Software Engineering and Applications 软件工程与应用, 2025, 14(5), 1105-1112
Published Online October 2025 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2025.145098

Hans汉斯

基于PPO算法的链路不相交多路径路由优化研究

刘正堂

塔里木大学信息工程学院，新疆 阿拉尔

收稿日期：2025年9月22日；录用日期：2025年10月13日；发布日期：2025年10月23日

摘　要

链路不相交多路径路由是当前网络优化的重要方向，传统路由算法在面对网络动态变化时存在适应性差、效率低等问题。本文提出了一种基于强化学习的链路不相交多路径路由算法，具体采用PPO (近端策略优化)算法。实验结果表明该算法具有良好的收敛性与稳定性，所选路径集合的奖励显著优于随机方法，在不同网络状态下均表现出较强的泛化能力与适应能力。

关键词

链路不相交，路由，强化学习，PPO

Research on Link-Disjoint Multipath Routing Optimization Based on the PPO Algorithm

Zhengtang Liu

College of Information Engineering, Tarim University, Alar Xinjiang

Received: September 22, 2025; accepted: October 13, 2025; published: October 23, 2025

Abstract

Link-disjoint multipath routing is an important direction in current network optimization. Traditional routing algorithms often suffer from poor adaptability and low efficiency when facing dynamic changes in networks. This paper proposes a link-disjoint multipath routing algorithm based on reinforcement learning, specifically using the PPO (Proximal Policy Optimization) algorithm. Experimental results show that the proposed algorithm exhibits good convergence and stability. The reward of the selected path sets is significantly better than that of random methods, and the algorithm demonstrates strong generalization and adaptability under different network conditions.

文章引用: 刘正堂. 基于 PPO 算法的链路不相交多路径路由优化研究[J]. 软件工程与应用, 2025, 14(5): 1105-1112.
DOI: 10.12677/sea.2025.145098

Keywords

Link-Disjoint, Routing, Reinforcement Learning, PPO

1. 引言

链路不相交多路径路由是指在源节点与目的节点之间构建多条无共享链路的路径，以提高数据传输的容错性与稳定性[1]。为了解决链路不相交多路径路由问题，一些算法被提出。黄敏等[2]提出 PEMP-OLSR 算法，引入链路与节点生存时间并设计迭代控制机制，提升路径构建的稳定性与不相交性。然而，当网络规模扩大或节点高速移动时预测误差明显，路由维护开销过大。章刚等[3]针对算力网络中服务链部署需求，采用遗传算法在资源映射过程中搜索不相交路径集合，保障算力调度效率。该方法在网络状态高度动态或实时性要求较高的场景下，遗传算法收敛速度慢，结果容易延迟。徐忠根等[4]设计基于 HSV 色彩编码的路由方法，通过空间映射识别路径间缠绕度，实现路径可视化分离。但在链路频繁波动或大规模拓扑下，计算复杂度和状态更新滞后限制了算法效果。朱尚明等[5]结合最宽路径策略和瓶颈链路带宽提出一种 QoS 约束下的不相交路径搜索算法。不过，在大规模或链路 QoS 参数剧烈波动的场景下，路径容易失效且维护开销高。吴正宇等[6]则基于蚁群优化方法，引入节点剩余能量与寿命评估函数，实现能耗平衡的不相交路径选择。但在高速动态或干扰严重的环境中信息更新滞后，难以保持路径稳定。

