**Paper outline**

**Working Title:** Using Reinforcement Learning to Solve Capacitated Vehicle Routing Problem

**Authors:** Amandeep Rathee, Ronald Tinashe Nhondova, Tzu-Chun Hsieh

**Aim:** Apply new reinforcement learning (RL) algorithms to the capacitated vehicle routing problem (CVRP) that achieves better performance than the state-of-the-art RL techniques that have been applied on the CVRP. A better performance refers to finding a route whose distance is less than the distance achieved by other RL algorithms, and/or using less computational resources to arrive at the same solution in comparison to other RL algorithms.

**Objectives:** Solve the capacitated vehicle routing problem using various RL models, and try to outperform current stats-of-the-art RL methods that were applied on CVRP.

**Audience:** Researchers with experience in training reinforcement learning models.

**Unmet needs:** A more effective reinforcement learning model either takes less training time, requires less computational resources, or provides a better solution to solve CVRP.

**Style:** Academic paper

**Abstract**

Routing problems are optimization problems that are part of NP-hard problems. The best result is to find approximate solutions rather than exact optimal solutions because of the huge combinatorial space. There are numerous heuristics that could be applied to solve a given routing problem, but the solutions are not generalizable to a different configuration of the same problem. Reinforcement learning (RL) has shown promising results in coming up with approximate solutions to routing problems that are also generalizable. In this paper, we explore a suite of RL techniques and try to do better at solving a specific variant of routing problems known as the capacitated vehicle routing problem.

**Introduction**

Routing problems are a suite of combinatorial optimization problems that are NP-hard in nature, i.e. the amount of time it takes to arrive at a solution increases exponentially with the input graph. The goal of routing problems is to find an optimal route that covers the required locations in a graph. The most famous routing problem is the Traveling Salesperson problem (TSP) where the task is to find an optimal route starting from the specified source location in the graph, traverse through all the required locations specified in the problem exactly once, and come back to the source. A more general case of the routing problem is the Vehicle Routing Problem (VRP) which is the same as TSP except that VRP has multiple agents (referred to as vehicles) that visit all the required locations (exactly once), and end their journey at the source. the capacitated vehicle routing problem (CVRP), which is similar to VRP with an additional constraint that each node also needs a certain number of items to be delivered rather than just visiting it. Additionally, each vehicle has a capacity constraint. While one could optimize the solution for various metrics, we are only interested in the minimum trip distance because that is a commonly desired goal and is easy to generalize to other metrics. There are many potential applications of this problem. One of the most common use cases is to optimize the route for a fleet for an e-commerce company that is constantly delivering products to its customers. Because of this popular use case, the locations that need to be visited in CVRP are also referred to as customers.

There are various ways to solve CVRP. A naive way is to do a full combinatorial search over the search space and choose the path that minimizes the trip distance. However, the problem grows exponentially as the number of locations and vehicles increase and quickly becomes infeasible to solve using this naive approach. There are other heuristics that come up with an approximate solution. But a solution for one instance of CVRP cannot be used for a different instance of CVRP. One has to start again and apply the heuristic to get an approximate solution for each new instance. In the real world, solving each new instance becomes painstakingly hard and slow. With the recent advances in computing power, computer scientists have tried to apply RL methods to solve CVRP which have shown promising results (discussed later). An RL model (once trained) could be used to solve new instances of CVRP almost instantly. This is a huge advantage over conventional methods as there is no way to reuse a particular solution.

There are many more RL methods that are still unexplored. The goal of our project is to discuss some of these unexplored RL methods and see if they outperform the one that have already been tried to solve CVRP.

**Data**

Our dataset for CVRP was a fully connected synthetic graph with 102 vertices and XXX edges. A simple example is shown in Figure 1. One of the vertices would represent the source node (also known as the depot), other vertices would represent the customers including the demand information. Moreover, we will have a distance matrix which represents the Euclidean distance between every two vertices. Lastly, we will set the number of vehicles and the capacity of the vehicles for the problem.

Figure 1: A sample of simulated graph with 5 vertices and edges. The numbers in the matrix represent Euclidean distances between all pairs of vertices.

