symmetry

MDPI

Article

Multi-Task Multi-Objective Evolutionary Search Based on Deep Reinforcement Learning for Multi-Objective Vehicle Routing Problems with Time Windows

Jianjun Deng [1], Junjie Wang [2], Xiaojun Wang [2], Yiqiao Cai [2,*] and Peizhong Liu [3,*]

[1] Chengdu Aeronautic Polytechnic, Chengdu 610100, China; dengjianjun@cap.edu.cn
[2] College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China; 18816234790@163.com (J.W.); t_wxjmq@163.com (X.W.)
[3] College of Engineering, Huaqiao University, Quanzhou 362000, China
* Correspondence: caiyq@hqu.edu.cn (Y.C.); pzliu@hqu.edu.cn (P.L.)

Abstract: The vehicle routing problem with time windows (VRPTW) is a widely studied combinatorial optimization problem in supply chains and logistics within the last decade. Recent research has explored the potential of deep reinforcement learning (DRL) as a promising solution for the VRPTW. However, the challenge of addressing the VRPTW with many conflicting objectives (MOVRPTW) still remains for DRL. The MOVRPTW considers five conflicting objectives simultaneously: minimizing the number of vehicles required, the total travel distance, the travel time of the longest route, the total waiting time for early arrivals, and the total delay time for late arrivals. To tackle the MOVRPTW, this study introduces the MTMO/DRP-AT, a multi-task multi-objective evolutionary search algorithm, by making full use of both DRL and the multitasking mechanism. In the MTMO/DRL-AT, a two-objective MOVRPTW is constructed as an assisted task, with the objectives being to minimize the total travel distance and the travel time of the longest route. Both the main task and the assisted task are simultaneously solved in a multitasking scenario. Each task is decomposed into scalar optimization subproblems, which are then solved by an attention model trained using DRL. The outputs of these trained models serve as the initial solutions for the MTMO/DRL-AT. Subsequently, the proposed algorithm incorporates knowledge transfer and multiple local search operators to further enhance the quality of these promising solutions. The simulation results on real-world benchmarks highlight the superior performance of the MTMO/DRL-AT compared to several other algorithms in solving the MOVRPTW.

Keywords: multiobjective vehicle routing problem with time windows; deep reinforcement learning; evolutionary multi-task optimization; knowledge transfer

Citation: Deng, J.; Wang, J.; Wang, X.; Cai, Y.; Liu, P. Multi-Task Multi-Objective Evolutionary Search Based on Deep Reinforcement Learning for Multi-Objective Vehicle Routing Problems with Time Windows. *Symmetry* **2024**, *16*, 1030. https://doi.org/10.3390/sym16081030

Academic Editors: Hsien-Chung Wu and Sergei D. Odintsov

Received: 4 June 2024
Revised: 28 July 2024
Accepted: 2 August 2024
Published: 12 August 2024

1. Introduction

The vehicle routing problem with time windows (VRPTW) is a widely studied combinatorial optimization problem in logistics, encompassing areas such as supply chain management, production planning, waste collection, home healthcare, and so on [1–5]. As a crucial variant of the vehicle routing problem (VRP), the VRPTW involves servicing a set of customers with specific time windows and known demands using a fleet of vehicles [1]. The primary goal of the VRPTW is to minimize delivery costs by optimizing routes while adhering to all constraints. However, the VRPTW is computationally NP-hard [2], making it challenging to solve effectively.

Due to its practical significance in various applications in the real world, the VRPTW has emerged as a prominent research problem in the field of operations research [2,3]. Consequently, numerous optimization approaches have been developed to tackle this challenge [6–8]. Broadly, optimization approaches for the VRPTW are categorized into exact methods [6], suitable for small-scale problems, and meta-heuristic methods [7,8], preferred

for large-scale problems. Meta-heuristic methods, known for their capability and potential in tackling the VRPTW, encompass various search mechanisms [3,7,8]. These methods address both the single-objective VRPTW and the multi-objective VRPTW. Previous studies [9,10] have discussed the VRPTW as an inherently multi-objective optimization problem with many conflicting objectives relevant to real-world applications. As a result, research on the multi-objective VRPTW (MOVRPTW) problem has gained significant attention and is now considered a prominent area in the field of computational intelligence [3].

However, because of the high complexity of the MOVRPTW, most existing meta-heuristic methods still face significant challenges in effectively solving it [11]. These methods often require a significant number of iterations to update the population or conduct search, especially for optimization problems with many conflicting objectives. This will lead to lengthy computational times for optimization. Furthermore, meta-heuristic methods necessitate problem-specific experience and knowledge, requiring adjustments to yield favorable results when encountering new problems or even new instances of similar problems [12]. Therefore, there yet remains much room for proposing more efficient approaches to address the challenges brought by the MOVRPTW.

With the rapid advancement of artificial intelligence technology, deep reinforcement learning (DRL) has become increasingly prevalent and successful across various fields. Notably, it has made significant contributions in areas such as computer vision [13,14] and natural language processing [15]. In the realms of operations research and combinatorial optimization, DRL has also proven its advantages in terms of autonomous feature discovery, effective accumulation of problem information, and efficient decision optimization [16–19]. However, as discussed in [20], directly applying the trained model on unseen problem instances may be considered unreliable. Furthermore, the majority of DRL-based methodologies concentrate on resolving a single MOVRPTW problem by initiating the search from scratch, disregarding the similarities between disparate tasks. Consequently, the useful knowledge gained by addressing one problem cannot be fully leveraged for optimizing other similar problems. Therefore, it is crucial to explore ways to further enhance the quality of the output results obtained by the trained model, especially in the context of DRL-based approaches.

Recently, a new paradigm called evolutionary multitask optimization (EMTO) has emerged in the field of evolutionary algorithms. EMTO aims to optimize multiple tasks simultaneously using a shared search space [21]. By leveraging the latent synergies among those tasks, EMTO has been shown to outperform single-task optimization methods, yielding superior performance in both continuous and combinatorial optimization problems [21,22]. Furthermore, the efficacy of EMTO has been demonstrated in successfully solving a wide range of combinatorial optimization problems [23,24]. It can, thus, be seen that the integration of the EMTO framework with DRL-based approaches presents a compelling proposition for addressing complex combinatorial optimization problems.

Building upon the aforementioned findings, this study introduces the MTMO/DRL-AT, a multi-task multi-objective evolutionary search algorithm for solving the MOVRPTW with five conflicting objectives. The proposed algorithm combines DRL and the multi-tasking mechanism. In the MTMO/DRL-AT, a two-objective VPRTW is constructed as an assisted task based on the characteristics of the main MOVRPTW task. Both the main task and the assisted task are decomposed into scalar optimization subproblems, each addressed by an attention model trained using DRL. The output results of these trained models serve as the initial solutions for the MTMO/DRL-AT. Subsequently, the proposed algorithm optimizes both tasks simultaneously under a multitasking framework. To further improve the quality of the solutions, multiple local search operators are employed. Experimental studies on 45 real-world instances are conducted to validate the effectiveness of the proposed algorithm. The simulation results clearly demonstrate the superiority of the MTMO/DRL-AT over other compared approaches in solving MOVRPTWs.

In summary, the main contributions of this study are as follows:

- A novel evolutionary optimization algorithm, termed the MTMO/DRL-AT, is presented for solving MOVRPTWs involving five conflicting objectives. The MTMO/DRL-AT conducts a multitasking search over both the main task and an assisted task, utilizing an attention model trained through DRL.
- The synergy between DRL-based model training and the multitasking-based search mechanism is built up. Attention models are trained using DRL for subproblems in both the main and assisted tasks, serving as the starting point for the algorithm. Knowledge transfer strategies and objectivewise local search operators are then employed to further refine the optimization of both tasks, ultimately improving the quality of solutions derived from the trained models.

The remainder of this paper is structured as follows: Section 2 describes the formulation of the MOVRPTW. Section 3 reviews related work. Section 4 shows the DRL-based modeling and training for the MOVRPTW. Section 5 presents the details of the MTMO/DRL-AT. Then, the experimental results and analysis are provided in Section 6. Finally, Section 7 gives the conclusions and future work.

2. Problem Formulation of MOVRPTW

The MOVRPTW is a complex multi-objective optimization problem with practical applications and multiple constraints. It can be mathematically represented by a complete undirected graph, denoted as G = $\{V, E\}$, where V represents the node set and E represents the edge set. The node set $V = \{v_i | i = 0, 1, \ldots, N\}$, consists of a depot, denoted as v_0, and other customer nodes, $v_1, v_2, \ldots, v_N$. The edge set $E = \{e_{i,j} | i, j \in V, i \neq j\}$, where each edge $e_{i,j}$ is linked to a travel time $t_{i,j}$ and a travel distance $d_{i,j}$. Similarly, each customer is assigned to a demand q_i, a service time window $[b_i, e_i]$, and a service time s_i.

In the MOVRPTW, each vehicle is assigned a route, $r_k = (c_0^k, c_1^k, \ldots, c_{N_k}^k, c_{N_{k+1}}^k)$, that consists of a sequence of N_k customers to be visited, denoted as r_k and $c_0^k = c_{N_k+1}^k = 0$, where c_j^k represents the jth customer to be visited in r_k and $c_0^k = c_{N_k+1}^k = 0$ (depot). Each customer is exclusively serviced by a single vehicle. Moreover, it is essential to ensure that the cumulative demand of customers assigned to each vehicle does not exceed its maximum capacity, denoted as Q. Additionally, all vehicles are obligated to depart from and return to the depot within the time window specified as $[0, e_0]$. To allow for some flexibility, a soft time window constraint is implemented, permitting a vehicle to arrive at a customer's location after the specified latest service time, e_i, within a maximum allowed delay time, denoted as md. The delay time experienced by vehicle k at the jth customer is defined as $dt_{c_j^k} = \max\{0, a_{c_j^k} - e_{c_j^k}\}$, where $a_{c_j^k}$ represents the arrival time at customer c_j^k. In case a vehicle arrives prior to the earliest service time (b_i), it is required to wait until b_i to initiate service, resulting in a waiting time. The waiting time for vehicle k at the jth customer is determined by $w_{c_j^k} = \max\{0, b_{c_j^k} - a_{c_j^k}\}$.

Figure 1 provides an example of the solution representation for the MOVRPTW. As illustrated in Figure 1, the MOVRPTW consists of one depot (i.e., 0) and nine customers to be serviced (i.e., 1 to 9). A solution comprising three routes is denoted as $x = (r_1, r_2, r_3)$, where $r_1 = (0, 5, 7, 1, 0)$, $r_2 = (0, 9, 6, 3, 0)$, and $r_3 = (0, 8, 4, 2, 0)$.