以上传统算法在网络发展阶段的初期都取得了很好的结果，但随着网络的快速发展，很难适应网络状态信息高度动态变化的特点。因此，人们开始将强化学习引入到路由优化中，来提升算法对复杂动态环境的适应能力。文献[7]提出了一种名为 RLMR 的 SDN 多路径路由方案，基于马尔可夫决策过程和 Q 学习构建路径选择模型，将其建模为状态–动作交互过程，提升了链路利用率，在抖动与丢包率方面展现出明显优势，RLMR 方案使用 Q 学习算法，面对大规模复杂状态空间时，泛化能力有限。文献[8]提出了一种基于深度强化学习的多路径路由规划算法 m-DQN，利用神经网络拟合 Q 值函数，构建状态到动作的映射模型，在提升链路带宽利用率的同时，有效减少带宽损耗，并满足不同服务流的 QoS 要求，m-DQN 算法引入深度神经网络增强了对高维网络状态特征的建模，但在训练过程中对样本数量与学习率较为敏感，收敛速度及对网络动态变化的适应性仍有待进一步优化。文献[9]针对数据中心网络中多路径资源调度问题，提出了一种子流自适应多路径路由算法 SAMP，利用深度强化学习中的 DDPG 算法构建状态到动作的映射策略，实现子流路径分配的协同调度。实验结果显示，在数据中心典型流量场景下，SAMP 相较于 ECMP 和 MPTCP 在降低延迟和提升任务完成率方面表现更优。该算法是在数据中心网络环境下设计与训练的，对场景特征具有较强依赖性，当网络拓扑结构或流量模式发生变化时，其迁移能力相对有限。文献[10]提出了一种基于深度强化学习的路径集合调度器，采用 PPO 算法，在多路径 TCP (MPTCP) 传输中实现吞吐量最大化与能耗最小化。通过状态–动作映射训练调度策略，实现了能效与性能的自适应平衡。实验结果表明，该方法在不同网络环境下具有良好的策略稳定性和环境适应性，但在面对复杂路径结构或任务密集型调度场景时，其策略扩展性与泛化能力仍需进一步验证。

尽管现有强化学习方法在多路径路由策略优化中取得一定成效，但普遍聚焦于路径选择性能，较少考虑路径集合的结构性约束(如链路不相交)的显式建模。在多路径并发调度场景下，这种忽略易导致链路

重叠与资源瓶颈，从而降低调度效率。

针对上述问题，本文提出一种基于 PPO 的链路不相交多路径路由算法。该算法以路径瓶颈带宽最大化为目标，通过网络拓扑进行实验评估，验证了该算法的收敛性与稳定性。

2. 链路不相交多路径路由建模

本文用无向图 $\mathfrak{G}=(\mathcal{V},\mathcal{E})$ 来表示网络拓扑，其中 $\mathcal{V}=\{v_i\}$ 是网络中的节点构成的集合，v_i 可以是网络中的各种设备(路由器或交换机等)，用 $n_v=|\mathcal{V}|$ 表示网络节点的个数。$\mathcal{E}=\{e_{ij}\}$ 是网络中节点间的所有链路构成的集合，e_{ij} 表示从节点 v_i 到节点 v_j 的链路，用 $n_e=|\mathcal{E}|$ 表示网络节点间的链路条数，链路通常包含了带宽、延迟等属性，不失一般性，本文将剩余带宽作为主要讨论对象，用 b_{ij} 表示链路 e_{ij} 的剩余带宽。

本文主要关注不相交的多径传输的实现，为此，用 $\mathcal{P}^{sd}=\{\mathfrak{P}_k^{sd}\}$ 表示从源节点 s 到目标节点 d 的所有简单路径的集合，用 $n_p=|\mathcal{P}^{sd}|$ 表示从源节点 s 到目标节点 d 的所有简单路径的条数，其中，$\mathfrak{P}_k^{sd}=(\mathcal{V}_k^{sd},\mathcal{E}_k^{sd}),k=1,2,\cdots,n_p$ 表示 $\mathcal{P}^{sd}$ 中编号为 k 的简单路径，这里 $\mathcal{V}_k^{sd}=\{v_{i'}\}\subset\mathcal{V}$ 表示路径 $\mathfrak{P}_k^{sd}$ 所经过的节点集合，$v_{i'}$ 表示 $\mathfrak{P}_k^{sd}$ 中编号为 i' 的节点，$\mathcal{E}_k^{sd}=\{e_{i'j'}\}\subset\mathcal{E}$ 表示路径 $\mathfrak{P}_k^{sd}$ 所有边的集合，$e_{i'j'}$ 表示 $\mathfrak{P}_k^{sd}$ 中从编号为 i' 的节点到编号 j' 的节点的链路。为了获得不相交路径，本文定义如下二元变量和约束条件：

