Proceedings of the IISE Annual Conference & Expo 2023
K. Babski-Reeves, B. Eksioglu, D. Hampton, eds.

Analysis of Decoding Strategies for Transformer-Based Solution of Multi-Depot Vehicle Routing Problems

Seyedeh Shaghayegh Rabbanian[a], Hao Wang[a], and Gerald M. Knapp[a]

[a] *Louisiana State University, Baton Rouge, LA, USA*

Abstract

Multi-Depot Vehicle Routing Problem (MDVRP) is a well-known problem with many real-world applications in supply chains. In MDVRP, a set of customer nodes needs to be served by a fleet of vehicles supplied by multiple depots. Each vehicle starts from a depot, visits a subset of customers in a sequential order, and then returns to the original depot. Finding the optimal solution to this NP-hard problem can be challenging in the area of distribution and logistics. The goal is to find (near-)optimal routes that minimize the total transportation costs. We solve and improve the model using a transformer-based encoder-decoder architecture that has been originally proposed for neural machine translation. Our proposed method produces a set of consecutive actions which assign each vehicle to a certain customer node using a decoding strategy at each decoding step. After training, the model has low inference time, and produce high-quality solutions in a few seconds for variant number of customers which is faster compared to other exact and approximate approaches. In this paper, we investigate the impact of different decoding strategies, including greedy, multinomial sampling, and beam search, on solution quality. An ensemble decoding strategy is presented which further improves the efficiency of the transformer architecture in finding near-optimal MDVRP solutions in 2 out of every 3 cases on average. We compare the results of our proposed transformer architecture to original transformer and an extension of Lin-Kernighan heuristic solver (LKH-3) which has found many best-known solutions for well-known benchmarks.

Keywords
Transformer architecture, decoding strategies, vehicle routing problem, ensemble methods

1. Introduction

The Vehicle Routing Problem (VRP) is defined as a number of vehicles with capacity constraints, starting from a depot, that are to be routed to service a set of geographically distributed customers with differing demands and then returning to the depot after servicing the last customer in their scheduled route [1, 2]. VRP is a well-known NP-hard problem which makes it computationally difficult to find the exact solution in a reasonable timeframe particularly for large-scale VRP. Multi-Depot Vehicle Routing Problem (MDVRP) is a variant of VRP which involves multiple depots or starting nodes for the vehicles. There are various constraints in MDVRP such as the number of vehicles, time windows, and the capacity of each delivery truck. The goal is to find (near-)optimal permutation of visiting customer nodes to minimize the total tour length traveled by the delivery trucks and satisfy all customer demands.

2. Problem Description

Due to computational complexity of MDVRP, many approximate methods including heuristic and metaheuristic approaches have been proposed. Approximate solutions can find good-quality solutions but are not guaranteed to be optimal. Recently, researchers have focused on solving VRP variants using learning-based approaches. Learning-based approaches can solve VRP with on par or better solution quality than many non-learning based approximate methods. Through learning-based approaches, we can extract internal features of the input data (depots and customers coordinate locations, and customers' demands).

The Transformer architecture [3] is a sequence-to-sequence deep learning model that can handle large amounts of data and variable sequence lengths, both important factors for VRPs. Transformer models are based on a self-attention mechanism which allows weighing the importance of different elements in a sequence. This architecture is well-suited choice for solving VRP because of their ability to effectively process sequential data, such as the order of the customers to be visited, as well as varying sequence lengths. Although transformer architecture can achieve (near-)optimal solutions in sequence modeling, training them is computationally intensive. Our experiments show that by employing an ensemble of decoding strategies in the inference phase, the same architecture with the same learned parameters can generate outputs close to optimal with no significant increase in computational cost.