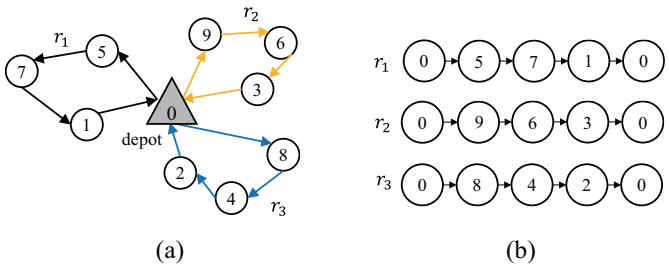

Figure 1. Solution representation for the MOVRPTW. (**a**) A solution for the MOVRTPW. (**b**) The solution representation.

To provide a clear mathematical model of the MOVRPTW, the basic notations used in this study are summarized in Table 1.

Table 1. Notations for MOVRPTW.

Notation	Description
Property sets:	
C	The set of customers: $C = \{1, 2, 3, \ldots, N\}$;
V	The set of vertices: $V = C \cup \{0\}$;
E	The set of edges between vertices: $E = \{e_{ij} \mid i, j \in V\}$;
D	The set of distances between customers: $D = \{d_{ij} \mid i, j \in C\}$;
T	The set of travel times between customers: $T = \{t_{ij} \mid i, j \in C\}$.;
Problem parameters:	
Q	The maximum capacity of the vehicle;
q_i	The demand of customer i;
md	The maximum allowable delay time at each customer;
$[b_i, e_i]$	The time window of customer i;
b_i	The earliest service time for customer i;
e_i	The latest service time for customer i;
s_i	The service time for customer i.
Problem variables:	
x_{ij}^k	e_{ij} is traversed by the kth vehicle (i.e., $x_{ij}^k = 1$) or not (i.e., $x_{ij}^k = 0$);
K	The number of routes in x;
r_k	The kth route consisting of a sequence of N_k customers $r_k = \{c_0^k, c_1^k, \ldots, c_{N_k}^k, c_{N_{k+1}}^k\}$;
c_j^k	The jth customer visited in the kth route;
a_i	The time the vehicle arrives at customer i;
w_i	The waiting time incurred by the vehicle at customer i;
dt_i	The delay time generated by the vehicle at customer i.

In general, the mathematical model of the MOVRPTW, which includes five objectives, is defined as follows [9,10]:

$$\min F(x) = (f_1, f_2, f_3, f_4, f_5) \tag{1}$$

$$f_1 = K \tag{2}$$

$$f_2 = \sum_{i=1}^{N} \sum_{j=0, j \neq i}^{N} \sum_{k=1}^{K} d_{ij} x_{ij}^k \tag{3}$$

$$f_3 = \max_{k=1,\ldots,K} \{ \sum_{i=1}^{N} \sum_{j=0, j \neq i}^{N} x_{ij}^k (t_{ij} + w_i + s_i) \} \tag{4}$$

$$f_4 = \sum_{i=1}^{N} \sum_{j=0, j \neq i}^{N} \sum_{k=1}^{K} w_i x_{ij}^k \tag{5}$$

$$f_5 = \sum_{i=1}^{N} \sum_{j=0, j \neq i}^{N} \sum_{k=1}^{K} dt_i x_{ij}^k \tag{6}$$

The MOVRPTW mathematical model, described by Equation (1), is a multi-objective problem that encompassed five objectives. These objectives are defined as follows: In Equation (2), the first objective aims to minimize the number of vehicles required. In Equation (3), the second objective focuses on minimizing the total travel distance. In Equation (4), the third objective aims to minimize the travel time of the longest route. In Equation (5), the fourth objective seeks to minimize the total waiting time for early arrivals. In Equation (6), the fifth objective aims to minimize the total delay time for late arrivals.

The constraints of the MOVRPTW are defined as follows:

$$\sum_{i=1}^{N} x_{i0}^{k} = \sum_{j=1}^{N} x_{0j}^{k} = 1, \quad k = 1, \ldots, K \tag{7}$$

$$\sum_{j=0, j\neq i}^{N} x_{ij}^{k} = \sum_{j=0, j\neq i}^{N} x_{ji}^{k} \leq 1, \quad i \in C, k = 1, \ldots, K \tag{8}$$

$$\sum_{i=0, i\neq j}^{N} \sum_{k=1}^{K} x_{ij}^{k} = \sum_{j=0, j\neq i}^{N} \sum_{k=1}^{K} x_{ij}^{k} = 1, \quad i \in C, j \in C \tag{9}$$

$$\sum_{i=0}^{N} q_i \sum_{j=0, j\neq i}^{N} x_{ij}^{k} \leq Q, \quad k = 1, \ldots, K \tag{10}$$

$$\sum_{j=0, j\neq i}^{N} dt_i x_{ij}^{k} \leq md, \quad i \in C, k = 1, \ldots, K \tag{11}$$

$$(t_{i0} + a_i + w_i + s_i) x_{i0}^{k} \leq e_0, \quad i \in C, k = 1, \ldots, K \tag{12}$$

$$x_{ij}^{k} \in \{0, 1\}, \quad i \in C, j \in C, k = 1, \ldots, K \tag{13}$$

Constraints (7) and (8) ensure that each vehicle starts from the depot and then returns to the depot. Constraint (9) guarantees that each customer is served only once by one vehicle. Constraint (10) ensures that the total demand served by a vehicle does not exceed its maximum capacity Q. Constraint (11) limits the delay time for each customer to the specified value md. Constraint (12) states that each vehicle must return to the depot before it closes. Constraint (13) defines the range of the decision variable.

3. Literature Review

This section begins by providing an overview of the existing studies conducted on the VRPTW. Subsequently, it briefly examines recent DRL approaches applied to combinatorial optimization problems (COPs), with a specific focus on the VRP and its variants. Lastly, it reviews the applications of EMTO to the VRP.

3.1. Meta-Heuristic Approaches for VRPTW

Broadly, optimization approaches for the VRPTW are categorized into exact methods [6], suitable for small-scale problems, and meta-heuristic methods [7,8], preferred for large-scale problems. Meta-heuristic methods, known for their capability and potential in tackling the VRPTW, encompass various search mechanisms [3,7,8]. These methods address both the single-objective VRPTW and the multi-objective VRPTW. Previous studies [9,10] have discussed the VRPTW as an inherently multi-objective optimization problem with many conflicting objectives relevant to real-world applications. Therefore, this subsection provides only a brief overview of the related work on the MOVRPTW, which is summarized in Table 2. Other related work on the VRP and its variants can be found in [2,3,6–8].

Researchers have proposed various multi-objective optimization algorithms with various optimization frameworks and local search strategies to address the MOVRPTW. For example, Qi et al. [25] introduced a decomposition-based multi-objective evolutionary algorithm, which included a specially designed selection operator and three local searches. Moradi [26] proposed a discrete learnable evolution model for multi-objective optimization that integrated machine learning and a new priority-based representation scheme. In addition to the above evolutionary optimization methods, DRL-based methods are also used to solve the MOVRPTW. In [19], Zhang et al. introduced the MODRL/D-EL, an approach that combines the decomposition technique with attention models. They also employed evolutionary learning to further fine-tune the parameters of the trained model.

Table 2. Summary of the methods for solving the MOVRPTW.

Reference	Authors	Problem	Approach
		MOVRPTW with two objectives	
[25]	Qi et al.	MOVRPTW with f_1 and f_2	Decomposition-based EA Specially designed selection operator Three novel local searches
[26]	Moradi	MOVRPTW with f_1 and f_2	The strength Pareto evolutionary algorithm (SPEA) Discrete learnable evolution model A priority-based representation scheme
[19]	Zhang et al.	MOVRPTW with f_2 and f_3	Multiobjective DRL with evolutionary learning (MODRL/D-EL) Decomposition technique Attention models Evolutionary learning to further fine-tune the model's parameters
		MOVRPTW with many objectives	
[9]	Gutiérrez et al.	MOVRPTW with f_1–f_5	Nondominated sorting genetic algorithm (NSGA-II) New instances from real-world data
[10]	Zhou and Wang	MOVRPTW with f_1–f_5	Local-search-based multiobjective optimization algorithm (LSMOVRPTW) Objectivewise local searches
[27]	Zhang et al.	MOVRPTW with f_1–f_5	Multi-objective memetic algorithm based on adaptive local search chains (MMA-ALSC) Enhanced local search chain techniques Multi-directional local search strategy
[28]	Cai et al.	MOVRPTW with f_1–f_5	Hybrid evolutionary multitasking algorithm (HEMT) Simultaneously optimize multiple distinct instances Knowledge transfer and knowledge reuse strategies

Efforts have also been made to tackle the VRPTW with more than three objectives (also called the many-objective VRPTW [29]). To address this problem, Gutiérrez et al. [9] proposed a nondominated sorting genetic algorithm (NSGA-II) and developed new instances from real-world data to address weak dependence relationships among objectives. Followed that, Zhou and Wang [10] designed multiple objectivewise local searches for distinct objectives of the VRPTW, thereby proposing a local search-based multiobjective optimization algorithm (LSMOVRPTW). Recently, Zhang et al. [27] presented a multi-objective memetic algorithm based on adaptive local search chains (MMA-ALSC). This approach combined enhanced local search chain techniques with a multi-directional local search strategy to guide the search process. By exploiting the similarity between different MOVRPTWs, Cai et al. [28] proposed a hybrid evolutionary multitasking algorithm (HEMT). Their approach involved solving multiple different MOVRPTWs concurrently, employing an exploration stage that incorporated knowledge transfer and an exploitation stage that used a knowledge reuse strategy.

3.2. The DRL-Based Approaches for the COPs

In recent years, DRL has proven successful in addressing complex COPs across various fields. For single-objective optimization, Vinyals et al. [30] proposed a Pointer network (Ptr-Net) model based on the sequence-to-sequence (Seq2Seq) model, achieving good results on the Traveling Salesman Problem (TSP). Bello et al. [16] trained a Ptr-Net model to solve TSPs using reinforcement learning and a critic network as a baseline. Nazari et al. [31] used Ptr-Net to solve dynamic VRPs by dividing the instances into dynamic and static parts and then trained the model with reinforcement learning algorithms. Nowak et al. [32] proposed a Graph Neural Network (GNN) using supervised training and beam search. Deudon et al. [33] improved the traditional Pointer network based on a Transformer with MHA and reinforcement learning.

Kool et al. [34] introduced an attention-based approach for solving various COPs, outperforming Ptr-Net on the TSP, CVRP, PCTSP, and others. Zhao et al. [17] designed an

adaptive discriminator to optimize the parameters of DRL models and a routing simulator to aid in training and evaluating the effectiveness of DRL models. Peng et al. [35] proposed a dynamic attention model for the VRP using a dynamic encoding–decoding structure with reinforcement learning. Wang et al. [18] proposed a feedback mechanism integrating an iterative greedy algorithm for flow shop scheduling problems based on DRL.