$$x_{i'j'}^{\mathfrak{P}_k^{sd}}=\begin{cases}1, & \text{如果 } e_{i'j'}\in\mathcal{E}_k^{sd}\\ 0, & \text{如果 } e_{i'j'}\notin\mathcal{E}_k^{sd}\end{cases} \tag{1}$$

$$\sum_{\mathfrak{P}_k^{sd}\in\mathcal{P}^{sd},e_{i'j'}\in\mathcal{E}_k^{sd}} x_{i'j'}^{\mathfrak{P}_k^{sd}}\le 1 \tag{2}$$

满足条件(1)和(2)的简单路径称为不相交路径，用 $\tilde{\mathcal{P}}^{sd}=\{\mathfrak{P}_{k'}^{sd}\}$ 表示不相交路径集合。对于路径 $\mathfrak{P}_k^{sd}$，假设 $\tilde{b}_k^{sd}$ 为该路径 $\mathfrak{P}_k^{sd}$ 上所有链路的最小带宽，即瓶颈带宽，它可以表示为：

$$\tilde{b}_k^{sd}=\min_{e_{i'j'}\in\mathcal{E}_k^{sd}} b_{i'j'} \tag{3}$$

其中 $b_{i'j'}$ 表示从节点 i' 到节点 j' 的链路的剩余带宽。本文的目标是使所选路径的瓶颈带宽之和最大，具体来说，是最大化以下目标函数：

$$\max\sum_{\mathfrak{P}_{k'}^{sd}\in\tilde{\mathcal{P}}^{sd}}\tilde{b}_k^{sd}=\max\sum_{\mathfrak{P}_{k'}^{sd}\in\tilde{\mathcal{P}}^{sd}}\min_{e_{i'j'}\in\mathcal{E}_k^{sd}} b_{i'j'} \tag{4}$$

3. 基于 PPO 的链路不相交多路径路由算法

本节分为三个部分，3.1 节介绍状态空间，动作空间，奖励函数，路径集合更新机制，3.2 节介绍网络设计，3.3 节介绍网络更新过程。

3.1. 状态空间，动作空间，奖励函数，路径集合更新机制

3.1.1. 状态空间

用 $\mathcal{P}_0^{sd}$ 表示从源节点到目标节点的初始化路径集合，用 $n_{p_0}=|\mathcal{P}_0^{sd}|$ 表示初始化路径集合的大小，用 $M_{ij}^{\mathcal{P}^{sd}}(t)=\left[m_{ij}^{\mathcal{P}^{sd}}(t)\right]^{n\times n}$ 表示路径矩阵，$m_{ij}^{\mathcal{P}^{sd}}(t)$ 表示边 e_{ij} 在 $\mathcal{P}^{sd}$ 里出现的次数，定义如下：

$$m_{ij}^{\mathcal{P}^{sd}}(t)=\sum_{p\in\mathcal{P}^{sd}} I^p(e_{ij}) \tag{5}$$

其中 $I^p(e_{ij})$ 为示性函数。

用 $B_{ij}^{\mathcal{P}^{sd}}(t)=\left[b_{ij}^{\mathcal{P}^{sd}}(t)\right]^{n\times n}$ 表示带宽矩阵，$b_{ij}^{\mathcal{P}^{sd}}(t)$ 表示边 e_{ij} 在 $\mathcal{P}^{sd}$ 中的可用带宽，本文将状态空间定义为 $S=\left(M_{ij}^{\mathcal{P}^{sd}}(t),B_{ij}^{\mathcal{P}^{sd}}(t)\right)$，其维度为 $N^{n\times n}\times R^{n\times n}$。用 $s=\left(m_{ij}^{\mathcal{P}^{sd}}(t),b_{ij}^{\mathcal{P}^{sd}}(t)\right)$ 表示状态空间 S 的元素。