In this paper, we demonstrate that if we train the network using a multinomial sampling strategy only once, utilizing an ensemble of sampling (multinomial distribution), greedy and beam search decoding strategies can enhance the results in approximately two-thirds of the cases when compared to using a single decoding strategy alone. For the remaining one-third of the cases, the results are as good as a single decoding strategy. This approach helps us extract more information from the trained model simply by decoding the context vector generated by the encoder in different ways and further balancing the exploration-exploitation dilemma.

3. Related Research

In the last decade, deep learning has remarkably improved computer vision (CV), natural language processing (NLP), and speech recognition by replacing handcrafted features by features learned from data. Consequently, using deep learning approaches has become increasingly attractive for solving NP-hard combinatorial optimization problems such as Traveling Salesman Problem (TSP) and VRP to improve distribution performance and customer satisfaction, as well as cost reduction.

As a pioneering work in neural-based VRP solvers, Vinyals *et al.* [4] used pointer networks, a combination of Recurrent Neural Networks (RNN) and supervised learning to solve TSP. As few labeled 'optimal' sequence datasets are available for large problems to use supervised learning, Bello *et al.* [5] proposed an RL-based pointer network to solve TSP. Nazari *et al.* [6] extended their work to solve Capacitated VRP (CVRP). Transformers are a recent innovation in deep learning models that have quickly been widely adapted for use in NLP, CV, speech processing, and most recently combinatorial optimization. The benefits of transformers over standard RNN architectures lies in handling long range dependencies and more efficient representation and computation [7, 8]. Kool *et al*. [9] proposed an early application of transformers to VRP using an attention mechanism. Xin *et al.* [10] developed a multi decoder attention mechanism that learns multiple diverse policies which increases the chance of better performance. Unlike other neural architectures, transformers are capable of handling sequences or sets of variable length for both inputs and outputs.

4. Methodology

The transformer architecture (aka Encoder-Decoder Architecture) consists of an encoder and a decoder. This architecture was initially developed to perform Neural Machine Translation (NMT) from one language to another language. The similarity between language translation and VRP led to increased focus on using transformers for routing problems. In NMT, both input and output are sequences; but in MDVRP, only the output is a sequence; the input is a set of feature vectors, not a sequence. The encoder extracts internal (learned) features from the input. The encoder block is mainly composed of a multi-head self-attention sublayer and a position-wise feed-forward network (FFN) sublayer to update the initial input embedding with L sequential layers. Each sublayer has a batch normalization (BN) process leading to a higher learning rate and a more effective optimization [11, 12]. The decoder block inserts attention modules and the position-wise FFNs. The decoder uses the encoded features to select the next node to add to a vehicle sequence. The decoder step repeats until all customer nodes are traversed and the delivery trucks are routed back to the depot. A partial solution is produced at each decoder step. This process continues until the solution is constructed step by step. A masking mechanism is used to remove already visited nodes from consideration. In the beginning, all the delivery trucks are at the depots; therefore, the depots are masked at this first step. The depots are subsequently unmasked in later steps to allow each vehicle to return to its original depot. The reward (negative tour length) is calculated at end of decision sequence when all the nodes are visited and propagated backwards to update the encoder-decoder architecture weights.

Input and Output representation are as follows:

- *Input representation*. The input to the network is a depot and N customer nodes $\mathrm{X} = \{x_0, x_1, \dots x_N\}$ with their coordinate locations (x_i, y_i), customer demand (cd_i) where $CD = \{cd_1, cd_2, \dots, cd_N\}$, number of delivery trucks and, delivery truck capacity.

- *Output representation*. Given the input, we seek to find the (near-)optimal sequence of visiting nodes for each vehicle to distribute the orders as the output. In fact, we define the solution as multiple tours and each tour is the permutation of the nodes which the sequence of nodes to be visited for each vehicle.

