

基于 OISL-OSU 的通算一体卫星路由优化算法

周伟¹, 罗青松¹, 谢晓兰^{2,3}, 敬超^{2,3}, 程小辉^{2,3}, 潘必韬⁴

(1. 中国电子科技集团公司 第三十四研究所, 广西 桂林 541004; 2. 桂林理工大学 计算机科学与工程学院, 广西 桂林 541000;

3. 桂林理工大学 广西嵌入式技术与智能系统重点实验室, 广西 桂林 541000;

4. 北京邮电大学 信息光子学与光通信国家重点实验室, 北京 100876)

摘要: 为了进一步解决多星及星地间通信和计算系统资源的高度整合与互馈增强问题, 提出了一种基于卫星间光链路光业务单元(OISL-OSU)的通算一体卫星路由优化算法——OISL-DDPG 算法。该算法采用深度确定性策略梯度(DDPG)优化方法, 提高了卫星光网络 QoS 性能。仿真实验表明: 采用基于时间窗口的网络切片方法符合国内低轨星座真实网络运行状态; 与最小跳数算法相比, 所提出的算法以增加微小时延为代价, 获得更高的链路利用率和链路占用率; 与多约束最短路由优先(MCSPF)算法相比, 该算法的收敛性更好, 时延和丢包率更低。

关键词: 卫星光网络; 光业务单元; 业务承载; 路由优化

中图分类号: TN256 **文献标志码:** A **文章编号:** 1002-5561(2024)05-0051-09

DOI: 10.13921/j.cnki.issn1002-5561.2024.05.009

开放科学(资源服务)标识码(OSID):



Communication-computing-integrated satellite routing optimization algorithm based on OISL-OSU

ZHOU Wei¹, LUO Qingsong¹, XIE Xiaolan^{2,3}, JING Chao^{2,3}, CHENG Xiaohui^{2,3}, PAN Bitao⁴

(1. The 34th Research Institute of CETC, Guilin Guangxi 541004, China; 2. College of Computer Science and Engineering,

Guilin University of Technology, Guilin Guangxi 541000, China; 3. Guangxi Key Laboratory of

Embedded Technology and Intelligent System, Guilin Guangxi 541000, China; 4. State Key Laboratory of

Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: In order to tackle the issue of high integration and mutual feedback enhancement of the resources of multi-satellite and inter-satellite communication and computing systems, an integrated satellite routing algorithm OISL-DDPG is proposed based on optical inter-satellite link-optical service unit(OISL-OSU). OISL-DDPG adopts the deep deterministic policy gradient (DDPG) optimization method to enhance the quality of service(QoS) performance of satellite optical networks. The simulation experiments indicat that the network slicing method based on time windows is consistent with the actual network operation status of domestic low-orbit constellations. Additionally, the proposed algorithm achieves higher link utilization and link occupancy rates at the cost of increasing slight delay than the minimum hop algorithm, and gets a better convergence rate, lower latency, and lower packet loss rate than the multi-constrained shortest path first(MCSPF) algorithm.

Key words: satellite optical network, optical service unit, service carrying, routing optimization

收稿日期: 2024-04-08.

基金项目: 2024 年广西重点研发计划项目(批准号:2023AB07026)资助。

作者简介: 周伟(1982—), 男, 四川成都人, 博士, 高级工程师。主要研究方向为网络通信协议、卫星光网络、机器学习、算法研究, 主持并完成省部级项目 4 项。



0 引言

研究确定性网络技术对于提升低轨卫星光网络服务质量, 保障国家基础设施安全, 提升信息覆盖能力具有重要的战略意义。确定性卫星通信网络除了提供可预测、可定量的网络服务, 还能够凭借定量的算力资源, 提供可持续的算力服务, 提升业务稳定性及可扩展性^[1]。未来卫星光网络的发展方向之一是将算力网络与传输网络融合。目前, 基于光业务单元光传

