Learning to Segment for Capacitated Vehicle Routing Problems

Wenbin Ouyang, Sirui Li, Yining Ma, and Cathy Wu
MIT, {oywenbin, siruil, yiningma, cathywu}@mit.edu

Extended abstract submitted for presentation at the 12th Triennial Symposium on Transportation Analysis conference (TRISTAN XII)
June 22-27, 2025, Okinawa, Japan

February 13, 2025

Keywords: Vehicle Routing, Machine Learning, Neural Combinatorial Optimization

1 INTRODUCTION

Vehicle Routing Problems (VRPs) are combinatorial optimization problems with profound applications in logistics, transportation, and supply chain systems (Laporte, 2009). Iterative search heuristics, e.g., LKH-3 (Helsgaun, 2017) and HGS (Vidal, 2022), are state-of-the-art VRP solvers, which progressively refine solutions through local search and metaheuristics. However, as depicted in Figure 1, existing methods often involve redundant searches, with a significant portion of edges remaining unchanged during search steps, especially in later search stages. This limits both efficiency and scalability, particularly for large-scale instances with over 1,000 customers.

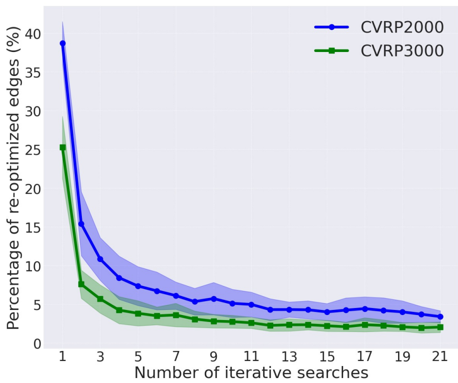

Figure 1 – *Percentage of re-optimized edges during iterative search process: using LKH-3 on 100 CVRP instances of sizes 2,000 and 3,000, we observe that many edges remain unchanged, particularly in later optimization stages. See Section 3 for CVRP setting details.*

Building on this critical observation, we present **Learn-to-Segment (L2Seg)**, a novel approach to accelerate iterative VRP solvers by leveraging deep learning to dynamically identify stable solution segments, enabling re-optimization of only the remaining parts. We first introduce a generic decomposition technique, First-Segment-Then-Aggregate (FSTA), for VRPs. FSTA first segments the VRP solutions by grouping stable portions, and then represents each segment into hypernodes with aggregated attributes of nodes in the segment. It preserves optimality while enabling problem reduction across iterations. We then design a customized graph neural network with a tailored loss function to predict segments by estimating edge re-optimization probabilities dynamically at different optimization stages. Empirical results on Capacitated VRP (CVRP) show that L2Seg accelerates various traditional and learning-guided state-of-the-art VRP solvers by up to **32%** while also enhancing solution quality. Notably, our L2Seg has strong potential for extension to other VRP variants and integrating with other decomposition frameworks, holding promise to offer great flexibility and performance gains for various iterative VRP solvers.

2 METHODOLOGY

For brevity, we focus on the CVRP here, with our ongoing work extending the framework to other VRP variants. A CVRP instance, denoted as P, is defined on an undirected graph $G = (V, E)$, where $V = \{0\} \cup \{1, ..., n\}$ represents depot (0) and customers; and E is the distance matrix. Each customer $v \in V$ has a demand d_v. The objective is to suggest routes with minimal length L for a fleet of vehicles with capacity C, such that all vehicles start and end at the depot, each customer is served exactly once, and the total demand on each route does not exceed C.

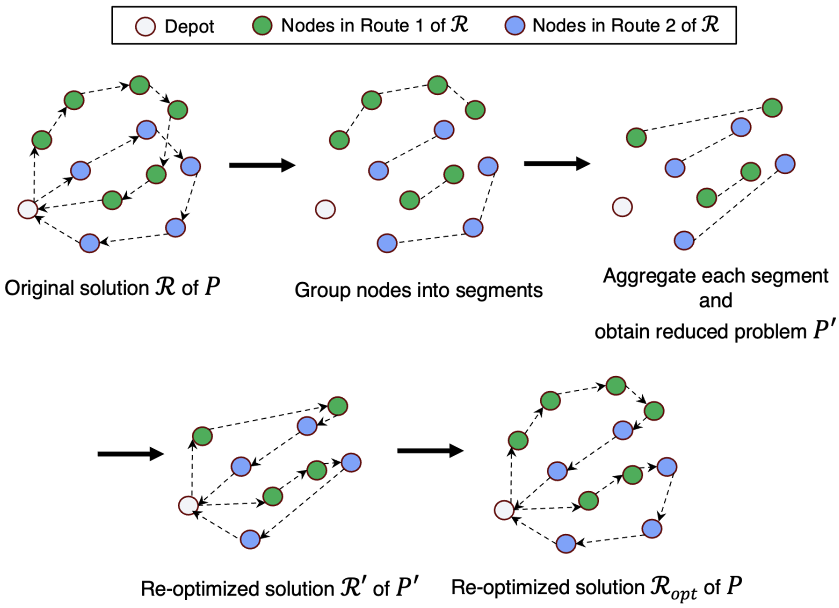

Figure 2 – *Overview process of our First-Segment-Then-Aggregat (FSTA) technique. Depot is shown in white; nodes with the same color belong to the same routes of the original solution $\mathcal{R}$.*

