Hierarchical Reinforcement Learning with Topology-Aware Exploration Framework for Multi-path Commodity Flow Problem

Jingchen Jiang[1], Xuan Zhou[1*], Jiayuan Li[1,2], Geng Han[1], Xiang Shi[3†], Fang Deng[1]
[1]Beijing Institute of Technology
[2]Zhongguancun Academy
[3]Tsinghua University
{3120215447, 3120225455, lijiayuan, 3120215446, dengfang}@bit.edu.cn, shi-xiang@tsinghua.edu.cn

Abstract

The multi-path commodity flow problem (MPCFP) is crucial for ensuring reliable and high-speed data transmission in communication networks. However, existing studies that employ pre-generated routing paths neglect real-time load state and the coupling among decisions, thus hindering the achievement of high-quality solutions. To overcome this, we propose **H**ierarchical **R**einforcement **L**earning with **T**opology-**A**ware **E**xploration (HRL-TAE), which is the first fully end-to-end framework that dynamically produces high-quality solutions based on real-time network states. HRL-TAE integrates an exploration mechanism and utilizes the **S**tate **T**ransition **G**uiding **L**ist (STGL) to guide state transitions, thereby transforming topology exploration into a Markov decision process. Guided by STGL, two closely coupled layers in HRL-TAE, that is, the path construct layer and the ratio allocate layer, construct multiple subpaths for each flow and allocate traffic ratios among them. Subsequently, adaptive constraint-driven masks exclude infeasible actions during decision making, thereby guaranteeing that all constraints are satisfied. We also adopt a tailored training approach to obtain accurate gradient estimates and improve training efficiency. Simulations and real-world experiments demonstrate that HRL-TAE achieves superior performance.

Introduction

In recent years, communication networks are advancing driven by the explosive growth of network services. With devices and user demands increasing dramatically, there is an urgent need for efficient network resource allocation and traffic management strategies (Galliera 2024; Berger et al. 2025). In this context, the Multi-Path Commodity Flow Problem (MPCFP), a powerful tool for addressing network scheduling, has gained increasing attention. It involves transporting multiple commodities through a network while respecting resource limitations, such as capacity (Fukugami 2023; Sui et al. 2024). Moreover, high-quality solutions to MPCFP ensure efficient and reliable data transmission within the network, and are suitable for addressing the diverse demands of communication networks.

Currently, many researchers address MPCFP in communication scenarios using heuristic algorithms, evolutionary computation, or linear programming for solutions. Gomes et al. (2024) propose an artificial algae algorithm to solve MPCFP in the embedding problem of 5G network, and find the real physical paths between network nodes. Zhang et al. (2021) use game theory to transform the problem into a Nash equilibrium problem, and improve the communication efficiency of the network. Farrugia et al. (2023) design an ERA algorithm that considers all traffic and link capacity on the network, and improves the throughput and latency of the network. Sui et al. (Sui et al. 2024) apply the CSO algorithm with a problem characteristic-based crossover operator to optimize bandwidth allocation, thereby enhancing network load balancing. However, these algorithms face difficulties in balancing the quality and efficiency of the solution (Tessler et al. 2022). Learning-based algorithms, on the other hand, have the potential to achieve a better balance by maintaining quality while reducing computational time (Zuo et al. 2018; Jiang et al. 2024).

With the advancement of neural network research, many scholars have also applied learning-based algorithms to solve similar problems in communication network scenarios. Xu et al. (2023b) propose the TEAL algorithm, which uses RL to allocate flows across multiple subpaths to achieve a better distribution and solve MPCFP. Casas et al. (2022) introduce the DRSIR algorithm to make routing decisions, considering path metrics to adapt dynamic traffic changes without prior knowledge of the underlying network. Liu et al. (2024) employ a distributed training deep learning algorithm, which implements routing path selection for network aggregation. Lu et al. (2023) apply the SAMP algorithm to handle dynamic traffic changes and redundant links in data center networks, thus improving the efficiency of data transmission. However, existing algorithms pre-generate flow paths rely on algorithms such as K-Shortest Path (KSP), and use learning-based modules only for traffic allocation. Such strategies ignore the real-time network load state and the coupling among decisions for different flows, thereby producing suboptimal solutions.

To overcome the aforementioned limitations, we propose HRL-TAE, a fully end-to-end approach with two-layer network for MPCFP. In HRL-TAE, the path construct layer constructs subpaths for each flow, and the ratio allocate

*These authors contributed equally.
†Corresponding author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

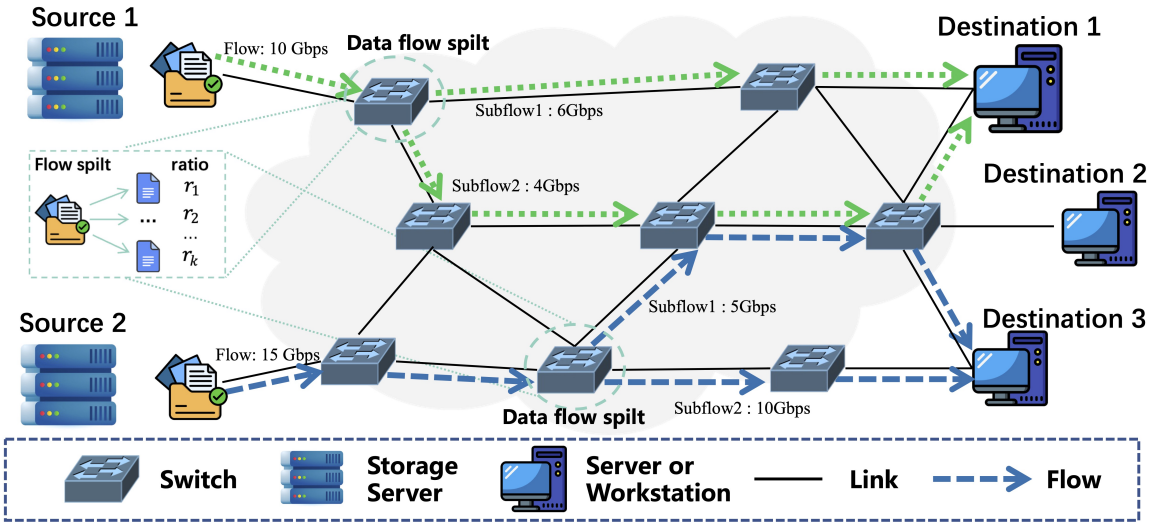

Figure 1: Schematic Diagram of MPCFP within Communication Networks Scenario

layer allocates traffic ratios among them. Driven by our proposed Tree-Like Exploration Process (TLEP), the agent can achieve topology awareness, exploration, and decision-making. Throughout the process, STGL is dynamically updated to guide state transitions and information observation. Thus, HRL-TAE addresses the two critical sub-tasks in a more adaptive way, which improves overall performance. The contribution of this paper lies in the following aspects:

