

CFR-RL: Traffic Engineering with Reinforcement Learning in SDN

摘要

传统的流量工程 (TE) 解决方案可以通过重新路由尽可能多的流量来实现最优或接近最优的性能。然而, 它们通常没有考虑到频繁重新路由流量时可能产生的负面影响, 如数据包乱序。为了缓解网络干扰的影响, 一种有前景的 TE 解决方案是通过使用等效多路径转发大多数流量, 并通过 SDN 选择性地重新路由一些关键流, 以平衡网络的链路利用率。然而, 关键流的重新路由并不简单, 因为关键流选择的解空间非常庞大。此外, 由于基于规则的启发式算法无法适应流量矩阵和网络动态的变化, 因此不可能基于固定和简单的规则设计该问题的启发式算法。在本文中, 我们提出了 CFR-RL (Critical Flow Rerouting-Reinforcement Learning), 一种基于强化学习的方案, 能够为给定的流量矩阵自动学习选择关键流的策略。CFR-RL 通过构建并求解一个简单的线性规划问题, 重新路由这些选择的关键流, 以平衡网络的链路利用率。结果表明, CFR-RL 通过重新路由仅 10%-21.3% 的总流量, 即可实现接近最优的性能。

关键词: 强化学习; 软件定义网络; 流量工程; 负载均衡; 网络干扰缓解

1 引言

软件定义网络 (Software-Defined Networking, SDN) 为提高网络性能提供了新的机会 [1]。在 SDN 中, 控制平面可以基于其网络全局视图生成路由策略, 并通过在 SDN 交换机上安装和更新流条目来在网络中部署这些策略。流量工程 (TE) 是 SDN 中的一项重要网络功能 [2-4], 通常在 SDN 的控制平面中实现。TE 的目标是通过配置骨干网络中的路由来控制流量分布, 帮助互联网服务提供商优化网络性能和资源利用率 [5, 6]。由于节点间负载动态波动, 传统 TE [7-9] 通常通过周期性地重新路由大量流量来平衡各链路上的负载, 从而最小化网络拥塞的概率, 其中一个流量被定义为一个源-目的对。通常, 人们会将流量路由问题与特定的性能指标一起表述为一个优化目标函数。在给定的流量矩阵下, 人们通常希望通过路由所有流量的方式, 使网络中的最大链路利用率最小化。

尽管传统的 TE 解决方案可以通过尽可能多地重新路由流量实现最优或接近最优的性能, 但它们并未考虑在网络中重新路由流量时可能带来的负面影响, 例如数据包乱序。为了达到最优性能, TE 解决方案可能会重新路由大量流量, 仅仅是为了稍微降低最拥塞链路的链路利用率, 这可能会导致显著的网络干扰和服务中断。例如, 在骨干网络中, 两个节点之间的流量通常由许多不同应用的微流 (例如基于五元组的 TCP 流) 聚合而成。更改某个流的路径可

能会暂时影响许多 TCP 流的正常运行。数据包丢失或乱序可能引发重复 ACK 传输，从而触发发送端响应并缩小其拥塞窗口大小，进而降低发送速率，最终增加流的完成时间并降低流的服务质量。此外，在网络中重新路由所有流量可能会对 SDN 控制器造成巨大负担，因为控制器需要计算并部署新的流路径 [4]。由于在骨干网络中通过重新路由流量来缓解拥塞可能会对用户体验质量产生不利影响，网络运营商在设计 TE 解决方案时，除非将减少网络干扰纳入考量，否则并不倾向于在其网络中部署这些传统 TE 解决方案。

为减轻网络干扰的影响，一种有前景的 TE 解决方案是使用等价多路径（ECMP）转发大部分流量，同时利用 SDN 选择性地重新路由少量关键流量，从而平衡网络的链路利用率。其中，关键流量被定义为对网络性能具有主导影响的流量（例如，位于最拥塞链路上的流量）。现有研究表明，在给定的流量矩阵中存在关键流量 [4]。ECMP 通过在等价路径上平均分配流量来降低拥塞概率，而关键流量的重新路由则旨在以较低的网络干扰实现进一步的性能提升。

关键流量重路由问题可以分解为两个子问题：(1) 识别关键流量，以及 (2) 重路由这些关键流量以获得良好的性能。尽管通过将子问题 (2) 表述为线性规划（LP）优化问题可以相对容易地解决，但解决子问题 (1) 并不简单，因为其解空间非常庞大。例如，如果需要从 100 个流量中找出 10 个关键流量，解空间将有 $C_{100}^{10} \approx 17$ 万亿种组合。考虑到流量矩阵在分钟级别上会发生变化，高效的解决方案应能够快速且有效地识别每个流量矩阵中的关键流量。不幸的是，基于固定和简单规则设计启发式算法来解决这一算法复杂问题是不可能的。这是因为基于规则的启发式方法无法适应流量矩阵和网络动态的变化，当其设计假设被违反时，其性能无法得到保证。