2.1 First-segment-then-aggregate technique

At each search step, we employ FSTA to reduce the search redundancy and the scales of decision variables. Given a solution $\mathcal{R} = \{R^{(1)}, ..., R^{(m)}\}$ with m vehicles and total route length $L_{\mathcal{R}}$, $R^{(i)} = \left(0, r_1^{(i)}, ..., r_{|R^{(i)}|-2}^{(i)}, 0\right)$ is the route for the i^{th} vehicle. A segment is defined as a sequence of consecutive edges within any route $R^{(i)}$ and is represented by an ordered list of customers, $S = (v_1, v_2, \ldots, v_{|S|}) \subseteq R^{(i)}$. The FSTA process consists of three steps. Specifically,

1) **Segment Partitioning:** FSTA groups the customers in each route $R \in \mathcal{R}$ into several distinct segments, ensuring that no nodes are shared between any two segments.

2) **Hypernode Aggregation:** FSTA aggregates each segment S and replaces it with two hypernode $S' = (v'_1, v'_2)$, with the locations of the two nodes being the first and the last nodes of S ($v'_1 = v_1$, $v'_2 = v_{|S|}$). The demands are set as $d_{v'_1} = d_{v'_2} = \frac{1}{2}\sum_{v \in S} d_v$. Additional constraints are imposed to keep including the edge (v'_1, v'_2) in the solutions.

3) **Solution Recovery:** We solve the reduced problem P', obtain the refined solution $\mathcal{R}'$ for P', and recover the corresponding solution $\mathcal{R}_{\text{opt}}$ for P by replacing each S' back to S.

Figure 2 demonstrates a overview of our FSTA. Note that step 2 of FSTA accurately preserves the structure of the original solution, allowing FSTA to create smaller subproblems at each step without affecting search optimality. In the literature, FSTA is somewhat conceptually related to the path decomposition in Santini *et al.* (2023), which typically clusters geographically close routes using route barycenters. However, FSTA permits segments of arbitrary lengths, allows reversal of the segments, and aggregates segment attributes for a more compact representation. Moreover, we introduce an automated L2Seg approach that leverages deep learning to address the challenge of identifying stable segments (step 1 of FSTA) at each search step dynamically.

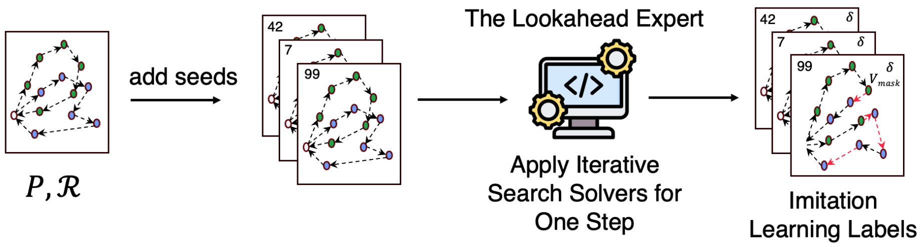

Figure 3 – *Overview of our label generation process. To enhance label generation, we assign different random seeds to the Lookahead Expert for diverse solution refinements.*

2.2 Learning to segment (L2Seg) framework

Predicting stable segments corresponds to estimating the re-optimization probability for each edge. Our L2Seg network takes the CVRP instance P and current solution $\mathcal{R}$ as input and outputs the re-optimization probability per edge. Node features include coordinates, demands, and a one-hot encoding indicating whether a node is a depot; edge features include length and a one-hot indicator for solution membership. We design an architecture consisting of a graph neural network (GNN) encoder and a transformer-based decoder with one transformer followed by a sigmoid activation function to produce the edge-re-optimization probabilities. Note that we simplify edge selection by treating it as node selection, removing both connected edges for the selected nodes. Once edges are removed, we apply FSTA to aggregate the remaining segments in the route, creating a simpler subproblem and accelerating the search.