1. To the best of our knowledge, the proposed HRL-TAE framework represents the first fully end-to-end solution that solves MPCFP by addressing its two coupled sub-problems: path construction and ratio allocation. HRL-TAE employs the two-layer network to make respective decisions, which enhances the solution quality and demonstrates a novel paradigm for similar problems.

2. A unique TLEP rule is designed to facilitate agents in topology awareness, exploration and decision making. During exploration, the dynamically updated STGL is used to guide state transitions and information observations, thereby transforming MPCFP into Markov Decision Process (MDP).

3. The adaptive constraint-driven mask mechanism is developed to eliminate infeasible actions during exploration, guaranteeing strict adherence to problem constraints without performing heuristic repair for infeasible ones.

4. The Quadruple Collaboration Training (QCT) method is proposed to train the two layers jointly. By recombining the training and baseline networks, it obtains accurate gradients, thereby tackling the credit assignment and enhancing training efficiency.

Problem Description

Fig. 1 illustrates a typical MPCFP scenario within communication networks. The switches correspond to the nodes in graph, the links between switches represent the edges, and the data flows are equivalent to different commodity flows. In network systems, controllers forward data flows based on routing schemes. Data flows, which originate from the source nodes, split the bandwidth into several subflows according to the allocated ratios. These subflows are transmitted along corresponding paths and ultimately reach the destination nodes. During transmission, data flows cannot reuse the same link, and must avoid exceeding link capacity. The algorithm needs to generate a routing scheme that includes K subpaths for each flow and the corresponding allocation ratios. MPCFP has proved to be NP-hard (Barnhart, Hane, and Vance 2000; Masri, Krichen, and Guitouni 2019), and to provide a clearer overview, we present a mathematical description of the problem.

The network is modeled as a graph $\mathcal{G} = \{N, E, C\}$, where N indicates the nodes, $E(i,j)$ is the edges and $C(i,j)$ is the edge capacity, where $i, j, \in N, i \neq j$. The q-th commodity flow $f_q = \{S_q, D_q, B_q\}$, which indicates that a commodity with demand of B_q needs to be transported from the source node S_q to the destination node D_q. In MPCFP, the inputs are the network topology $\mathcal{G}$ and the commodity flows $F = \{f_1, ..., f_Q\}$. The objective is to find subpaths for each commodity, allocate ratios among subpaths (i.e. $P_{q,k}$ and $u_{q,k}$), and minimize the cost. The path $P_{q,k}$ is defined as a sequence of edges, and $p_q^k(i,j)$ serves as a Boolean variable, indicating whether the k-th subflow of flow q traverses edge $E(i,j)$. The main constraints involved are as follows:

Edge Capacity Limitation

$$\sum_{q \in Q, k \in K} (p_q^k(i,j) \times u_{q,k}) \leq C(i,j), \forall i, j \in N \quad (1)$$

Flow Split Conservation

$$\sum_{i \in N} p_q^k(i,j) = \sum_{z \in N} p_q^k(j,z), \quad (2)$$
$$\forall j \in N \backslash \{s_q, d_q\}, \forall q \in Q, \forall k \in K$$