2 相关工作

2.1 传统 TE 解决方案

在多协议标签交换网络中，路由问题被建模为一个优化问题，通过为每个源-目的对生成显式路由来分配流量 [7, 8]。使用开放式最短路径优先（OSPF）和等价多路径（ECMP）协议的研究 [10–12] 尝试通过调整链路开销来优化路径选择，从而尽可能平衡链路利用率。加权 ECMP [9] 扩展了 ECMP 协议，允许在每个节点对流量进行加权分配，并在性能上显著优于传统 ECMP。两阶段路由方法通过选择一组中间节点并调整流量分配比例，优化路由性能 [13, 14]。在第一阶段，每个源节点根据预定的分配比例将流量发送到中间节点；在第二阶段，中间节点再将流量传递到最终目的地。该方法需要在每个阶段中使用 IP 隧道、光层电路或标签交换路径。

2.2 基于 SDN 的 TE 解决方案

得益于新兴的 SDN 所提供的灵活路由策略，动态混合路由 [4] 通过预配置的路由策略动态调整流量分布，应对流量波动，从而实现了在多种流量场景下的负载均衡。Agarwal 等人 [2] 研究了一种部分部署 SDN 交换机的网络，通过战略性地布置控制器和 SDN 交换机，提高了网络利用率并减少了数据包丢失。Guo 等人 [3] 提出了一种名为 SOTE 的新算法，用于最小化 SDN/OSPF 混合网络中的最大链路利用率。

2.3 基于机器学习的 TE 解决方案

机器学习已被用于提升骨干网络和数据中心网络的性能。在骨干网络方面，Geyer 等人 [15] 设计了一种基于半监督深度学习的自动化网络协议。Sun 等人 [16] 选择性地控制一组节点，并使用基于强化学习的策略动态调整流经这些节点的流量的路由决策。为了在大规模 SDN 中最小化信令延迟，Lin 等人 [17] 提出了一种结合基于 RL 的 QoS 感知自适应路由解决方案的分布式三级控制平面架构。Xu 等人 [18] 利用 RL 优化了流量工程中的吞吐量和时延。在数据中心网络中，AuTO [19] 通过双层 RL 优化路由流量，其中外围系统用于主机部署和小流量的路由，而中央系统则用于收集全局流量信息并路由大流量。

然而，以上所有研究都未考虑因重路由导致的网络干扰和服务中断的影响。

3 本文方法

3.1 本文方法概述

通过训练 CFR-RL，以学习在多种历史流量矩阵上进行选择策略，其中流量矩阵可以由 SDN 交换机测量并定期收集到 SDN 中央控制器 [20]。CFR-RL 将选择策略表示为一个神经网络，该网络将“原始”观测（例如流量矩阵）映射到关键流量的组合。神经网络为将各种流量矩阵纳入选择策略提供了一种可扩展且具有表达力的方法。CFR-RL 基于 REINFORCE 算法 [21] 对该神经网络进行训练，并结合了一些定制化调整。

训练完成后，CFR-RL 定期将关键流选择策略应用于 SDN 控制器提供的每个实时流量矩阵，其中选择少量关键流（例如 K ）。VI-B1 节中的评估结果表明，选择总流量的 10% 作为关键流量（大约是总流量的 11%-21%）足以让 CFR-RL 实现接近最优的性能，而网络扰动（即与传统 TE 重新路由所有流量相比，重新路由总流量的百分比至少减少 78.7%。然后，SDN 控制器通过流重新路由优化方法在交换机上安装和更新相应的流条目来重新路由选定的关键流。其余流将继续通过默认 ECMP 路由进行路由。需要注意的是，上一周期选择的关键流在交换机上的流表项将会超时，并且该流将被默认 ECMP 路由或当前周期新安装的流表项路由。图 1 显示了一个说明性示例。CFR-RL 通过在 SDN 路径上相应的交换机上安装转发表项，将流量从 $S0$ 重新路由到 $S4$ ，以平衡链路负载。

不采用强化学习来解决流量重路由问题的原因有两个。首先，由于关键流量的集合较小，线性规划是一种高效且最优的方法来解决重路由问题。其次，一个路由解决方案包括每条链路上每个流量的分流比例（即流量需求的百分比）。在具有 E 条链路的网络中，路由解决方案总共会包含 $E \times K$ 个分流比例，其中 K 是关键流量的数量。由于分流比例是连续数值，我们不得不采用适用于连续动作域的 RL 方法 [22]。然而，由于高维的连续动作空间，当输出参数（即 $E \times K$ ）的数量较大时，已有研究表明这类 RL 方法会导致学习速度慢且效率低下 [18, 23]。