送网(OSU-OTN)技术构建通算一体化系统将面临业务调度/映射复杂度增加的问题,具体如下:1)通算一体化网络需要考虑业务需求、网络资源、设备能耗等多方面的调度因素,并根据不同业务的服务质量(QoS)需求,如时延、抖动、隔离、安全、鲁棒性,提供定制化的网络通信与算力供应^[2]。2)由于 OSU-OTN 的小颗粒度传送单元,业务数量预计会出现激增^[3-7]。例如,OSU-OTN 的最小传送粒度为 2.6 Mb/s,它能够光通道数据单元(ODU)划分为更小的带宽颗粒,这将使得 OTN 管控层面业务的规模大幅增长。在这种大规模业务场景下,实现高效的多业务并行调度和资源协调将变得异常复杂。3)在网络通算一体化设计中,平衡多个目标具有挑战性。在算力方面,需要同时考虑资源效率、服务质量和可持续发展等因素;而在传输方面,则需提高资源利用效率,减少单个业务对资源的占用,并实现网络传输与算力资源间的负载均衡^[8]。

为此,根据空间环境特性并结合深度确定性策略梯度(DDPG)算法,提出一种应用于卫星间光链路光业务单元(OISL-OSU)的通算一体化卫星路由优化算法——OISL-DDPG 算法。该算法旨在解决业务服务资源利用率、计算资源负载均衡以及传输资源负载均衡等方面的快速路由收敛计算问题。

1 OISL-OSU 业务调度优化模型

地面 OTN 协议正处于不断迭代的演进过程中,受限于卫星的功耗和处理能力,不能将 OTN 协议直接应用于天基场景,需要针对星上应用场景对协议进行裁剪优化。

本文的卫星光网络场景为多个算网业务并行调度,承载网络为由 OISL-OSU 组成的卫星光网络(N_c, N_s, L),其中 N_c, N_s, L 分别为卫星计算节点集合、卫星传输节点集合与星间链路集合。每个卫星计算节点 n_c ($n_c \in N_c$) 包含 M 种异构计算资源。将基于 OISL-OSU 光层各种物理传输特性与资源进行抽象,假设每个波长信道传输容量为 X_{ch} Gb/s。所有卫星算力节点设定为业务的入口,都具备交换转发能力,单个 OSU 的容量设定为 X_{OSU} Mb/s,且作为业务最小映射单元支持交叉功能,卫星传输节点转发交换最小单元设定为 O- DUK ($k \in \{0, 1, 2, 3, 4\}$),对应速率分别约为 1.25、2.5、10、40、104 Gb/s。在上述业务模型中,每个业务都有计算需求,卫星光网络需要为每个业务分配一个计算节点与相应的数据传输路由。假定第 i 个业务模型为 $r_i = (s_i, d_i, b_i, D_i, t_i, c_i)$, 其中, s_i 为业务源卫星节点, d_i 为业

务目的卫星节点, b_i, D_i, t_i 分别为第 i 个业务的传输带宽要求、端到端传输时延要求、业务持续时间要求; c_i 为第 i 个业务对 M 种资源的使用量, $c_i = \{c_i^1, \dots, c_i^M\}$ 。对于每条业务,分配一个卫星算力节点承载算力需求,在业务源卫星节点 s_i 与分配的卫星算力节点之间建立一条工作路由(和一条保护路由)。在 OISL-OSU 传输业务时,先将业务数据按照 OSU $\leftrightarrow$ ODUK $\leftrightarrow$ OTUK (光通路传送单元)层级进行映射,映射业务 i 需要的 OSU 数量 $n_{OSU}^i = b_i / X_{OSU}$ 。

