Test-Time Search in Neural Graph Coarsening Procedures for the Capacitated Vehicle Routing Problem

NEURAL INFORMATION PROCESSING SYSTEMS

Yoonju Sim
KAIST

Hyeonah Kim
Mila

Changhyun Kwon*
KAIST

Problem Definition

◆ Capacitated Vehicle Routing Problem

$$\text{minimize} \sum_{(i,j)\in E} c_{ij}x_{ij}$$

$$\text{subject to} \quad x(\delta(\{i\})) = 2 \quad \forall i \in V_C$$

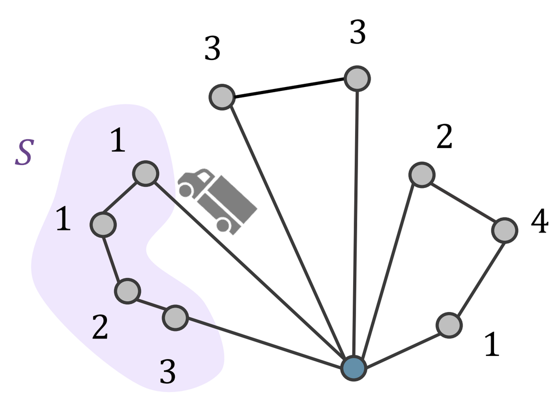
$$x(\delta(\{0\})) = 2K$$

$$x(\delta(S)) \geq 2\,r(S) \quad \forall S \subseteq V_C \quad \text{(Capacity Inequality)}$$

$$x_{ij} \leq 1 \quad \forall 1 \leq i < j \leq |V|$$

$$x_{0j} \leq 2 \quad \forall j \in V_C$$

$$x_{ij} \in \mathbb{Z}_+ \quad \forall j \in V,$$



(e.g.) a CVRP instance with a vehicle capacity of 7 to serve 9 customers

where K is the number of available vehicles to serve all customers

➔ **Too many capacity inequalities ➔ handled via cutting plane methods!**

▪ **Rounded Capacity Inequalities (RCIs)**

$$x(\delta(S)) \geq 2\left\lceil \sum_{i\in S} \frac{d_i}{Q} \right\rceil$$

▪ **Framed Capacity Inequalities (FCIs)**

$$x(\delta(H)) + \sum_{i\in I} (\delta(S_i)) \geq 2r(\Omega) + 2\sum_{i\in I} \left\lceil \frac{d(S_i)}{Q} \right\rceil$$

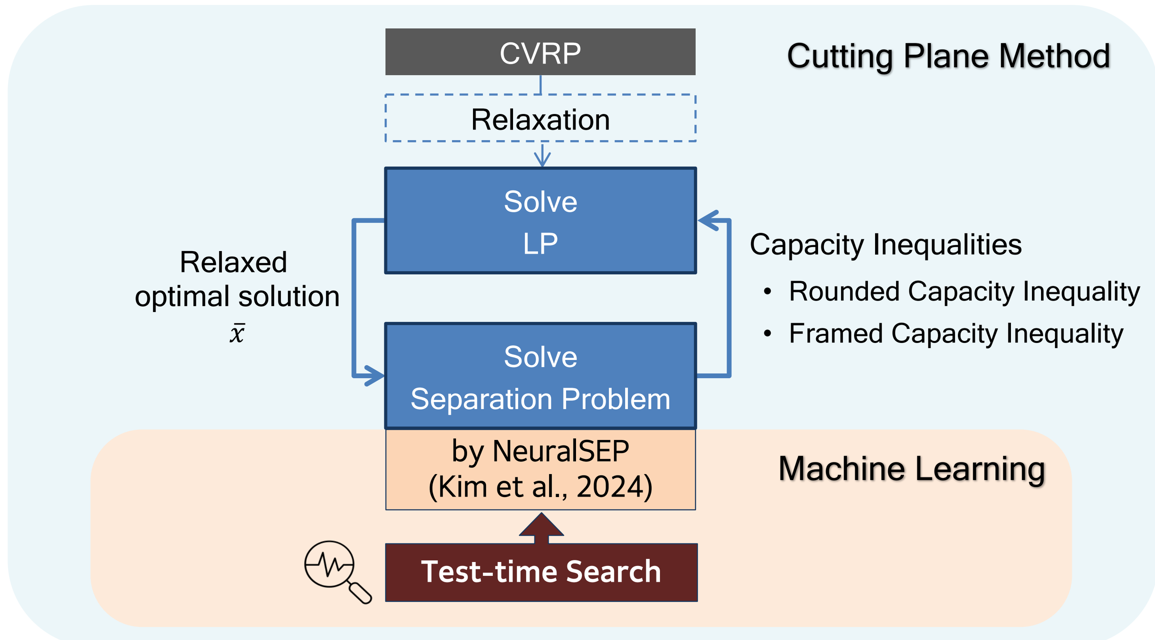
➔ **Find RCIs and FCIs by using test-time search method in neural graph coarsening!**

Motivation & Overview

◆ Motivation

– **Overcome** the **NP-hard exact separation** of RCI/FCI, which limits solver scalability.
– **Replace** traditional heuristics (CVRPSEP) with an efficient learning-based algorithm, **NeuralSEP**.
– **Fully leverage** the trained model's potential by employing a **Test-Time Search (TTS)** technique during inference to **enhance the performance**.

