Deep Reinforcement Learning and Heuristic Methods Integrated in Capacity-Constrained Vehicle Routing Problem Use of

A Hybrid Method by Integrating Deep Reinforcement Learning and Heuristics Approach for Capacitated Vehicle Routing Problem

Yasin Furkan Aktaÿ1,2 , Ahmet Murat Özbayoÿlu ¤ 1
1Department of Computer Engineering, TOBB ETÜ, Ankara, Turkey {yaktas,mozbayoglu}@etu.edu.tr
2ASELSAN Research Center, ASELSAN, Ankara, Turkey {furkanaktas}@aselsan .com.tr

Abstract—With the widespread use of online platforms, problems such as manned/ unmanned food delivery, cargo delivery, raw material delivery increase the importance of logistics day by day. The vehicle routing problem, which is one of the most important problems in the field of logistics, is a combinatorial problem and as the problem space grows, it takes a long time to find a solution with human effort, and in most cases it is not possible to find a solution. For this reason, it is important that the solution of this problem is autonomous. Although it is possible to solve the problem with classical heuristic optimization methods, it takes a long time from minutes to hours depending on the problem size and sometimes cannot give a good enough solution. Deep reinforcement learning models with attention mechanisms hold great potential in this regard. However, in cases where the problem space of deep reinforcement learning methods is large, in case of not getting enough information, the methods can sometimes diverge from the optimal solution, even if the methods give good results. In this study, better results are obtained in an acceptable time by using the attention mechanism deep reinforcement learning models and heuristic methods in a hybrid way.

Keywords—Deep reinforcement learning, attention model advanced, heuristics, optimization, vehicle routing problem

Abstract—With the spread of online platforms, problems such as manned/unmanned food delivery, cargo delivery, raw material delivery, are increasing the importance of logistics day by day.
Vehicle routing problem, which is one of the most important problems in the field of logistics, is a combinatorial problem and as the problem space grows, it takes a long time to find a solution with human effort and in most cases it is not even possible .

Thus, it becomes essential for the solution of this problem to be autonomous. Although it is possible to solve the problem with classical heuristic optimization methods, it takes a long time and sometimes does not give a good enough solution. Deep reinforcement learning models with attention mechanisms have great potential in this regard. However, in case of insufficient training in large problem space, it is possible to get away from the optimal solution. In this study, better results are taken in an acceptable time by using the deep reinforcement learning models with attention-model and heuristic methods in a hybrid way.

Keywords—Deep reinforcement learning, attention, heuristics, optimization, CVRP

I. INTRODUCTION

Today, with the spread of the use of online shopping and supply platforms, studies on the vehicle routing problem are increasing rapidly in direct proportion to the importance of supply chain and logistics issues. Although CVRP (Capacity Constrained Vehicle Routing problem) is an NP-hard problem, it is almost impossible to solve by trying all possibilities. Finding the optimal solution with mathematical exact solvers such as Gurobi [1] and CPLEX [2] takes as long as 15 minutes in a small problem space such as 20 customers, as stated in [3], and as the problem space grows, problems with 50,100 customers It becomes almost impossible to find the optimal solution in dimensions. In order to avoid the scalability problem of VRP, methods that can approach the optimal solution have been proposed.

While the solutions found by the studies on the Simpler version of VRP and the NP-Hard Problem Traveling Salesman Problem (TSP) can find or converge to the optimal in most cases, they approach the optimal solution due to the greater problem space and constraints in VRP. Studies on the work still continue.

In the past studies, methods such as genetic algorithm, ant colonies algorithm, which are classical heuristic methods, have been applied. But these methods take too long to give an acceptable and good result to the problem. For example, in the study conducted in [4], optimization results were found for 50,100,150 customer problems for 2,10,25 minutes, respectively. Lin, who was subsequently asserted with the TSP

978-1-6654-8894-5/22/$31.00 ©2022 IEEE

It has been observed that better results are obtained with heuristic methods specific to routing problems, such as the Kernighan heuristic algorithm, but the computation time is longer in the order of hours [3].

