electronics

MDPI

Article

A Neural Multi-Objective Capacitated Vehicle Routing Optimization Algorithm Based on Preference Adjustment

Liting Wang [1], Chao Song [1], Yu Sun [2], Cuihua Lu [1] and Qinghua Chen [1,*]

[1] The Third Faculty, Naval Aviation University, Yantai 264001, China; litingwang_start@163.com (L.W.); schhh_1983@126.com (C.S.); llu1978@163.com (C.L.)
[2] Institute of Marine Science and Technology, Shandong University, Qingdao 266237, China; 202236973@mail.sdu.edu.cn
* Correspondence: 201813678@sdtbu.edu.cn; Tel.: +86-15066386779

Abstract: The vehicle routing problem (VRP) is a common problem in logistics and transportation with high application value. In the past, many methods have been proposed to solve the vehicle routing problem and achieved good results, but with the development of neural network technology, solving the VRP through neural combinatorial optimization has attracted more and more attention by researchers because of its short inference time and high parallelism. PMOCO is the most state-of-the-art multi-objective vehicle routing optimization algorithm. However, in PMOCO, preferences are often uniformly selected, which may lead to uneven Pareto sets and may reduce the quality of solutions. To solve this problem, we propose a multi-objective vehicle routing optimization algorithm based on preference adjustment, which is improved from PMOCO. We incorporate the weight adjustment method in PMOCO that is able to adapt to different approximate Pareto fronts and to find solutions with better quality. We treat the weight adjustment as a sequential decision process and train it through deep reinforcement learning. We find that our method could adaptively search for a better combination of preferences and have strong robustness. Our method is experimented on multi-objective vehicle routing problems and obtained good results (about 6% improvement compared with PMOCO with 20 preferences).

Keywords: vehicle routing problem; logistic and transportation; neural combination optimization; multi-objective optimization

Citation: Wang, L.; Song, C.; Sun, Y.; Lu, C.; Chen, Q. A Neural Multi-Objective Capacitated Vehicle Routing Optimization Algorithm Based on Preference Adjustment. *Electronics* **2023**, *12*, 4167. https://doi.org/10.3390/electronics12194167

Academic Editor: Rashid Mehmood

Received: 10 August 2023
Revised: 19 September 2023
Accepted: 20 September 2023
Published: 7 October 2023

1. Introduction

The vehicle routing problem is a common problem in logistics and transportation. With the rapid development of the e-commerce industry and intelligent transportation, the VRP has become increasingly popular with researchers, and its purpose is to design a series of routes to make vehicles move orderly under certain constraints [1]. The multi-objective vehicle routing problem requires us to optimize two contradictory objectives at the same time, and it has many application scenarios in practice. For example, in the task of transporting dangerous goods, we need to reduce transportation risks and to reduce transportation costs. In the task of green transportation, we need to reduce the distribution cost of goods while trying to avoid environmental problems.

In the real world, the basic VRP often fails to meet diversified requirements, so a large number of similar but more complex vehicle routing problems have been proposed to adapt to practical applications. In the vehicle routing problem with a time window, the delivery time of each customer is limited [2]. In the capacitated vehicle routing problem, there is a limit to the maximum capacity of the vehicle [3]. In the multi-depot vehicle routing problem, there are multiple depots in the distribution network, and the same goods can be picked up at multiple depots [4]. In the split delivery vehicle routing problem, the same customers' goods can be delivered by multiple vehicles [5].

The VRP is also a classical combinatorial optimization problem; since the problem was first proposed by Dantig and Ramser in 1959 [6], many different types of algorithms have been proposed to solve it. In general, these algorithms can be divided into three categories: exact algorithm, approximate algorithm, and heuristic algorithm. The exact algorithm can obtain the theoretical optimal solution, but the VRP is an NP-hard problem [7]. An NP problem refers to the fact that validation can be found in polynomial time of the problem. NP-hard problems are those in which all NP problems can be reduced in polynomial time. NP-hard problems are usually very complex, and it is difficult to find an exact solution in polynomial time. Therefore, it is usually not advisable to use exact solution methods on large-scale VRP problems; the time cost brought by the exact algorithm is often unacceptable with the increase in the problem size. Approximation algorithms hope to find an approximate solution in an acceptable time, but many combinatorial optimization problems do not have approximation guarantees. In the past, heuristic algorithms were more often studied to solve the VRP. A heuristic algorithm is an algorithm based on experience design; it could give a feasible solution in an acceptable time, but the optimality of the feasible solution is often not guaranteed. Furthermore, heuristic algorithms are often designed for specific problem, so they may lack flexibility. In recent years, with the development of deep learning and reinforcement learning technology [8–14], neural combinatorial optimization has become more and more popular among researchers for its short inference time, high parallelism, and strong robustness [15–18]. However, neural combinatorial optimization algorithms often require elaborate designs, esoteric domain knowledge, and long training times. In many related studies, the VRP is modeled as a sequential decision problem, which may lead to the problem of a long computation time when solving large-scale problems. Therefore, solving the VRP with neural combinatorial optimization methods both has potential and is challenging [19].

In our work, we mainly studied the multi-objective capacitated vehicle routing problem (MOCVRP) [20], where there are two contradictory objectives: one objective is the total tour length, and the other one is the tour length for the longest route. PMOCO [21] is the most state-of-the-art algorithm in solving MOCVRP. PMOCO takes Transformer [22] as the backbone structure of the model. It uses an encoder to encode the location information of customers and the depot and uses a decoder to output the next selected customer one by one, thus forming a route. In order to solve the multi-objective problem, PMOCO proposes a preference-conditioned model using hypernetwork [23] to generate decoder parameters, which can solve the multi-objective problem with any number of preferences. Although PMOCO has a good effect on the multi-objective vehicle routing problem, PMOCO does not take into account the importance of weight adjustment. Our method proposes an adaptive weight adjustment method on the basis of PMOCO to obtain a better weight combination. In addition, the weight adjustment strategy of most traditional methods adopts a heuristic method [24], which has a high time complexity. The combination of traditional weight adjustment strategies and deep learning algorithms may make deep learning algorithms lose their advantages of a short inference time. Therefore, we propose an end-to-end weight adjustment strategy, which is more suitable for multi-objective algorithms based on deep learning, filling the gaps in weight adjustment strategies and further improving the effect of PMOCO.