Commodity Demand Constraints

$$\sum_{k \in K} u_{q,k} = B_q, \forall q \in Q \quad (3)$$

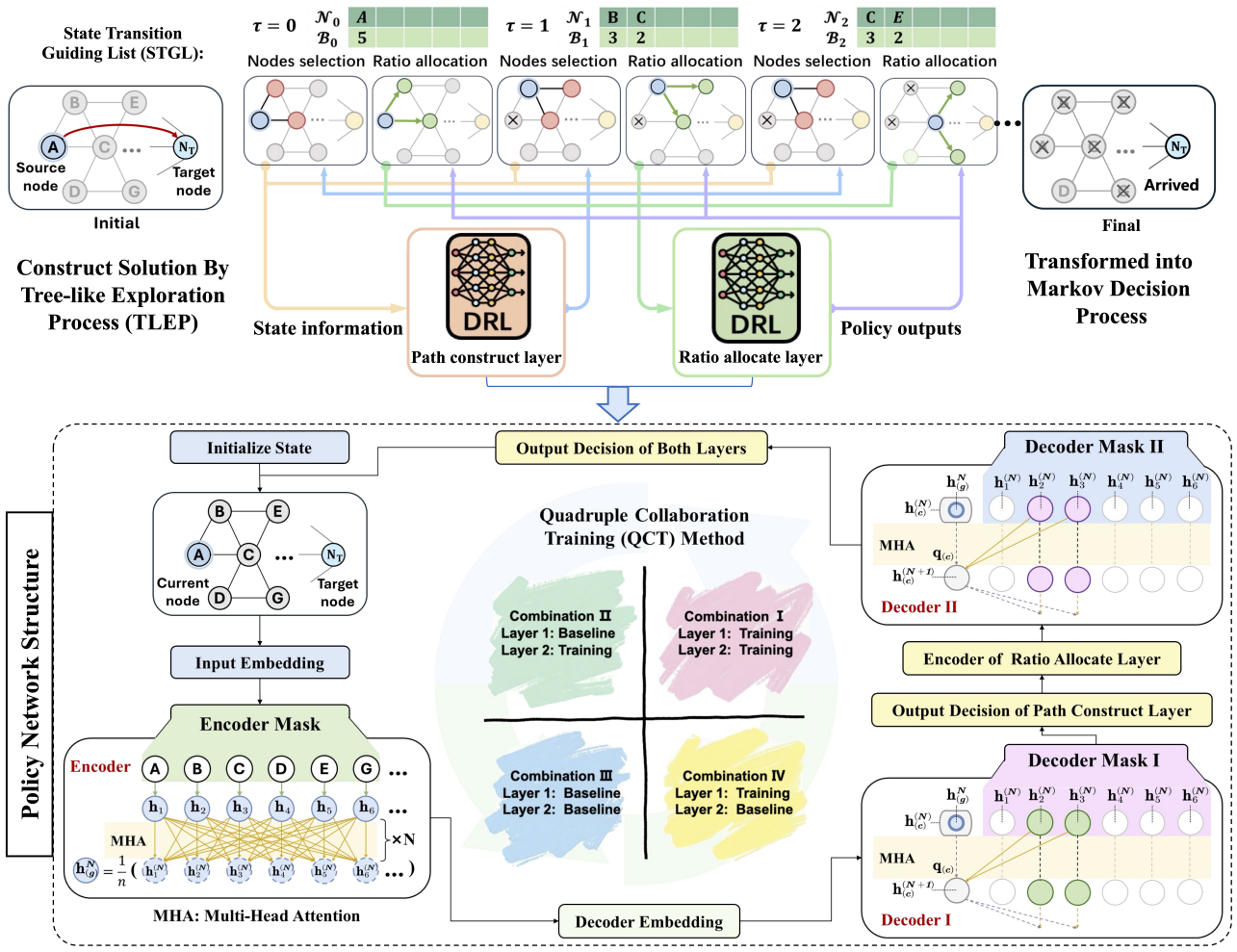

Figure 2: Architecture of HRL-TAE

In this paper, we define the minimization of Maximum Link Utilization (MLU) as the optimization objective, which is a classical indicator in network optimization to measure load balancing (Chen et al. 2024; Zhang et al. 2020; Jiang et al. 2025). MLU can be calculated as Eq. (4):

$$MLU = max(\frac{\sum_{q\in Q,k\in K} p_q^k(i,j) * u_{q,k}}{C(i,j)}), \forall i,j \in N \quad (4)$$

Hierarchical Reinforcement Learning with Topology-Aware Exploration

The dynamic nature of communication network poses significant challenges in obtaining optimal routing schemes. Most existing algorithms pre-generate paths by using KSP, which may neglect real-time load states and the coupling among decisions for different flows, thus leading to suboptimal solutions. Moreover, existing studies lack effective inference and training schemes, thus leaving the development of an end-to-end framework in urgent need.

Thereby, we propose the fully end-to-end HRL-TAE framework to solve MPCFP, as shown in Fig. 2. However, using RL to make decisions on creating routing schemes requires the problem to exhibit the Markov property. In response to the requirement, we introduce TLEP to facilitate the topology awareness and exploration by agent. During the process, STGL is dynamically updated to guide state transitions and information observation. These mechanisms transform the problem into an MDP that meets the requirement.

To reduce decision complexity, the problem is decomposed into two coupled parts: path construction and ratio allocation, which are addressed by two layers in HRL-TAE. At each step, the path construct layer selects one or more nodes for future visits, and the ratio allocate layer allocates traffic ratios among them. Once exploration is complete, the selected nodes are sorted in sequence to form subpaths, and the allocated ratios become the ratios between subpaths. Meanwhile, to filter infeasible actions during decision-making and ensure all solutions meet the constraints, an adaptive constraint-driven mask mechanism is designed.

In addition, we adopt QCT method to jointly train the two-layer network. It recombines the training and baseline network to obtain accurate gradient estimates, thereby reducing

learning fluctuations and accelerating convergence.

Tree-Like Exploration Process

To transfer the problem into an MDP and enable RL agents to comprehensively explore and be aware of the network topology, we introduce TLEP. During multiple subpaths are explored, an auxiliary list, STGL, is employed to guide state transitions and information observation. The general process of TLEP is shown in Fig. 3.

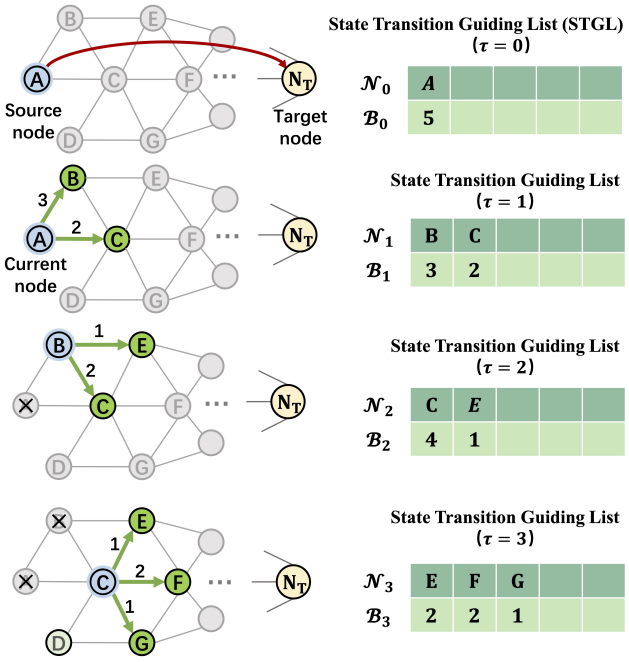

Figure 3: Tree-like Exploration Process

For each flow at step t_τ, STGL includes two-dimensional information $\mathcal{L}_\tau = (\mathcal{N}_\tau, \mathcal{B}_\tau)$, where $\mathcal{N}_\tau = \{n_1^\tau, ..., n_k^\tau\}$ is the IDs of nodes that will visit in the future, and $\mathcal{B}_\tau = \{b_1^\tau, ..., b_k^\tau\}$ is the quantities that are allocated to those nodes. The upper limit of $\mathcal{L}_\tau$ length is K, indicating that a maximum of K subflows are allowed to exist simultaneously.