Since it takes too long to solve the problem with heuristic and classical optimization methods, artificial learning methods have become widespread in recent years to solve the scalability problem of CVRP. Systems that learn well enough with supervised learning can quickly find the solution to the problem. However, it is necessary to produce the necessary data for training and it is not a useful method to produce a sufficient amount of near-optimal solutions. While finding the route solution of each vehicle separately by clustering the problem in the multi-vehicle traveling salesman problem, this method is not suitable for CVRP. Therefore, recently, deep reinforcement learning models with attention mechanisms have become widespread on CVRP, and they have converged to the methods that produce the best known solution in these studies and have shortened the analysis time considerably.

In this study, a two-layer solution method has been created by integrating deep reinforcement learning methods with attention mechanisms and heuristic methods, and it is aimed to obtain better results with acceptable calculation time.

The remainder of the paper is organized as follows. In the 2nd chapter, the methods examined in the literature are summarized, in the 3rd chapter the problem definition is made, in the 4th chapter the setup and application of the experiments are explained, and in the 5th chapter the analysis and evaluation of the test results are made. Finally, by mentioning the possible future studies related to the work done in the 6th chapter, the paper is concluded with the 7th chapter.

II. RELATED STUDIES

A. Heuristic and Classical Optimization Methods

For many years, heuristic methods have been studied on CVRP and routing problems with its variants.

Local search algorithms[5], genetic algorithm [6], ant colony algorithms [4], which are generally applied in many optimization problems and observed to give good results, have also been applied on these problems. Although heuristic methods such as swap [7], 2-opt [8] give good results in TSP problem, they have not been applied on CVRP. The method named LKH3, which was created and extended by applying it to the TSP problem, [9] The method called Lin Kernighan Heuristics, which is basically an improved version of the 2-opt algorithm, was able to find the best known solutions in TSP problems and even found better ones than some. has been observed. The LKH method has been extended and adapted to different vehicle routing problems. In the CVRP problem, as in TSP, it can find the best solutions, and therefore, it is seen that the performance criterion is made according to this method in many studies such as [10], [3]. In the study in reference [11], separate routing problem solving methods for each vehicle were tried by clustering the problem into sub-problems with methods such as k-means and factor analysis in order to reduce the problem space.

Tabu search with low-level heuristics in vehicle routing problems involving a larger problem space [12] Pre-heuristic methods such as method, guided local search [13] methods are combined and solutions are produced. Since the use of pre-heuristics and classical optimization methods is very wide, commonly used software tools have also been developed in this regard. Concorde [14] tool is a famous software library that can give precise results on a large scale for TSP, and there are optimal solutions for combinatoric problems that can be modeled mathematically with Gurobi [1] and CPLEX [2], but no optimal solution can be found as the problem space grows. OR-tools [15] is a common library developed by Google and used for vehicle routing. However, it is a customizable library that includes pre-heuristics and low-level heuristics, and is included in comparisons in many CVRP studies [3, 16].

B. Learning Based Methods

After the studies with heuristic methods, artificial learning methods have been frequently encountered in recent years in order to make the problem more scalable. In the artificial learning approach, it is thought that by feeding and training the model with a sufficient amount of CVRP problems and a dataset containing the appropriate solution to the problem, it can give quality results very quickly in the testing phase of the problem [17]. In the work in reference [18], a learning-based method is used and more successful Marker Network Architecture (observed to be Pointer Net- work) is presented on combinatorial optimization problems. The architecture proposed in this study has an attention mechanism and a regenerative neural network (RNN) based encoder-decoder structure. Although it seems possible to use supervised learning methods in combinatorial problems, since a large amount of data and time is needed [3], the RNN structure in the attention mechanism of the marker architecture is similar to that in the trans former architectures. with attention). In addition, the reinforcement learning method is used to update the model parameters unsupervised by interacting with the surrounding environment. In many heuristics. In his study [19], as a different variant of CVRP, the problem of vehicles with deep reinforcement learning method and was inspired by the architecture mentioned in [3]. Since it plans the route of heterogeneous vehicles, it has been applied by combining two separate decoders that perform vehicle selection and route selection instead of a decoder, in a single action.