Our paper follows the following structure. In the Introduction section, we give a brief overview of the problem background and the proposed approach. In the Related Work section, we introduce some classical vehicle routing algorithms and neural multi-objective optimization algorithms. In the Formulation section, we introduce the problem formulation of MOCVRP and multi-objective problems, as well as several key concepts in multi-objective optimization. In the Scalarization Method section, we introduce three common Scalarization methods in solving the multi-objective problem. In the Methodology section, we describe the details of the algorithms and model structures we used. In the Experiment Setting section, we introduce the setting of our experiments, including the baselines, problems and settings, and inference and metrics. In the Results section,

we analyze the results of the experiment. In the Conclusion and Future Work section, we summarize the proposed methods and propose some work worth improving in the future.

Our contributions of this paper are threefold:

- We analyze the shortcomings of applying the traditional weight adjustment method directly to the neural combinatorial optimization algorithm.
- We propose an end-to-end weight adjustment method that is more suitable for solving the VRP.
- We demonstrate the effectiveness of the proposed method by performing experiments on standard MOCVRP instances.

2. Related Work

There have been many feasible algorithms for solving the VRP and multi-objective optimization problems. In this section, we will provide a review of traditional vehicle routing algorithms, heuristic VRP algorithms based on machine learning, and neural multi-objective optimization algorithms.

Traditional VRP Algorithms. There are many traditional methods to solve the VRP. The exact algorithms for solving the VRP mainly include the branch and bound method [25], the mixed integer linear programming method [26], and the dynamic programming method [27]. Exact algorithms can obtain exact solutions, but this method is often limited by the size and form of the problem, and the performance is not good when solving large-scale or complex problems. Renaud et al. [28] tried to combine the sweep algorithm and the mixed integer programming method to improve the speed and quality of the solution. Clark and Wright [29] sorted the routes according to the size of the savings value and discharged the corresponding two customer points into the paths according to certain rules. Shaw et al. [30] used the ruin and repair operators to improve the original solution and achieved better local optimality. Dorigo et al. [31] designed an ant colony optimization algorithm inspired by ant colonies and used it in the vehicle routing problem. Osman combined tabu search and simulated annealing (SA) to solve the vehicle routing problem [32]. There are also many other studies that combined different heuristic methods to adapt to more complicated problems [19].

Heuristic VRP Algorithms based on Machine Learning. Machine learning-assisted heuristic VRP algorithms could be divided into two categories: one is end-to-end methods, and the other is methods that assist traditional heuristics. Inspired by the problem of machine translation, Vinyals [33] designed a pointer network, with a long short-term memory network as an encoder and an attention mechanism network as a decoder, and trained with supervised learning. Scarselli et al. [34] combined a pointer network with graph neural networks to increase the ability to generalize to large-scale problems. Kool et al. [35] used Transformer as the backbone of his network. Meanwhile, in order to avoid expensive label acquisition, reinforcement learning was used to train the network. Xin et al. [36] proposed to update the learned node representations in the Transformer architecture in [35] at each step of the decoding process, so as to obtain more precise information and better solutions. Xin et al. [37] further proposed a multi-decoder architecture, which learns multiple decoding policies at the same time, which increases the chance of finding better solutions. Despite the potential of the end-to-end approach, the machine learning-assisted heuristic has better performance [19]. Wu et al. [38] learned a method to improve the original solution. A new search operator was designed by Chen [39] based on deep learning and achieved good results. Santana et al. [40] adopted a heatmap to guide search. Feng et al. [41] adopted transfer learning to optimize VRP.

Neural Multi-Objective Optimization Algorithms. Common methods for solving multi-objective optimization problems fall into three categories: decomposition-based, dominance-based, and indicate-based [42]. Neural multi-objective optimization algorithms usually follow the first one. The decomposition-based method uses an aggregation function to decompose a multi-objective problem into multiple single-objective problems corresponding to different preferences and optimizes them with a neural network, which is

similar to the idea of an MOEA\D [43]. DRL-MOA [44] adopts preference transfer training to avoid repeated training for similar subproblems. PMOCO uses hypernetworks to integrate preference information into neural networks to establish a preference-conditioned model. COSMOS [45] concatenated preferences and instances as input features of neural networks. MDRL [46] treats the single-objective problems corresponding to different preferences as subtasks and trains a meta model to optimize these subtasks simultaneously. PMTL [47] considers both the Pareto solution corresponding to the preference and the angle of the preference itself to make the approximate Pareto front more uniform. Most of these methods do not change the preference but split the preference evenly, and our method is able to adjust the weights to further improve the effect of the neural algorithm based on preference.

3. Formulation

In this section, we introduce the problem formulation of MOCVRP and multi-objective problems, as well as several key concepts in multi-objective optimization.