Furthermore, DRL has been applied to solve multi-objective COPs. Li et al. [12] developed the DRL-MOA, a framework using decomposition and Ptr-Net for a multi-objective TSP. Wu et al. [36] extended the DRL-MOA with the MODRL/DAM, constructing an attention model for each subproblem and training them with reinforcement learning. Similarly, Zhang et al. [19] presented the MODRL/D-EL, combining decomposition, attention models, and evolutionary algorithms for parameter fine-tuning.

3.3. The EMTO Approaches for VRP

In contrast with traditional optimization approaches that focus solely on a single optimization problem, EMTO aims to address multiple optimization tasks concurrently within a unified representation space [21,22]. By leveraging the underlying synergies among different optimization tasks, EMTO has demonstrated its potential in achieving superior performance for both continuous and combinatorial optimization problems when compared to its single-task counterparts [21,22]. The effectiveness and promising capabilities of EMTO in addressing multiple related optimization tasks have garnered significant interest from researchers, resulting in the development of various EMTO algorithms in the fields of science and engineering [22]. In the literature, EMTO has been successfully applied to solve the VRP and its variants.

In [37], a permutation-based multifactorial evolutionary algorithm (P-MFEA) was proposed to address multiple capacitated VRPs simultaneously. In the P-MFEA, a permutation-based unified representation was introduced as a replacement for the random key unified representation. Additionally, a split-based decoding operator was utilized to translate the solutions from the unified space to the problem-specific space.

In [24], an explicit EMTO (EEMTO) approach was presented to solve the capacitated VRP. EEMTO incorporates a weighted l_1-norm regularized learning process to capture the transfer mapping and uses a solution-based knowledge transfer process across different VRPs.

In [23], an EMTO was applied to address a novel variant of the VRP, called the VRP with heterogeneous capacity, time window, and occasional driver (VRPHTO). The proposed EMTO algorithm optimizes multiple VRPHTOs simultaneously and employs four operators: permutation-based common representation, split procedure, routing information exchange, and chromosome evaluation.

In [28], a hybrid evolutionary multitask algorithm (HEMT) was proposed to solve multiple MOVRPTWs in a multitasking scenario. The HEMT incorporates an exploration stage for global search with knowledge transfer, an exploitation stage for local search with knowledge reuse, and a tradeoff mechanism to balance these search processes.

The aforementioned related works highlight the advantages of using the EMTO framework in solving VRPs. However, it is worth noting that most existing EMTO approaches primarily focus on addressing VRPs with a single objective or two objectives. The application of EMTO in the context of the VRP with many objectives (more than three) is relatively limited, which motives our interest to further investigate its potential.

4. DRL-Based Modeling and Training

As reviewed above, DRL has shown its advantages in solving VRPs [36,38]. However, most of the works only focus on the extraction of node features, ignoring the fact that the distances and traveling times between customers in the real world are asymmetric. To efficiently solve the real-world MOVRPTW considered in this study, a multiobjective DRL method [36] is employed. This method uses the decomposition strategy and the attention model to enhance the optimization process.

In this section, the decomposition and parameter-transfer strategies for the MOVRPTW are firstly introduced. Then, the encoder and decoder of the attention model for each subproblem are presented. Finally, the training process for the models through DRL is given.

4.1. Decomposition and Parameter-Transfer Strategies

In this study, the MOVRPTW is decomposed into M subproblems using the weighted sum approach [39]. Specifically, a set of weight vectors W is generated for the MOVRPTW using Das and Dennis's method [40]. These weight vectors are then used to define the objective function of the jth subproblem by the weighted sum approach, as follows:

$$\min g^{ws}(\pi|\lambda_i) = \sum_{j=1}^{m} \lambda_{ij} \bar{f}_j(\pi) \tag{14}$$

where $\boldsymbol{\lambda}_i = (\lambda_{i1}, \ldots, \lambda_{iM})$ represents the weight vector of the ith subproblem, with the constraints that $\sum_{j=1}^{M} \lambda_{ij} = 1$.

After the decomposition, each scalar optimization subproblem is modeled by a neural network, which is then solved using DRL methods. Additionally, to expedite model training, a neighborhood-based parameter-transfer strategy [12] is utilized, as depicted in Figure 2. This strategy involves transferring the parameters from the model of a solved subproblem to the model of its neighboring subproblem. The neighboring subproblem's model is then trained using these transferred parameters as the initial starting point. More details of the parameter-transfer strategy can be found in [12].

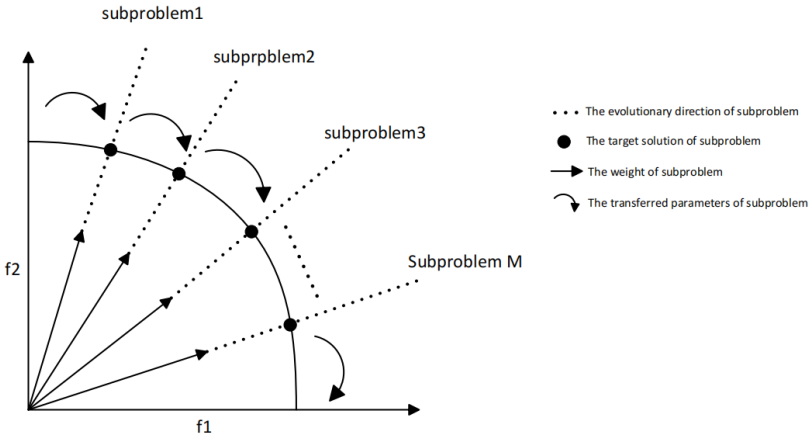

Figure 2. The neighborhood-based parameter-transfer strategy.

4.2. Encoder of Model

The encoder comprises three components. The first component uses a fully connected layer to transform the feature vectors. These vectors consist of the coordinates (x_i, y_i), time windows $[b_i, e_i]$, demands q_i, service duration w_i, travel time t_{ij}, and travel distance d_{ij} between customers, the initial embedding being h_i^0 and h_{ij}^0. The second component incorporates a multi-head attention mechanism to aggregate the information features from both node and edge embeddings. The processed data are then further updated and transformed in the last component through a combination of a residual network and a fully connected feedforward layer. This results in the generation of the final embedding h_i^N and h_{ij}^N. The structure of the encoder can be visualized in Figure 3.

As shown in Figure 3, the input data are split into two parts: node embeddings (i.e., $c_i = [(x_i, y_i), q_i, (b_i, e_i), s_i]$), which contain node-specific information, and edge embeddings (i.e., $e_{ij} = (d_{ij}, t_{ij})$), which include distance and time information between customers.

In the encoder of the model, the feature vector is transformed into the initial node embedded in the network by linear transformation as follows:

$$h_i^0 = W_N c_i + b_N \tag{15}$$

$$h_{ij}^0 = W_E e_{ij} + b_E \tag{16}$$

where $i, j \in N$, N is the No. of customers, and W_N and b_N are trainable network parameters.

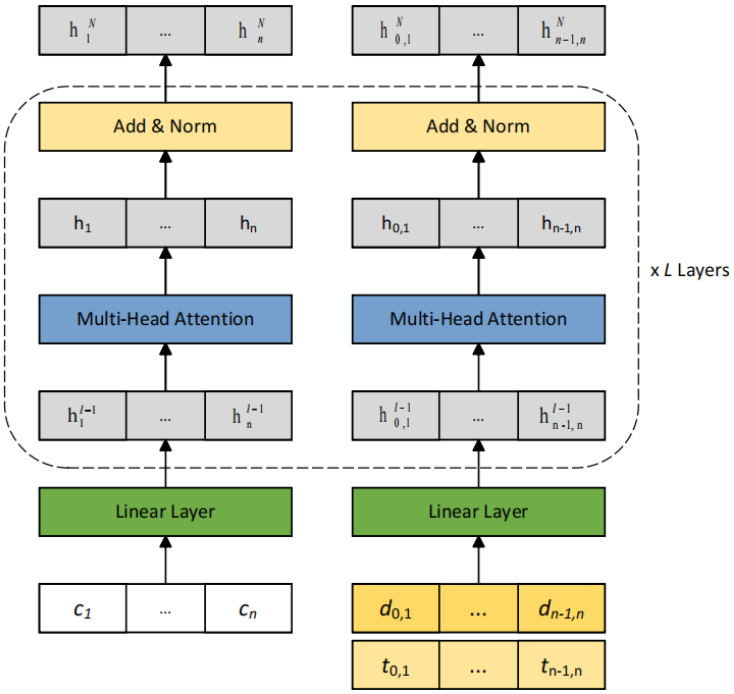

Figure 3. The structure of the encoder in the model.

Then, the node embeddings $h_0^l, \ldots, h_N^l$ and edge embeddings $h_{0,1}^l, \ldots, h_{N-1,N}^l$ are aggregated using the multi-head attention operator, as follows:

$$\widetilde{h}_i^l = BN(h_i^{l-1} + MHA^l(W_N^l h_i^{l-1})) \tag{17}$$

$$\widetilde{h}_{ij}^l = BN(MHA^l(W_E^l h_{ij}^{l-1})) \tag{18}$$

where $BN(\cdot)$ represents the batch normalized layer and $MHA(\cdot)$ refers to the multi-head attention layers. Note that the MHA is related to the three vectors q_i^l, k_i^l, and v_i^l. These vectors can be calculated as follows: $q_i^l = W_q^l h_i^{l-1}, k_i^l = W_k^l[h_i^{l-1}; h_{ij}^{l-1}], v_i^l = W_v^l[h_i^{l-1}; h_{ij}^{l-1}]$. Here, the trainable parameters W_q^l, W_k^l, and W_v^l are used to map the embeddings to the query, key, and value vectors, respectively.

After that, the embeddings of nodes and edges are combined by the residual network layer (add and norm) to update the embeddings of each node, as follows:

$$h_i^l = ReLu(h_i^{l-1} + FF^l(\hat{h}_i^l)) \tag{19}$$

$$h_{ij}^l = ReLu(h_{ij}^{l-1} + FF^l(\hat{h}_{ij}^l)) \tag{20}$$

where $FF(\cdot)$ (feedforward) is a fully connected feedforward layer, which can further improve the expression capability of the network.

Finally, the final embedding vector of each node is obtained through N attention layers, as follows:

$$\overline{h}_o^N = \frac{1}{n}\sum_{i=0}^{n} h_i^N \tag{21}$$

$$\overline{h}_e^N = \frac{1}{n}\sum_{i=0}^{n} h_{ij}^N \tag{22}$$

where $\overline{h}_o^N$ and $\overline{h}_e^N$ are the final embedding vectors of the node feature and edge feature, respectively. The node feature, the edge feature, and the final embedded vector will be output from the encoder to the decoder.