由于采用 OISL-OSU 承载,将导致卫星光网络中业务数量规模激增。因此,需要针对业务调度问题进行建模和优化,模型参数和变量定义如表 1 所示。

表 1 业务调度模型参数定义

参数	定义
c_m^i	第 i 个业务对第 m 种计算资源的需求量
C_m^n	第 n 个卫星算力节点中第 m 种计算资源的剩余量
P_m^n	第 n 个卫星算力节点中第 m 种计算资源的单位成本
l_{sd}	从源节点 s 到目的节点 d 的星间工作链路
l_{op}	从源节点 o 到目的节点 p 的星间工作链路
S_p^+	从源节点 s 到目的节点 d 的所有星间工作链路集合, $l_{sd} \in S_p^+$
S_p^-	从源节点 o 到目的节点 p 的所有星间工作链路集合, $l_{sd} \in S_p^-$
$P_{l_{sd}}$	星间链路 l_{sd} 带宽单位成本
$D_{l_{sd}}$	星间链路 l_{sd} 传输时延, $D_{l_{sd}} = l_{sd} / c$, c 为光在真空中的传输速度
$L_{l_{sd}}$	星间链路 l_{sd} 传输误码/丢包率
$\delta_k^{l_{sd}}$	星间链路 l_{sd} 中第 k 个 ODU 是否被使用
$r_k^{l_{sd}}$	星间链路 l_{sd} 中第 k 个 ODU 中剩余的 OSU 数量
D_s	卫星传输节点转发时延
o_n^m	第 n 个节点的第 m 种计算资源已经被使用的量
Z_m^n	第 n 个节点的第 m 种计算资源的总资源量
x_i^n	第 i 个业务是否使用第 n 个卫星算力节点
$y_i^{l_{sd}}$	第 i 个业务的工作路由是否使用星间链路 l_{sd}
$y_{i,prot}^{l_{sd}}$	第 i 个业务的保护路由是否使用星间链路 l_{sd}
$z_{i,k}^{l_{sd}}$	第 i 个业务的工作路由是否使用星间链路 l_{sd} 中的第 k 个 ODU
$z_{i,prot,k}^{l_{sd}}$	第 i 个业务的保护路由是否使用星间链路 l_{sd} 中的第 k 个 ODU

优化模型包括业务服务资源利用量 f_{cost} 、计算资源负载均衡指标 f_{LB} 和传输资源负载均衡指标 f_{QoS} 等多个优化目标,优化模型目标函数如式(1)所示。

$$\min F(x) = (f_{\text{cost}}, f_{\text{LB}}, f_{\text{QoS}})^T \quad (1)$$

1)业务服务资源利用量包含算力资源和带宽资源成本,可表示为

$$f_{\text{cost}} = \sum_i \sum_n \sum_m c_m^i x_i^n p_m t_i + \sum_i \sum_{l_{\text{sd}}} b_i \left(y_i^{l_{\text{sd}}} + y_{i,\text{prot}}^{l_{\text{sd}}} \right) p_{l_{\text{sd}}} t_i \quad (2)$$

其中,等式右侧第一项算力资源成本计算方式为“资源使用量×资源单位成本×业务持续时间”,等式右侧第二项带宽资源成本包含工作路由和保护路由的资源占用量。

2)计算资源负载均衡指标是为了避免业务集中占用卫星算力节点资源,对于第 n 个节点的第 m 种计算资源的使用率为

$$r_n^m = (o_n^m + \sum_i c_m^i x_i^n) / Z_m^n \quad (3)$$

其中, $\sum_i c_m^i x_i^n$ 为新业务使用的计算资源量。计算资源负载均衡指标可采用不同卫星计算节点每种计算资源使用率的方差进行评估,计算方法如式(4)所示。

$$f_{\text{LB}} = \sum_n \sum_m (r_n^m - (1/N_c) \sum_n r_n^m)^2 / (nm) \quad (4)$$

3)传输资源负载均衡指标是为了避免业务集中占用某些星间链路传输资源,从而因拥堵导致网络服务水平降低、引起误码或丢包。为了获得 QoS 的最小时延和最低丢包率,该指标可以表示为

$$f_{\text{QoS}} = \sum_i \sum_{l_{\text{sd}}} (D_{l_{\text{sd}}} + D_s) + D_s \left(y_i^{l_{\text{sd}}} + y_{i,\text{prot}}^{l_{\text{sd}}} \right) + L_{\text{sd}} \left(y_i^{l_{\text{sd}}} + y_{i,\text{prot}}^{l_{\text{sd}}} \right) \quad (5)$$