We generate the vertices in a two-dimensional Euclidean space using a uniform distribution. In other words, the probability that a particular point will be generated as a vertex is the same for all points. After creating the vertices, we compute the adjacency matrix that contains the distance between all pairs of nodes. Note that we do not generate the edges explicitly since we assume the graph is fully-connected (i.e. there exists a direct path between every pair of vertices). We could easily put a restriction that any two given pairs are not directly connected by replacing the Euclidean distance corresponding to the appropriate row and column vertex to infinity. We plan to do it at later stages of our research.

**Previous work**

In recent years, there have been so many studies using various kinds of methods to solve CVRP problems. Salimans et al., 2017 used highly parallelizable Evolution Strategies (ES) to achieve competitive performance to RL algorithms with less wallclock time. An attention model trained with REINFORCE applied by Kool et al., 2019 is suitable for multiple routing problems and as effective as problem-specific approaches. Lu et al., 2020 introduced a learning-based algorithm for solving CVRP that iteratively improves or perturbs the initialized feasible solution to explore better solutions. However, these previous models still require much time or computational resources to train. Moreover, there are some new methods such as Trust Region Policy Optimization (TRPO) and Proximal Policy Optimization (PPO) that haven’t been applied to CVRP. Therefore, our goal is to implement these new methods to the training process of these models to construct new algorithms that would either use less resources to achieve the same performance or provide a better solution for CVRP.

**Methodology**

We started by replicating research work from past papers that apply reinforcement learning to solve routing problems. These replicated models would serve as baseline models for the rest of our research.

* Reinforcement learning model that includes an attention-based neural network.
* Reinforcement learning model that learns to improve (L2I) current solution to a routing problem by making atomic changes to the graph such as removing an edge and moving an edge from one node to some other node.
* Attention-Based reinforcement learning model with Evolution strategy (ES) as training method.
* L2I reinforcement learning model with Evolution strategy (ES) as training method.

We started by replicating the attention-based and L2I models with classic deep RL training methods as was done in the original papers. Additionally, we also trained both the models with ES to compare the result with the classic version of the attention-based and L2I models. Then, we tried combining both the attention-based and L2I methods. The reason for applying ES as a training method and comparing the results with original models is ES can be parallelized and using multiple CPUs to reduce the training time. If a model with ES can perform similarly to the models using classic training methods, then we can use ES to replace the original methods and further parallelize the work to reduce training time.

The aim of all the reinforcement learning models was to minimize the total distance covered by the vehicles in the capacitated vehicle routing problem. The next section talks about the experiment and results that followed our replication experiments.

**Experiments**

**Experiment configuration**

The configuration that was used for our experiment to replicate previous work described earlier is as follows. Firstly, the problem specification had a 100 nodes that represent the number of customers. The demand of each of the nodes is uniformly distributed from 1 to 9 (inclusively). The location of each of the nodes is sampled from a unit square space (Figure 2).. In addition, the vehicle capacity is constrained to 40, with this being on the same measure or scale as the demand of each node. The total travelling cost, which is the key metric being optimized, is calculated as the Euclidean distance between nodes. In terms of the hardware, Google Colab in standard CPU environment was used for ES models, with NVIDIA P100 GPU used for the standard models without ES.

Figure 2: A sample graph shows the solution to CVRP with 11 vehicles.

**Experiment results**

Figure 3 compares the total trip distance achieved (during model training) by the routes created by the attention-based model, L2I model and a combination of both models. The individual models were trained for 10 hours whereas the combined model was trained for only 5 hours to account for the combination. It is clear that the L2I model achieves the optimal solution quickest, followed by the combination of L2I and attention-based model followed by the attention-based model. This result is consistent with the results of the original work.

Figure 3: Training curve of Attention and L2I models and a combination of the two.

Figure 4 compares the attention-based model and L2I model ran with and without ES as the training method. All the models were trained for 10 hours. We observe that L2I with and without ES both converge to the same trip distance and give the optimal solution as compared to the attention-based model. On the other hand, the attention-based model does perform worse if it is run with ES as compared to running it without ES.

Figure 4: Training curve of Attention and L2I models with and without evolution strategy (ES).

**Conclusion/Discussion**

(Place holder)

**References**

* Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, Hya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning, 2017.
* Wouter Kool, Herke van Hoof, Max Welling. Attention, learn to solve routing problems!, 2019.
* Hao Lu, Xingwen Zhang \* & Shuang Yang. A learning-based iterative method for solving vehicle routing problems, 2020.