III. PROBLEM DEFINITION

As seen in Table I, CVRP basically consists of a warehouse, N number of customers and a variable amount of demand from each customer. The aim of the problem is to find the best solution to minimize the total distance traveled with the minimum number of vehicles with the maximum demand limit (c), ie it minimizes the objective function in Table I. In another analogy, we have a fleet of vehicles consisting of only one vehicle type and the maximum cargo volume that the vehicles can take is limited to c.

When the amount of cargo to be left to each customer is matched with the amount of demand, it is aimed to distribute cargo to customers with the minimum number of roads and vehicles to be taken.

Variable	Explanation
N	Node Number
c	Maximum Demand Capacity Per Vehicle
m = (mx, my)	Warehouse Coordinate
X = {x1, x2, .., xi, .., xN } xi = (xi, yi)	Node List
	Node Coordinates
P = {p1, p2, .., pi, .., pN } pi = (xi, yi)	Request List
	Demand Quantity at Node
V = {v1, v2, .., vi, .., vk} vi = (xi, ..,	Vehicle List
xj , .., xk)	Vehicle Route
dvi	Total euclidean distance of the vehicle route
min dvi viÿV	objective function
pj ÿ c xj ÿVi	Capacity Constraint for Each Vehicle

Table I: Variables of Capacity-Limited Vehicle Routing Problem

In the basic CVRP problem, vehicles start from the warehouse and it is assumed that there are enough vehicles in the fleet. ÿ

IV. METHOD APPLIED

As seen in Figure 1, the proposed method consists of 4 main modules. The first module, the data generation module, is responsible for generating the data of the problem and putting it into the appropriate form that the initial solution method can run on the second. The initial solution method, which is the module, is responsible for obtaining a good initial solution so that the final solution method can reach a solution. The third module, the final solution method, is responsible for taking the result produced by the initial solution method as input and improving it. The last module, the result evaluation module, compares the output obtained by the result solution method before the calculation time limit defined for it, and the initial solution, and presents the better performing solution as the final solution.

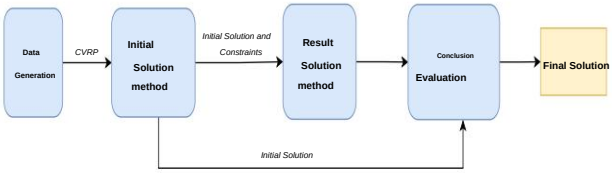

Figure 1: Flowchart of the Proposed Method

Reference In the studies in [16],[17],[20] it has been shown that using deep reinforcement learning methods, it can produce fast results close to the best solutions for CVRP. However, although it is not very common in these studies, deep reinforcement learning models can sometimes be encountered with situations where they learned the wrong behaviors or could not adequately discover them [21]. On the other hand, when starting with a good initial solution in heuristic optimization methods, the problem gives a near-optimal solution much faster.

Although LKH3 and OR-tools are the most widely used heuristic optimization methods in CVRP, LKH3 can often give the best results despite its long computation time. On the other hand, OR-tools gives results in a shorter time compared to LKH3 [3], and it is below the performance of deep reinforcement learning and LKH3.

Heuristics and DRL methods are hybrid

When used, it is ensured that the heuristic methods can optimize the problem faster and obtain a more reliable solution compared to the cases where the heuristic methods are used plainly. Thus, deep reinforcement learning's fast solution generation and heuristic optimization algorithms' ability to improve the given solution iteratively complement each other.

V. EXPERIMENTS

While applying the method, LKH3 [9], OR-tools [15] and DRL (deep reinforcement learning) [3] methods, which have been observed to be the most used in the past studies, are based on. By hybridizing these baseline methods with their combinations, the computation time and result improvement metrics were evaluated. Comparisons were made with greedy, beam and sample decoder methods using the attention mechanism applied in the study in reference [3].

Algorithms are tested with 10, 20, 50, 100 customers.

Demand values of customers (demand) are {5-50} when the number of customers is 10, {3-33} when 20, {2-25} when 50, and {2 when 100 Randomly distributed values in the range of -20} were determined. Warehouse and customer locations are randomly assigned and the selected experiment. The maximum capacity of each vehicle is assigned as 100.