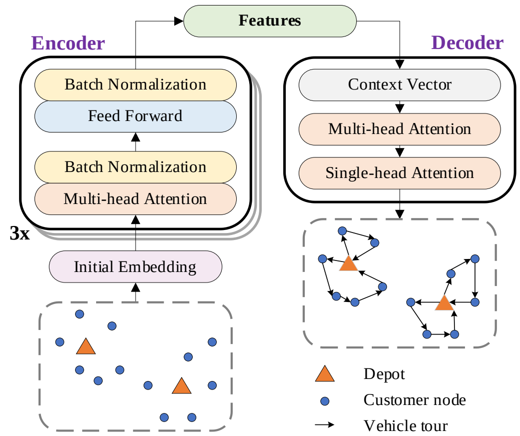


Figure 1: Transformer architecture for VRP

During the construction of the solution at each step, the decoder predicts the probability (really a stand-in for expected reward) of selecting each node and delivery truck. Using a decoding strategy in the decoder, a node will be selected to be visited by a specific delivery truck and it will be added to the solution. There are different decoding strategies including greedy, sampling and beam search in the literature. However, for solving VRP, sampling decoding strategy based on multinomial distribution has been mainly used, as it can address the exploration-exploitation trade-off during training. We should note that the performance of different decoding strategies is different for different problems based on the problem characteristics. Therefore, in our research, we propose an ensemble based decoding strategy which solves the problem using all decoding strategies simultaneously, and finally, the model reports the tours with minimum cost (total tour length). By combining different models, the resulting ensemble is typically more accurate and reliable than any single model. In our research, we train the model using sampling decoding strategy, and apply the ensemble decoding strategy during inference. Training the model using the beam search decoding strategy leads to high computational complexity which considerably increases training time. The main goal of this research is to improve the performance of the original transformer architecture to solve MDVRP without the need to train the model multiple times under different decoding strategies. The decoder has an environment which is updated at each decoding step. Based on the selected node and vehicle, the current location of the delivery trucks, the sequence of visited nodes, masked depots and customer nodes, and traversed customers are updated at each step in the environment. Here, we define all decoding strategies that we use in our proposed ensemble method applied to improve the performance of the model outputs during the inference time:

- **Greedy:** Choosing the single index with the highest predicted probability at each step is called greedy decoding. For a 'perfectly' trained neural network (one trained on sufficient examples from all areas of the problem space), this should yield close to optimal results as probabilities are based on expected reward. As the network is not perfectly trained, the chosen point might not be globally optimum [13]. We have used the greedy decode type in rollout greedy baseline step (during training time). This approach is identical to pure exploitation policy.

- **Sampling:** In this sampling strategy, a single index is chosen using a multinomial distribution based on the action probabilities. Sampling using multinomial distribution is an exploration policy.

- **Beam search:** In beam search, instead of choosing the best index with the highest predicted probability at each step in generating the sequence, we keep k possible 'most probable' indices at each step. The value of k is called beam width, metaphorically referring to a flashlight beam that can be parameterized to be wider or narrower. Combining different decoding strategies including greedy and sampling is a valid approach. Beam search decoding strategy increases the search space. In this research, we propose two types of beam search strategies: (1) Beam search-ss using sampling (with multinomial distribution) for building the branches of tree, and sampling (with multinomial distribution) for choosing the branches based on beam width and scores, and (2) Beam search-sg using sampling (with multinomial distribution) for building the branches, and greedy (top *k* nodes with highest probabilities, with *k* being the beam width) for choosing the branches based on beam width and scores. We also test the beam search case where the branches are built and chosen based on greedy approach. However, based on our experiments, their results were identical to greedy approach. Therefore, we do not report their results in the paper.

