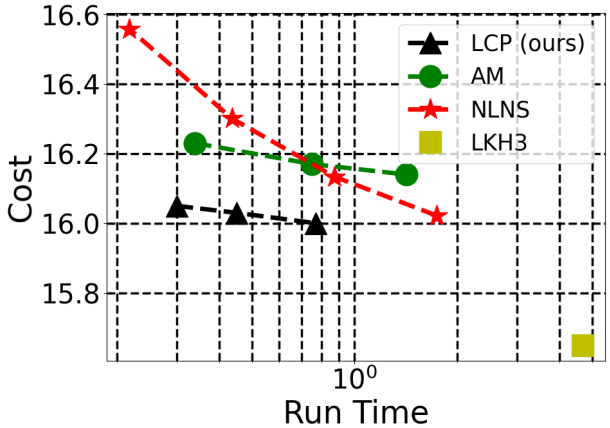
Scalability and Time-performance



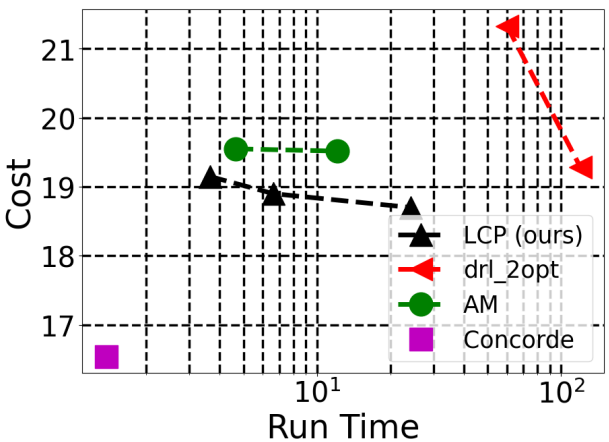
(a) TSP100



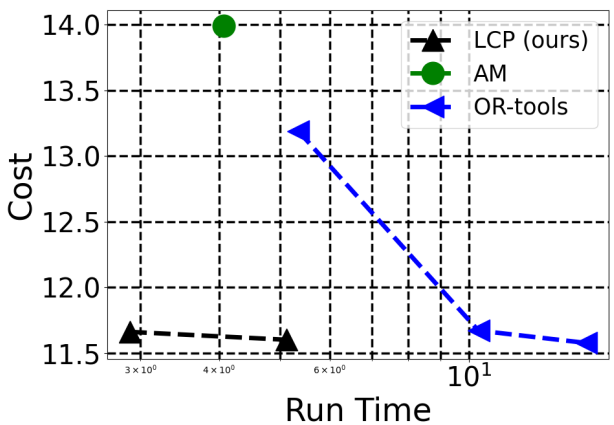
(b) PCTSP100



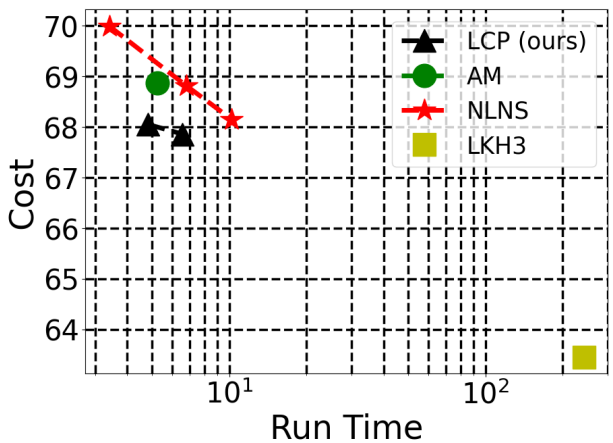
(c) CVRP100



(d) TSP500



(e) PCTSP500



(f) CVRP500

Results in TSPLIB

Instance	Opt.	AM			DRL-2opt			LCP *(ours)*		
		Cost	Gap	Time	Cost	Gap	Time	Cost	Gap	Time
eil51	426	435	2.11%	13s	**427**	0.23%	460s	429	0.73%	13s
berlin52	7,542	8663	14.86%	14s	7974	5.73%	460s	**7550**	0.10%	13s
st70	675	690	2.18%	23s	680	0.74%	540s	680	0.74%	13s
eil76	538	555	3.18%	27s	552	2.60%	540s	**547**	1.64%	18s
pr76	108,159	110,956	2.59%	27s	111,085	2.60%	540s	**108,633**	0.44%	18s
rat99	1,211	1,309	8.09%	44s	1,388	14.62%	680s	**1,292**	6.67%	24s
rd100	7,910	8,137	2.87%	46s	7,944	0.43%	680s	**7,920**	0.13%	26s
KroA100	21,282	23,227	9.14%	46s	23,751	11.60%	680s	**21,910**	2.95%	26s
KroB100	22,141	23,227	8.23%	46s	23,790	7.45%	680s	**22,476**	1.51%	26s
KroC100	20,749	21,868	5.40%	46s	22,672	9.27%	680s	**21,337**	2.84%	26s
KroD100	21,294	22,984	7.94%	46s	23,334	9.58%	680s	**21,714**	1.97%	26s
KroE100	22,068	22,686	2.80%	46s	23,253	5.37%	680s	**22,488**	1.90%	26s
eil101	629	654	4.03%	46s	**635**	0.95%	680s	645	2.59%	26s
lin105	14,379	16,516	14.87%	49s	16,156	12.36%	680s	**14,934**	3.86%	26s
pr124	59,030	63,931	8.30%	68s	**59,516**	0.82%	700s	61,294	3.84%	37s
bier127	118,282	125,256	5.90%	72s	**121,122**	2.40%	720s	128,832	8.92%	37s
ch130	6,110	6,279	2.76%	77s	6,175	1.06%	790s	**6,145**	0.57%	38s
pr136	96,772	101,927	5.33%	84s	98,453	1.74%	820s	**98,285**	1.56%	38s
pr144	58,537	63,778	8.95%	93s	61,207	4.56%	720s	**60,571**	3.47%	43s
kroA150	26,524	28,658	8.05%	102s	30,078	13.40%	900s	**27,501**	3.68%	44s