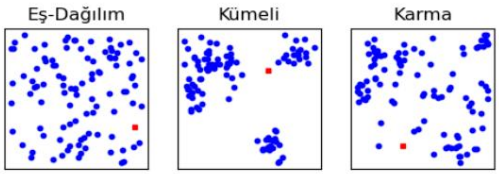

Figure 2: Example CVRP datasets

In order to test whether the method is dependent on customer locations, as exemplified in Figure 2, tests were carried out in the case of three separate distributions: uniformly distributed, clustered and mixed. Clustered customers by choosing 3-3-4-5 random cluster centers in the number of 10-20-50-100 customers within the clustered distribution area, with equal weights in the evenly dispersed locations are generated, in a mixed distribution, half of the customers are clustered and the other half are chosen with an the optimization distribution of OR-tools, acceptable time limit is defined as 10, 20, 50, and 10, 12, 15, 20 for the number of 100 customers, respectively. The experiments were with Intel i7-10700 16 core processor, 128 GB RAM and Nvidia Quadro RTX4000 8 GB graphics card.

While performing the experiments, the parameters of the LKH3 algorithm were selected as max_trial=50000, move_type=5_special,runs=1. It was observed that the Runs parameter changed the result minimally and was assigned as 1 to get the results faster. The DRL (AM/attention Model) models we used were taken as reference from the study [3]. Width=1280 parameters are used for DRL(AM)- Beam and DRL(AM)[3] sample. CVRP problem OR-tools basically integer and and 1000 for length calculations. multiplied by

Method	N=10						N=20					
			Clustered		Mixed				Cluster		Mixed	
	Equal Cost	Duration (s)	Cost	Time (s) 4667.4	Cost	Time (sec) 4957.2	Equal Cost	Duration (s)	Cost	Time (s) 6302.2	Cost	Time (s) 6882.4
LKH3	4392.1 3.531		2.88 4469.8 4[illegible]	[illegible]	3.86		5846.1 12.56 [illegible]	[illegible]	11.15 6274.9 [illegible]	[illegible]	11.20 6857.9 [illegible]	[illegible]
OR-tools	4234.0	10.01		10.01	4932.1	10.01	**5821.9** 5921.1[illegible]	[illegible]		12.01		12.01
DRL(AM)-Greedy	4345.3	0.0187		0.02	4906.6	0.014	5871.5 5828.0[illegible]	[illegible]		0.03		0.021
DRL(AM)-Beam	**4189.6**	0.0273		0.022	**4887.9**	0.023		0.04		0.04		0.04
DRL(AM)-Sampling	4276.0	0.0241	4575.2	0.018	4898.4	0.019		0.031	6264.4	0.03	6847.2	0.03
LKH3-OR-tools	4234.0	13,531	4469.8	12.88	4932.1	13.86		24.56	6271.7	23.15	6845.5	23.2
DRL(AM)-Greedy - OR-tools	4234.0	10.02	4469.8	10.02	4932.1	10.01		12.02	6260.2	12.02	6857..9	12.03
DRL(AM)-Beam - OR-tools	4206.0	10.03	4469.8	10.02	4932.1	10.02		12.04	6259.4	12.04	6858.5	12.04
DRL(AM)-Sample - OR-tools	4210.0	10.03	4457.4	10.02	4932.1	10.02		12.03	6259.4	12.03	6857.9	12.03

Method	N=50						N=100					
			Cluster Mixed Equal Cost Time (s) Cost Time(s) Cost Time (s)						Clustered		Mixed	
	Equal Cost	Duration (s)	8552.3 56.9 [illegible]	[illegible]	[illegible]	[illegible]	15.01 16927.8		Cost Time (sec) 14154.2		Cost Time (s) 15235.9	
LKH3	**10941.8** 66.04 15.01								159.8		148.16 16237.[illegible]	[illegible]
OR-tools	11479.1		8772.2					20.01	14724	20.01	0.1 15818.8 0.34 15695.1	
DRL(AM)-Greedy	11522.7 0.06		9271.7	0.05		0.05			15103.2	0.09	**15205.4** 1569[illegible]	[illegible]
DRL(AM)-Beam	11187.9 0.116		8771.4	0.13		0.12			14698.3	0.33		
DRL(AM)-Sampling	11213.1	0.118	8739.2	0.11	10248.0 0.10		16324.5	0.30	14611.6	0.3		0.31
LKH3+OR-tools	10949.1	81.01	**8533.0**	71.93	10007.8	67.76	**15876.0** 190.63		**13996.5**	179.8		168.2
DRL(AM)-Greedy +OR-tools	11147.0	15.06	8669.5	15.06	10118.1	15.05	16400.9 20.09 16121.3		14348.8	20.1		20.1
DRL(AM)-Beam + OR-tools	11059.0	15.118	8641.8	15.13	10047.5	15.12	20.33 16053.1 20.30		14334.6	20.33		20.34
DRL(AM)-Sample + OR-tools	11092.5	15.12	8643.5	15.11	10079.1	15.11			14245.7	20.30		20.32