4.3. Decoder of Model

The primary function of the decoder is to estimate the probability distribution of the remaining nodes being selected based on the embedding vector of the nodes and edges that are output from the encoder. This process is repeated iteratively until all customers are served. More specifically, at each time step $t \in N$, the decoder determines the optimal decision on π_t by considering the partial tour $\pi_{1:t-1}$ and the embedding vector of the nodes and edges. Figure 4 shows the structure of the decoder.

First, a context embedding representing the relationships between contexts is needed. The initial context (i.e., $t = 1$) includes the node features ($h_o^{N'}$) and the embedding vectors of the edge features ($h_e^{N'}$), both obtained from the encoder. Additionally, the current vehicle's remaining capacity (Q_t) and the last customer served by the vehicle ($h_{\pi_{t-1}}^N$) are incorporated into the initial context. The description of the initial context is as follows:

$$h_c^{N'} = \begin{cases} [\overline{h}_o^N, \overline{h}_e^N, Q_t, h_0], & t = 1 \\ [\overline{h}_o^N, \overline{h}_e^N, Q_t, h_{\pi_{t-1}^N}], t > 1 \end{cases} \tag{23}$$

where $[\dots]$ denotes the vector connection operator.

Then, a new context vector $h_c^{N'}$ is calculated using the MHA network layer. For each node, its key vector (q_c) and the value vector (v_c) are derived from the embedding vectors of the encoder. The transformation process is as follows:

$$\begin{aligned} q_c &= W^Q h_c^{N'}, \\ k_i &= W^K [h_i; h_{ij}] + q_c, \\ v_i &= W^V [h_i; h_{ij}] + q_c \end{aligned} \tag{24}$$

where W^Q, W^K, and W^V are the trainable parameters. Subsequently, the compatibility of each node is computed by masking the nodes that have been visited. The compatibility values are within the range of $[-1, 1]$ and are determined as follows:

$$u_{(c)i} = \begin{cases} \frac{q_c^T k_i}{\sqrt{d_k}}, i \notin \pi_t \\ -\infty, \text{otherwise} \end{cases} \tag{25}$$

where i denotes the node index and d_k is the dimension of q_c/k_i. Then, based on Equation (13), the compatibility of each node is recalculated by transforming the context vector and the embedding vectors of nodes and edges into the corresponding q, k, and v, with the range of $[-C, C]$, as follows:

$$u_{(c)i} = \begin{cases} C \cdot tanh(\frac{q_c^T k_i}{\sqrt{d_k}}), i \notin \pi_t \\ -\infty, \text{otherwise} \end{cases} \tag{26}$$

Finally, the probability of selecting node x_i as the next node to be visited is calculated as follows:

$$p_i = p_\theta(\pi_t = i|\pi_{1:t-1}, s) = \frac{e^{u_{(c)i}}}{\sum_j e^{u_{(c)j}}} \tag{27}$$

The decoder repeats the steps mentioned above, where each time, the selected node is masked. This process continues until all customers are selected.

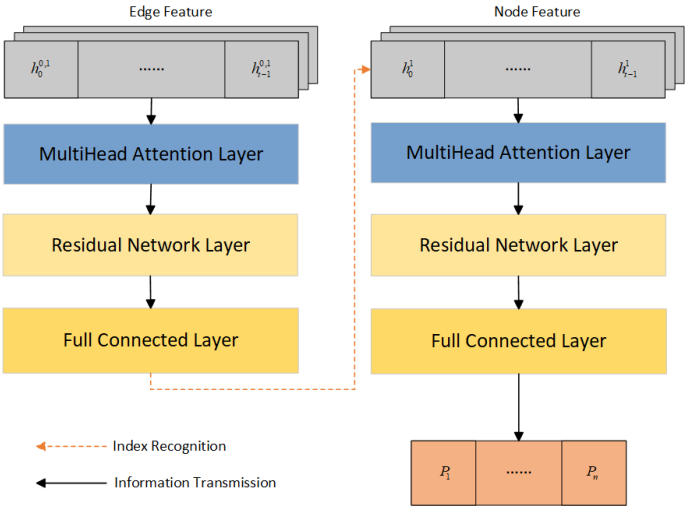

Figure 4. The structure of the decoder in the model.

4.4. Training Driven by DRL

In this section, we adopt the well-known actor–critic method [41] to train the model of each subproblem. The training process, employing the actor–critic method, is outlined in Algorithm 1. To train both the actor and critic networks, the Adam optimizer [42] is employed in this study.

In the algorithm, the actor network, which is an attention model, is responsible for learning the strategy gradient and selecting actions based on the probability distributions of nodes generated by the decoder. On the other hand, the critic network acts as a baseline to predict an estimation of the objective function for the subproblem and evaluate the results obtained from the actor network's strategy. This evaluation assists the actor in making action selections. Therefore, the training parameters of each subproblem (W_{λ_i}) include an actor network parameterized by θ and a critic network parameterized by ϕ.

Suppose that the processing sequence π generated by the actor network obeys the distribution $\pi \sim p_\theta(\cdot|X)$, where $p_\theta(\cdot|X)$ is the policy given by the actor network for an instance X. The objective $\ell(\theta|X)$ is the expected $g^{ws}(\pi|\lambda, X)$:

$$\ell(\theta|X) = \mathbb{E}_{\pi \sim p_\theta(\cdot|X)} g^{ws}(\pi|\lambda, X) \tag{28}$$

where $g^{ws}(\pi|\lambda, X)$ represents the min value calculated through the sequence π for X.

Then, the gradients of the parameters θ are calculated as follows:

$$\nabla_\theta \ell(\theta|X) = \frac{1}{B} \sum_{j=1}^{B} [(g^{ws}(\pi_i|\lambda_j, X_j) - b_\phi(X_j)) \nabla_\theta log p_\theta(\pi_j|X_j)] \tag{29}$$

Here, B represents the batch size, which is the number of samples for each training iteration. X_j is a randomly selected instance of the subproblem, and π_j represents the solution for X_j obtained from the actor network. In addition, $b_\phi(X_j)$ refers to a baseline function, which is computed by the critic network. Its purpose is to estimate the expected objective value, which helps to reduce the variance of the gradients.

For the critic network, its goal is to learn how to estimate the expected objective value for a given instance X_j. Therefore, the objective function of the critic network can be defined as a mean-squared error function between the actual objective function generated by the actor network for X_j and the predicted objective value $b_\phi(X_j)$ from the critic network. This can be expressed as follows:

$$L_\phi = \frac{1}{B}\sum_{j=1}^{B}(b_\phi(X_j) - g^{ws}_{\min}(\pi_j|\lambda_i, X_j))^2 \tag{30}$$

Algorithm 1 Actor–critic training method [41]

Input: Number of problem instances T, number of iterations E, parameters of actor network θ and critic network ϕ. **Output**: Trained parameter θ, ϕ

1: $\theta, \phi \leftarrow$ initialized parameter as Ref. [12];
2: **For** $iteration = 1$ to E
3: **For** $k = 1$ to T
4: **For** $j = 1$ to B
5: $\pi_j \leftarrow p_\theta(X_j)$;
6: $b_j \leftarrow b_\phi(X_j)$;
7: **End**
8: $d_\theta = \frac{1}{B}\sum_{j=1}^{B}[g^{ws}(\pi_j|\lambda_i, X_j) - b_j \nabla_\theta log p_\theta(\pi_j|X_j)]$;
9: $L_\phi = \frac{1}{B}\sum_{j=1}^{B}(b_j - g^{ws}(\pi_j|\lambda_i, X_j))^2$;
10: $\theta \leftarrow ADAM(\theta, d_\theta)$;
11: $\phi \leftarrow ADAM(\phi, \nabla_\phi L_\phi)$;
12: **End**
13: **End**

5. MTMO/DRL-AT

In this section, a multi-task multi-objective evolutionary search algorithm based on DRL (MTMO/DRL-AT) is presented. Specifically, the general framework of the proposed algorithm is firstly outlined. Then, three main components of the MTMO/DRL-AT, i.e., the construction of the assisted task, knowledge transfer across tasks, and local search, are elaborated.

5.1. General Framework of MTMO/DRL-AT

The MTMO/DRL-AT framework is outlined in Algorithm 2. As can be seen, the MTMO/DRL-AT consists of three main phases: the initialization phase, the transfer reproduction phase, and the local search phase. In the initialization phase (Lines 3–6), the populations of the main task and the constructed assisted task are initialized using the trained DRL-based models, as described in Section 3. Specifically, n models are selected from the trained models for the main task, and thus, the population with n solutions is directly obtained by these models. Similarly, for the assisted task, its population with n solutions is produced using the selected trained models of the assisted task. In the transfer reproduction phase (Line 8), the knowledge transfer process is applied to update the solutions in the external archive A by leveraging the knowledge from both the main task and the assisted task. In the local search phase (Line 9), the objectivewise local searches [10] are employed to further refine the solutions in the archive A. Finally, when the stopping condition is met, the external archive A is returned as the approximate Pareto set for the MOVRPTW.

Algorithm 2 MTMO/DRL-AT

Input: Maximum running time of the target task T, population size $popsize$, training batch size $batch$, number of transferred solutions N_f, number of subproblems for main task N_m, number of subproblems for assisted task N_a, the trained models for the main task $Model_m$, the trained models for the assisted task $Model_a$.

Output: The external archive A.

$A = \emptyset$; //Define the external archive for the main task
$n = popsize/batch$; // Calculate the number of submodels from M
$SetIdx1 \leftarrow Random^n(1, N_m)$; // Randomly select n values from [1, N_m] as the indexes of the models for the main task;
$SetIdx2 \leftarrow Random^n(1, N_a)$; // Randomly select n values from [1, N_a] as the indexes of the models for the assisted task;
$Pop_m \leftarrow Model_m(SetIdx1, popsize)$; // Initialize Pop_m with the selected models for the main task
$Pop_a \leftarrow Model_a(SetIdx2, popsize)$; // Initialize Pop_a with the selected models for the assisted task
While $t < T$ **Do**
    $Transfer\_reproduction(Pop_m, Pop_a, A, N_f)$; // see Algorithm 3
    $Local\_search(Pop_m, A)$; // see Algorithm 4
End while

5.2. Construction of the Assisted Task

When solving an MOVRPTW with many objectives, most multi-objective evolutionary algorithms perform poorly due to a significant proportion of incomparable and mutually nondominated solutions [29]. To address this issue, an assisted task is constructed in a simpler search space for the MOVRPTW. This enables efficient assistance in optimizing the original problem through knowledge transfer. By leveraging the simpler task, the search process for the main task becomes more effective in finding high-quality solutions.

In the MOVRPTW, optimizing the objectives related to the total travel distance and the travel time of the longest route greatly impacts the optimization of other objectives. Therefore, the construction of the assisted task focuses on these two objectives. By selecting them as the optimization objectives for the assisted task, the aim is to effectively optimize these crucial factors, which in turn can positively influence the optimization of other related objectives in the MOVRPTW problem.