3.1.2. 动作空间

动作 a 表示从初始化路径集合中选择一条路径，动作空间 $A=\{a\}\in\{0,1\}^{n_{P0}}$，其中 $a=\left[a(i)\right]^{n_{P0}}$，$a(i)$ 定义如下：

$$a(i)=\begin{cases}1, & \text{选择第 } i \text{ 条路径}\\ 0, & \text{未选择第 } i \text{ 条路径}\end{cases}\quad i=1,2,3,\cdots,n_{P0} \tag{6}$$

3.1.3. 奖励函数

单步奖励函数 r 本文定义为在状态 s 下，根据动作 a 选择一条路径 p_k 所得的瓶颈带宽，即

$$r=r(s,a)=\tilde{b}^{sd}(p_k),p_k=p(s,a) \tag{7}$$

3.1.4. 路径集合更新机制

为了避免不同路径之间出现重复使用同一链路的情况，本文在动作空间的基础上引入了一种路径集合更新机制。该机制通过逐步移除已经使用链路相关的路径，构建出一组互不重叠的可选路径序列。更新步骤如下：

1. 初始化路径路径集合为 $\mathcal{P}_0^{sd}$；
2. 在第 k 次迭代中，在状态 s 下，根据动作 a 从路径集合 $\mathcal{P}_k^{sd}$ 中选择一条路径 p_k，即 $p_k=p(s,a)$；
3. 从路径集合 $\mathcal{P}_k^{sd}$ 中删除所有与路径 p_k 存在公共边的路径，得到 $\mathcal{P}_{k+1}^{sd}$；
4. 重复步骤 2~3，直到 $\mathcal{P}_k^{sd}=\varnothing$。

3.2. 网络设计

本文使用 PPO 算法来进行路径选择优化，PPO 属于 Actor-Critc 架构，通过策略网络(Actor)和价值网络(Critic)进行路径选择与动作评估。用 $\pi(a|s;\theta)$ 表示策略网络，θ 为 $\pi(a|s;\theta)$ 的网络参数，其含义为在输入状态 s 下，输出一个动作概率分布，所以满足 $\sum_{a\in A}\pi(a|s;\theta)=1$。用 $V(s;\psi)$ 表示价值网络，其中 ψ 为 $V(s;\psi)$ 的网络参数，其含义为在状态 s 下的估计值，$V(s;\psi)$ 与 $r(s,a)$ 的关系如下：

$$V(s;\psi)=E_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)|s_0=s\right] \tag{8}$$

其中 $r(s,a)$ 为单步奖励函数，$\gamma\in(0,1)$ 为折扣因子，期望 E_π 是关于策略 $\pi(a|s;\theta)$ 的期望，因此 $V(s;\psi)$ 表示在状态 s 下预期的长期累积奖励的估计值。

由于状态空间和动作空间的维度不一样，因此无法把状态动作对 (s,a) 直接输入策略网络 $\pi(a|s;\theta)$ 中，为了解决这个问题，本文将动作重新编码，用 $\Phi=\left[\phi_{ij}\right]\in\{0,1\}^{n_{P0}\times n_e}$ 表示动作编码矩阵，其中 ϕ_{ij} 表示路径 p_j 是否包含边 e_{ij}，定义如下：

$$\phi_{ij}=\begin{cases}1, & e_{ij}\in p_j\\ 0, & e_{ij}\notin p_j\end{cases} \tag{9}$$

3.3. 网络更新

在网络更新过程中，从经验池 D 中采样 N 个样本 $\{(s_i,a_i,r_i,s_{i'})\}$。价值网络的目标值计算如下：

$$\delta_i = r_i + \gamma V\left(s_{i'};\psi\right) - V\left(s_i;\psi\right) \tag{10}$$