We train the model by imitating a Lookahead Expert to accelerate a backbone iterative search heuristic (LKH-3, LNS, and L2D in this work). At each search step, the Lookahead Expert performs a one-step search based on the backbone heuristic that changes the current solution R to a potentially improved solution R'. We then record the edges that were re-optimized, along with the associated objective value improvements δ. These re-optimized edges serve as labels for our network, with δ as weights to highlight label significance. We also incorporate a regularization term to guide the model in estimating the number of edges needing re-optimization, based on the count determined by the Lookahead Expert. Let p_v represent the output probability of re-optimizing the two edges connected to node v, and let $V_{\text{re-opt}}$ denote the set of nodes selected by the lookahead expert for re-optimization. We design the loss function as follows:

$$Loss(\mathbf{p}) = -\frac{\mathbf{1}}{|\mathbf{V}_{\text{re-opt}}|}\sum_{\mathbf{v}\in\mathbf{V}_{\text{re-opt}}} \delta\log(\mathbf{p_v}) + \alpha(\sum_{\mathbf{v}\in\mathbf{V}} \mathbf{p_v} - |\mathbf{V}_{\text{re-opt}}|)^{\mathbf{2}}$$

where α is a hyperparameter that balances the two loss terms. The first term employs an entropy-like loss to increase the re-optimization probability of suboptimal edges based on the importance weights δ, while the second term prevents the model from disrupting too many stable edges.

Lastly, to generate the training dataset, we use multiple seeds to augment each instance P as shown in Figure 3, with P sampled from a problem distribution. We train a single model that can accelerate the backbone solver for any new instance from this distribution at each time step.

3 RESULTS & DISCUSSION

In this section, we show that L2Seg can improve the efficiency and effectiveness of existing iterative search heuristics, demonstrating its broad applicability for boosting various solvers.

Table 1 – *Objective value comparisons, with improvement % over each heuristic w/o L2Seg.*

Methods	CVRP2000		CVRP3000		CVRP5000	
	Obj.↓	Impr.↑	Obj.↓	Impr.↑	Obj. ↓	Impr.↑
LKH-3	43.33	-	46.12	-	51.23	-
LKH-3+L2Seg	42.30	2.38%	45.89	0.50%	50.12	2.17%
LNS	41.21	-	45.85	-	49.97	-
LNS+L2Seg	40.75	1.12%	45.42	0.94%	49.52	0.90%
L2D	41.71	-	45.78	-	49.67	-
L2D+L2Seg	40.65	2.54%	45.24	1.18%	49.22	0.91%

Table 2 – *Solve time comparisons, with improvement % over each heuristic w/o L2Seg.*

Methods	CVRP2000		CVRP3000		CVRP5000	
	Time(s) ↓	Impr. ↑	Time(s) ↓	Impr. ↑	Time(s) ↓	Impr. ↑
LKH-3	45.12	-	37.25	-	35.45	-
LKH-3+L2Seg	37.51	16.87%	26.33	29.32%	27.44	22.60%
LNS	41.85	-	33.52	-	32.45	-
LNS+L2Seg	28.35	32.26%	27.12	19.09%	25.66	20.92%
L2D	31.42	-	31.24	-	26.33	-
L2D+L2Seg	24.12	23.23%	25.21	19.30%	24.21	8.05%

Data. We evaluate L2Seg on clustered CVRPs with sizes 2000, 3000, and 5000 and capacity 500, 1000, and 1500, respectively. The clustered distribution is created by sampling a centre (x_c, y_c) in the unit square, an angle ϕ from 0° to 180°, and assigning 4 more nodes at locations of $(x_c + i \times r_c \cos\phi, y_c + i \times r_c \sin\phi), i \in \{-2, -1, 1, 2\}$. We set r_c to 0.01 for this work.

ML Setup. We train L2Seg with learning rate 10^{-3} on a machine equipped with one V100 GPU and 48 CPU cores. The training dataset contains around 20,000 labels. We use $\alpha = 5$ and training takes around 12 hours. We test our model on 100 CVRP instances.

Iterative Search Heuristics. We employ LKH-3, large neighborhood search (LNS), and Learning to delegate (L2D) from Li *et al.* (2021), which represent state-of-the-art iterative search heuristics in their respective categories: heuristics, metaheuristics, and learning-guided heuristics.

Metrics. We either fix the time budget to 200 seconds and compare the resulting solution quality, or we fix the solution performance at 95% of the improvement achieved by LKH-3 within 200 seconds and compare the time required by each method to reach this benchmark.

The empirical results corroborate that L2Seg not only accelerates state-of-the-art iterative solvers up to 32% (Table 2) but also enhances solution quality (Table 1). It also shows that L2Seg has the potential to enhance various iterative VRP solvers. Ongoing work aims to extend L2Seg to additional VRP variants, more realistic instances, and benchmark it against more baselines.

References

Helsgaun, Keld. 2017. An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. *Roskilde: Roskilde University*, **12**, 966–980.

Laporte, Gilbert. 2009. Fifty years of vehicle routing. *Transportation science*, **43**(4), 408–416.

Li, Sirui, Yan, Zhongxia, & Wu, Cathy. 2021. Learning to delegate for large-scale vehicle routing. *Advances in Neural Information Processing Systems*, **34**, 26198–26211.

Santini, Alberto, Schneider, Michael, Vidal, Thibaut, & Vigo, Daniele. 2023. Decomposition strategies for vehicle routing heuristics. *INFORMS Journal on Computing*, **35**(3), 543–559.

Vidal, Thibaut. 2022. Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. *Computers & Operations Research*, **140**, 105643.