Initially, STGL contains only source nodes S_q and quantity B_q. For t_τ, agent moves to the node at the first position in STGL, i.e., the agent's current position corresponds to $X_\tau = n_1^\tau$, and the commodity quantity to be processed is $Q_\tau = b_1^\tau$. The state information is fed to the two-layer network in HRL-TAE, whose outputs determine the next nodes to visit $V_\tau = \{v_1^\tau, ...v_{k'}^\tau\}$, and the allocated ratios between these nodes $R_\tau = \{r_1^\tau, ...r_{k'}^\tau\}$, where $k' \leq K - k$ to ensure that STGL will not overflow. Thus, the quantities $Q'_\tau = \{q_1^\tau, ..., q_{k'}^\tau\}$ from X_τ to selected nodes V_τ can be calculated by Eq. (5).

$$q_\sigma^\tau = \frac{r_\sigma^\tau \times Q_\tau}{\sum_1^{i=k'} r_i^\tau}, \sigma \in [1, k'] \tag{5}$$

After HRL-TAE outputs the decisions on node selection and ratio allocation, STGL guides the state transition in the following ways. First, STGL clears the data at the first position and shifts the subsequent data forward. Second, the current node X_τ is marked as reached, ensuring it won't be reselected, thereby avoiding the formation of path loops. Third, the remaining capacity of the edges passed through minus the corresponding Q'_τ, as Eq. (6).

$$C(X_\tau, v_\sigma^\tau)' = C(X_\tau, v_\sigma^\tau) - q_\sigma^\tau, \sigma \in [1, k'] \tag{6}$$

Fourth, the decisions are incorporated into STGL. For each node $v_i^\tau \in V_\tau$, check if there exists a matching $n_j^\tau \in \mathcal{N}_\tau$. If found, set $b_j^\tau = b_j^\tau + q_i^\tau$; otherwise, append v_i^τ and q_i^τ to the end of STGL. Finally, STGL is sorted based on the distance from each n_j^τ to D_q, resulting in the new STGL $\mathcal{L}_{\tau+1}$. In the new list, the first element $n_1^{\tau+1}$ is farthest from D_q, and it also indicates the agent's position $X_{\tau+1}$ for the next step $\tau + 1$. As per the TLEP, the agent repeatedly explores and makes decisions until only D_q remains in STGL, indicating the exploration is accomplished.

Elements and Network Structure

When applying RL to make decisions, the problem must exhibit the Markov property (Sutton, Barto et al. 1998). By utilizing TLEP and STGL outlined above, MPCFP can be transformed into an MDP that meets the requirement. The key elements for HRL-TAE are as follows:

- State: The state information includes the network topology, remaining capacity of edges, current position of flow, destination node, and STGL list.
- Action: At each step, the action of the path construct layer is to select the next nodes to visit, while the ratio allocate layer allocates the traffic ratios among these nodes.
- Transitions: State transitions are deterministic. Once the next nodes and allocated ratios are determined, the state is updated according to TLEP rules.
- Reward: Reward is provided until current flow is fully routed, both two layers will receive an identical and undiscounted reward. The reward is defined as the negative of MLU, incentivizing HRL-TAE to generate paths and allocate ratios that promote a balanced load distribution.

Both layers in HRL-TAE adopt the similar structure as shown in Fig. 4. When making a decision, the state information is fed into the path construct layer, which selects the next nodes to visit. Subsequently, the selected nodes and the state information passed to the ratio allocate layer, which then allocates ratios among the nodes. Together, the decisions made by the two layers form the output of HRL-TAE.

Adaptive Constraint-Driven Mask Mechanism

To ensure that all solutions satisfy the problem constraints, infeasible actions are filtered out at decision time. Meanwhile, maintaining specific infeasible actions in the encoder enhances the understanding of the underlying associations between states. This leads to a better comprehension of the relative value of actions in different states, thereby improving decision-making quality. Therefore, an adaptive constraint-driven mask mechanism is proposed.

Specifically, in the path construct layer, the encoder mask blocks all nodes that have already been visited, while the decoder is allowed to select only the unvisited neighbors of the

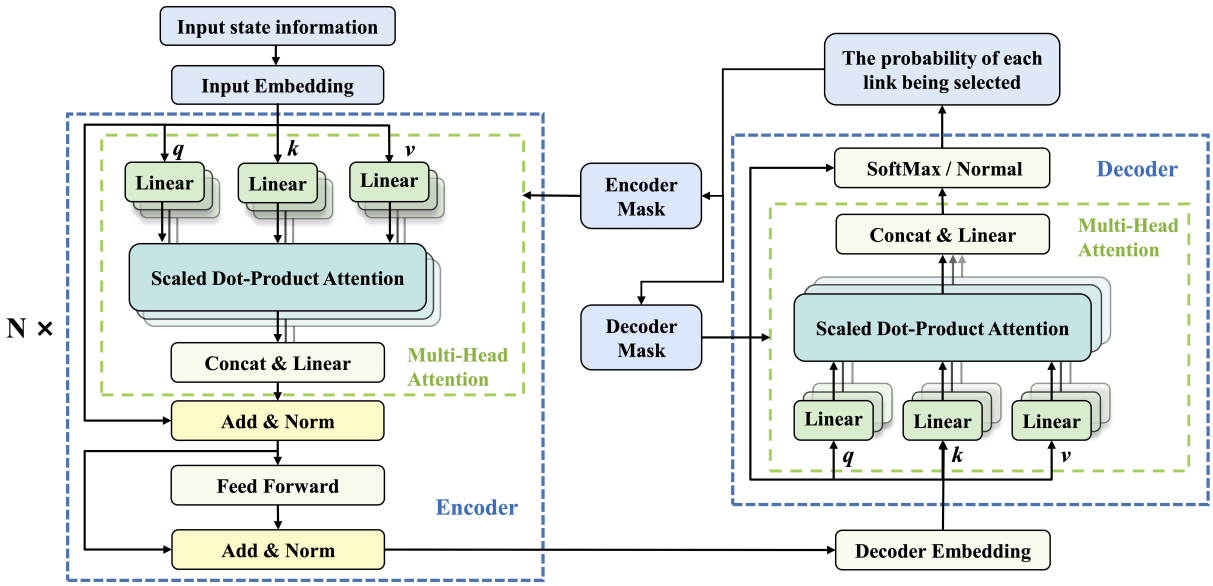

Figure 4: Structure of single layer network

current node. In the ratio allocate layer, the encoder mask remains identical to that of the path construct layer; the decoder, however, is restricted to the nodes selected by path construct layer. The adaptive constraint-driven mask mechanism is shown in Fig. 5, in which encoder and decoder I for path construct layer, while II for ratio allocate layer.