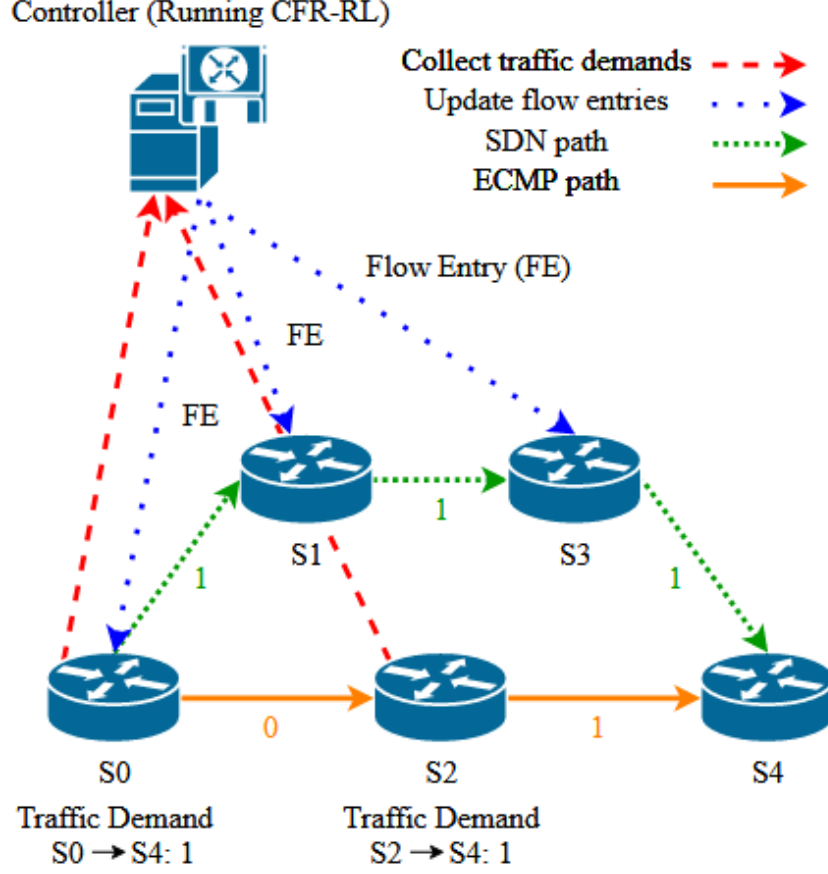


图 1. CFR-RL 重路由过程的一个示例说明。每条链路的容量均为 1。

3.2 关键流选择

3.2.1 强化学习公式

状态空间：智能体将状态 $s_t = TM_t$ 作为输入，其中 TM_t 是时间步 t 的流量矩阵，包含每个流量的流量需求信息。通常，网络拓扑保持不变，因此我们没有将拓扑信息作为输入的一部分。

动作空间：对于每个状态 s_t ，CFR-RL 将选择 K 个关键流量。在具有 N 个节点的网络中，总共有 $N*(N-1)$ 个流量，因此该强化学习问题将需要一个大小为 $C_{N*(N-1)}^K$ 的大型动作空间。受 [24] 的启发，将动作空间定义为 $0, 1, \dots, (N*(N-1))-1$ ，并允许智能体在每个时间步 t 采样 K 个不同的动作，即 $(a_t^1, a_t^2, \dots, a_t^K)$ 。

奖励：在给定状态 s 的情况下，CFR-RL 采样出 K 个不同的关键流量（即 f_k ），然后对这些关键流量进行重路由，并通过求解重路由优化问题 (4a)（将在后续部分描述）获得最大链路利用率 U 。奖励 r 定义为 $1/U$ ，用以反映关键流量重路由后网络负载均衡的性能。 U 越小（即奖励 r 越大），性能越好。

3.2.2 训练算法

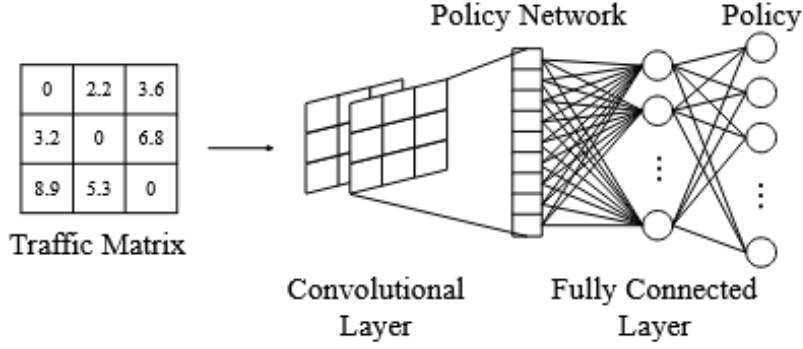


图 2. 策略网络架构

关键流量选择策略由一个神经网络表示。该策略网络以状态 $s_t = TM_t$ 作为输入，并输出所有可用动作上的概率分布 $\pi(a_t | s_t)$ 。图 2 展示了策略网络的架构。由于每个状态 s_t 会采样 K 个不同的动作且其顺序无关紧要，我们将解决方案 $a_t^K = (a_t^1, a_t^2, \dots, a_t^K)$ 定义为 K 个采样动作的组合。对于给定状态 s_t 下选择解决方案 a_t^K ，可以通过以下公式 (1) 近似参数化的随机策略 $\pi(a_t^K | s_t)$ 。其中， θ 表示策略网络的参数。