◆ Overall Structure

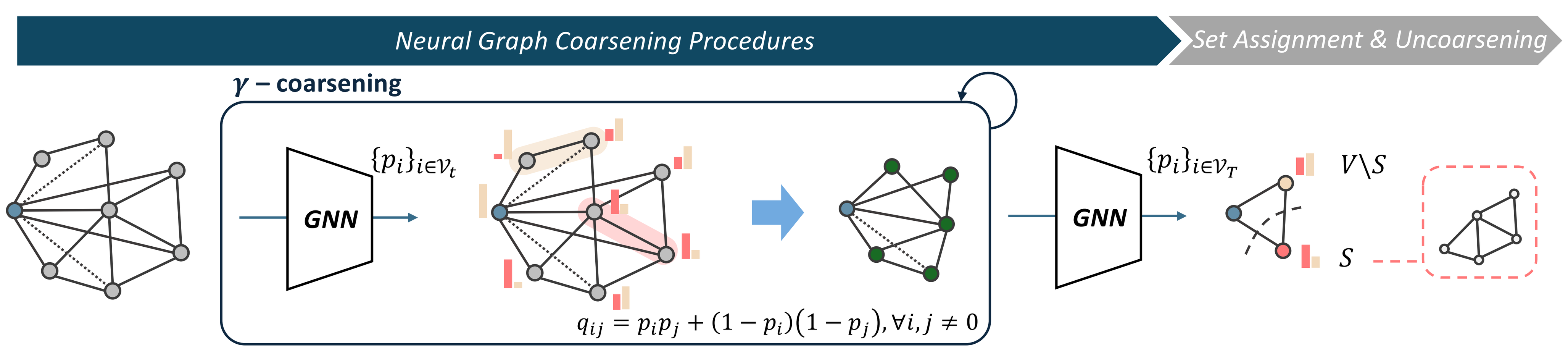


Employ NeuralSEP as the separation algorithm within **a cutting plane method** to generate valid capacity inequalities (cuts) for the CVRP, further integrating a Test-Time Search (TTS) technique.

Kim, H., Park, J., & Kwon, C. (2024). A neural separation algorithm for the rounded capacity inequalities. *INFORMS Journal on Computing*, *36*(4), 987-1005.

Methodology

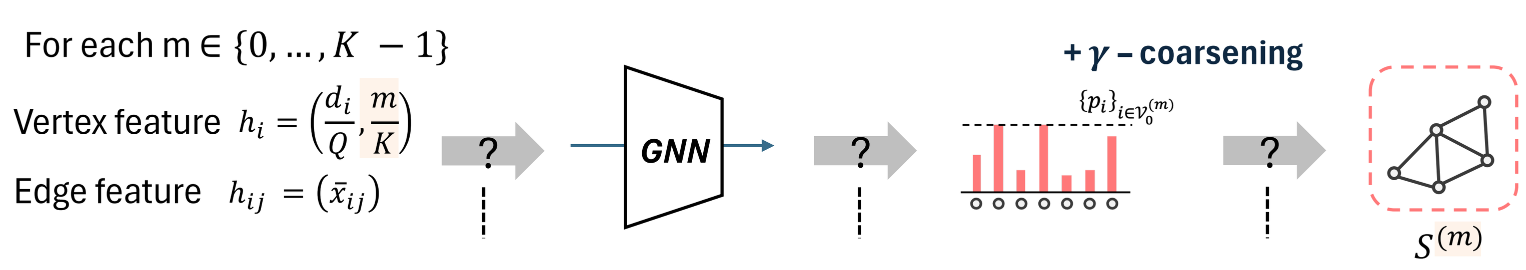
◆ Neural Graph Coarsening Procedures (Kim et al, 2024)



– **NeuralSEP** learns from the optimal solutions of the **exact separation problem**, encoding the fractional solution ($\bar{x}$) and a demand & vehicle-related parameter as graph features for the input.
– The GNN predicts probabilities for **vertex inclusion**, which drive an iterative graph coarsening process to simplify the graph and determine the final **candidate subset** (S).
– **Check for the RCI violation** of these candidate subsets.