Training Method

In HRL-TAE, the two-layer network is trained using synthetic datasets that are randomly generated based on the connectivity patterns of each topology. At each step, the path construct layer outputs selection probabilities for each node, while the ratio allocate layer outputs the weight ratio of the normal distribution. The training of both layers employs the REINFORCE algorithm with a rollout baseline, where the baseline is greedily updated after each batch. The training process entails 1000 epochs, each comprising 10 batches of size 128. We utilize the Adam optimizer with a dynamically decaying learning rate that begins at $lr = 0.0001$.

The credit assignment problem is a vital challenge in HRL. As rewards depend on decisions from multiple layers, it is difficult to judge whether a single layer's decision is beneficial or not. If not addressed, it will lead to fluctuations and reduced efficiency (Pateria et al. 2021; Zhou et al. 2024; Vezhnevets et al. 2017; Xu et al. 2023a; Gao et al. 2024).

To provide a universal training method to address this challenge and enhance training efficiency, we propose the QCT method. Inspired by the concept of *control variates method* in scientific experiments, In QCT, the training networks and baseline networks of two layers are combined to form four different combinations, as shown in Fig. 6. These combinations are utilized to make decisions and calculate rewards independently. Taking the training of path construct layer's neural parameters as an example, the loss value can be calculated using Eq. (7).

$$loss = -\sum_{i=1}^{n}(\mathcal{R}(\pi_i^{\mathrm{I}}) - \mathcal{R}(\pi_i^{\mathrm{II}}))logP_\theta(\pi_i^{\mathrm{I}}) - \sum_{i=1}^{n}(\mathcal{R}(\pi_i^{\mathrm{IV}}) - \mathcal{R}(\pi_i^{\mathrm{III}}))logP_\theta(\pi_i^{\mathrm{IV}}) \quad (7)$$

where, π_i^{I} indicates the solution output by using combination I in Fig. 6, while $\mathcal{R}(\pi)$ represents the obtained rewards corresponding to the solution π.

Moreover, upon completing the training of each batch, it is imperative to determine whether to update the baseline network. Similarly, directly comparing rewards $\mathcal{R}(\pi_i^{\mathrm{I}})$ and $\mathcal{R}(\pi_i^{\mathrm{III}})$ is inadequate to accurately assess a single-layer network. In HRL-TAE, for path construct layer, if $\mathcal{R}(\pi_i^{\mathrm{IV}})$ outperforms $\mathcal{R}(\pi_i^{\mathrm{III}})$ and passes a one tailed T-test at the 5% significance level, the baseline network will be updated to synchronize with the training network.

Experiments

In this section, experiments are conducted to verify the effectiveness of HRL-TAE and its components. To ensure experimental rigor, all experiments are conducted on a computer equipped with a 9th Intel Core i9 processor, an NVIDIA GeForce GTX1660Ti GPU, and 32GB of RAM, using Python 3.10. In all experiments, the test data are randomly generated according to the connectivity patterns of each topology, with each case consisting of 20 instances for decision-making.

Simulation Experiments

Simulation experiments are conducted in four real network topologies (Abilene: 11 nodes, 14 links; ATT: 25 nodes, 56 links; Geant: 37 nodes,56 links; DFN: 58 nodes, 87 links),