$$\pi(a_t^K | s_t; \theta) = \prod_{k=1}^K \pi(a_t^k | s_t; \theta) \quad (1)$$

训练的目标是最大化在各种流量矩阵下的网络性能，即最大化期望奖励 $E[r_t]$ 。因此，我们通过梯度上升优化 $E[r_t]$ ，使用带有基准 $b(s_t)$ 的 REINFORCE 算法。策略参数 θ 根据公式 (2) 更新：

$$\theta \leftarrow \theta + \alpha \sum_t \nabla_{\theta} \log \pi(a_{t,K} | s_t; \theta) (r_t - b(s_t)) \quad (2)$$

其中， α 是策略网络的学习率。一个好的基准 $b(s_t)$ 可以减少梯度的方差，从而提高学习速度。本文中，我们使用每个状态 s_t 的平均奖励作为基准。 $(r_t - b(s_t))$ 表示在给定状态 s_t 下，特定解决方案与“平均解决方案”相比的优越性。直观地讲，公式 (2) 可以解释如下：如果 $(r_t - b(s_t))$ 为正， $\pi(a_{t,K} | s_t; \theta)$ （即解决方案 $a_{t,K}$ 的概率）通过以步长 $\alpha(r_t - b(s_t))$ 更新策略参数 θ 在方向 $\nabla_{\theta} \log \pi(a_{t,K} | s_t; \theta)$ 上增加。否则，解决方案的概率将减少。公式 (2) 的净效应是强化那些在经验上导致更好奖励的动作。

为了确保在训练过程中，强化学习代理能够充分探索动作空间并发现较好的策略，熵项被加入到公式 (2) 中。这一方法通过鼓励探索而避免了过早收敛至次优的确定性策略，从而提升了探索性。因此，公式 (2) 被修改为以下形式：

$$\theta \leftarrow \theta + \alpha \sum_t (\nabla_{\theta} \log \pi(a_{t,k} | s_t; \theta) (r_t - b(s_t))) + \beta \nabla_{\theta} H(\pi(\cdot | s_t; \theta)) \quad (3)$$

其中， H 是策略的熵（即动作的概率分布）。超参数 β 控制熵正则化项的强度。

3.3 关键流重路由

默认情况下，流量是根据 ECMP 路由分配的。我们通过对这些关键流 $\langle s, d \rangle \in f_K$ 进行显式路由优化，重新路由少量的关键流（即 f_K ）。关键流重路由问题可以描述如下。给定一个网络 $G(V, E)$ ，其中包括选定关键流的流量需求集 $D_{s,d} (\forall \langle s, d \rangle \in f_K)$ 和通过默认 ECMP 路由所贡献的背景链路负载 $\{\bar{l}_{i,j}\}$ ，我们的目标是为每个关键流获得最优的显式路由比例 $\{\sigma_{s,d}^{i,j}\}$ ，以使得最大链路利用率 U 最小化。

为了搜索所有可能的未充分利用路径以供选择的关键流使用，我们将重路由问题表述为如下的优化问题。

$$\text{minimize } U + \epsilon \cdot \sum_{\langle i,j \rangle \in E} \sum_{\langle s,d \rangle \in f_K} \sigma_{i,j}^{s,d}$$

约束条件为：

$$l_{i,j} = \sum_{\langle s,d \rangle \in f_K} \sigma_{i,j}^{s,d} \cdot D_{s,d}^k + \bar{l}_{i,j} \quad i, j : \langle i, j \rangle \in E$$

$$l_{i,j} \leq c_{i,j} \cdot U \quad i, j : \langle i, j \rangle \in E$$

$$\sum_{k: \langle k, i \rangle \in E} \sigma_{k,i}^{s,d} - \sum_{k: \langle i, k \rangle \in E} \sigma_{i,k}^{s,d} = \begin{cases} -1 & \text{if } i = s \\ 1 & \text{if } i = d \\ 0 & \text{otherwise} \end{cases} \quad i \in V, s, d : \langle s, d \rangle \in f_K$$

$$0 \leq \sigma_{i,j}^{s,d} \leq 1 \quad s, d : \langle s, d \rangle \in f_K, i, j : \langle i, j \rangle \in E$$