Table II: Performance and Results Comparisons

converted. Finally, the cost values seen in the experiments were calculated based on this transformation.

VI. ANALYSIS AND ASSESSMENT

When we look at the results of the performed experiments listed in Table II, when we compare the generated datasets as iso-dispersed, clustered, mixed, the cost of the solutions obtained is Mixed>Cluster>E in the problem space N=10.20. It is seen that for the problem space N=50.100, it changes as Work Distributed > Mixed > Set as Uniform. For this reason, while it is advantageous for the problem to be uniformly distributed in small N=10.20, problems such as disadvantageous in larger-sized problems such as N=50.100. However, when we examine the cost variation of algorithms according to distributions, it is seen that they exhibit similar behavior with an increase and decrease wave that is close to each other. Thus, we can say that algorithms are robust to distribution. Especially in small problems such as N= 10, 20, deep reinforcement learning methods can give good enough results much faster than other heuristic optimization and hybrid solutions. On the other hand, as a disadvantage, however, learning methods produce solutions to CVRP problems quickly, [illegible] they are also be

In cases where the problem space is larger, we see that applying the hybrid method by giving an initial solution to OR-tools is beneficial by further reducing the cost value. This is because in optimization algorithms, the optimality gap increases with large problem sizes, and starting the solution from scratch with the initial solution, rather than from scratch, improves the performance much more. Although the use of hybrid with the LKH3 algorithm gives the best results [illegible] size of N=100,

Since the optimality gap for almost all problem sizes is very small, it could not provide a great improvement. In the results of N=50.100 solved with the hybrid approach, a significant improvement has been made to the DRL(AM) methods because the optimality gap is [illegible] N=50.100 for the LKH3 Although LKH3, it works much slower than other methods. From another point of view, with the DRL(AM)-Sample+OR-tools method, a solution very close to the solution produced by the LKH3 method can be reached in an acceptable time like 20 seconds rather than 148 seconds.

When we examine the DRL(AM)-Greedy, DRL(AM)-Beam and DRL(AM)-Sample methods, very good results are produced for small-sized problems. While DRL(AM)-Beam model gives better results for N=10.20, it gives very close results with DRL(AM)-Sample for N=50.

For N=100, DRL(AM)-Sample gives better results than other DRL(AM) ÿ methods. The reason for this is that when the problem size grows, the search tree boundary limits the solution when choosing the next output in the beam decoder structure. Since the maximum width of the search tree is fixed as 1280, the problem has grown, and the result performance per day has decreased a bit compared to the sample method.

If the width value is given a larger value, it has the disadvantage of slower operation.

In summary, the DRL(AM) method gives good results for small size (N=10.20) CVRP problems and although the hybrid method does not seem to benefit, the resulting improvement is obvious for larger size (N=50.100) CVRP problems. is seen in the figure.

VII. FUTURE WORKS

In this study, the capacity constrained vehicle routing problem is discussed. In future studies, problems with different vehicle capacities and heterogeneous speeds can be addressed. In addition, in the light of the results obtained from the experiments,

Intelligent mechanisms can be developed that select which method is more appropriate according to the procedure.

In the vehicle routing problem, it can be combined not only with the cargo pick-up from the warehouse to the customer, but also with the customer-to-customer/delivery problems.

VIII. CONCLUSION

The study was carried out on the capacity constrained vehicle routing problem (CVRP), which is a combinatorial and NP-hard problem. Within the scope of this study, first of all, a literature review summarized in II was conducted. Considering the methods examined in the literature, classical heuristic optimization methods such as ant colonies or genetic algorithms require a resolution time of 5-20 minutes to solve the problem. Although heuristic optimization methods specific to the route planning problem such as LKH3 produce very good solutions, the calculation time can take hours.