kroB150	26,130	27,565	5.49%	102s	28,169	7.80%	900s	**26,962**	3.18%	44s
pr152	73,682	79,442	7.82%	101s	**75,301**	2.20%	720s	75,539	2.52%	44s
u159	42,080	50.656	20.38%	111s	**42,716**	1.51%	840s	46,640	10.84%	45s
rat195	2,323	**2,518**	8.14%	168s	2,955	27.21%	1080s	2,574	10.81%	57s
kroA200	29,368	33,313	13.43%	173s	32,522	10.74%	1,120s	**31,172**	6.14%	86s
ts225	126,643	138,000	8.97%	223s	**127,731**	0.86%	1,110s	134,827	6.46%	113s
tsp225	3,919	4,837	23.42%	224s	**4,354**	11.10%	1,160s	4,487	14.50%	113s
pr226	80,369	90,390	12.47%	228s	91,560	13.92%	940s	**85,262**	6.09%	113s
gil262	2,378	2,588	8.81%	306s	**2,490**	4.71%	1380s	2,508	5.49%	134s
lin318	42,029	47,288	12.51%	397s	**46,065**	9.60%	1,470s	46,540	10.72%	158s
rd400	15,281	17,053	11.59%	458s	**16,159**	8.10%	1,870	16,519	8.10%	209s
pr439	107,217	160,594	49.78%	744s	143,590	33.92%	1760s	**130,996**	22.18%	228s
pcb442	50,778	58,891	15.98%	897s	57,114	12.48%	1,760s	**57,051**	12.35%	228s
avg. gap	0.00%		9.90%			7.63%			**5.14%**	

Ablation Study in AM and PointerNet

Component of the LCP			TSP		PCTSP		CVRP	
Entropy	Weight Scheduling	Reviser	cost	gap	cost	gap	cost	gap
			7.96	2.65%	6.08	1.64%	16.29	3.43%
✓			7.96	2.68%	6.08	1.76%	16.25	3.16%
✓	✓		7.94	2.45%	6.07	1.62%	16.20	2.86%
		✓	7.86	1.32%	6.04	1.13%	16.20	2.86%
✓		✓	7.84	1.17%	6.05	1.16%	16.16	2.59%
✓	✓	✓	**7.82**	**0.88%**	**6.04**	**1.02%**	**16.12**	**2.37%**

Component of the LCP			Pointer Network (greedy)		Pointer Network {1280}	
Entropy	Weight Scheduling	Reviser	cost	gap	cost	gap
			3.95	2.63%	7.33	90.75%
✓			3.95	2.71%	7.30	89.77%
✓	✓		3.95	2.62%	7.32	90.29%
		✓	3.89	1.27%	3.85	0.21%
✓		✓	3.89	1.18%	3.85	0.24%
✓	✓	✓	3.89	1.24%	**3.85**	**0.20%**

Thank You for Listening!

min-su@kaist.ac.kr