在 (4a) 式中 $\epsilon \cdot \sum_{\langle i,j \rangle \in E} \sum_{\langle s,d \rangle \in f_K} \sigma_{i,j}^{s,d}$ 是必要的，否则最优解可能会包含不必要的长路径，其中 $\epsilon (\epsilon > 0)$ 是一个足够小的常数，以确保 U 的最小化具有更高优先级。(4b) 式表示链路 $\langle i, j \rangle$ 上由显式路由的流量需求和默认的 ECMP 路由的流量需求所导致的流量负载。(4c) 式是链路容量利用率约束。(4d) 式是所选关键流的流量守恒约束。

通过使用线性规划求解器求解上述线性规划问题，我们可以得到所选关键流 $\{\sigma_{i,j}^{s,d}\} (\forall \langle s, d \rangle \in f_K)$ 的最优显式路由解。然后，SDN 控制器相应地在交换机上安装和更新流表项。

4 复现细节

4.1 与已有开源代码对比

本文相关 [源代码](#) 部分开源，本次复现工作着重于复现作者的实验结果，并在源代码的基础之上修改了部分参数以及神经网络结构。数据集使用了 Abilene 网络拓扑及其相关流量矩阵，下载地址为：[数据集下载地址](#)。

4.2 实验环境搭建

4.2.1 硬件环境

- CPU: Intel(R) Xeon(R) Gold 6430
- GPU: NVIDIA GeForce RTX 4090

4.2.2 软件环境

- 编程语言：Python 3.8.5
- 深度学习框架：Tensorflow 2.2.0
- CUDA 版本：12.4

4.3 具体实现细节

策略神经网络由三层组成。第一层是一个卷积层，具有 128 个滤波器，卷积核大小为 3×3 ，步幅为 1。第二层是一个全连接层，包含 128 个神经元。前两层使用的激活函数是 Leaky ReLU。最后一层是一个全连接线性层，包含 $N \times (N - 1)$ 个神经元，对应所有可能的关键流。对最后一层的输出应用 softmax 函数，以生成所有可用动作的概率。学习率 α 初始配置为 0.001，并且每 500 次迭代以 0.96 为基数衰减，直到达到最小值 0.0001。熵因子 β 配置为 0.1。

5 实验结果分析

本部分对实验所得结果进行分析，比较基于最优路由、Top-K Critical、Top-K Flows 和 ECMP 策略的负载分配情况。

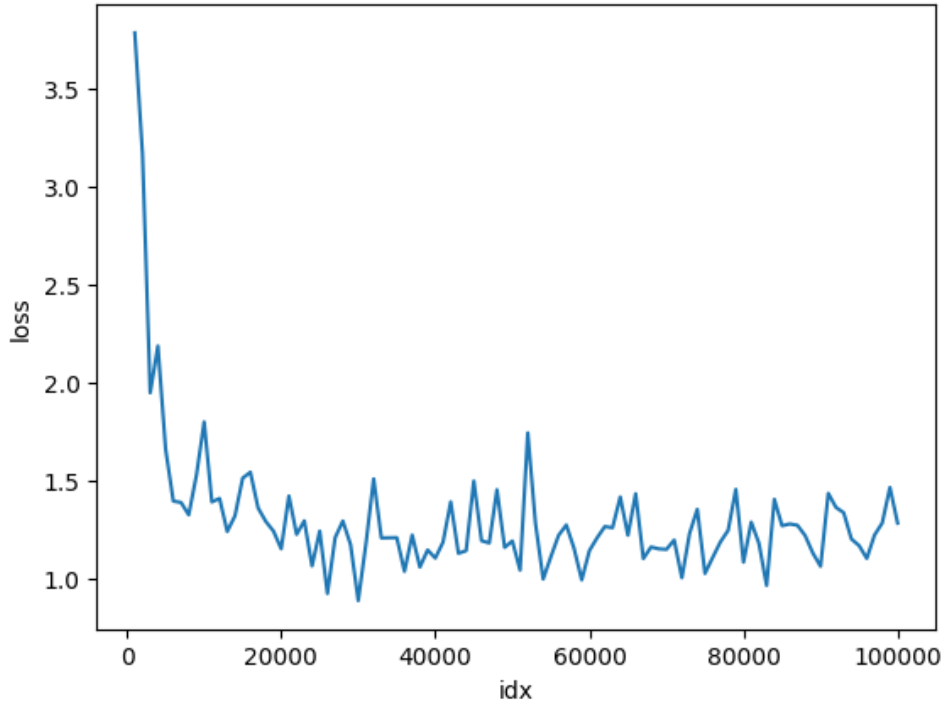


图 3. 损失变化图

图 3 展示了在训练过程中的损失变化图。初始阶段损失较高，随着训练进行，损失值迅速下降。在训练中后期，损失值趋于稳定，但仍有一定波动，范围在 1.0 到 1.5 之间。