◆ Limitation of NeuralSEP

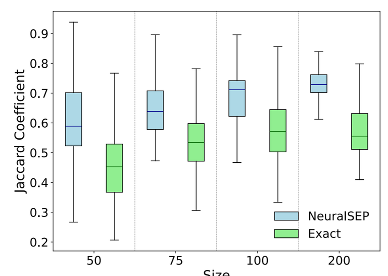
Although NeuralSEP performs effectively in large-scale instances, we observe an issue:
It finds substantially **fewer** cuts than the exact separation method, despite being trained to **imitate** it.



(a) Partial Derivative **(b) Cosine Similarity** **(c) Jaccard Coefficient**

Here, m is the integer-valued parameter in the exact separation problem.

We utilize three key metrics (including the **Jaccard Coefficient**) to show that **NeuralSEP** generates highly similar and thus **less diverse subsets** than the exact separation method.



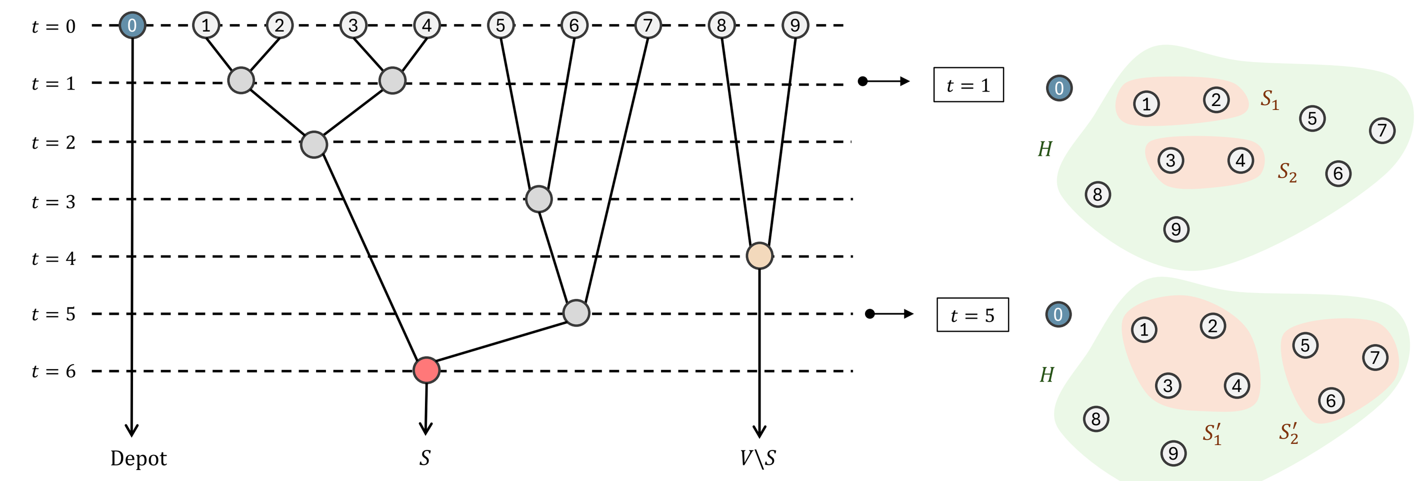
(c) Metric D_3: Jaccard Coefficient

◆ Solution: π-greedy method TTS

Calculate $q_{ij} = p_i p_j + (1-p_i)(1-p_j) + \pi_{ij}, \quad \forall i,j \neq 0 \quad \pi_{ij} \sim \mathcal{U}(0, 0.001)$

→ Find $(i,j) \in \bar{E}$ that maximizes q_{ij}

◆ Graph Coarsening History based Partitioning (GraphCHiP) algorithm TTS



Utilize the intermediate records of the **graph coarsening** to identify the candidate subsets & partitions.

Experimental Results

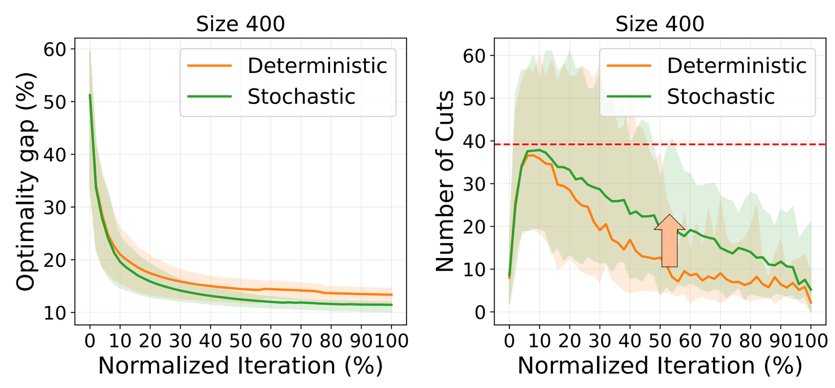
◆ Comparison of RCI separation algorithm