Definition 1 (Multi-Objective Capacitated Vehicle Routing Problem (MOCVRP))**.** *The VRP could be described as assuming that there are n customers and a depot in graph* $G = (V, E)$*, where* V *consists of customers node and depot node and* $E = \{(i, j) : i, j \in V, i < j\}$ *represents a edge set which contains connects to different nodes, such as* v_a *and* v_b $(a \in n, b \in n, a \neq b)$*. The solution of the VRP could be regarded as a trajectory* π*, and the objective of the VRP is to optimize the* π *under certain conditions. The vehicle needs to depart from the depot and to return back after completing the distribution task.*

MOCVRP is an extension of the VRP. The vehicle has a maximum capacity limit D, and each customer has a fixed amount of demand $d_i (i \in n)$ *to transport. For a distribution process of the same vehicle, the sum of demand* $\sum d_i$ *through customer nodes cannot be greater than D. A simple illustration is shown in Figure 1.*

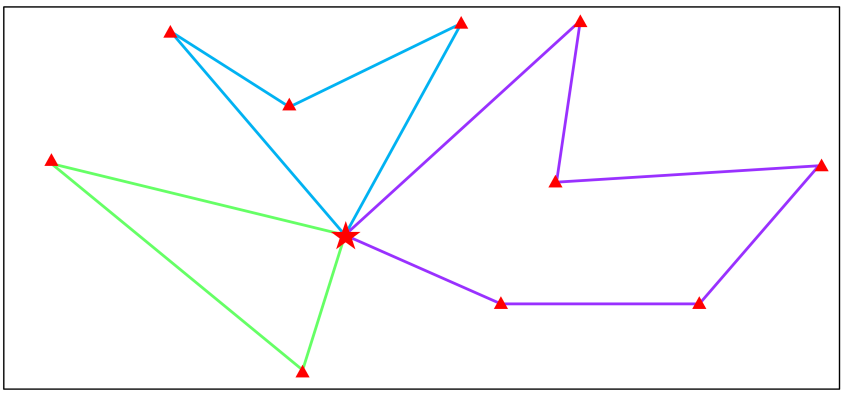

Figure 1. Example solution of CVRP. The five-pointed star represents the depot, and the triangle represents the customer's location. And, the blue, green, and purple lines represent different routes for different vehicles to deliver goods.

Definition 2 (Multi-Objective Problems (MOP))**.** *In general, a multi-objective problem can be defined as follow:*

$$\min_{x \in \mathcal{X}} F(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \tag{1}$$

where the $\mathcal{X}$ *is the decision space,* m *is the number of objectives,* $f_i(x)$ *is the ith objective* $(i = 1, 2, \ldots, m)$*, and* $F(x)$ *is an m-dimension objective vector of the multi-objective problem. In multi-objective optimization, the values of different objectives are often conflicting, which also means that it is difficult to optimize multiple objectives at the same time.*

Definition 3 (Pareto Dominance). *Let $x_1, x_2 \in \mathcal{X}$, if and only if $f_i(x_1) \leq f_i(x_2)$, $\forall i \in \{1, 2, \ldots, n\}$ and $f_i(x_1) < f_i(x_2), \exists i \in \{1, 2, \ldots, n\}$, we could said that x_2 is dominated by x_1 (i.e., $x_1 \prec x_2$).*

Definition 4 (Pareto Optimality). *If $x^* \in \mathcal{X}$ is Pareto optimality, then there is no $x \in \mathcal{X} (x \neq x^*)$ dominating x^*. And, we call $F(x^*)$ the Pareto optimal point.*

Definition 5 (Pareto Set). *In general, there exist multiple Pareto optimal solutions; we define the set of all Pareto optimal solutions as the Pareto set and the image of all Pareto optimal points as the Pareto front. But, in practice, the Pareto optimal solutions we can obtain is finite and it is difficult to obtain the theoretical optimal solution, so we often replace the Pareto set with an approximate Pareto set of finite Pareto optimal solutions and replace the Pareto front with an approximate Pareto front of finite Pareto optimal points.*

Definition 6 (Hypervolume (HV)). *To measure the effect of our method, we need to compare Pareto sets. However, Pareto sets are difficult to compare directly, so we choose the hypervolume (HV) [48] that is most commonly used in multi-objective optimization to measure the advantages and disadvantages of different methods. Since it is difficult for us to obtain the ground truth Pareto set truth, we use the approximate Pareto set instead.*

Hypervolume is the volume dominated by an approximate Pareto set V of reference points r^*. So, we could define HV(V) as follow:

$$S = \{r \in \mathbb{R}^m \mid \exists\, v \in V \text{ such that } v \prec r \prec r^*\} \tag{2}$$

where m is the number of objectives and HV (V) = VOL (S). A simple illustration is shown in Figure 2 for a more intuitive understanding of HV, where $V = \{v1, v2, v3, v4\}$ is dominated by r^* in our two-dimensional example.

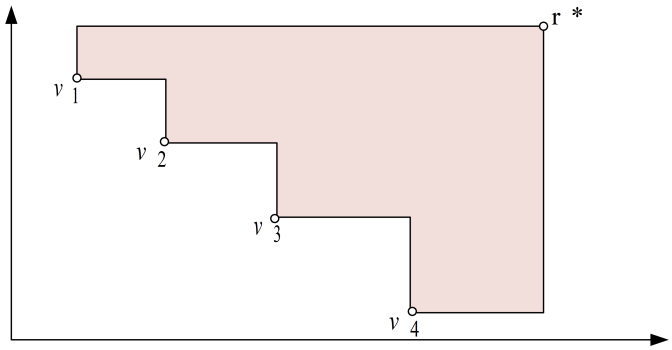

Figure 2. Example of HV. The part of r^* dominated by $v1, v2, v3$, and $v4$ is the HV of this example (i.e., the area of the pink).

4. Scalarization Method

It is very difficult to solve multi-objective problems directly, so it is a common method to transform a multi-objective optimization problem into a set of single-objective optimization problems. The specific method is to normalize each sub-objective in the multi-objective, so as to obtain a set of problems and to solve the set of problems. Three commonly used quantization methods are defined below.

- Weighted Sum:

$$\min_{x \in \mathcal{X}} g_{\text{ws}}(x \mid \lambda) = \min_{x} \in \mathcal{X} \sum_{i=1}^{m} \lambda_i f_i(x) \tag{3}$$

Weighted sum is the simplest scalarization method [49], where λ is the coefficient vector of the objective function, also known as preference. However, the weighted

sum method is not good at solving non-convex parts of the Pareto front [50], and more methods have been proposed to accommodate the more complex Pareto front.