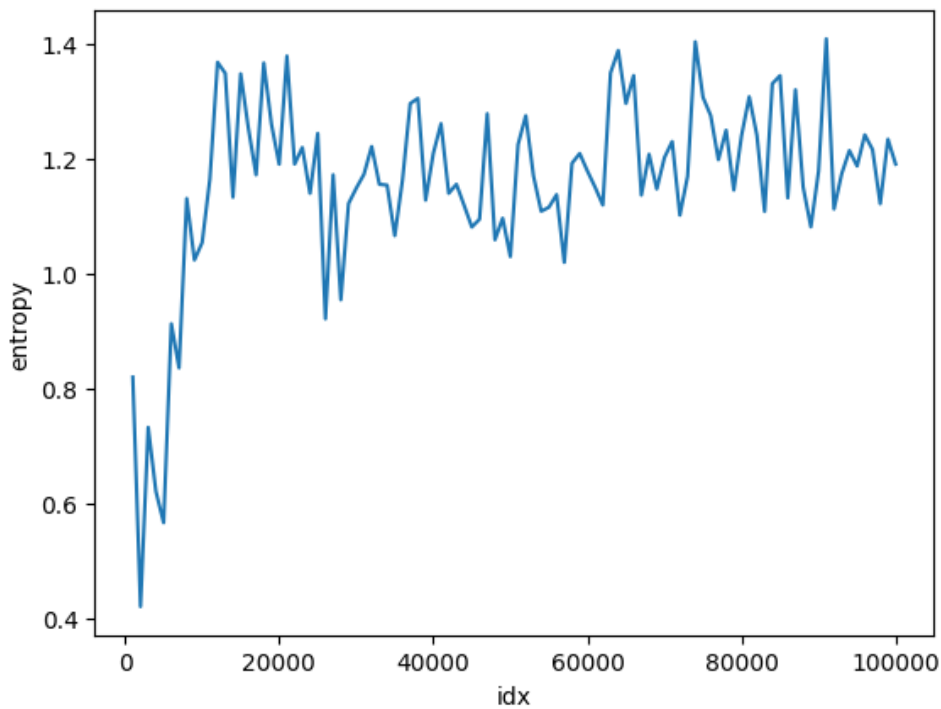


图 4. 熵变化图

图 4 展示了在训练过程中，熵值随训练轮数的变化。初始阶段熵值较低，随后熵值迅速上升，随着训练的进行，熵值呈现较为稳定的波动，波动范围在 0.8 至 1.4 左右。熵代表策略的随机性，初期熵较低可能表示模型在开始时策略较为确定性，但未探索到更多可能的策略。

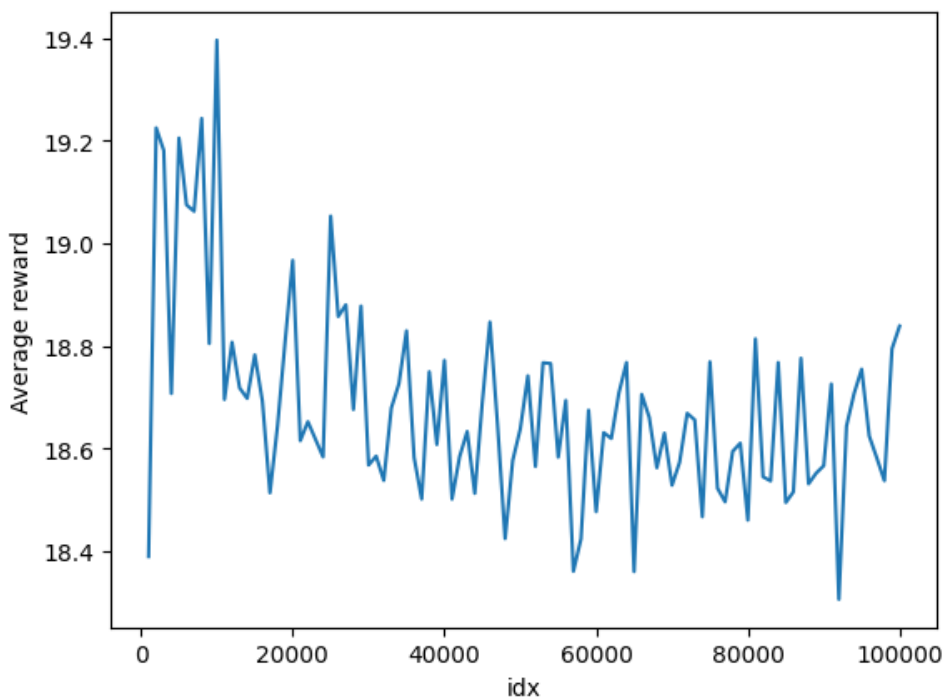


图 5. 平均奖励变化图

图 5 展示了在训练过程中，平均奖励值随训练轮数的变化。初始阶段奖励值较高，随后出现略微下降的趋势，整体波动范围在 18.4 到 19.4 之间。奖励下降可能表明随着训练的进