▪ Dataset: Evaluation on Randomly Generated CVRP
▪ Baseline: CVRPSEP, Original NeuralSEP, NeuralSEP with migrated library
▪ Metric: Optimality Gap (%)

Size	CVRPSEP		NeuralSEP$_1$		NeuralSEP$_2$		π-NeuralSEP$_2$ + GC	
	Gap	Time/Iter	Gap	Time/Iter	Gap	Time/Iter	Gap	Time/Iter
50	**1.970%**	0.009	4.151%	0.830	5.250%	0.120	3.679%	0.133
75	**2.769%**	0.054	5.305%	1.066	5.164%	0.209	4.393%	0.246
100	**4.539%**	0.145	6.611%	1.440	6.410%	0.378	5.953%	0.394
200	**6.280%**	2.001	9.214%	3.411	8.314%	1.293	7.683%	1.594
300	**7.903%**	10.431	10.515%	12.006	10.087%	4.607	8.714%	7.482
400	12.618%	16.936	12.848%	26.714	13.632%	13.518	**10.970%**	19.850
500	16.357%	16.947	15.413%	41.227	14.826%	26.705	**13.429%**	39.125
750	25.783%	16.603	22.553%	102.623	22.187%	90.436	**20.956%**	111.835
1,000	30.408%	23.321	28.777%	161.183	26.434%	139.826	**26.136%**	159.042

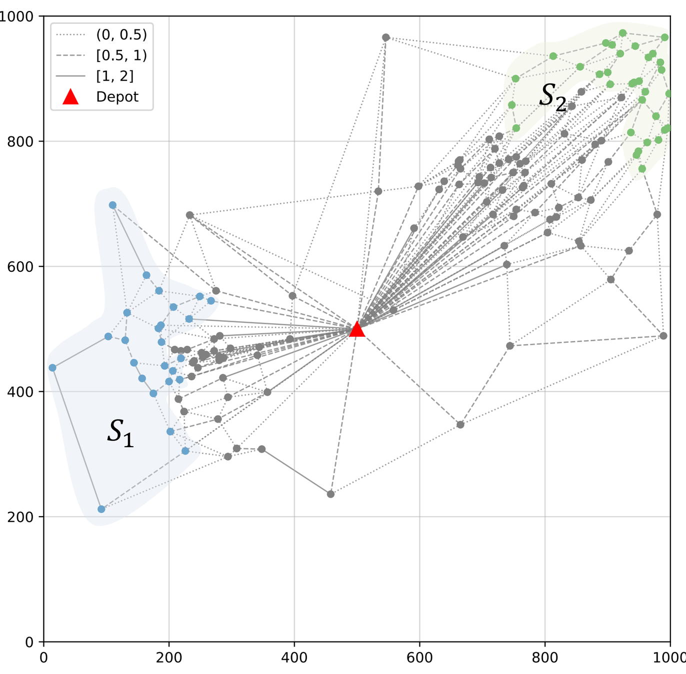
Maximum runtime: 1 hour

◆ Performance of test-time search method for RCIs



Both the π -greedy method and the **GraphCHiP** algorithm significantly **increase** the yield of high-quality **RCI cuts**, resulting in a substantial **reduction of the optimality gap**.

◆ Performance of GraphCHiP algorithm for FCIs



▪ A example of an FCI found by GraphCHiP on 'X-n153-k22' instance ($Q = 144$)
▪ Calculate FCI violation

$$\bar{x}(\delta(H)) + \bar{x}(\delta(S_1)) + \bar{x}(\delta(S_2)) = 44.0 + 10.5 + 11.19 = 65.69$$

$$\ngeq 2r(\Omega) + 2\left(\left\lceil \frac{\sum_{i\in S_1} d_i}{Q} \right\rceil + \left\lceil \frac{\sum_{i\in S_2} d_i}{Q} \right\rceil\right) = 2(23+5+5) = 66$$

▪ The existence of FCI cuts is highly dependent on the problem structure.
▪ As a result, adding FCI results in further reduction of the optimality gap.

Conclusion

❖ **We observe the limitations and potential improvements of NeuralSEP based on three key evaluation metrics.**

❖ **We propose a π-greedy method and the GraphCHiP algorithm to generate not only RCIs but also FCIs without retraining the model.**

❖ **To our knowledge, this is the first learning-based approach to find FCIs.**

❖ **The proposed test-time search method can be applied to other learning-based algorithms that employ iterative graph coarsening.**

KAIST COMPUTATIONAL OPTIMIZATION METHODS LAB