Algorithm

```
Input: batch size (B), no. customers (n), no. vehicles (k), depots and customers' locations (x,y), customers' demands (CD)
Initialize trained parameter θ
for decode_type = 'Greedy', 'Sampling', 'Beam search_ss','Beam search_sg', 'Beam search_gg'
    for step = 1, 2, …, B do
        for encoder_layer = 1, 2, 3
            Multi head attention in encoder
            Feedforward
        for decoding_step=1, 2, …all nodes are traversed
            Multi head attention in decoder
            Single head attention in decoder
            Apply decoding strategy on log-probabilities and choose next node to add to a vehicle sequence.
        Generate output routes and calculate cost (total tour length).
Compare final outputs for all decoding strategies and report the best result with minimum total tour length.
```

5. Results

We trained our model using the Louisiana State University high-performance computing (HPC) Deep Bayou cluster for 20, 50 and 100 customers using sampling decoding strategy. Experiments for the testing phase were performed on an Intel® Core™ i7-3770 CPU @ 3.40GHz 3.40 GHz processor and 8 GB RAM. We used the Python PyTorch package for our implementation. In our formulation, the node tensor contains the coordinate locations of depots first and then customers $i \in \{0, \dots, n\}$. Demand values for depots are set to zero, and for the customers the demand should be less than the capacity of each vehicle ($0 < cd_i \leq D$). We follow [6] for instance generation in training process; therefore, we use $Uniform(0,1)$ to generate customers and depots' coordinate locations. The demand values are discrete and sampled uniformly from $\{1, \dots, 9\}$, and the capacity of each vehicle depending on the problem size are as follows $D^{20} = 30, D^{50} = 40$ and $D^{100} = 50$. We consider the normalized capacity of $\widehat{D} = 1$ and normalized demands of $0 < \widehat{cd_i} \leq 1$ where $\widehat{cd_i} = \frac{cd_i}{D}$. The distance between the nodes are calculated based on the Euclidean distance.

In Table 1, we compare the results of transformer architecture for each decoding strategy to the extension of a well-known heuristic approach known as Lin-Kernighan heuristic (LKH3). This heuristic approach found many optimal and best-known solutions (BKS) for many benchmark instances. We use Augerat *et al.* [14] benchmark instances (set A), available in the CVRP library, to test the performance of our model. Since we use normalized depots and customers coordinate locations and demands during training, we normalized the values in Augerat *et al.* benchmark instances as well. In the literature, sampling decoding strategy is mainly used for solving MDVRP using transformer architecture. The ensemble decoding strategy finds out the results of each decoding strategy, and reports the best solution with minimum tour length. Based on Table 1, by using ensembled decoding strategy, the average total tour length for set A is decreased by 1.17% compared to sampling decoding strategy. Using ensemble decoding strategy, we reduced the gap between the results of transformer architecture to the best-known solutions. The minimum tour length for each sample is indicated by the numbers highlighted in bold within Table 1. In Figure *2*, we present a comparison between the ensemble and sampling decoding strategy costs by normalizing the costs relative to the BKS. As illustrated in the figure, the ensemble decoding strategy outperforms the commonly used sampling strategy in 67% (8 out of 12) of the cases by delivering better results. On the other hand, the sampling strategy proves to be the best in 33% (4 out of 12) of the cases with the minimum tour length in the ensemble method. It is noteworthy to mention that using the ensemble-based decoding strategy does not incur an increase in computation time compared to the sampling strategy, with an average computation time of 5-10 seconds for problem sizes ranging from 20 to 100 customers.

Table 1: Solution quality performance comparison for n nodes and k vehicles (total tour lengths based on normalized distances)