优化模型约束条件如下:

1)节点选择约束。式(6)要求所有业务计算请求都映射到一个卫星算力节点中。式(7)要求当多个业务请求映射到某个卫星算力节点中,每种异构算力的需求之和不能超过该种算力的剩余量。

$$\sum_{n \in N_c} x_i^n = 1 \quad (6)$$

$$\sum_i x_i^n c_m^i \leq C_m^n \quad (7)$$

2)路由一致性约束。式(8)和式(9)保证所有业务的工作路由与保护路由都是连通的。

$$\sum_{l_{\text{wp}} \in S_p^i} y_i^{l_{\text{wp}}} - \sum_{l_{\text{ld}} \in S_p^i} y_i^{l_{\text{ld}}} = x_i^p - \delta_i^p \quad (8)$$

$$\sum_{l_{\text{wp}} \in S_p^i} y_{i,\text{prot}}^{l_{\text{wp}}} - \sum_{l_{\text{ld}} \in S_p^i} y_{i,\text{prot}}^{l_{\text{ld}}} = x_i^p - \delta_i^p \quad (9)$$

3)ODU 分配约束。式(10)保证所有业务占用的星间链路 l_{sd} 的第 k 个 ODU 中的 OSU 数量小于星间链路 l_{sd} 的第 k 个 ODU 中剩余的 OSU 数量。 n_{OSU}^i 表示业务 i 需要的 OSU 数量。式(11)和式(12)要求是:如果某个业务的工作路由或保护路由使用了一条星间链路 l_{sd} , 则该业务需要使用这条星间链路 l_{sd} 上的某个 ODU k 。

$$\sum_i \left(z_{i,k}^{l_{\text{sd}}} + z_{i,\text{prot},k}^{l_{\text{sd}}} \right) n_{\text{OSU}}^i \leq r_k^{l_{\text{sd}}} \quad (10)$$

$$\sum_k z_{i,k}^{l_{\text{sd}}} = y_i^{l_{\text{sd}}} \quad (11)$$

$$\sum_k z_{i,\text{prot},k}^{l_{\text{sd}}} = y_{i,\text{prot}}^{l_{\text{sd}}} \quad (12)$$

4)保护路由约束。式(13)保证,如果某个业务的工作路由使用了某条星间链路 l_{sd} , 则该业务的保护路由不会使用同一条星间链路 l_{sd} 。此约束是为了确保业务的传输可靠,避免因星间链路失效而导致工作路由和保护路由同时中断的情况发生。

$$y_i^{l_{\text{sd}}} + y_{i,\text{prot}}^{l_{\text{sd}}} \leq 1 \quad (13)$$

5)时延约束。式(14)和式(15)要求,对于任何业务,其工作路由和保护路由的星间链路 l_{sd} 传输时延与卫星节点转发时延之和,都不能大于该业务的端到端传输时延要求。

$$\sum_{l_{\text{sd}}} y_i^{l_{\text{sd}}} (D_{l_{\text{sd}}} + D_s) + D_s \leq D_i \quad (14)$$

$$\sum_{l_{\text{sd}}} y_{i,\text{prot}}^{l_{\text{sd}}} (D_{l_{\text{sd}}} + D_s) + D_s \leq D_i \quad (15)$$

将基于 OISL-OSU 的业务调度方法进行形式化建模,该模型表现为一个多目标整数二次规划问题(MO-IQP)。在此模型中,所有决策变量均为 0~1 整数变量。目标函数涉及 3 个方面的多目标协同优化:业务服务资源的利用量、计算资源的负载均衡以及传输负载均衡的均衡。