Therefore, the mathematical model of the assisted task is defined below:

$$\min H = (h_1, h_2) \tag{31}$$

$$h_1 = \sum_{k=1}^{|R|} \sum_{i=0}^{N_k} d_{c_i^k, c_{i+1}^k} \tag{32}$$

$$h_2 = \max\{t_{N_k|k=1,2,\ldots,|R|}\} \tag{33}$$

where h_1 and h_2 correspond to the f_2 and f_3, respectively, of the main task (i.e., Equation (1)). Additionally, the constraints of the assisted task are identical to those of the main task, as shown in Equation (7).

Furthermore, due to that the assisted task having a similar structure and characteristics as the main task, the DRL-based modeling and training methods described in Section 3 are also adopted for the assisted task.

5.3. Transfer Reproduction Operator

To effectively exploit the useful search experiences obtained from the constructed assisted task, a transfer reproduction operator is employed to transfer knowledge between the main and assisted tasks. The procedure of the transfer reproduction operator is shown in Algorithm 3.

Algorithm 3 Transfer reproduction

Input: Population of the main task Pop_m, population of the assisted task Pop_a, number of transferred solutions N_f, the external archive A.
Output: The updated A.

1: $C \leftarrow \emptyset$;
2: $O \leftarrow \emptyset$;
3: Use the fast nondominated sorting method [43] for the solutions in Pop_m and Pop_a, respectively;
4: $C \leftarrow$ the best N_f solutions in Pop_a;
5: $C \leftarrow C \bigcup$ the worst $popsize - N_f$ solutions in Pop_m;
6: Re-evaluate all solutions in C with the main task;
7: **For** $x_i \in C, i = 1, \ldots, popsize$
8: $o_i \leftarrow Genetic\_operator(x_i)$;
9: Evaluate o_i with the main task;
10: $O \leftarrow O \bigcup o_i$;
11: **End**
12: Update Pop_m with $C \bigcup O$;
13: Update A with $C \bigcup O$.

As shown in Algorithm 3, in Line 3, all solutions in Pop_m and Pop_a are ranked, respectively, using the fast nondominated sorting approach [43]. Subsequently, as shown in Lines 4 and 5, the best N_f solutions in Pop_a and the worst $popsize - N_f$ solutions in Pop_m are selected to form the set C. Next, in Line 6, each solution in C is re-evaluated under the main task environment. Note that the duplicate solutions are removed from the set. Afterwards, Line 8 employs the genetic operators on the solutions in C to generate offspring. In this study, the mutation strategy and the crossover operator of differential evolution (DE) [44] are adopted as the $Genetic\_operator(\cdot)$. Specifically, for each solution $x_i \in C$, a mutant vector (v_i) is first generated through the "DE/rand/1" mutation strategy, as follows:

$$v_i = x_{r1} + F \times (x_{r2} - x_{r3}) \tag{34}$$

where F is the mutation factor and $r1$, $r2$, and $r3 \in \{1, 2, \ldots, |C|\} \setminus \{i\}$ are randomly selected indices. Following that, a trial vector (u_i) is generated by using the binomial crossover operator for the pair of x_i and v_i, as follows:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } rand(0,1) \leq Cr \text{ or } j = j_{rand} \\ x_{i,j}, & \text{otherwise.} \end{cases} \tag{35}$$

Here, $Cr \in [0, 1]$ represents the crossover rate, $rand(0, 1) \in (0, 1)$ denotes a randomly generated variable, and $j_{rand} \in [1, D_{\max}]$ indicates a randomly selected integer. Additionally, a random initialization will be performed if u_i exceeds the range of $[0, 1]$. It is worth noting that the solution to the problem is a customer sequence vector, whereas the solution obtained by the genetic operator is a continuous vector. To convert a continuous vector into a customer sequence, a ranked order value (ROV) mapping method [45] is employed.

Once each solution in O has been evaluated with the main task, Pop_m is updated with the solutions of $C \bigcup O$ using the nondominated sorting and crowding distance, as described in Line 12, following the approach in [43]. As for updating A, the ϵ-dominance relation suggested in [10] is adopted.

5.4. Local Search Operator

To further refine the solutions in A and achieve better performance for the main task, we used the objectivewise local searches [10] in the MTMO/DRL-AT, which is presented in Algorithm 4.

First, in Line 2, an initial solution x is randomly selected from A for the subsequent local searches. After that, in Line 4, the objectivewise local searches are conducted on

x. Following the approach in [10], the local search is independently performed for each objective, denoted as $LS_{f_i}(x)$ ($i = 1, \ldots, 5$), to enhance the quality of x with respect to the corresponding objective (f_i). Additionally, three neighborhood operators are integrated into the local searches for $f_2(x) - f_5(x)$. Specifically, in each search step, a random neighborhood operator is conducted on x to produce a new solution x'. If $f_i(x')$ is superior to $f_i(x)$, x is substituted by x'. Concurrently, x' is immediately used to update A through the ϵ-dominance relation in Line 5. For more details of the objectivewise local searches, please refer to [10]. Finally, in Line 8, the solutions in C are used to update Pop_m by directly replacing its inferior solutions.

Algorithm 4 Local search

Input: Population of the main task Pop_m, the external archive A.
Output: The updated A, the updated Pop_m.

1: $C \leftarrow \emptyset$;
2: $x \leftarrow Rndselect(A)$;
3: **For** $i = 1$ to 5
4: Perform $LS_{f_i}(x)$;
5: Update A with the obtained solutions;
6: Add the best solution in LS_{f_i} to C;
7: **End**
8: Replace the worst five solutions in Pop_m with the solutions in C.

6. Experiment

To assess the effectiveness of the MTMO/DRL-AT, a series of experiments was performed on a set of 45 real-world MOVRPTW instances. This section begins with a brief description of the MOVRPTW instances. Subsequently, the experimental setup is outlined, detailing the procedures and methodologies employed. Following that, a comprehensive comparison between the MTMO/DRL-AT and the representative algorithms is conducted. Finally, an in-depth analysis is presented to examine the influence of the main components of the MTMO/DRL-AT on its overall performance.

6.1. MOVRPTW Instances

To evaluate the effectiveness of the proposed algorithm, a set of 45 real-world instances of the MOVRPTW was adopted in this study. These instances, as described in [9], were derived from data obtained from an actual distribution company. Consequently, they reflect the complex and challenging nature of real-world MOVRPTW scenarios.

Table 3 provides an overview of the properties of these 45 MOVRPTW instances. As the table shows, these instances were generated by combining various features, including the number of customers (CN), the profile of time windows (PT), and the capacity of each vehicle (Q). The number of customers can be set to 50, 150, or 250, while the time window profile can range from 1 to 5. The capacity of each vehicle is determined using a formula that incorporates the lower and upper bounds ($\underline{D}$ and $\overline{D}$) and a modulation factor δ. Each MOVRPTW instance is labeled as "$a - b - c$", where a represents NC, b represents the index of the δ type, and c represents the index of the TW profile. For further details, please refer to [9,10].

Table 3. The real-world MOVRPTW instances.

Instance	*CN*	*Q*	*PT*	Instance	*CN*	*Q*	*PT*	Instance	*CN*	*Q*	*PT*
50-0-0	50	690	1	150-0-0	150	1854	1	250-0-0	250	3078	1
50-0-1	50	690	2	150-0-1	150	1854	2	250-0-1	250	3078	2
50-0-2	50	690	3	150-0-2	150	1854	3	250-0-2	250	3078	3
50-0-3	50	690	4	150-0-3	150	1854	4	250-0-3	250	3078	4
50-0-4	50	690	5	150-0-4	150	1854	5	250-0-4	250	3078	5

Table 3. *Cont.*

Instance	*CN*	*Q*	*PT*	Instance	*CN*	*Q*	*PT*	Instance	*CN*	*Q*	*PT*
50-1-0	50	250	1	150-1-0	150	638	1	250-1-0	250	1046	1
50-1-1	50	250	2	150-1-1	150	638	2	250-1-1	250	1046	2
50-1-2	50	250	3	150-1-2	150	638	3	250-1-2	250	1046	3
50-1-3	50	250	4	150-1-3	150	638	4	250-1-3	250	1046	4
50-1-4	50	250	5	150-1-4	150	638	5	250-1-4	250	1046	5
50-2-0	50	85	1	150-2-0	150	182	1	250-2-0	250	284	1
50-2-1	50	85	2	150-2-1	150	182	2	250-2-1	250	284	2
50-2-2	50	85	3	150-2-2	150	182	3	250-2-2	250	284	3
50-2-3	50	85	4	150-2-3	150	182	4	250-2-3	250	284	4
50-2-4	50	85	5	150-2-4	150	182	5	250-2-4	250	284	5

6.2. Experimental Setup

To train the models of the MTMO/DRL-AT, training instances of different sizes for the MOVRPTW were generated using a data simulator. The process involves randomly generating the coordinates of the depot and customer within the range $[0,1] \times [0,1]$. The distance and time matrices for travel between customers were randomly generated within the range of [0, 1]. For each customer, the demand was randomly generated within the range of [1, 9], the time window was randomly set as $b_i \in [0,5]$ and $e_i \in [0,5]$, and the service time was randomly selected from the set $\{1,5,2\}$. In addition, the maximum capacity of vehicles (Q) was set as follows: $Q = 20$ if $CN = 10$, $Q = 30$ if $CN = 20$, and $Q = 50$ if $CN = 40$. During the model-training process, problem instances with 40 nodes were used, and the dataset was generated based on the aforementioned process, with asymmetric distance and time matrices.

The parameter settings for the model and training were mostly similar to those described in [12,36], which are shown in Table 4. In addition, the parameter settings for the evolutionary search are also summarized in Table 4.

It is important to acknowledge that these parameter settings may not be optimal for the proposed algorithm, as finding the optimal settings can be challenging and often problem-specific. However, the effectiveness of these parameter settings has been demonstrated in the following experiments. In future work, the impact of these parameters on the performance of the MTMO/DRL-AT will be further investigated.

In the experiments, all the algorithms were implemented using Python, and the maximum running times of different instances were set according to the suggestions in [10]. Additionally, all the test experiments were conducted in the same configuration environment, as outlined in Table 5.

To evaluate the performance of the compared algorithms, two measures were employed: the inverted generational distance (IGD) [46] and hypervolume (HV) [47]. The IGD metric assesses both the convergence and diversity of the obtained nondominated solutions, while the HV metric evaluates the volume of the union of hypercubes determined by each nondominated solution and the reference point. A smaller value of the IGD or a larger value of the HV suggests better performance achieved by the corresponding algorithm in the approximation of the true Pareto front. For more detailed information on the IGD and HV metrics, please refer to [46,47].