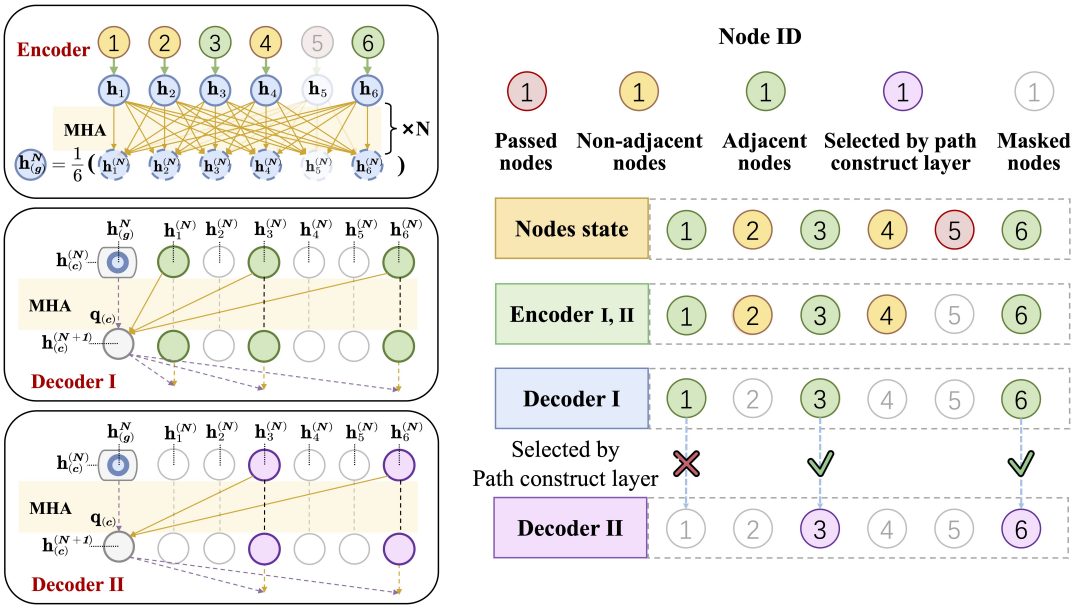

Figure 5: Adaptive Constraint-Driven Mask Mechanism

Case	Topo.	HRL-TAE		TEAL		ECMP		DRSIR	
		Obj.	Time(s.)	Obj.	Time(s.)	Obj.	Time(s.)	Obj.	Time(s.)
1	Abilene	**0.346**	0.12	0.368	0.16	0.411	0.04	0.400	0.13
2	Abilene	**0.376**	0.13	0.390	0.13	0.414	0.03	0.422	0.15
3	Abilene	**0.385**	0.13	0.405	0.14	0.434	0.03	0.425	0.15
4	Abilene	**0.383**	0.11	0.420	0.14	0.427	0.03	0.412	0.16
5	Abilene	**0.343**	0.11	0.367	0.13	0.400	0.03	0.396	0.15
6	ATT	**0.356**	0.23	0.388	0.32	0.432	0.04	0.416	0.13
7	ATT	**0.342**	0.21	0.372	0.32	0.410	0.04	0.396	0.16
8	ATT	**0.355**	0.17	0.395	0.29	0.385	0.04	0.401	0.15
9	ATT	**0.315**	0.18	0.363	0.28	0.376	0.04	0.377	0.15
10	ATT	**0.344**	0.20	0.391	0.31	0.415	0.04	0.410	0.15
11	Geant	**0.416**	0.17	0.438	0.55	0.503	0.04	0.438	0.13
12	Geant	**0.425**	0.26	0.449	0.54	0.511	0.04	0.443	0.16
13	Geant	**0.405**	0.22	0.445	0.53	0.483	0.04	0.442	0.14
14	Geant	**0.408**	0.26	0.435	0.50	0.487	0.03	0.445	0.15
15	Geant	**0.417**	0.17	0.420	0.57	0.484	0.03	0.450	0.15
16	DFN	**0.406**	0.12	0.435	1.13	0.500	0.03	0.438	0.13
17	DFN	**0.409**	0.20	0.470	1.07	0.507	0.04	0.445	0.16
18	DFN	**0.383**	0.20	0.398	1.13	0.463	0.03	0.424	0.15
19	DFN	**0.412**	0.17	0.441	1.19	0.472	0.03	0.420	0.14
20	DFN	**0.406**	0.18	0.425	1.12	0.504	0.04	0.455	0.14

Table 1: Results of Simulation Experiments

which are obtained from the Internet topology zoo data set (Knight et al. 2011). The topologies are shown in Fig. 7.

Three different comparison algorithms are adopted to verify the effectiveness of HRL-TAE:

1.TEAL: Xu et al. (2023b) introduce TEAL, this algorithm pre-generates paths by KSP and allocates the flows to subpaths by a deep reinforcement learning network.

2.ECMP: It is a simple, fast, and effective rule that constructs K subpaths for each flow, and evenly allocates flow across these paths (Zhang et al. 2020; Li et al. 2022).

3.DRSIR: Casas et al. (2022) propose DRSIR, which use DRL network to select paths and allocate flows among pre-generated in software defined network.

Table 1 presents the results and running times for all algorithms across different cases. As shown in the table, HRL-TAE demonstrates superior performance with comparable execution time across diverse cases. Specifically, compared with the TEAL, ECMP, and DRSIR algorithms, the routing

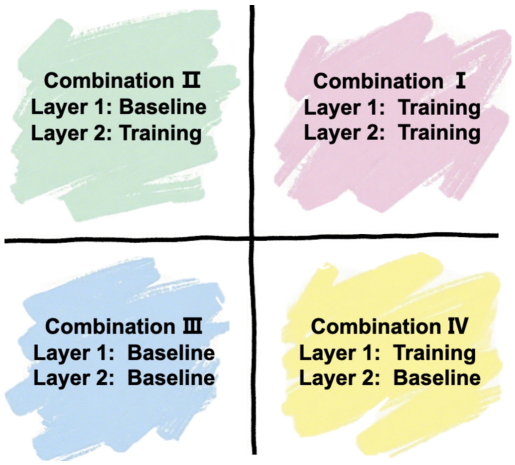

Figure 6: Combinations for QCT (For example, in Combination II, the path construct layer adopts the baseline network, while the ratio allocate layer adopts the training network.)

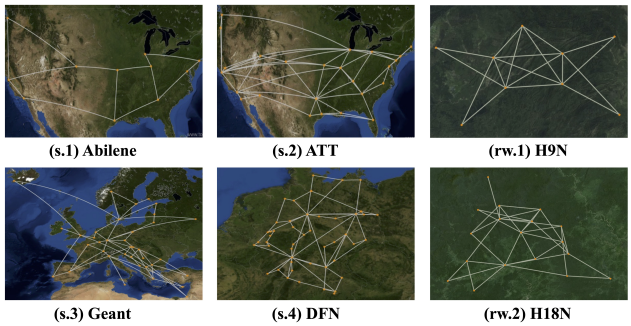

Figure 7: Different Topologies for Experiment (Abilene, ATT, Geant and DFN for simulation experiments; H9N and H18N for real-world experiments)

schemes generated by HRL-TAE achieve average improvements of 7.08%, 15.35%, and 9.74%, respectively. These results confirm that HRL-TAE has a performance advantage across varying scenarios.

Ablation Experiments

To verify the effectiveness of the components proposed for HRL-TAE, ablation experiments are conducted. In ablation experiments, two comparison algorithms are adopted: Abl.1: Adopting HRL-TAE framework, while QCT is not employed. Instead, the basic REINFORCE algorithm is utilized to update both two layers together as Eq. (8).

$$loss = -\sum_{i=1}^{n}(\mathcal{R}(\pi_i) - \mathcal{R}(\pi_i^{BL}))logP_\theta(\pi_i) \qquad (8)$$

Abl.2: The two layers are trained alternately, switching every 10 epochs. When the path construct layer is training, the ratio allocation layer remains static, and vice-versa.

The two ablation algorithms adopt the same parameters as HRL-TAE. In the ablation experiments, the same test data as in the simulation experiment is used, and the results are shown in Table 2.