Although the deep reinforcement learning method with attention mechanism, which is widely studied in current publications, does not perform as well as LKH3, it can obtain a better solution than OR-tools faster than other methods in the literature. However, the training of deep reinforcement learning methods can take days and there is no guarantee of learning the problem adequately. Therefore, a two-layer hybrid solution method, which is considered to be safer and specified in IV, is proposed.

In order to test the proposed hybrid method and compare it with the methods in the literature, experimental setups have been created with problem sizes of N=10,20,50,100 and configurations where the problem set is uniformly distributed, clustered and mixed. Methods such as DRL, LKH3, OR-tools have been tested in experimental setups and presented in V. In the light of the results obtained, since the lean methods give good enough results in the small problem space (N=10,20), there is not much improvement with hybrid methods, but the results in the large CVRP space (N=50,100) improve in terms of time and results. It has been seen that it provides growth. ÿ

QUOTES

[1] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual. 2022. URL: https://www.gurobi.com.

[2] IBM ILOG Cplex. “V12. 1: User's Manual for CPLEX”. In: International Business Machines Corporation 46.53 (2009), p. 157.

[3] Wouter Kool, Herke van Hoof, and Max Welling. At tention, Learn to Solve Routing Problems! 2018.

[4] John E. Bell and Patrick R. McMullen. “Ant colony op timization techniques for the vehicle routing problem”. In: Advanced Engineering Informatics 18.1 (2004), pp. 41–48.

[5] Emile Aarts and Jan K. Lenstra. Local Search in Com binatorial Optimization. 1st. USA: John Wiley & Sons, Inc., 1997. ISBN: 0471948225.

[6] Barrie M. Baker and MA Ayechew. “A genetic algo rthm for the vehicle routing problem”. In: Computers & Operations Research 30.5 (2003), pp. 787–800.

[7] Cathy Wu et al. “Optimizing the diamond lane: A more tractable carpool problem and algorithms”. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). 2016, p. 1389–1396.

[8] GA Croes. “A Method for Solving Traveling Salesman Problems”. In: Operations Research 6 (1958), pp. 791–812.

[9] S. Lin and BW Kernighan. “An Effective Heuristic Algorithm for the Traveling-Salesman Problem”. In: 21.2 (Apr. 1973), pp. 498–516.

[10] Sirui Li, Zhongxia Yan, and Cathy Wu. Learning to Delegate for Large-scale Vehicle Routing. 2021.

[11] Basma Hamdan, Hamdi Bashir, and Ali Cheaitou. “A novel clustering method for breaking down the symmet ric multiple traveling salesman problem”. In: Journal of Industrial Engineering and Management 14.2 (2021).

[12] Gulay Barbarosoglu and Demet Ozgur. “A taboo search algorithm for the vehicle routing problem”. In: Computers & Operations Research 26.3 (1999), pp. 255–270.

[13] Philip Kilby, Patrick Prosser, and Paul Shaw. “Guided Local Search for the Vehicle Routing Problem with Time Windows”. In: Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. Ed. by Stefan Voß et al. Boston, MA: Springer US, 1999, pp. 473–486.

[14] Concorde-A TSP Solver. URL: https://www. math uwaterloo.ca/tsp/ concorde.html.

[15] Laurent Perron and Vincent Furnon. OR Tools. Version 7.2. Google. URL: https://developers.google.com/optimization/.

[16] Mohammadreza Nazari et al. Reinforcement Learning for Solving the Vehicle Routing Problem. 2018.

[17] Hanjun Dai et al. Learning Combinatorial Optimization Algorithms over Graphs. 2017.

[18] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer Networks. 2015.

[19] Jingwen Li et al. “Deep Reinforcement Learning for Solving the Heterogeneous Capacitated Vehicle Routing Problem”. In: IEEE Transactions on Cybernetics (2021), pp. 1–14.

[20] Irwan Bello et al. Neural Combinatorial Optimization with Reinforcement Learning. 2016.

[21] Alex Irpan. Deep Reinforcement Learning Doesn't Work Yet.