- Tchebycheff Approach:

$$\min g_{\text{te}}(x \mid \lambda, z^*) = \max_{1 \le i \le m} \{\lambda_i | f_i(x) - z_i^* |\} \tag{4}$$

Tchebycheff is another commonly used scalarization method [49], which is roughly based on the idea of reducing the maximum gap and thus approximating the individual to Pareto front.

- Penalty-based Boundary Intersection (PBI) Approach:

$$\min_{x \in \mathcal{X}} g_{\text{pbi}}(x \mid \lambda, z^*) = d_1 + \theta d_2 \tag{5}$$

$$d_1 = \frac{\left\| (F(x) - z^*)^T w\lambda \right\|}{\|\lambda|} \tag{6}$$

$$d_2 = \left\| F(x) - \left(z^* + d_1 \frac{\lambda}{\|\lambda\|} \right) \right\| \tag{7}$$

PBI aims to make $F(x)$ as close to the theoretical optimal solution as possible on the premise of ensuring the convergence and diversity of the method [43]. However, PBI needs to set hyperparameters in advance, which limits the flexibility of this method.

5. Methodology

In this section, we propose a method to automatically adjust preferences according to the solution of the problem in the objective space, which hardly increases the quality of solution of neural combinatorial optimization algorithm. We define the method as a Markov Decision Process (MDP) and train it through reinforcement learning.

5.1. Markov Decision Process

Preference adjustment is an important part of multi-objective optimization. A good adjustment method can ensure the diversity of solutions in multi-objective algorithms. Traditional weight adjustment method includes several steps, an AdaW [24] summary for archive maintenance, weight addition, weight generation, weight deletion, and five weight update steps. The algorithm complexity of the traditional method will increase with the number of objectives, number of populations, number of archives, number of weights, and other parameters, and it is not suitable for the new neural method. However, the algorithmic complexity of our proposed method is only related to the amount of fine-tuning. If no fine-tuning is selected, the algorithm complexity at inference time is O_1. We define the MDP formulation $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$ as follows:

State space $\mathcal{S}$: $s_t = \{O_1, O_2\}$ is a state in $\mathcal{S}$, and it consists of two parts: the first part is the objective values corresponding to the uniform preferences ou, and the second part is the objective value corresponding to the adjustable weights oa. A more detailed definition is as follows:

$$O_1 = \{(ou_1^1, \ldots, ou_N^1), \ldots, (ou_1^M, \ldots, ou_N^M)\} \tag{8}$$

$$O_2 = \{(oa_1^1, \ldots, oa_N^1), \ldots, (oa_1^M, \ldots, oa_N^M)\} \tag{9}$$

where N is the number of objective and M is the number of preferences.

Action space $\mathcal{A}$: the action $a_t \in \mathcal{A}$ is to find a new preference, and the objective values corresponding to the old weight in O_2 will be replaced by the objective value corresponding to the new weight.

Transition probability $\mathcal{P}$: The transition probability here is obtained with the learned method and is determined in our method.

Reward function r: The reward function is designed as the difference between the HV of the current state and the next state. That is, we learned a strategy of adjusting preferences to make HV larger.

Discount rate γ: The discount rate γ is set to 1.0, which means that future rewards are not discounted for the current state.

We define the process of weight adjustment as a sequential process. In this process, we learn a strategy that aims to maximize the numerical value of HV by gradually adjusting the preferences and learning it through a reinforcement learning algorithm; we apply the Soft Actor Critic algorithm (SAC) [51] algorithm as our reinforcement learning algorithm. A simple illustration is shown in Figure 3. To further improve the effect of our method, in the experiment, we fixed instances and fine-tuned them. Of course, it is okay not to fine-tune. The specific method of fine-tuning is that we will update the neural network parameters several times online for a given instance to achieve better results, and the fine-tuning time is usually not very long.

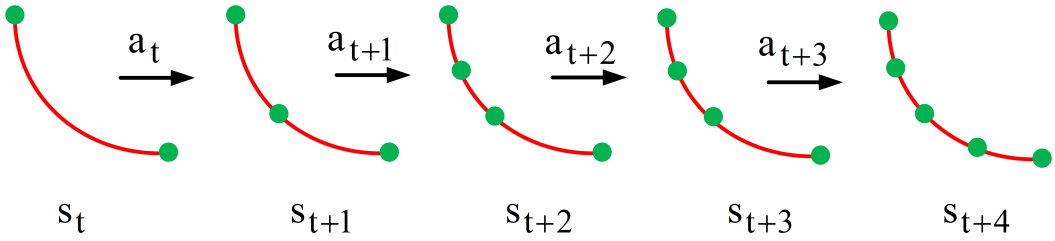

Figure 3. A simplified illustration of the MDP. Given state s, action a will choose the appropriate preference at each step and obtain reward r.

5.2. Model Design

5.2.1. Basic Model

We selected the state-of-the-art preference-conditioned multi-objective model in PMOCO as our basic model, and its structure is a improvement and extension of the famous Attention Model [35].

Encoder The decoder mainly consists of multi-head attention, a batch norm layer, and a feed forward layer [22]. Unlike Transformer, we do not use position embeddings in the input to the model, instead taking the geographic location of n customers and a depot as input. So, we could obtain the embedded location information in the encoder.