其中 δ_i 表示第 i 次采样的目标值，r_i 表示第 i 次采样的奖励，s_i 表示第 i 次采样的状态，$s_{i'}$ 表示执行动作后的下一个状态，$\gamma \in (0,1)$ 为折扣因子，$V\left(s_i;\psi\right)$ 表示对当前状态 s_i 的估计值，$V\left(s_{i'};\psi\right)$ 表示对下一个状态 $s_{i'}$ 的估计值，ψ 为网络参数。

为了充分利用历史经验，本文采用广义优势估计(GAE)计算优势函数 A_i，其定义如下：

$$A_i = \delta_i + (\gamma\lambda)\delta_{i+1} + (\gamma\lambda)^2 \delta_{i+2} + \cdots \tag{11}$$

其中 δ_i 表示第 i 次采样的目标值，$\gamma \in (0,1)$ 为折扣因子，$\lambda \in (0,1)$ 为 GAE 的平滑参数。

策略网络的目标是是最大化以下剪切目标函数：

$$L^{CLIP}(\theta) = \frac{1}{N}\sum_{I=1}^{N} \min\left(r_i(\theta)A_i, clip\left(r_i(\theta),1-\varepsilon,1+\varepsilon\right)A_i\right) \tag{12}$$

其中 $r_i(\theta) = \dfrac{\pi\left(a_i \mid s_i;\theta\right)}{\pi\left(a_i \mid s_i;\theta_{old}\right)}$ 表示策略概率比，θ 表示当前策略参数，θ_{old} 表示上一轮策略参数，A_i 表示优势函数，$clip\left(r_i(\theta),1-\varepsilon,1+\varepsilon\right)$ 表示剪切项，$\varepsilon \in (0,1)$ 为剪切系数，$clip\left(r_i(\theta),1-\varepsilon,1+\varepsilon\right)$ 定义如下：

$$clip\left(r_i(\theta),1-\varepsilon,1+\varepsilon\right) = \begin{cases} r_i(\theta), 1-\varepsilon \le r_i(\theta) \le 1+\varepsilon \\ 1-\varepsilon, r_i(\theta) < 1-\varepsilon \\ 1+\varepsilon, r_i(\theta) > 1+\varepsilon \end{cases} \tag{13}$$

对于价值网络定义损失函数如下：

$$L = \frac{1}{N}\sum_{i=1}^{N}\left(V\left(s_i;\psi\right) - \delta_i\right)^2 \tag{14}$$

其中 $V\left(s_i;\psi\right)$ 为状态 s_i 的估计值，δ_i 为目标值，N 为样本数量。

4. 伪代码

算法 1：基于 PPO 的链路不相交多路径路由优化算法

输入：初始化路径集合 $\mathcal{P}_0^{sd}$，状态空间 S，动作空间 A，最大迭代次数 K，折扣因子 γ，剪切系数 ε，GAE 参数 λ，策略网络学习率 α_1，价值网络学习率 α_2

输出：优化后的策略网络 $\pi\left(a \mid s;\theta\right)$

1. 初始化策略网络 $\pi\left(a \mid s;\theta\right)$，价值网络 $V\left(s;\psi\right)$，旧策略网络 $\pi\left(a \mid s;\theta_{old}\right) \leftarrow \pi\left(a \mid s;\theta\right)$
2. 初始化经验池 $D \rightarrow \varnothing$
3. 初始化路径集合 $\mathcal{P}_0^{sd} \leftarrow \mathcal{P}^{sd}$
4. **for** 迭代轮次 $k = 1$ to K **do**
5. 　初始化状态 $s_0 \in S$
6. 　　**while** $\mathcal{P}_k^{sd} \neq \varnothing$ **do**
7. 　　根据当前策略网络采样动作：$a_t \sim \pi\left(a \mid s_t;\theta\right)$
8. 　　执行动作 a_t，获得奖励 r_t，下一个状态 s_{t+1}