行，模型在某些特定场景中为了探索策略而暂时牺牲了收益。

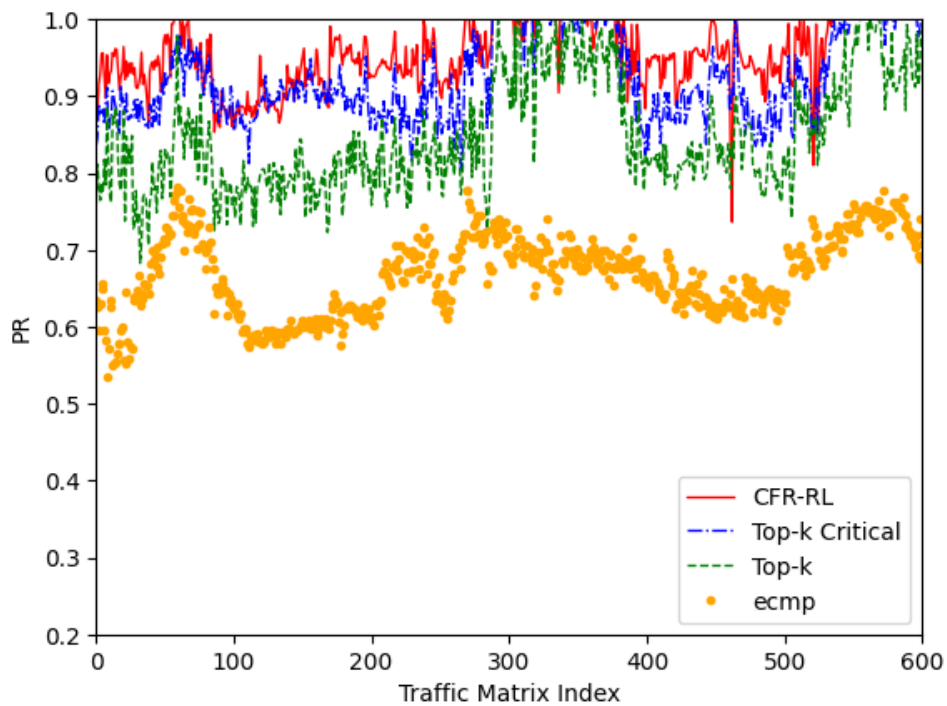


图 6. 复现结果

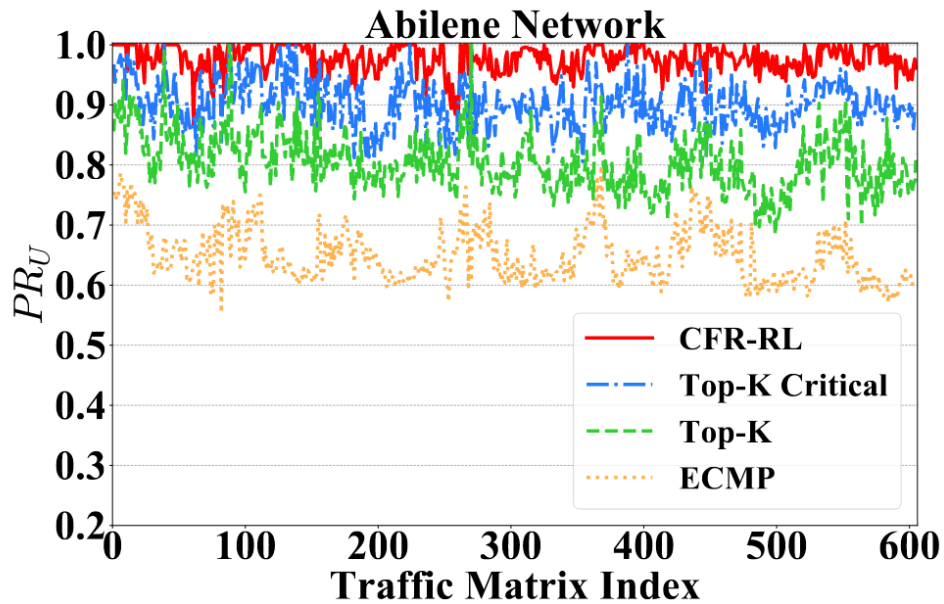


图 7. 原文结果

图 6 和图 7 分别展示了复现结果与原文结果的效果对比，从图中可以看出复现结果与原文结果大体一致，表明 CFR-RL 算法能够通过重路由 10% 左右的关键流，有效的平衡网络的链路利用率。

6 总结与展望

CFR - RL 通过强化学习自动选择关键流并重路由，实现接近最优性能且泛化能力良好。未来工作将从优化目标、可扩展性和重训练等方面改进 CFR - RL，如考虑最小化端到端延迟、减少重路由流量，通过多种方式提升对大规模网络的支持，以及确定重训练时机和数据选择等。