To further demonstrate the significant differences between the compared algorithms, the KEEL software [48] was employed to conduct single-problem and multiple-problem analysis using the Wilcoxon test [49,50]. The results of single-problem analysis are summarized as "$w/t/l$", indicating that the considered algorithm is significantly better and performs equally to or performs worse than the competitor on the w, t, and l instances, respectively, at the 0.05 significance level. In the multiple-problem analysis, $R+$ and $R-$ represent the sum of ranks where the considered algorithm is significantly better than and worse than the competitor for all the instances, respectively. Additionally, the average ranking values of the considered algorithms for all instances were analyzed using

Friedman's test [49,50]. For brevity, this paper only presents the statistical results of the comparisons. For those interested in the detailed numerical values, please contact the corresponding author.

Table 4. Parameter settings.

Parameter	Value
For the model and training	
Input dimension	7
Node-embedding dimension	128
Batch size during training	500
Size of problem instances	5×10^6
Number of epochs for training the model for the first subproblem	5
Number of epochs for training the model for each remaining subproblem	1
Critic network architecture	four 1D convolutional layers with the following channels (7, 128), (128, 20), (20, 20), and (20, 1) $kernelsize = 1$, $stride = 1$
Number of attention layers	1
Number of heads	8
Dimension of the query vector and value vector	16
Learning rate for the Adam optimizer	0.0001
Number of decomposed subproblems for the main task	100
Number of decomposed subproblems for the assisted task	70
For the evolutionary search	
Population size ($popsize$)	50 for each task
Number of transferred solutions (TN)	15
Crossover rate (Cr)	0.9
Mutation factor (F)	0.5
Number of independent runs for each instance	30

Table 5. Experimental configuration.

Operating Environment	Version
Server	
System	Ubuntu 7.5.0
CPU	Intel Xeon Processor
GPU	GeForce RTX 2080 (8 G)
Memory	12 GB
CUDA	11.0
Local host	
System	Windows 10
CPU	Intel Xeon W-2223 (3.60 GHz)
Memory	16 GB

6.3. Performance Comparison

6.3.1. Comparison with LSMOVRPTW

In this section, we aim to demonstrate the effectiveness of the MTMO/DRL-AT for solving the MOVRPTW by comparing it with the LSMOVRPTW [10]. To provide a comprehensive overview of the performance comparisons, Table 6 presents the statistics summarizing these comparisons on all the instances.

As depicted in Table 6, the MTMO/DRL-AT demonstrates a significant improvement over the LSMOVRPTW in terms of both the IGD and HV. Specifically, based on the single-problem analysis conducted using the Wilcoxon test, the MTMO/DRL-AT significantly outperforms the LSMOVRPTW on 41 instances in terms of the IGD and on all 45 instances in terms of the HV. In the multiple-problem analysis carried out with the Wilcoxon test, the MTMO/DRL-AT achieves a higher $R+$ than $R-$ for both the IGD and HV. Additionally, based on the p-value, significant differences between the MTMO/DRL-AT and LSMOVRPTW are observed at both $\alpha = 0.05$ and $\alpha = 0.1$, indicating that the MTMO/DRL-AT outperforms the LSMOVRPTW overall.

Moreover, to visually illustrate the distinct characteristics of the competing algorithms, the approximate Pareto fronts of several representative instances obtained by the MTMO/DRL-AT and LSMOVRPTW are projected at the $f_1 - f_3$ and $f_2 - f_3$ planes, as shown in Figure 5. As the figure shows, the superiority of the MTMO/DRL-AT in achieving better Pareto fronts than the LSMOVRPTW for the selected instances is evident. The solutions generated by the MTMO/DRL-AT more accurately approximate the Pareto front and demonstrate a wider distribution along it. This further validates the superior convergence and diversity properties of the MTMO/DRL-AT in comparison to the LSMOVRPTW.

Table 6. Results of the single- and multiple-problem analysis by the Wilcoxon test between the MTMO/DRL-AT and LSMOVRPTW.

Algorithm	Metric	*w*/*t*/*l*	$R+$	$R-$	p-Value	$\alpha = 0.05$	$\alpha = 0.1$
MTMO/DRL-AT	IGD	41/4/0	1035.0	0.0	0.0	Yes	Yes
vs. LSMOVRPTW	HV	45/0/0	1035.0	0.0	0.0	Yes	Yes

Based on the aforementioned results, it is evident that the MTMO/DRL-AT outperforms the LSMOVRPTW on the majority of instances. This performance difference can be attributed to several factors that contribute to their varying performances: (1) The MTMO/DRL-AT incorporates attention models specifically designed for the subproblems of the MOVRPTW using DRL. These attention models are capable of adapting to MOVRPTW instances of varying scales. By leveraging the advantages of DRL, the attention models can learn to focus on critical aspects of the MOVRPTW and make more informed decisions during the optimization process. Furthermore, the output of the attention models in the MTMO/DRL-AT serves as high-quality initial solutions for the subsequent evolutionary process. These initial solutions provide a strong starting point for the algorithm, which can lead to faster convergence and better overall performance. (2) Unlike the MSMOVRPTW, which focuses solely on solving a single MOVRPTW formulation, the MTMO/DRL-AT introduces multitasking optimization. This means that the MTMO/DRL-AT can simultaneously solve multiple related optimization tasks, including the assisted task of the MOVRPTW. By incorporating multitasking optimization, valuable knowledge and insights gained from solving one task can be shared and utilized to improve the performance on other related tasks. This knowledge transfer and sharing contribute to the enhanced performance of the MTMO/DRL-AT compared to the LSMOVRPTW. (3) By combining attention models through DRL and multitasking optimization, the MTMO/DRL-AT offers a more robust and adaptive approach to solving the MOVRPTW. The attention models provide a finer grained focus on problem-specific details, while the multitasking optimization allows for the utilization of shared knowledge and insights across related tasks.

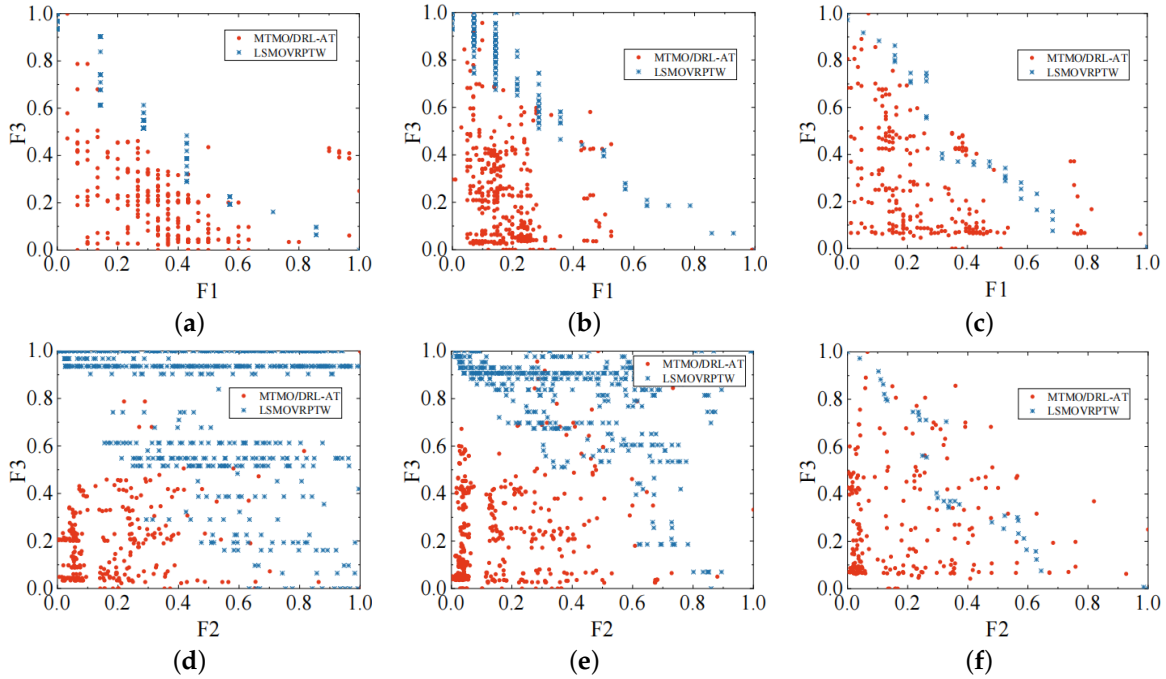

Figure 5. Distributions of the approximate Pareto fronts obtained by the MTMO/DRL-AT and LSMOVRPTW on the representative real-world instances: (**a**) 50-1-2 at f1-f3 plane; (**b**) 150-1-1 at f1-f3 plane; (**c**) 250-2-2 at f1-f3 plane; (**d**) 50-1-2 at f2-f3 plane; (**e**) 150-1-1 at f2-f3 plane; (**f**) 250-2-2 at f2-f3 plane.

6.3.2. Comparison with MMA-ALSC and HEMT

Two advanced approaches have recently been proposed to address the challenges of the MOVRPTW: the multiobjective memetic algorithm based on adaptive local search chains (MMA-ALSC) [27] and the hybrid evolutionary multitask algorithm (HEMT) [28]. The MMA-ALSC combines a multi-directional local search strategy with an enhanced local search chain technique. This allows for the search to be conducted in multiple directions in a chain-based way [27]. On the other hand, the HEMT takes a different approach by simultaneously considering multiple distinct MOVRPTWs within an evolutionary multitasking framework [28]. For this experiment, only the HEMT-hm5t variant is considered due to its promising performance. The comparisons between the MTMO/DRL-AT and MMA-ALSC (or HEMT) were conducted, and the results of the statistical tests are shown in Table 7.

Table 7. Results of the single- and multiple-problem analysis by the Wilcoxon test between the MTMO/DRL-AT and two recently proposed algorithms.