2 OISL-DDPG 算法

DeepMind 团队提出了深度确定性策略梯度 (DDPG) 的深度强化学习机制^[9],用于解决连续动作空间的强化学习问题。算法的整体框架如图 1 所示。

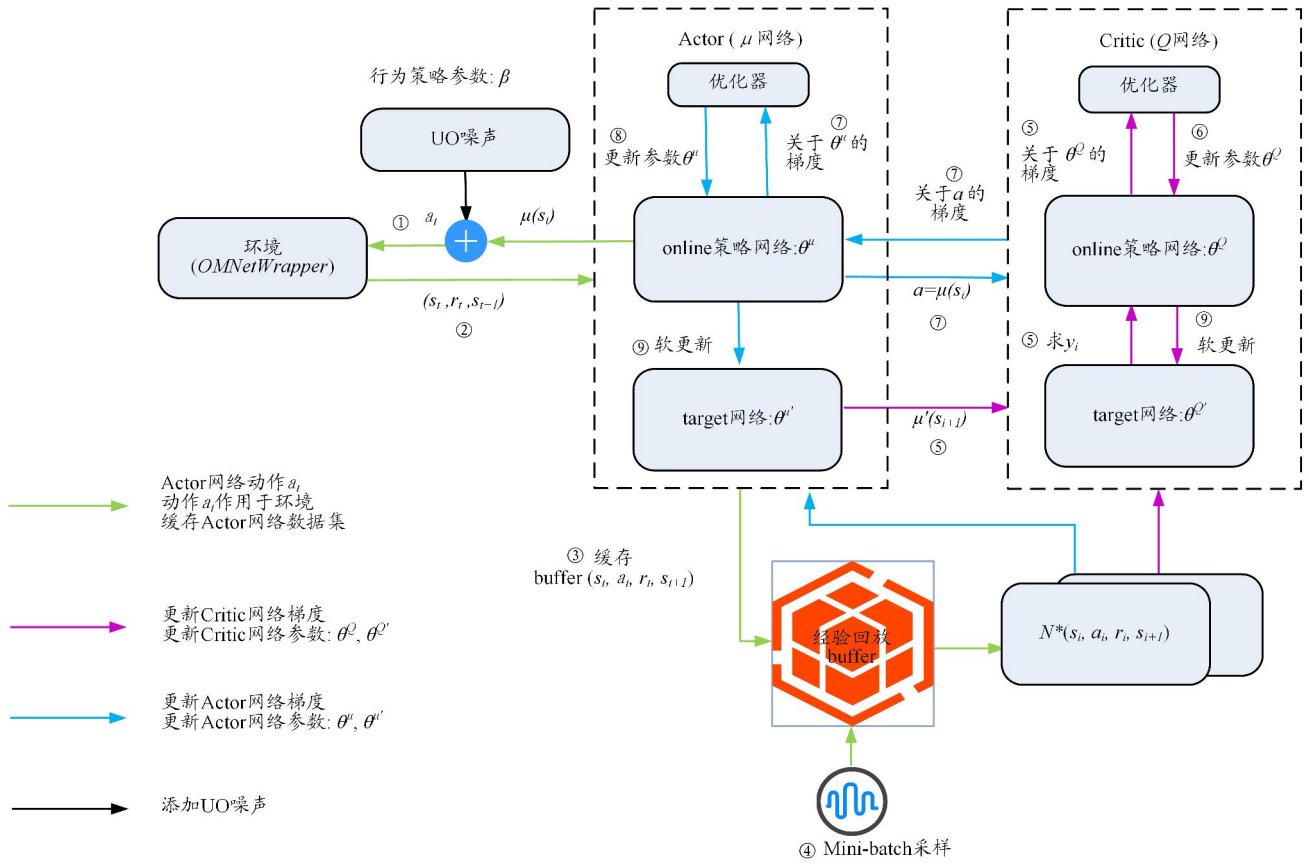


图1 DDPG 算法框架

DDPG 算法的 Actor 模型用于输出动作,而 Critic 模型则用于评估状态-动作对的价值函数。不同于传统的 Actor-Critic 模型,DDPG 算法的 Actor 模型输出连续的动作,并且 Critic 模型的目标值是通过使用 Actor 模型来确定的,从而形成了所谓的“确定性策略梯度”方法。此外,在训练过程中,DDPG 算法采用了“经验回放”技术,即把历史经验存储在一个回放缓存中,之后随机抽取一批经验(Mini-batch)进行训练,这种方法能有效提升训练的稳定性 and 效率。