Case	HRL-TAE		Abl.1		Abl.2	
	Obj.	Time	Obj.	Time	Obj.	Time
1	**0.346**	0.12	0.383	0.18	0.357	0.17
2	**0.376**	0.13	0.398	0.15	0.381	0.16
3	**0.385**	0.13	0.398	0.15	0.398	0.13
4	**0.383**	0.11	0.419	0.16	0.393	0.14
5	**0.343**	0.11	0.358	0.12	0.353	0.12
6	**0.356**	0.23	0.383	0.26	0.372	0.20
7	**0.342**	0.21	0.369	0.17	0.358	0.15
8	**0.355**	0.17	0.386	0.20	0.355	0.12
9	**0.315**	0.18	0.341	0.18	0.323	0.19
10	**0.344**	0.20	0.393	0.18	0.379	0.16
11	**0.416**	0.17	0.447	0.22	0.449	0.21
12	**0.425**	0.26	0.452	0.18	0.448	0.22
13	**0.405**	0.22	0.422	0.17	0.418	0.28
14	**0.408**	0.26	0.440	0.24	0.430	0.21
15	**0.417**	0.17	0.452	0.26	0.441	0.27
16	**0.406**	0.12	0.488	0.32	0.432	0.33
17	**0.409**	0.20	0.486	0.30	0.451	0.33
18	**0.383**	0.20	0.439	0.34	0.416	0.35
19	**0.412**	0.17	0.458	0.38	0.439	0.39
20	**0.406**	0.18	0.463	0.32	0.420	0.32

Table 2: Results of Ablation Experiments

The results indicate that HRL-TAE outperforms the ablation algorithms in all cases and shows average improvements of 8.86% and 4.75%, respectively. In addition, the training curves of the three algorithms on Geant topology are presented in Fig. 8. The ablation experiments demonstrate that QCT markedly reduces learning fluctuations and enhances training efficiency.

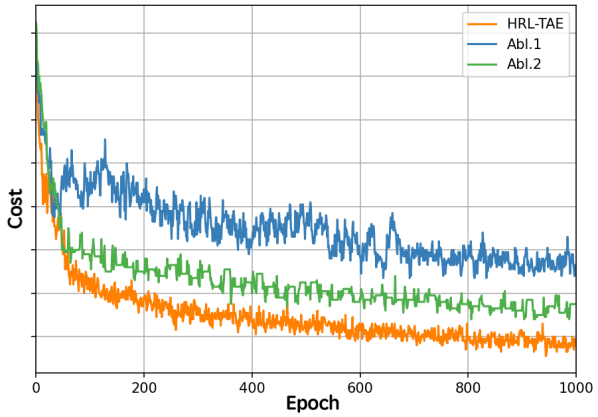

Figure 8: Training Curves of Different Algorithms

Real-World Experiment

To verify the effectiveness of HRL-TAE in real-world system, experiments are conducted on the software-defined network system, which mainly includes the following devices:

Case	Topo.	HRL-TAE		TEAL		ECMP		DRSIR	
		Obj.	Time(s.)	Obj.	Time(s.)	Obj.	Time(s.)	Obj.	Time(s.)
rw1	H9N	**0.462**	0.07	0.510	0.12	0.552	0.03	0.536	0.14
rw2	H9N	**0.473**	0.10	0.512	0.13	0.597	0.03	0.515	0.16
rw3	H9N	**0.478**	0.07	0.532	0.13	0.540	0.03	0.546	0.15
rw4	H9N	**0.482**	0.10	0.518	0.13	0.576	0.03	0.520	0.16
rw5	H9N	**0.474**	0.09	0.517	0.11	0.591	0.03	0.543	0.14
rw6	H18N	**0.484**	0.15	0.513	0.19	0.594	0.03	0.522	0.12
rw7	H18N	**0.500**	0.19	0.528	0.19	0.575	0.03	0.533	0.14
rw8	H18N	**0.471**	0.17	0.497	0.17	0.562	0.03	0.522	0.15
rw9	H18N	**0.482**	0.10	0.498	0.22	0.531	0.04	0.492	0.16
rw10	H18N	**0.493**	0.14	0.528	0.20	0.581	0.03	0.527	0.15

Table 3: Results of Real-world Experiments

BigTao Network Tester: It has 18 full-duplex ports for data transfer. With test software, it allows programmable definition of transmission rates and collection of real-time network metrics such as packet loss rates and throughput.

Controller: Open Network Operating System (ONOS) platform is utilized. It can obtain network information and push flow tables to control routing paths.

Switches: The core switch is adopted to push control commands, while other switches are used for data transmission.

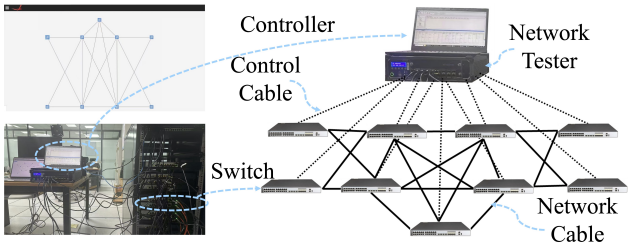

Figure 9: Real-world Experiment System

As shown in Fig. 9, the system corresponds to the H9N topology. Deployed in controller, the algorithm reads information via ONOS platform. It generates a routing scheme, converts it into flow tables, and sends them to switches to control flow transmission. The real-world experiments are conducted with the same settings as simulation experiments, and are conducted on the H9N (9 nodes, 18 links) and H18N (18 nodes, 37 links) topologies generated via the well-known Waxman algorithm (Waxman 1988) and shown in Fig. 7. The results of experiment are presented in Table 3.

According to the results, HRL-TAE outperforms the comparison algorithms in all cases, with average improvements of 6.90%, 15.80%, and 8.72%, respectively. These confirm the effectiveness of HRL-TAE in the real-world system.

Conclusion

This paper presents HRL-TAE, an end-to-end framework to solve MPCFP. To avoid the disadvantages of using pre-generated path, HRL-TAE employs a two-layer network to construct paths and allocate ratios separately. Guided by TLEP, the problem is transferred into MDP and enables the agent to explore and make decisions. Additionally, a constraint-driven mask mechanism filters out infeasible options, and QCT method boosts training efficiency. Finally, experiments conducted on both simulation and real-world systems demonstrate the effectiveness of HRL-TAE.

In the future, we will continue to conduct research on MPCFP, apply it to more complex scenarios, such as joint optimization in multiple sub-nets with hybrid protocols, and adopt more advanced algorithms.