Algorithm	Metric	*w*/*t*/*l*	*R*+	*R*−	*p*-Value	$\alpha = 0.05$	$\alpha = 0.1$
MTMO/DRL-AT vs. MMA-ALSC	IGD	35/9/1	1014.0	21.0	0.0	Yes	Yes
	HV	45/0/0	1035.0	0.0	0.0	Yes	Yes
MTMO/DRL-AT vs. HEMT	IGD	40/5/0	990.0	0.0	0.0	Yes	Yes
	HV	45/0/0	1035.0	0.0	0.0	Yes	Yes

According to the results presented in Table 7, the MTMO/DRL-AT demonstrates superior performance compared to both the MMA-ALSC and HEMT across all instances. These findings provide a deeper understanding of the comparative performance of the algorithms: (1) In terms of the IGD, the MTMO/DRL-AT outperforms the MMA-ALSC on 35 instances and performs worse on only 1 instance. This indicates that the MTMO/DRL-AT consistently achieves better convergence and diversity in the obtained Pareto front solutions compared to the MMA-ALSC. The superior performance on the majority of instances suggests the effectiveness of the MTMO/DRL-AT in capturing a more diverse and high-

quality set of solutions. (2) In terms of the HV, the MTMO/DRL-AT significantly surpasses the MMA-ALSC on all 45 instances. The consistent superiority of the MTMO/DRL-AT over the MMA-ALSC in the HV demonstrates that the MTMO/DRL-AT can generate solutions that are both close to the true Pareto front and well-distributed across the problem space. (3) The results of the Wilcoxon test in the multiple-problem analysis indicate that the MTMO/DRL-AT outperforms the MMA-ALSC significantly in terms of both the IGD and HV. This statistical analysis strengthens the claim of the superior performance of the MTMO/DRL-AT compared to the MMA-ALSC. The significance of the difference further reinforces the effectiveness of the MTMO/DRL-AT in solving the MOVRPTW. (4) When compared to the HEMT, the MTMO/DRL-AT consistently exhibits strong performance on the majority of instances. The consistent strong performance suggests that the MTMO/DRL-AT outperforms the HEMT in terms of both the IGD and HV. This indicates that the MTMO/DRL-AT can generate a more diverse set of high-quality solutions compared to the HEMT.

Overall, the observations from these comparisons provide strong evidence that the MTMO/DRL-AT is a highly effective approach for solving the MOVRPTW. The superior performance over the MMA-ALSC and HEMT, as indicated by both the quantitative metrics and statistical analysis, highlights the advantage of the MTMO/DRL-AT in achieving better convergence, diversity, and solution quality.

6.3.3. Overall Comparisons

To assess the overall performance of the proposed algorithm, a comparison was conducted between the MTMO/DRl-AT and the above competing algorithms. The results of Friedman's test are summarized in Table 8.

Based on the results in Table 8, the MTMO/DRl-AT emerges as the top algorithm for both the IGD and HV, outperforming all other algorithms. The HEMT achieves the second-best ranking for the IGD, followed by the MMA-ALSC. In terms of the HV, the LSMOVRPTW achieves the second-best ranking, followed by the MMA-ALSC.

Moreover, when considering the characteristics of various MOVRPTW instances, several observations can be derived from the detailed numerical values presented in the Supplementary File. Firstly, it is evident that the MTMO/DRL-AT outperforms its competitors in terms of both the HV and IGD values for the instances with different customer sizes. This showcases the algorithm's strengths in terms of convergence and diversity. Secondly, the performance improvement achieved by the proposed algorithm is more significant in large-scale instances compared to small-scale ones. This can be attributed to the favorable initial solution provided by DRL.

In general, these results emphasize the competitive and exceptional performance of the proposed algorithm when compared to other state-of-the-art algorithms for the MOVRPTW.

Table 8. Average ranking values of the compared algorithms on all the instances.

	IGD		**HV**	
Algorithm	**Average Ranking**	**Final Ranking**	**Average Ranking**	**Final Ranking**
MTMO/DRL-AT	1.00	1	1.09	1
LSMOVRPTW	3.49	4	2.63	2
MMA-ALSC	2.81	3	3.00	3
HEMT	2.70	2	3.28	4

6.4. Impact of Main Components in MTMO/DRL-AT

In this section, we conducted additional experiments to address the following issues:

- Are the solutions generated by the trained models as initial solutions better for solving the MOVRPTW compared to randomly generated initial solutions?
- Can the knowledge transfer between the main and assisted tasks effectively enhance the performance of the MTMO/DRL-AT for the MOVRPTW?

- Can the local search phase further improve the performance of the MTMO/DRL-AT?

Each of the above issues will be explored and discussed in the subsequent subsections.

6.4.1. Effect Analysis of Initializing Population Using the Trained Models

To verify the effectiveness of initializing the population with the trained models, a variant of the MTMO/DRL-AT with a random initial population, denoted as the MTMO-AT, was considered for comparison. In the MTMO-AT, the population for both the main task and assisted task is initialized in a random manner, replacing the generated solutions by the trained models. The statistical comparison results between the MTMO/DRL-AT and MTMO-AT are given in Table 9.

From Table 9, we can find that the MTMO/DRL-AT outperforms the MTMO-AT significantly overall. Specifically, the MTMO/DRL-AT shows significant improvements over the MTMO-AT on 40 and 31 instances in terms of the IGD and HV, respectively, based on single-problem analysis using the Wilcoxon test. Moreover, the results of the multiple-problem analysis reveal that the MTMO/DRL-AT achieves a higher $R+$ than $R-$ with the p-values below 0.05 in both cases, indicating significant differences between the MTMO/DRL-AT and MTMO-AT for all the instances.

In general, the superior performance of the MTMO/DRL-AT compared to the MTMO-AT highlights the promising potential of DRL-based approaches in addressing multi-objective optimization problems. The results clearly indicate that leveraging deep reinforcement learning techniques can lead to significant improvements in solving complex multi-objective optimization tasks.

Table 9. Results of the single- and multiple-problem analysis by the Wilcoxon test between the MTMO/DRL-AT and MTMO-AT.

Algorithm	Metric	$w/t/l$	$R+$	$R-$	p-Value	$\alpha = 0.05$	$\alpha = 0.1$
MTMO/DRL-AT	IGD	40/5/0	1035.0	0.0	0.0	Yes	Yes
vs. MTMO-AT	HV	31/5/9	923.0	112.0	5.0×10^{-5}	Yes	Yes

6.4.2. Effect Analysis of Knowledge-Transfer Strategy

To evaluate the influence of the knowledge-transfer strategy on the performance of the MTMO/DRL-AT, a comparison was made between the MTMO/DRL-AT and its variant, the MTMO/DRL-AT_ST, which does not include the knowledge-transfer strategy. Unlike the proposed algorithm, the MTMO/DRL-AT_ST does not generate an assisted task for the main task, and there is no knowledge sharing between the main and assisted tasks during the transfer reproduction phase. Table 10 provides a statistical summary of the performance comparisons between the MTMO/DRL-AT and MTMO/DRL-AT_ST.

According to the results shown in Table 10, the MTMO/DRL-AT consistently exhibits better performance than the MTMO/DRL-AT_ST in terms of both the IGD and HV. To be specific, in terms of the IGD, the MTMO/DRL-AT achieves significant improvement over the MTMO/DRL-AT_ST on 17 instances, while it performs worse on 13 instances. In terms of the HV, the MTMO/DRL-AT outperforms the MTMO/DRL-AT_ST on 24 instances, but is outperformed by it on 7 instances. Additionally, the multiple-problem analysis reveals that the MTMO/DRL-AT obtains a higher $R+$ value than the $R-$ value in both the IGD and HV measures. Notably, the p-values indicate that the MTMO/DRL-AT performs significantly better than the MTMO/DRL-AT_ST in terms of the HV, at both α levels of 0.05 and 0.1.

Overall, these findings clearly demonstrate the efficacy of the knowledge-transfer strategy in improving the performance of the MTMO/DRL-AT. Additionally, the advantages of constructing an assisted task with a simpler search space are also validated. In general, these results highlight the benefits and effectiveness of integrating a knowledge-transfer strategy and utilizing a simplified search space in the MTMO/DRL-AT.

Table 10. Results of the single- and multiple-problem analysis by the Wilcoxon test between the MTMO/DRL-AT and MTMO/DRL-AT_ST.

Algorithm	Metric	*w*/*t*/*l*	$R+$	$R-$	p-Value	$\alpha = 0.05$	$\alpha = 0.1$
MTMO/DRL-AT	IGD	17/15/13	705.0	330.0	3.38×10^{-2}	No	Yes
vs. MTMO/DRL-AT_ST	HV	24/14/7	745.0	155.0	4.20×10^{-5}	Yes	Yes

6.4.3. Effect Analysis of Local Search Operators

To further evaluate the effectiveness of local searches for the proposed algorithm, a comparison was conducted between the MTMO/DRL-AT and its variant without local search phase, referred to as the MTMO/DRL-ATw/oLS. Unlike the proposed algorithm, the MTMO/DRL-ATw/oLS does not utilize the local search for additional optimization after the transfer reproduction phase. The comparison results between the MTMO/DRL-AT and its variant are presented in Table 11.

Table 11 clearly indicates that the MTMO/DRL-AT exhibits a significant advantage over the MTMO/DRL-ATw/oLS in overall performance. Specifically, based on the single-problem statistical analysis, the MTMO/DRL-AT significantly outperforms the MTMO/DRL-ATw/oLS on 27 instances for the IGD and 45 instances for the HV. The multiple-problem statistical analysis also reveals that the MTMO/DRL-AT obtains a higher $R+$ value than the $R-$ value compared to its variant. Furthermore, significant differences between these two variants are observed at both $\alpha = 0.05$ and $\alpha = 0.1$. Therefore, these results convincingly demonstrate the positive impact of the local searches in further enhancing the performance of the MTMO/DRL-AT when tackling the MOVRPTW.

Table 11. Results of the single- and multiple-problem analysis by the Wilcoxon test between the MTMO/DRL-AT and MTMO/DRL-ATw/oLS.

Algorithm	Metric	*w*/*t*/*l*	$R+$	$R-$	p-Value	$\alpha = 0.05$	$\alpha = 0.1$
MTMO/DRL-AT	IGD	27/7/11	903.0	232.0	1.25×10^{-3}	Yes	Yes
vs. MTMO/DRL-ATw/oLS	HV	45/0/0	1035.0	0.0	0.0	Yes	Yes

7. Conclusions and Future Work

In this study, we have proposed the MTMO/DRL-AT, a multi-task multi-objective evolutionary search algorithm based on deep reinforcement learning (DRL), for solving the MOVRPTW. Unlike traditional evolutionary algorithms, the MTMO/DRL-AT constructs an assisted task for the MOVRPTW with a simpler search space and simultaneously optimizes both the main and assisted tasks in a multitasking scenario. Additionally, attention models specifically designed for the subproblems of the MOVRPTW are incorporated, allowing for adaptation to instances of varying scales and providing high-quality initial solutions. Experimental studies on 45 real-world MOVRPTW instances have demonstrated the outstanding and competitive performance of the proposed algorithm.

In future work, our main focus will be on enhancing the DRL-based modeling and training process by incorporating more informative structural information extracted from problem instances. We also aim to explore effective strategies for leveraging the knowledge acquired from the assisted tasks to further improve the performance of the proposed algorithm. Additionally, we intend to conduct a thorough investigation into the impact of key parameters on the performance of the MTMO/DRL-AT. Lastly, we plan to extend the application of the MTMO/DRL-AT to solve other multi-objective combinatorial optimization problems.

Supplementary Materials: The following Supporting Information can be downloaded at https://www.mdpi.com/article/10.3390/sym16081030/s1.