存储样本 (s_t, a_t, r_t, s_{t+1}) 到经验池 D

路径集合更新：移除与 a_t 存在公共边的所有路径，得到 $\mathcal{P}_{k+1}^{sd}$

若 $\mathcal{P}_{k+1}^{sd} = \varnothing$，终止当前轮路径选择

令 $s_t \leftarrow s_{t+1}$，继续路径选择

end while

从经验池中采样 N 个样本 $\{(s_i, a_i, r_i, s_{i'})\}$

根据(10)计算价值网络的目标值 δ_i

根据(11)计算优势函数 A_i

计算策略概率比：

$$r_i(\theta) = \frac{\pi(a_i \mid s_i; \theta)}{\pi(a_i \mid s_i; \theta_{old})}$$

根据(13)计算剪切目标函数 $L^{CLIP}(\theta)$

更新策略网络参数：

$$\theta \leftarrow \theta + \alpha \nabla_\theta L^{CLIP}(\theta)$$

更新价值网络参数：

$$\psi \leftarrow \psi - \alpha \nabla_\psi \frac{1}{N} \sum_{i=1}^{N} \left(V(s_i; \psi) - \delta_i\right)^2$$

更新旧策略网络参数：

$$\theta_{old} \leftarrow \theta$$

end for

5. 实验结果

本节分为两个部分，5.1 给出奖励收敛曲线分析，5.2 给出本文算法与随机路径选择方法对比。本文实验结果在图 1 所示的 14 个节点的拓扑图上进行，其中节点 1 为源节点，节点 13 为目标节点。拓扑中存在多个可选路径，链路间存在交叉与瓶颈，适合用于评估基于强化学习的链路不相交多路径路由算法的性能。在实验设置中，本文算法关键超参数如下：折扣因子 $\gamma = 0.98$，GAE 参数 $\lambda = 0.95$，剪切系数 $\varepsilon = 0.2$，策略网路学习率 $\alpha_1 = 0.0004$，价值网络学习率 $\alpha_2 = 0.0005$。

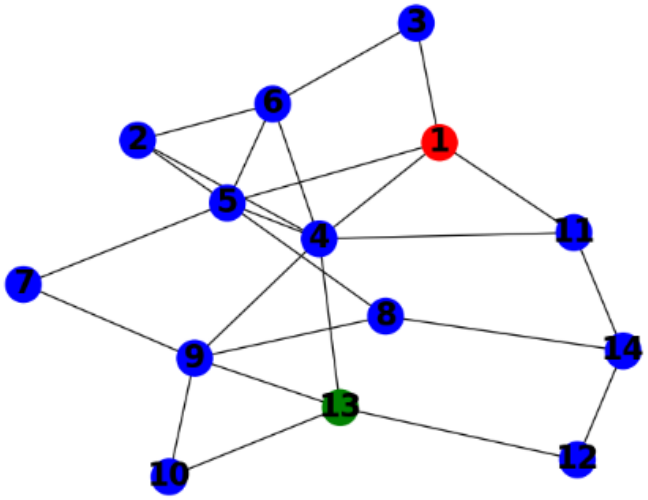

Figure 1. Topology diagram
图 1. 拓扑图

5.1. 奖励收敛曲线分析

图 2 显示了训练过程中的即时奖励(蓝色曲线)与滑动平均奖励(红色曲线)的变化趋势，在训练初期(前 300 轮)，奖励值波动较大，表明策略仍处于探索阶段，随着训练轮数增加，奖励值显著上升并趋于稳定，到 400 轮后基本收敛，最终滑动平均奖励值稳定在 75 左右。实验结果表明：在该网络拓扑下，所提出的链路不相交多路径路由算法不仅具备良好的收敛性能，还能有效适应链路瓶颈变化，体现出较强的环境适应能力。