References

Barnhart, C.; Hane, C. A.; and Vance, P. H. 2000. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2): 318–326.

Berger, J.; Friedrich, T.; Lenzner, P.; Machaira, P.; and Ruff, J. 2025. Strategic Network Creation for Enabling Greedy Routing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 13622–13630.

Casas-Velasco, D. M.; Rendon, O. M. C.; and da Fonseca, N. L. 2022. DRSIR: A deep reinforcement learning approach for routing in software-defined networking. *IEEE Transactions on Network and Service Management*, 19(4): 4807–4820.

Chen, J.; Xiao, W.; Zhang, H.; Zuo, J.; and Li, X. 2024. Dynamic routing optimization in software-defined networking based on a metaheuristic algorithm. *Journal of Cloud Computing*, 13(1): 41.

Farrugia, N.; Briffa, J. A.; and Buttigieg, V. 2023. Solving the multicommodity flow problem using an evolutionary routing algorithm in a computer network environment. *Plos one*, 18(4): e0278317.

Fukugami, T. 2023. Improvement of Network Flow Using Multi-Commodity Flow Problem. *Network*, 3(2): 239–252.

Galliera, R. 2024. Deep reinforcement learning for communication networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 23387–23388.

Gao, X.; Liu, J.; Wan, B.; and An, L. 2024. Hierarchical reinforcement learning from demonstration via reachability-based reward shaping. *Neural Processing Letters*, 56(3): 184.

Gomes, R.; Vieira, D.; and Pereira, M. B. 2024. Artificial algae optimization for Virtual Network Embedding problems in 5G network slicing scenarios. *Expert Systems with Applications*, 239: 122436.

Jiang, J.; Shi, X.; Zhou, X.; Han, G.; and Deng, F. 2025. CRL-KEA: A Deep Reinforcement Learning Assisted Evolutionary Algorithm for Multipath Routing Optimization Problem. In *2025 IEEE 19th International Conference on Control Automation (ICCA)*, 142–149.

Jiang, W.; Han, H.; Zhang, Y.; Wang, J.; He, M.; Gu, W.; Mu, J.; and Cheng, X. 2024. Graph neural networks for routing optimization: Challenges and opportunities. *Sustainability*, 16(21): 9239.

Knight, S.; Nguyen, H. X.; Falkner, N.; Bowden, R.; and Roughan, M. 2011. The Internet Topology Zoo. *IEEE Journal on Selected Areas in Communications*, 29(9): 1765–1775.

Li, J.; Giotsas, V.; Wang, Y.; and Zhou, S. 2022. Bgp-multipath routing in the internet. *IEEE Transactions on Network and Service Management*, 19(3): 2812–2826.

Liu, J.; Zhai, Y.; Zhao, G.; Xu, H.; Fang, J.; Zeng, Z.; and Zhu, Y. 2024. InArt: In-Network Aggregation with Route Selection for Accelerating Distributed Training. In *Proceedings of the ACM on Web Conference 2024*, 2879–2889.

Lu, Y.; Chen, Y.; Xu, X.; Fu, Q.; Chen, J.; and Liu, L. 2023. A sub-flow adaptive multipath routing algorithm for data centre network. *International Journal of Computational Intelligence Systems*, 16(1): 25.

Masri, H.; Krichen, S.; and Guitouni, A. 2019. Metaheuristics for solving the biobjective single-path multicommodity communication flow problem. *International Transactions in Operational Research*, 26(2): 589–614.

Pateria, S.; Subagdja, B.; Tan, A.-h.; and Quek, C. 2021. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 54(5): 1–35.

Sui, J.; Jiang, J.; Shi, X.; Liang, M.; and Deng, F. 2024. Multi-commodity Flow Optimization Algorithm Among Multiple Communication Protocols. In *2024 36th Chinese Control and Decision Conference (CCDC)*, 6161–6166. IEEE.

Sutton, R. S.; Barto, A. G.; et al. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Tessler, C.; Shpigelman, Y.; Dalal, G.; Mandelbaum, A.; Haritan Kazakov, D.; Fuhrer, B.; Chechik, G.; and Mannor, S. 2022. Reinforcement learning for datacenter congestion control. *ACM SIGMETRICS Performance Evaluation Review*, 49(2): 43–46.

Vezhnevets, A. S.; Osindero, S.; Schaul, T.; Heess, N.; Jaderberg, M.; Silver, D.; and Kavukcuoglu, K. 2017. Feudal networks for hierarchical reinforcement learning. In *International conference on machine learning*, 3540–3549. PMLR.

Waxman, B. M. 1988. Routing of multipoint connections. *IEEE journal on selected areas in communications*, 6(9): 1617–1622.

Xu, Z.; Bai, Y.; Zhang, B.; Li, D.; and Fan, G. 2023a. Haven: Hierarchical cooperative multi-agent reinforcement learning with dual coordination mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11735–11743.

Xu, Z.; Yan, F. Y.; Singh, R.; Chiu, J. T.; Rush, A. M.; and Yu, M. 2023b. Teal: Learning-accelerated optimization of wan traffic engineering. In *Proceedings of the ACM SIGCOMM 2023 Conference*, 378–393.

Zhang, J.; Jin, L.; and Yang, C. 2021. Distributed cooperative kinematic control of multiple robotic manipulators with an improved communication efficiency. *IEEE/ASME Transactions on Mechatronics*, 27(1): 149–158.

Zhang, J.; Ye, M.; Guo, Z.; Yen, C.-Y.; and Chao, H. J. 2020. CFR-RL: Traffic engineering with reinforcement learning in SDN. *IEEE Journal on Selected Areas in Communications*, 38(10): 2249–2259.

Zhou, X.; Shi, X.; Zhang, L.; Chen, C.; Li, H.; Ma, L.; Deng, F.; and Chen, J. 2024. Scalable Hierarchical Reinforcement Learning for Hyper Scale Multi-Robot Task Planning. arXiv:2412.19538.

Zuo, Y.; Wu, Y.; Min, G.; and Cui, L. 2018. Learning-based network path planning for traffic engineering. *Future Generation Computer Systems*, 92(MAR.): 59–67.