Author Contributions: Conceptualization, Y.C. and P.L.; methodology, J.D. and X.W.; software, J.W. and X.W.; validation, J.D. and Y.C.; writing—original draft preparation, J.D. and J.W.; writing—review and editing, Y.C.; visualization, J.W.; supervision, P.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the Natural Science Foundation of Fujian Province of China (No. 2021J01318), the Fujian Provincial Science and Technology Major Project (No. 2020HZ02014), and the Quanzhou Science and Technology Major Project (No. 2021GZ1).

Data Availability Statement: Data are contained with the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; nor in the decision to publish the results.

References

1. Kallehauge, B.; Larsen, J.; Madsen, O.B.; Solomon, M.M. Vehicle routing problem with time windows. In *Column Generation*; Springer: Boston, MA, USA, 2005; pp. 67–98.
2. Braekers, K.; Ramaekers, K.; Nieuwenhuyse, I.V. The vehicle routing problem: State of the art classification and review. *Comput. Ind. Eng.* **2016**, *99*, 300–313. [CrossRef]
3. Mańdziuk, J. New Shades of the Vehicle Routing Problem: Emerging Problem Formulations and Computational Intelligence Solution Methods. *IEEE Trans. Emerg. Top. Comput. Intell.* **2019**, *3*, 230–244. [CrossRef]
4. Fathollahi-Fard, A.M.; Ahmadi, A.; Karimi, B. Multi-objective optimization of home healthcare with working-time balancing and care continuity. *Sustainability* **2021**, *13*, 12431. [CrossRef]
5. Mojtahedi, M.; Fathollahi-Fard, A.M.; Tavakkoli-Moghaddam, R.; Newton, S. Sustainable vehicle routing problem for coordinated solid waste management. *J. Ind. Inf. Integr.* **2021**, *23*, 100220. [CrossRef]
6. Baldacci, R.; Mingozzi, A.; Roberti, R. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *Eur. J. Oper. Res.* **2012**, *218*, 1–6. [CrossRef]
7. Braeysy, O.; Gendreau, M. Vehicle Routing Problem with Time Windows, Part II: Metaheuristics. *Transp. Sci.* **2005**, *39*, 119–139. [CrossRef]
8. Dixit, A.; Mishra, A.; Shukla, A. Vehicle Routing Problem with Time Windows Using Meta-Heuristic Algorithms: A Survey. In *Harmony Search and Nature Inspired Optimization Algorithms*; Advances in Intelligent Systems and Computing; Springer: Singapore, 2019; Volume 741, pp. 539–546.
9. Gutiérrez, J.; Landa-Silva, D.; Moreno-Pérez, J. Nature of real-world multi-objective vehicle routing with evolutionary algorithms. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Anchorage, AK, USA, 9–12 October 2011; pp. 257–264.
10. Zhou, Y.; Wang, J. A Local Search-Based Multiobjective Optimization Algorithm for Multiobjective Vehicle Routing Problem with Time Windows. *IEEE Syst. J.* **2017**, *9*, 1100–1113. [CrossRef]
11. Sun, Y.; Yen, G.G.; Yi, Z. IGD indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Trans. Evol. Comput.* **2019**, *23*, 173–187. [CrossRef]
12. Li, K.; Zhang, T.; Wang, R. Deep reinforcement learning for multiobjective optimization. *IEEE Trans. Cybern.* **2020**, *51*, 3103–3114. [CrossRef] [PubMed]
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1–9. [CrossRef]
14. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
15. Li, J.; Monroe, W.; Ritter, A.; Galley, M.; Gao, J.; Jurafsky, D. Deep reinforcement learning for dialogue generation. *arXiv* **2016**, arXiv:1606.01541.
16. Bello, I.; Pham, H.; Le, Q.V.; Norouzi, M.; Bengio, S. Neural combinatorial optimization with reinforcement learning. *arXiv* **2016**, arXiv:1611.09940.
17. Zhao, J.; Mao, M.; Zhao, X.; Zou, J. A hybrid of deep reinforcement learning and local search for the vehicle routing problems. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 7208–7218. [CrossRef]
18. Wang, L.; Pan, Z. Scheduling optimization for flow-shop based on deep reinforcement learning and iterative greedy method. *Control Decis.* **2021**, *36*, 2609–2617.
19. Zhang, Y.; Wang, J.; Zhang, Z.; Zhou, Y. MODRL/D-EL: Multiobjective deep reinforcement learning with evolutionary learning for multiobjective optimization. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8.
20. Tang, K.; Yao, X. Learn to Optimize-A Brief Overview. *Natl. Sci. Rev.* **2024**, *11*, nwae132. [CrossRef] [PubMed]

21. Gupta, A.; Ong, Y.S.; Feng, L. Multifactorial evolution: Toward evolutionary multitasking. *IEEE Trans. Evol. Comput.* **2015**, *20*, 343–357. [CrossRef]
22. Ong, Y.S. Towards evolutionary multitasking: A new paradigm in evolutionary computation. In *Computational Intelligence, Cyber Security and Computational Models*; Springer: Singapore, 2016; pp. 25–26.
23. Feng, L.; Zhou, L.; Gupta, A.; Zhong, J.; Zhu, Z.; Tan, K.; Qin, K. Solving Generalized Vehicle Routing Problem with Occasional Drivers via Evolutionary Multitasking. *IEEE Trans. Cybern.* **2021**, *51*, 3171–3184. [CrossRef] [PubMed]
24. Feng, L.; Huang, Y.; Zhou, L.; Zhong, J.; Gupta, A.; Tang, K.; Tan, K.C. Explicit Evolutionary Multitasking for Combinatorial Optimization: A Case Study on Capacitated Vehicle Routing Problem. *IEEE Trans. Cybern.* **2021**, *51*, 3143–3156. [CrossRef]
25. Qi, Y.; Hou, Z.; Li, H.; Huang, J.; Li, X. A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows. *Comput. Oper. Res.* **2015**, *62*, 61–77. [CrossRef]
26. Moradi, B. The new optimization algorithm for the vehicle routing problem with time windows using multi-objective discrete learnable evolution model. *Soft Comput.* **2020**, *24*, 6741–6769. [CrossRef]
27. Zhang, K.; Cai, Y.; Fu, S.; Zhang, H. Multiobjective memetic algorithm based on adaptive local search chains for vehicle routing problem with time windows. *Evol. Intell.* **2022**, *15*, 2283–2294. [CrossRef]
28. Cai, Y.; Cheng, M.; Zhou, Y.; Liu, P.; Guo, J.M. A hybrid evolutionary multitask algorithm for the multiobjective vehicle routing problem with time windows. *Inf. Sci.* **2022**, *612*, 168–187. [CrossRef]
29. Li, B.; Li, J.; Tang, K.; Yao, X. Many-objective evolutionary algorithms: A survey. *ACM Comput. Surv. (CSUR)* **2015**, *48*, 1–35. [CrossRef]
30. Vinyals, O.; Fortunato, M.; Jaitly, N. Pointer networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9.
31. Nazari, M.; Oroojlooy, A.; Snyder, L.; Takác, M. Reinforcement learning for solving the vehicle routing problem. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1–13.
32. Nowak, A.; Villar, S.; Bandeira, A.S.; Bruna, J. A note on learning algorithms for quadratic assignment with graph neural networks. *Stat* **2017**, *1050*, 22.
33. Deudon, M.; Cournut, P.; Lacoste, A.; Adulyasak, Y.; Rousseau, L.M. Learning heuristics for the tsp by policy gradient. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 15th International Conference, CPAIOR 2018, Delft, The Netherlands, 26–29 June 2018, Proceedings*; Springer: Cham, Switzerland, 2018; pp. 170–181.
34. Kool, W.; Van Hoof, H.; Welling, M. Attention, learn to solve routing problems! *arXiv* **2018**, arXiv:1803.08475.
35. Peng, B.; Wang, J.; Zhang, Z. A deep reinforcement learning algorithm using dynamic attention model for vehicle routing problems. In *Artificial Intelligence Algorithms and Applications: 11th International Symposium, ISICA 2019, Guangzhou, China, 16–17 November 2019, Revised Selected Papers*; Springer: Singapore, 2020; pp. 636–650.
36. Wu, H.; Wang, J.; Zhang, Z. MODRL/D-AM: Multiobjective deep reinforcement learning algorithm using decomposition and attention model for multiobjective optimization. In *Artificial Intelligence Algorithms and Applications: 11th International Symposium, ISICA 2019, Guangzhou, China, 16–17 November 2019, Revised Selected Papers*; Springer: Singapore, 2020; pp. 575–589.
37. Zhou, L.; Feng, L.; Zhong, J.; Ong, Y.S.; Zhu, Z.; Sha, E. Evolutionary multitasking in combinatorial search spaces: A case study in capacitated vehicle routing problem. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016; pp. 1–8.
38. Liu, M.; Wang, Z.; Li, J. A deep reinforcement learning algorithm for large-scale vehicle routing problems. In Proceedings of the International Conference on Electronic Information Technology (EIT 2022), Chengdu, China, 18–20 March 2022; Volume 12254, pp. 824–829.
39. Zhang, Q.; Hui, L. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.* **2008**, *11*, 712–731. [CrossRef]
40. Das, I.; Dennis, J.E. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J. Optim.* **1998**, *8*, 631–657. [CrossRef]
41. Grondman, I.; Busoniu, L.; Lopes, G.A.; Babuska, R. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2012**, *42*, 1291–1307. [CrossRef]
42. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
43. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
44. Storn, R.; Price, K. Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
45. Liu, B.; Wang, L.; Jin, Y.H. An effective PSO-based memetic algorithm for flow shop scheduling. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2007**, *37*, 18–27. [CrossRef] [PubMed]
46. Coello, C.A.C.; Sierra, M.R. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In Proceedings of the Mexican International Conference on Artificial Intelligence, Mexico City, Mexico, 26–30 April 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 688–697.
47. Zitzler, E.; Thiele, L. Multiobjective optimization using evolutionary algorithms—A comparative case study. In Proceedings of the International Conference on Parallel Problem Solving from Nature, Amsterdam, The Netherlands, 27–30 September 1998; Springer: Berlin/Heidelberg, Germany, 1998; pp. 292–301.

48. Alcalá-Fdez, J.; Sanchez, L.; Garcia, S.; del Jesus, M.J.; Ventura, S.; Garrell, J.M.; Otero, J.; Romero, C.; Bacardit, J.; Rivas, V.M.; et al. KEEL: A software tool to assess evolutionary algorithms for data mining problems. *Soft Comput.* **2009**, *13*, 307–318. [CrossRef]
49. García, S.; Fernández, A.; Luengo, J.; Herrera, F. A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability. *Soft Comput.* **2009**, *13*, 959–977. [CrossRef]
50. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [CrossRef]