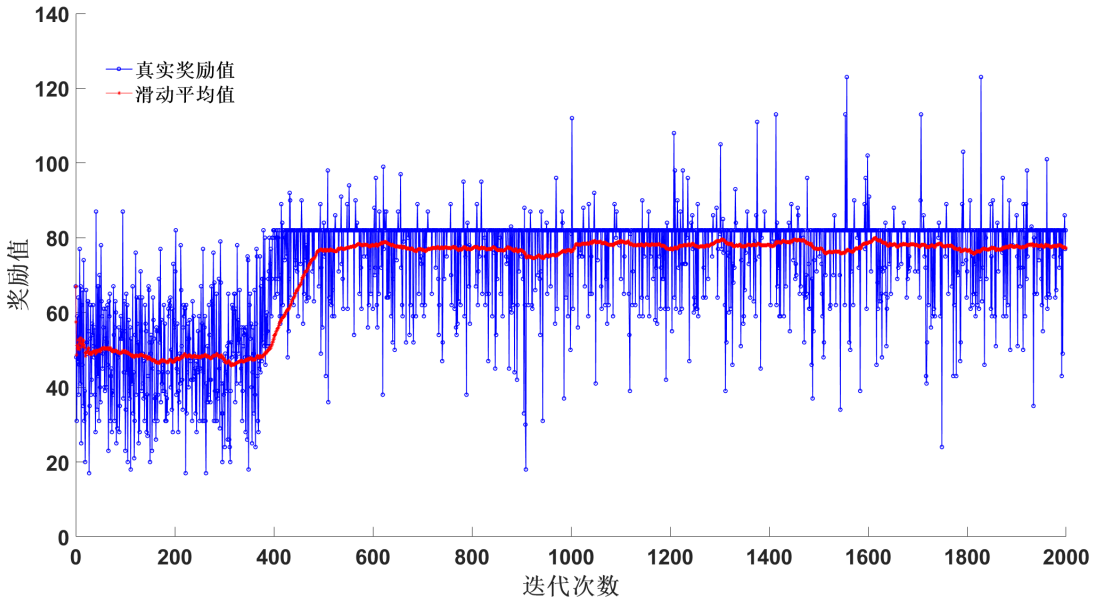

Figure 2. Reward convergence curve
图 2. 奖励收敛曲线

5.2. 本文算法与随机路径选择方法对比

表 1 显示了本文算法与随机选择路径方法的对比结果。对比实验共设置了 5 组不同网络状态，在每组测试中，本文算法都明显优于随机方法。本文算法选出的路径集合在拓扑结构中分布更均匀，有效避免了低带宽链路与路径重叠，对网络状态具有良好的泛化能力和鲁棒性。

Table 1. Comparison between the proposed algorithm and the random path selection method
表 1. 本文算法与随机路径选择方法对比

视图(不同的网络状态)	随机选择路径	随机选择路径的奖励	本文算法选择路径	本文算法选择路径的奖励
1	[1, 11, 4, 6, 5, 7, 9, 8, 14, 12, 13], [1, 4, 9, 13], [1, 3, 6, 2, 4, 5, 8, 9, 10, 13], [1, 5, 2, 6, 4, 13]	48.9987	[1, 4, 9, 13], [1, 11, 4, 6, 5, 8, 14, 12, 13], [1, 5, 7, 9, 10, 13], [1, 3, 6, 2, 4, 13]	73.9156
2	[1, 5, 8, 14, 11, 4, 9, 10, 13], [1, 11, 14, 8, 9, 7, 5, 2, 6, 4, 13], [1, 3, 6, 2, 5, 7, 9, 13]	33.9990	[1, 4, 9, 13], [1, 11, 4, 6, 5, 8, 14, 12, 13], [1, 5, 7, 9, 10, 13], [1, 3, 6, 2, 4, 13]	81.9989
3	[1, 4, 11, 14, 8, 9, 13], [1, 11, 4, 9, 10, 13], [1, 3, 6, 5, 7, 9, 8, 14, 12, 13], [1, 5, 2, 6, 4, 13]	38.9987	[1, 4, 9, 13], [1, 11, 4, 6, 5, 8, 14, 12, 13], [1, 5, 7, 9, 10, 13], [1, 3, 6, 2, 4, 13]	81.9988