Sample	BKS	LKH-3	Greedy	Sampling	Beam search-ss	Beam search-sg	Ensemble
A-n32-k5	7.84	7.84	8.479	8.479	**8.155**	8.176	**8.155**
A-n33-k5	6.61	6.61	**7.196**	**7.196**	**7.196**	**7.196**	**7.196**
A-n33-k6	7.42	7.42	7.625	7.571	**7.487**	**7.487**	**7.487**
A-n34-k5	7.78	7.78	8.382	8.382	8.382	**8.205**	**8.205**
A-n36-k5	7.99	7.99	8.508	8.549	8.508	**8.490**	**8.490**
A-n37-k5	6.69	6.69	**7.197**	7.236	7.236	**7.197**	**7.197**
A-n39-k5	8.22	8.22	10.492	**8.745**	9.076	9.275	**8.745**
A-n45-k7	11.46	11.46	12.290	**11.830**	12.128	12.192	**11.830**
A-n46-k7	9.14	9.14	9.960	9.758	9.620	**9.618**	**9.618**
A-n48-k7	10.73	10.73	14.324	**11.723**	11.930	11.753	**11.723**
A-n69-k9	11.59	11.59	**12.547**	12.744	12.886	13.064	**12.547**
A-n80-k10	17.63	17.63	19.388	19.509	19.604	**19.101**	**19.101**
Average	9.425	9.425	10.532	10.144	10.184	10.146	**10.025**

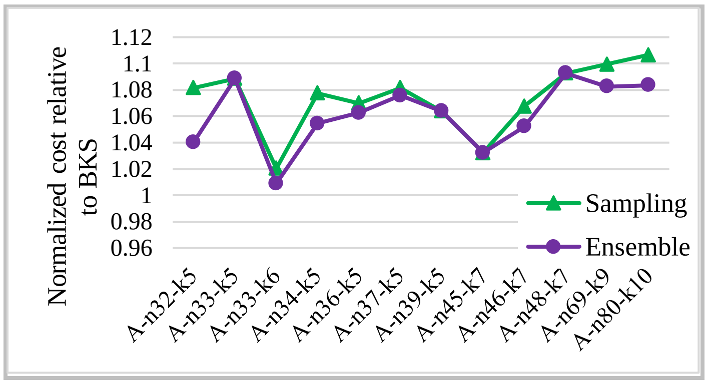
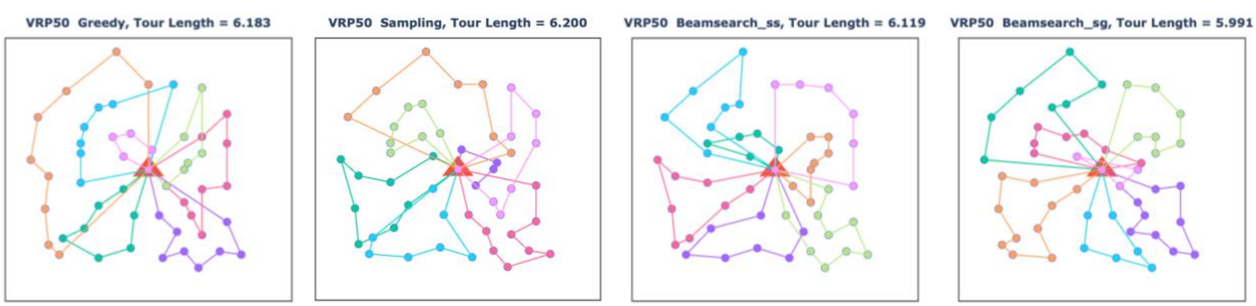
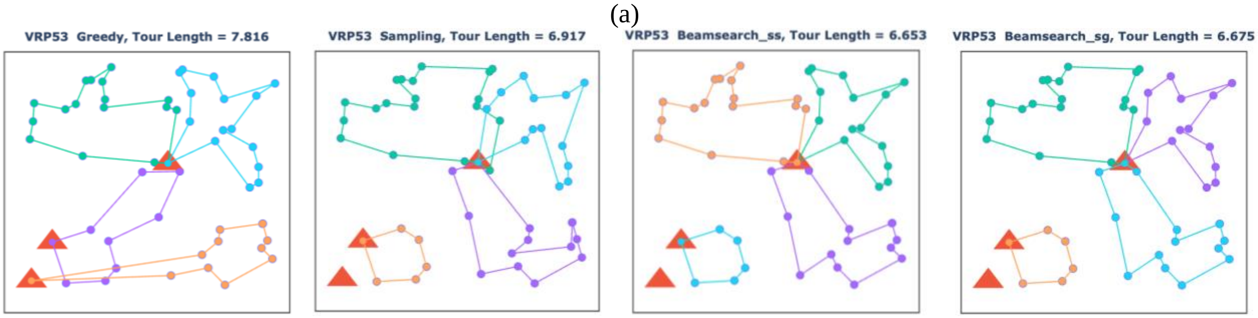