参考文献

- [1] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review*, 38(2):69–74, 2008.
- [2] Sugam Agarwal, Murali Kodialam, and TV Lakshman. Traffic engineering in software defined networks. In *2013 Proceedings IEEE INFOCOM*, pages 2211–2219. IEEE, 2013.
- [3] Yingya Guo, Zhiliang Wang, Xia Yin, Xingang Shi, and Jianping Wu. Traffic engineering in sdn/ospf hybrid network. In *2014 IEEE 22nd international conference on network protocols*, pages 563–568. IEEE, 2014.
- [4] Junjie Zhang, Kang Xi, Min Luo, and H Jonathan Chao. Dynamic hybrid routing: Achieve load balancing for changing traffic demands. In *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*, pages 105–110. IEEE, 2014.
- [5] Junjie Zhang, Kang Xi, and H Jonathan Chao. Load balancing in ip networks using generalized destination-based multipath routing. *IEEE/ACM Transactions on Networking*, 23(6):1959–1969, 2014.
- [6] Zehua Guo, Weikun Chen, Ya-Feng Liu, Yang Xu, and Zhi-Li Zhang. Joint switch upgrade and controller deployment in hybrid software-defined networks. *IEEE Journal on Selected Areas in Communications*, 37(5):1012–1028, 2019.
- [7] Yufei Wang and Zheng Wang. Explicit routing algorithms for internet traffic engineering. In *Proceedings Eight International Conference on Computer Communications and Networks (Cat. No. 99EX370)*, pages 582–588. IEEE, 1999.
- [8] Eric D Osborne and Ajay Simha. *Traffic engineering with MPLS*. Cisco Press, 2002.
- [9] Junjie Zhang, Kang Xi, Liren Zhang, and H Jonathon Chao. Optimizing network performance using weighted multipath routing. In *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, pages 1–7. IEEE, 2012.
- [10] Bernard Fortz and Mikkel Thorup. Optimizing ospf/is-is weights in a changing world. *IEEE journal on selected areas in communications*, 20(4):756–767, 2002.

- [11] K. Holmberg and D. Yuan. Optimization of internet protocol network design and routing. *Networks: An International Journal*, 43(1):39–53, 2004.
- [12] J. Chu and C.-T. Lea. Optimal link weights for ip-based networks supporting hose-model vpns. *IEEE/ACM Transactions on Networking*, 17(3):778–788, 2009.
- [13] M. Kodialam, T. Lakshman, J. B. Orlin, and S. Sengupta. Oblivious routing of highly variable traffic in service overlays and ip backbones. *IEEE/ACM Transactions on Networking*, 17(2):459–472, 2008.
- [14] M. Antic, N. Maksic, P. Knezevic, and A. Smiljanic. Two phase load balanced routing using ospf. *IEEE Journal on Selected Areas in Communications*, 28(1):51–59, 2009.
- [15] F. Geyer and G. Carle. Learning and generating distributed routing protocols using graph-based deep learning. In *ACM Workshop on Big Data Analytics and Machine Learning for Data Communication Networks’ 18*, pages 40–45. ACM, 2018.
- [16] P. Sun, J. Li, Z. Guo, Y. Xu, J. Lan, and Y. Hu. Sinet: Enabling scalable network routing with deep reinforcement learning on partial nodes. In *ACM SIGCOMM’ 19 Posters and Demos*, pages 88–89. ACM, 2019.
- [17] S.-C. Lin, I. F. Akyildiz, P. Wang, and M. Luo. Qos-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach. In *IEEE International Conference on Services Computing’ 16*, pages 25–33. IEEE, 2016.
- [18] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang. Experience-driven networking: A deep reinforcement learning based approach. In *IEEE International Conference on Computer Communications’ 18*, pages 1871–1879. IEEE, 2018.
- [19] L. Chen, J. Lingys, K. Chen, and F. Liu. Auto: Scaling deep reinforcement learning for datacenter-scale automatic traffic optimization. In *ACM SIGCOMM’ 18*, pages 191–205. ACM, 2018.
- [20] H. Xu, Z. Yu, C. Qian, X. Li, and Z. Liu. Minimizing flow statistics collection cost of sdn using wildcard requests. In *IEEE International Conference on Computer Communications’ 17*, pages 1–9. IEEE, 2017.
- [21] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.
- [22] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2015.
- [23] A. Valadarsky, M. Schapira, D. Shahaf, and A. Tamar. Learning to route. In *ACM Workshop on Hot Topics in Networks’ 17, HotNets-XVI*, pages 185–191. ACM, 2017.

- [24] H. Mao, M. Alizadeh, I. Menache, and S. Kandula. Resource management with deep reinforcement learning. In *ACM Workshop on Hot Topics in Networks' 16*, pages 50–56. ACM, 2016.