续表

4	[1, 3, 6, 2, 4, 5, 7, 9, 13], [1, 5, 4, 9, 8, 14, 12, 13], [1, 11, 14, 8, 9, 7, 5, 2, 6, 4, 13]	43.24600	[1, 4, 9, 13], [1, 11, 4, 6, 5, 8, 14, 12, 13], [1, 5, 7, 9, 10, 13], [1, 3, 6, 2, 4, 13]	75.5798
5	[1, 3, 6, 2, 4, 9, 10, 13], [1, 4, 2, 5, 8, 9, 13], [1, 11, 14, 12, 13], [1, 5, 7, 9, 8, 14, 11, 4, 13]	47.9708	[1, 4, 9, 13], [1, 3, 6, 2, 4, 11, 14, 8, 9, 10, 13], [1, 11, 4, 6, 5, 8, 14, 12, 13], [1, 5, 7, 9, 4, 13]	79.9985

6. 结束语

本文针对链路不相交多路径路由优化问题，提出一种基于 PPO 算法的路径选择方法。通过设计合理的状态空间，动作空间，奖励函数，实现了选择高质量链路不相交路径集合。实验结果表明，该算法在训练过程中展现出良好的收敛性与稳定性，在多组网络状态下，所选路径集合相较于随机方法获得了更高的奖励，说明该算法对路径选择具有合理性，对网络状态变化具有较强的适应能力。

基金项目

塔里木大学校长基金胡杨英才(硕士)项目《基于强化学习的 SDN 路由优化研究》资助，项目编号：TDZKSS202410。

参考文献

[1] 方效林, 石胜飞, 李建中. 无线传感器网络一种不相交路径路由算法[J]. 计算机研究与发展, 2009, 46(12): 2053-2061.

[2] 黄敏, 刘琼, 奚建清. 一种基于生存时间的 Adhoc 网络不相交多路径路由算法[J]. 计算机应用研究, 2010, 27(3): 1157-1160.

[3] 章刚, 黎曦. 基于算力网络的异构算力请求路由算法[J]. 电信科学, 2025, 41(2): 95-110.

[4] 徐忠根, 蒋琳. 无线传感器网络不相交多路径路由容错缠绕系统设计[J]. 现代电子技术, 2017, 40(13): 164-167.

[5] 朱尚明, 庄新华, 高大启. 一种端到端网络的不相交多路径 QoS 路由算法[J]. 计算机科学, 2007, 34(9): 35-38.

[6] 吴正宇, 宋瀚涛, 姜少峰, 等. 一种稳定的不相交多路径蚂蚁路由算法[J]. 北京理工大学学报自然版, 2007(4): 322-326.

[7] Chen, C., Xue, F., Lu, Z., Tang, Z. and Li, C. (2022) RLMR: Reinforcement Learning Based Multipath Routing for SDN. *Wireless Communications and Mobile Computing*, **2022**, Article ID: 5124960. https://doi.org/10.1155/2022/5124960

[8] Wang, Z., Lu, Z. and Li, C. (2020) Research on Deep Reinforcement Learning Multi-Path Routing Planning in SDN. *Journal of Physics*: *Conference Series*, **1617**, Article ID: 012043. https://doi.org/10.1088/1742-6596/1617/1/012043

[9] 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**, Article No. 25. https://doi.org/10.1007/s44196-023-00199-5

[10] Dong, P., Shen, R., Wang, Q., Zuo, Y., Li, Y., Zhang, D., *et al.* (2023) Multipath TCP Meets Reinforcement Learning: A Novel Energy-Efficient Scheduling Approach in Heterogeneous Wireless Networks. *IEEE Wireless Communications*, **30**, 138-146. https://doi.org/10.1109/mwc.013.2100658