在 DDPG 算法中,使用了一个参数为 θ^Q 的卷积神经网络,即 Q 网络(Q-network),来执行 Q 函数。它采用了一个称为 J 函数的性能目标来衡量策略 μ 的性能。J 函数定义为

$$J_{\beta}(\mu) = \int_S \rho^{\beta}(S) Q^{\mu}(S, \mu(S)) dS = E_{s \sim \rho^{\beta}}[Q^{\mu}(S, \mu(S))] \quad (16)$$

其中, $E_{s \sim \rho^{\beta}}[\cdot]$ 表示数学期望。式(16)中,环境状态 S 具有概率密度函数 ρ^{β} ,并且随着强化学习智能体的行为策略 μ 的变化而变化。 $\mu(S)$ 表示在状态 S 下采取的动作(action) a 。 $Q^{\mu}(S, \mu(S))$ 的值可以在每个环境状态 S

下根据策略 μ 选择相应的动作 a 获得。最终, $J_{\beta}(\mu)$ 是 $Q^{\mu}(S, \mu(S))$ 关于状态 S 的概率分布 ρ^{β} 的期望值,其中 S 服从 ρ^{β} 分布。训练 DDPG 智能体的目标是最大化 $J_{\beta}(\mu)$,同时最小化 Q 网络与真实值之间的误差。最优行为策略 μ 是为了最大化 $J_{\beta}(\mu)$,即

$$\mu = \arg\max_{\mu} J_{\beta}(\mu) \quad (17)$$

训练 μ 网络(μ -network)的过程是获得 μ 网络最优值 θ^{μ} 的过程。同时, Q 网络的训练过程也是搜索 Q 网络的最优值 θ^Q 的过程。

传统的启发式网络优化算法,如最短路径优先(SPF)算法,是专门为解决某些情况下的特定优化问题而设计的,具有一定的局限性。因此,本文设计了一种基于 OISL-OSU 承载的多目标整数二次型路由优化算法——OISL-DDPG 算法。采用 OISL-DDPG 算法取代传统的多约束最短路由优先策略,一步生成接近最优的路由选择,并实现连续动作空间下的通用可定制优化。OISL-DDPG 算法伪代码如表 2 所示。其中,步骤①~⑨对应于图 1 中的序号①~⑨,OMNetWrapper 表示除 OISL-DDPG 智能体以外的环境。OISL-DDPG 算法智能体通过 state(S)、action(a)和 reward(r)参数与环境进行交互。