Preference-Conditioned Decoder The decoder in our paper is different from that in Transformer. The preference-conditioned model is inspired by hypernetworks and trains a network capable of generating decoder parameters $\theta(\lambda) = \{W_Q(\lambda), W_K(\lambda), W_V(\lambda), W_{\text{MHA}}(\lambda)\}$ based on preferred input, where $W_Q(\lambda)$ is the query embedding, $W_K(\lambda)$ is the key embedding, $W_V(\lambda)$ is the value embedding, and $W_{\text{MHA}}(\lambda)$ is the multi-head attention embedding; more details can be found in Figure 4. In solving CVRP, the decoder obtains a complete route through multiple iterations. In every iteration t, the encoder will concatenate the first selected route embedding h_1 with the previous selected node embedding h_{t-1} and combines with the embedding parameters $\theta(\lambda)$, obtaining the complete context embedding:

$$\boldsymbol{h}_{(C)} = \textbf{MHA}\big(Q = [h_1, h_{t-1}]W_Q(\lambda), K = \{\boldsymbol{e}_{1,n}, d\}W_K(\lambda), V = \{\boldsymbol{e}_{1,n}, d\}W_V(\lambda)\big)W_{\text{MHA}}(\lambda) \quad (10)$$

where $e_{1,n}$ is the customer location embedding and d is the depot location embedding. The selected customer node will be masked, and other customer node will be selected according to the final $softmax$ layer. The decoder selects a customer node for each iteration. When the total amount of goods at the selected customer node exceeds the capacity of the vehicle, the vehicle will restart from the depot. And, when all customer nodes are selected, we obtain the full trajectory π.

5.2.2. Preference Adjustment Model

In the model part of preference adjustment, we use the multi-layer perceptron [52] (MLP) as the backbone of the model. We design a Y-shaped network, that is, the input layer is divided into two parts: one part accepts the objective values corresponding to the uniform embedding as input, and the other part accepts the objective values corresponding to the adjustable embedding as input, and the two parts of the input are concatenated together after the partial embedding layer (i.e., MLP). We predict the new weight at last. Such a design can not only effectively obtain the characteristics of different types of weights but also obtain their mixed characteristics, which helps neural networks make more comprehensive use of valid input information. Both of the models are shown in Figure 4. The pseudo-code of our training algorithm is given in Algorithm 1.

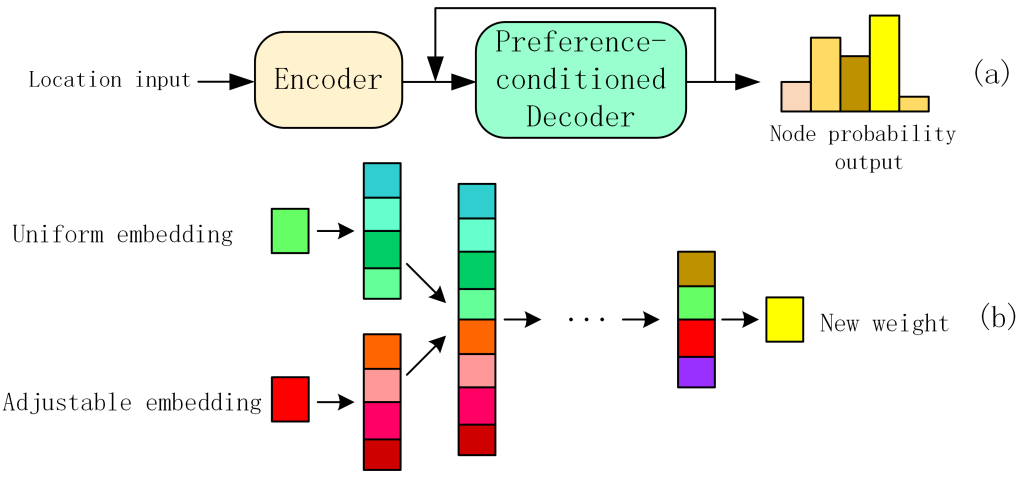

Figure 4. A simple illustration of the network structure. (**a**) is the preference-conditioned model, and (**b**) is the preference adjustment model

Algorithm 1 Preference adjustment model training.

Input: A preference-conditioned model P, a preference adjustment model $\mathcal{M}$ with parameter θ, training instance distribution D, number of preferences K, learning rate α, number of training iterations $\mathcal{N}$, batch size $\mathcal{B}$;

for $n = 1, \ldots, \mathcal{N}$ **do**
  Sample a mini batch of $\mathcal{B}$ instances $d_n \sim D$
  Initialize preference set $Q \leftarrow \emptyset$
  for $k = 1, \ldots, K$ **do**
    Generate a new preference $p_k = \mathcal{M}(Q|\theta)$
    Update preference set $Q' \leftarrow Q \cup \{p_K\}$
    Calculate reward $r = HV(P(Q', d_n)) - HV(P(Q, d_n))$
    Training the model using SAC: $\nabla_\theta \mathcal{M}(\theta) \leftarrow SAC(p_k, r, Q, Q'), \theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{M}(\theta)$
    $Q \leftarrow Q'$
  end for
end for
Return The trained preference adjustment model $\mathcal{M}$

6. Experiment Setting

In this section, we introduce the setting of our experiments, including the baselines, problems and settings, and inference and metrics.

6.1. Baseline

In this paper, we will compare our approach to the existing neural multi-objective combinatorial optimization approaches PMOCO [21] and DRL-MOA [44]. DRL-MOA is the first method to optimize multi-objective combinatorial optimization problems using neural networks. It regards a multi-objective optimization problem as an optimization

problem with multiple single objectives and uses transfer training method to improve the effect and to reduce the training amount. PMOCO is a newly proposed state-of-the-art method. Inspired by hypernetwork, it uses preference information to generate decoder's parameters and to obtain Pareto solutions under arbitrary preferences. In the experiments, a data augment method (Aug) proposed by POMO is used in PMOCO and our method adjustable preference (AP) to further improve the effect. The data augmentation method in POMO considers the symmetry of CVRP and amplifies one instance into eight different instances that have the same solution. The optimal solution of these eight instances is the solution after using the data augmentation method.