Figure 2: Ensemble vs Sampling decoding strategy



(a)



(b)

Figure 3. Graphical representation of different decoding strategies: (a) Single depot, and (b) multi-depot

In Figure *3* (a), we used a problem instance with one depot, 49 customers and 7 delivery vehicles from set P available in CVRP library [14]. Using ensemble decoding strategy, we find out that in this problem beam search-sampling and greedy (Beam search-sg) leads to best results (minimum total tour length). In Figure *3* (b) we solved a problem instance with 3 depots, 50 customers and 3 vehicles per depot. In this problem, beam search-sampling and sampling (Beam search-ss) decoding strategy finds the best tours. The ensemble decoding strategy can find the best decoding strategy during inference time based on problem characteristics.

6. Conclusions

In this research, we proposed a transformer architecture for solving multi-depot vehicle routing problem. This learning-based approach is beneficial in solving MDVRP within seconds. We improved the performance of the model during inference time by applying an ensemble-based decoding strategy, which combines the results of all decoding strategies simultaneously and reports the best solution with the lowest total tour length. We considered two types of beam search decoding strategies in our ensemble method. The proposed approach is beneficial in further improving the results of transformer architecture for solving MDVRP in 2 out of every 3 cases without the need to spend computational time and resources for training the model for different decoding strategies.

References

[1] A. N. Letchford and J.-J. Salazar-González, "The capacitated vehicle routing problem: Stronger bounds in pseudo-polynomial time," *European Journal of Operational Research,* vol. 272, no. 1, pp. 24-31, 2019.

[2] G. Laporte, "Fifty years of vehicle routing," *Transportation science,* vol. 43, no. 4, pp. 408-416, 2009.

[3] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems,* vol. 30, 2017.

[4] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," *Advances in neural information processing systems,* vol. 28, 2015.

[5] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," *arXiv preprint arXiv:1611.09940,* 2016.

[6] M. Nazari, A. Oroojlooy, L. Snyder, and M. Takác, "Reinforcement learning for solving the vehicle routing problem," *Advances in neural information processing systems,* vol. 31, 2018.

[7] Y. Ma *et al.*, "Learning to iteratively solve routing problems with dual-aspect collaborative transformer," *Advances in Neural Information Processing Systems,* vol. 34, pp. 11096-11107, 2021.

[8] A. Vaswani *et al.*, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998-6008.

[9] W. Kool, H. Van Hoof, and M. Welling, "Attention, learn to solve routing problems!," *7th International Conference on Learning Representations,* 2019.

[10] L. Xin, W. Song, Z. Cao, and J. Zhang, "Multi-decoder attention model with embedding glimpse for solving vehicle routing problems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, no. 13, pp. 12042-12049.

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015: PMLR, pp. 448-456.

[12] S. Santurkar, D. Tsipras, A. Ilyas, and A. Mądry, "How does batch normalization help optimization?," in *Proceedings of the 32nd international conference on neural information processing systems*, 2018, pp. 2488-2498.

[13] R. Klabunde, "Daniel jurafsky/james h. martin, speech and language processing," *Zeitschrift für Sprachwissenschaft,* vol. 21, no. 1, pp. 134-135, 2002.

[14] P. Augerat, J.-M. Belenguer, E. Benavent, A. Corbéran, and D. Naddef, "Separating capacity constraints in the CVRP using tabu search," *European Journal of Operational Research,* vol. 106, no. 2-3, pp. 546-557, 1998.