6.2. Problems and Setting

We considered MOCVRP as our experimental study, and the MOCVRP has three sizes: 20, 50, and 100. We set the learning rate at $\alpha = 10^{-4}$, the training iterations at $\mathcal{N}$ = 10,000, and batch size at $\mathcal{B} = 128$ in our method. We set $K = 20$ preferences in our experiments, both for our method and the baselines. We experimented on a single NVIDIA GeForce RTX 2080 super GPU and a single 3.6 GHZ intel i9-9900k CPU. All the results in our experiments are averaged over the results of 100 random instances.

6.3. Data Sets

In CVRP, it has a homogeneous fleet of vehicles with the same capacity D, which undertakes the task of transporting goods from one depot node to other n prespecified customer nodes and back to the depot node. Capacity D represents a certain capacity of the vehicle, such as the maximum load of the vehicle. Each customer node i is defined as a two-dimensional coordinate with the number of demands d_i to be satisfied. For the route traveled by each vehicle, the total demand of the customers it visits cannot exceed its capacity. The multi-objective CVRP (MOCVRP) considered in this paper involves the optimization of two conflicting objectives, namely minimizing the total trip length of all vehicles and minimizing the longest route length between vehicles.

Our dataset is generated in the same way as PMOCO [21]. We consider three problem sizes with 20, 50, and 100 customers. We generate the locations (i.e., the two-dimensional coordinates) of the customer nodes and depot node by uniformly sampling from the unit square $[0, 1]^2$. For the demand, we uniformly sample d_i from the set $\{1, \ldots, 9\}$, and the capacity D is set to 30, 40, and 50, respectively, for problems of size 20, 50, and 100, respectively.

6.4. Inference and Metrics

We use a variety of metrics to measure the effectiveness of our approach, including HV, gap, and model information. HV is one of the most commonly used metrics in multi-objective optimization. It can reflect not only the quality of the solution but also the uniformity of the solution, which is a comprehensive metric. More details about HV have been explained in detail in Section 3. Gap is the ratio of hypervolume difference between the current method and the optimal method under the same experimental conditions.

7. Results

We first discuss the number of trainable models and parameters of our method and the baselines. In Table 1, we list the model information of different methods. We can see that DRL-MOA [44], which is based on a pointer network [33] with 0.2 M trainable parameters, needs to train a model for each required preference. In contrast, the number of trainable models and parameters of PMOCO [21] and our method does not increase with the number of preferences. Although the base model, which is the attention model in [35], has 1.4 M parameters and is larger than that of DRL-MOA [44], the total number of trainable parameters is much smaller that of than DRL-MOA when the required number of preferences is large (20 in our case). This is convenient in practice because usually a large number of preferences is required to cover the Pareto front. In this case, the parameter training and storage cost of DRL-MOA [44] would quickly become infeasible. Compared with PMOCO,

though, our method needs to train one more model, i.e., the preference adjustment network; however, it does not add many trainable parameters since the preference adjustment model is very small.

Table 1. Model information for different methods ("#" means "numbers of").

Method	Base Model	#Models	#Params
DRL-MOA	Pointer-Network	#Pref	#Pref × 0.2 M
PMOCO	Attention Model	1	1.4 M
AP	Attention Model + MLP	2	1.4 M

In Table 2, we summarize the experimental results on MOCVRP. From the table, we can conclude that our method greatly improves the performance of the original method (PMOCO) in all experiments. Among all the methods, DRL-MOA is less effective than other methods. Compared with the state-of-the-art multi-objective neural combinatorial optimization method PMOCO, our approach achieved an average improvement of about 6%, without the data augmentation technique. When the usage data were augmented, the gap between our approach and PMOCO was narrowed, with an average improvement of about 3%. In Figure 5, we can also intuitively see that the preference distribution learned by our method is more uniform than that of PMOCO. Uniform preferences are unevenly distributed across the Pareto front of MOCVRP, and many overlapping points are generated. Our approach learns a more evenly distributed combination of weights across the Pareto front, thus improving the quality of the solution. We are pleasantly surprised that even without data augmentation, our approach did not lag far behind PMOCO using data augmentation, which further demonstrates the competitiveness of our approach for data augmentation, which typically requires additional GPU memory and a lot of inference time.

Table 2. Experimental results on MOCVRP with different input sizes (**bold** means the best among all methods).

	MOCVRP20		MOCVRP50		MOCVRP100	
Method	**HV**	**Gap**	**HV**	**Gap**	**HV**	**Gap**
DRL-MOA	0.141	39.74%	0.218	53.72%	0.199	56.07%
PMOCO	0.206	11.97%	0.410	12.95%	0.400	11.70%
PMOCO-Aug	0.231	1.28%	0.454	3.61%	0.428	5.52%
AP	0.225	3.85%	0.442	6.16%	0.420	7.28%
AP-Aug	**0.234**	**0.00%**	**0.471**	**0.00%**	**0.453**	**0.00%**

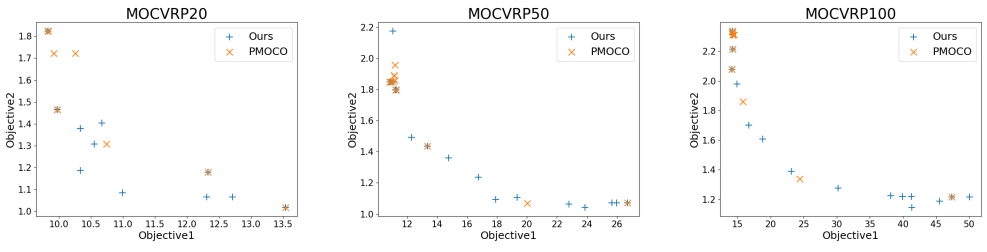

Figure 5. Pareto front of PMOCO and ours. We can intuitively see that the weights learned through our method are more evenly distributed across the Pareto front.

8. Conclusions and Future Work

We proposed a novel preference adjustable multi-objective vehicle routing optimization method. It treats the preference adjustment as a sequential decision-making process and was trained with common reinforcement learning methods. It makes the distribution of preferences more consistent with the problem, rather than a simple uniform distribution.

Because uniformly distributed preferences are often not guaranteed to be optimal preferences. We improved the PMOCO with our proposed method, which greatly improves the effect of PMOCO under 20 preferences. Furthermore, our proposed method solves the problem of a high time complexity when combining the traditional heuristic-based weight adjustment method and a deep learning algorithm. We also designed a series of experiments to verify the effectiveness of our method. The experimental results on MOCVRP with 20, 50, and 100 customers show that the proposed method can improve the results of the state-of-the-art PMOCO by about 6% with 20 preferences.

Although our method has achieved good results, there is still much room for improvement. Our approach treats preference adjustment as a sequential decision process, but in reality, there is no strict ordering of preferences selected in a certain order because these preferences are more similar to a set than a sequence. So, designing a set-based method to adjust preferences may be an important research direction in the future. In addition, our approach is less effective when there are a large number of preferences. Although this situation is relatively rare in practical application, it is still an urgent problem to be solved. Improving our approach so that it could work well even with a large number of preferences is also worth exploring in the future.

Author Contributions: Conceptualization, Y.S. and Q.C.; methodology, L.W., Y.S. and Q.C.; software, L.W., C.S. and C.L.; validation, L.W. and C.L.; formal analysis, L.W.; investigation, C.S.; resources, C.S.; data curation, L.W.; writing—original draft preparation, L.W.; writing—review and editing, L.W. and Q.C.; visualization, C.L.; supervision, Q.C.; project administration, Q.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Braekers, K.; Ramaekers, K.; Van Nieuwenhuyse, I. The vehicle routing problem: State of the art classification and review. *Comput. Ind. Eng.* **2016**, *99*, 300–313. [CrossRef]
2. Desrochers, M.; Desrosiers, J.; Solomon, M. A new optimization algorithm for the vehicle routing problem with time windows. *Oper. Res.* **1992**, *40*, 342–354. [CrossRef]
3. Lysgaard, J.; Letchford, A.N.; Eglese, R.W. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math. Program.* **2004**, *100*, 423–445. [CrossRef]
4. Ho, W.; Ho, G.T.; Ji, P.; Lau, H.C. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Eng. Appl. Artif. Intell.* **2008**, *21*, 548–557. [CrossRef]
5. Archetti, C.; Speranza, M.G.; Hertz, A. A tabu search algorithm for the split delivery vehicle routing problem. *Transp. Sci.* **2006**, *40*, 64–73. [CrossRef]
6. Laporte, G. Fifty years of vehicle routing. *Transp. Sci.* **2009**, *43*, 408–416. [CrossRef]
7. Archetti, C.; Feillet, D.; Gendreau, M.; Speranza, M.G. Complexity of the VRP and SDVRP. *Transp. Res. Part C Emerg. Technol.* **2011**, *19*, 741–750. [CrossRef]
8. Qin, Z.; Lu, X.; Nie, X.; Liu, D.; Yin, Y.; Wang, W. Coarse-to-fine video instance segmentation with factorized conditional appearance flows. *IEEE/CAA J. Autom. Sin.* **2023**, *10*, 1192–1208. [CrossRef]
9. Qin, Z.; Lu, X.; Nie, X.; Yin, Y.; Shen, J. Exposing the Self-Supervised Space-Time Correspondence Learning via Graph Kernels. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 2110–2118. [CrossRef]
10. Lu, X.; Wang, W.; Shen, J.; Crandall, D.J.; Van Gool, L. Segmenting objects from relational visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7885–7897. [CrossRef]
11. Song, W.; Cao, Z.; Zhang, J.; Xu, C.; Lim, A. Learning variable ordering heuristics for solving Constraint Satisfaction Problems. *Eng. Appl. Artif. Intell.* **2022**, *109*, 104603. [CrossRef]
12. Zhang, Z.; Song, W.; Li, Q. Dual-aspect self-attention based on transformer for remaining useful life prediction. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 2505711. [CrossRef]
13. Bhat, S.A.; Huang, N.F.; Hussain, I.; Sajjad, U. Correlating the Ambient Conditions and Performance Indicators of the LoRaWAN via Surrogate Gaussian Process based Bidirectional LSTM Stacked Autoencoder Showkat. *IEEE Trans. Netw. Serv. Manag.* **2023**. [CrossRef]
14. Zhang, L.; Bibi, F.; Hussain, I.; Sultan, M.; Arshad, A.; Hasnain, S.; Alarifi, I.M.; Alamir, M.A.; Sajjad, U. Evaluating the stress-strain relationship of the additively manufactured lattice structures. *Micromachines* **2022**, *14*, 75. [CrossRef]

15. Hottung, A.; Tierney, K. Neural large neighborhood search for the capacitated vehicle routing problem. *arXiv* **2019**. [CrossRef]
16. Kalakanti, A.K.; Verma, S.; Paul, T.; Yoshida, T. RL SolVeR pro: Reinforcement learning for solving vehicle routing problem. In Proceedings of the 2019 1st international conference on artificial intelligence and data sciences (AiDAS), Ipoh, Malaysia, 19 September 2019; pp. 94–99.
17. Nazari, M.; Oroojlooy, A.; Snyder, L.; Takác, M. Reinforcement learning for solving the vehicle routing problem. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 9839–9849.
18. Song, W.; Chen, X.; Li, Q.; Cao, Z. Flexible Job-Shop Scheduling via Graph Neural Network and Deep Reinforcement Learning. *IEEE Trans. Ind. Inform.* **2022**, *19*, 1600–1610. [CrossRef]
19. Liu, F.; Lu, C.; Gui, L.; Zhang, Q.; Tong, X.; Yuan, M. Heuristics for Vehicle Routing Problem: A Survey and Recent Advances. *arXiv* **2023**. [CrossRef]
20. Ye, T.; Zhang, Z.; Chen, J.; Wang, J. Weight-Specific-Decoder Attention Model to Solve Multiobjective Combinatorial Optimization Problems. In Proceedings of the 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Prague, Czech Republic, 9–12 October 2022; pp. 2839–2844.
21. Lin, X.; Yang, Z.; Zhang, Q. Pareto set learning for neural multi-objective combinatorial optimization. *arXiv* **2022**. [CrossRef]
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
23. Von Oswald, J.; Henning, C.; Grewe, B.F.; Sacramento, J. Continual learning with hypernetworks. *arXiv* **2019**. [CrossRef]
24. Li, M.; Yao, X. What weights work for you? Adapting weights for any Pareto front shape in decomposition-based evolutionary multiobjective optimisation. *Evol. Comput.* **2020**, *28*, 227–253. [CrossRef] [PubMed]
25. Toth, P.; Vigo, D. Branch-and-bound algorithms for the capacitated VRP. In *The Vehicle Routing Problem*; SIAM: Philadelphia, PA, USA, 2002; pp. 29–51.
26. Achuthan, N.; Caccetta, L. Integer linear programming formulation for a vehicle routing problem. *Eur. J. Oper. Res.* **1991**, *52*, 86–89. [CrossRef]
27. Novoa, C.; Storer, R. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *Eur. J. Oper. Res.* **2009**, *196*, 509–515. [CrossRef]
28. Renaud, J.; Boctor, F.F. A sweep-based algorithm for the fleet size and mix vehicle routing problem. *Eur. J. Oper. Res.* **2002**, *140*, 618–628. [CrossRef]
29. Lysgaard, J. *Clarke & Wright's Savings Algorithm*; Department of Management Science and Logistics, The Aarhus School of Business: Aarhus, Denmark, 1997; Volume 44.
30. Shaw, P. *A New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems*; APES Group, Dept. of 435 Computer Science, University of Strathclyde: Glasgow, UK, 1997; Volume 46.
31. Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1996**, *26*, 29–41. [CrossRef] [PubMed]
32. Thangiah, S.R.; Osman, I.H.; Sun, T. *Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Methods for Vehicle Routing Problems with Time Windows*; Technical Report SRU CpSc-TR-94-27; Computer Science Department, Slippery Rock University: Slippery Rock, PA, USA, 1994; Volume 69.
33. Vinyals, O.; Fortunato, M.; Jaitly, N. Pointer networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 2692–2700.
34. Scarselli, F.; Tsoi, A.C.; Hagenbuchner, M. The vapnik–chervonenkis dimension of graph and recursive neural networks. *Neural Netw.* **2018**, *108*, 248–259. [CrossRef]
35. Kool, W.; Van Hoof, H.; Welling, M. Attention, learn to solve routing problems! *arXiv* **2018**. [CrossRef]
36. Xin, L.; Song, W.; Cao, Z.; Zhang, J. Step-wise deep learning models for solving routing problems. *IEEE Trans. Ind. Inform.* **2020**, *17*, 4861–4871. [CrossRef]
37. Xin, L.; Song, W.; Cao, Z.; Zhang, J. Multi-decoder attention model with embedding glimpse for solving vehicle routing problems. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 12042–12049. [CrossRef]
38. Wu, Y.; Song, W.; Cao, Z.; Zhang, J.; Lim, A. Learning improvement heuristics for solving routing problems. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 5057–5069. [CrossRef] [PubMed]
39. Chen, X.; Tian, Y. Learning to perform local rewriting for combinatorial optimization. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 6281–6292.
40. Santana, Í.; Lodi, A.; Vidal, T. Neural Networks for Local Search and Crossover in Vehicle Routing: A Possible Overkill? In *Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*; Springer: Cham, Switzerland, 2023; pp. 184–199.
41. Feng, L.; Huang, Y.; Tsang, I.W.; Gupta, A.; Tang, K.; Tan, K.C.; Ong, Y.S. Towards faster vehicle routing by transferring knowledge from customer representation. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 952–965. [CrossRef]
42. Gunantara, N. A review of multi-objective optimization: Methods and its applications. *Cogent Eng.* **2018**, *5*, 1502242. [CrossRef]
43. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
44. Li, K.; Zhang, T.; Wang, R. Deep reinforcement learning for multiobjective optimization. *IEEE Trans. Cybern.* **2020**, *51*, 3103–3114. [CrossRef]

45. Ruchte, M.; Grabocka, J. Scalable pareto front approximation for deep multi-objective learning. In Proceedings of the 2021 IEEE international conference on data mining (ICDM), Auckland, New Zealand, 7–10 December 2021; pp. 1306–1311.
46. Zhang, Z.; Wu, Z.; Zhang, H.; Wang, J. Meta-learning-based deep reinforcement learning for multiobjective optimization problems. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**. [CrossRef]
47. Lin, X.; Zhen, H.L.; Li, Z.; Zhang, Q.F.; Kwong, S. Pareto multi-task learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 12060–12070.
48. Zitzler, E.; Brockhoff, D.; Thiele, L. The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. In *Evolutionary Multi-Criterion Optimization: Proceedings of the 4th International Conference, EMO 2007, Matsushima, Japan, 5–8 March 2007*; Proceedings 4; Springer: Cham, Switzerland, 2007; pp. 862–876.
49. Miettinen, K. *Nonlinear Multiobjective Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1999; Volume 12.
50. Ehrgott, M. *Multicriteria Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2005; Volume 491.
51. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference on Machine Learning (PMLR), Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.
52. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **1958**, *65*, 386. [CrossRef]