表 2 OISL-DDPG 算法伪代码

步骤	内容
	初始化 Actor 网络、Critic 网络的 online 策略网络参数: θ^{μ}, θ^Q ; 初始化经验回放缓存(buffer); for each episode:
①	for each step: Actor 网络基于行为策略参数 β , 在 t 时刻为状态 S_t 选择动作(action) a_t , 并作用于环境(Environment): $a_t = \mu(S_t \theta^{\mu}) + N_t$; 其中 N_t 为 t 时刻添加的 Uhlenbeck-Ornstein(UO)噪声;
②	环境得到 t 时刻的奖励(reward) r_t , 并生成 $t+1$ 时刻的状态(state) S_{t+1} : $S_{t+1}, r_t = \text{OMNeTWrapper}(S_t, a_t)$;
③	将状态、动作、奖励集(S_t, a_t, r_t, S_{t+1})作为经验回放缓存数据集;
④	随机选取 $N^*(S_t, a_t, r_t, S_{t+1})$ 数据集作为 mini-batch 数据, 用于训练 Actor 网络、Critic 网络的 online 策略网络;
	计算 Critic 网络的 online 策略网络参数 θ^Q 梯度; 使用类似监督学习的方法将 Q 网络的损失定义为均方误差(MSE):
⑤	$L_{\text{loss}} = \frac{1}{N} \sum_i y_i - Q(S_i, a_i \theta^Q)^2$ $y_i = r_i + \gamma Q'(S_{i+1}, \mu'(S_{i+1} \theta^{\mu}) \theta^Q)$
⑥	使用优化器(Adam)更新 Critic 网络的 online 策略网络参数 θ^Q ;
	为 Actor 网络的 online 策略网络参数 θ^{μ} 计算梯度:
⑦	$\nabla \theta^{\mu} J_{\beta}(\mu) \approx \frac{1}{N} \sum_i (\nabla_a Q(S, a \theta^Q) _{a=\mu(S)} \nabla \theta^{\mu} \mu(S \theta^{\mu}) _{\theta^{\mu}})$
⑧	使用优化器(Adam)更新 Actor 网络的 online 策略网络参数 θ^{μ} ;
	软更新(soft update)Actor 网络、Critic 网络的 target 策略网络参数 θ^{μ}, θ^Q : $\tau \theta^Q + (1-\tau) \theta^Q \rightarrow \theta^Q, \tau \theta^{\mu} + (1-\tau) \theta^{\mu} \rightarrow \theta^{\mu}, \tau=0.001$
⑨	end for step end for episode

1)state(S): S_t 表示 OISL-DDPG 算法智能体在某一时刻观察到的网络状态。在 OISL-DDPG 算法中, S_t 是管理层收集的网络节点对之间当前网络业务负载流量状态信息。

2)action(a): a_t 表示 OISL-DDPG 算法智能体在某个时刻的决策动作, 即星间链路权值 $[\omega_1, \omega_2, \dots, \omega_l]$, $l \in L$, 其中 L 为星间链路集合。根据该套权值配置重新计算数据流链路, 并通过软件定义网络(SDN)控制器实现路由规则下发。通过执行该动作, 网络状态变为 S_{t+1} 。在 OISL-DDPG 算法中, 使用 Uhlenbeck-Ornstein 随机过程, 作为引入的随机噪声, 其表达式为

$$dx_t = \theta(\mu - x_t)dt + \sigma dW_t \quad (18)$$

其中, 常数 $\theta > 0$, 方差 $\sigma > 0$, W_t 为维纳过程(或布朗运动)。由于 Uhlenbeck-Ornstein 随机过程在时序上具有较强的相关性, 这使得 OISL-DDPG 算法中的智能体能够更好地探索那些具有动量属性的环境。

3)reward(r): r_t 表示网络处于状态 S_t 时执行动作后

a_t 获得的奖励。在 OISL-DDPG 算法中, 奖励 r_t 的计算表达式如式(19)所示。

$$r_t = \gamma_1 f_{\text{cost}} + \gamma_2 f_{\text{LB}} + \gamma_3 f_{\text{QoS}} \quad (19)$$

其中, $\gamma_1, \gamma_2, \gamma_3$ 表示为网络优化策略调整因子, $\gamma_1, \gamma_2, \gamma_3 \in [-1, 0]$ 。

3 仿真实验

本文仿真采用基于知识定义网络(KDN)设计的卫星光网络架构如图 2 所示。该架构由知识平面、控制平面、数据平面三部分组成^[9]。其中:

1)知识平面基于深度强化学习算法构建了一个路由策略学习智能体, 该智能体利用上述网络状态信息进行策略函数的参数学习, 以计算出在当前业务状态下合理的路由路径。通过控制器将计算得到的路由信息下发到各个卫星节点, 从而实现了对整个网络的全局实时控制。

2)控制平面的控制器负责收集诸如业务流量、网

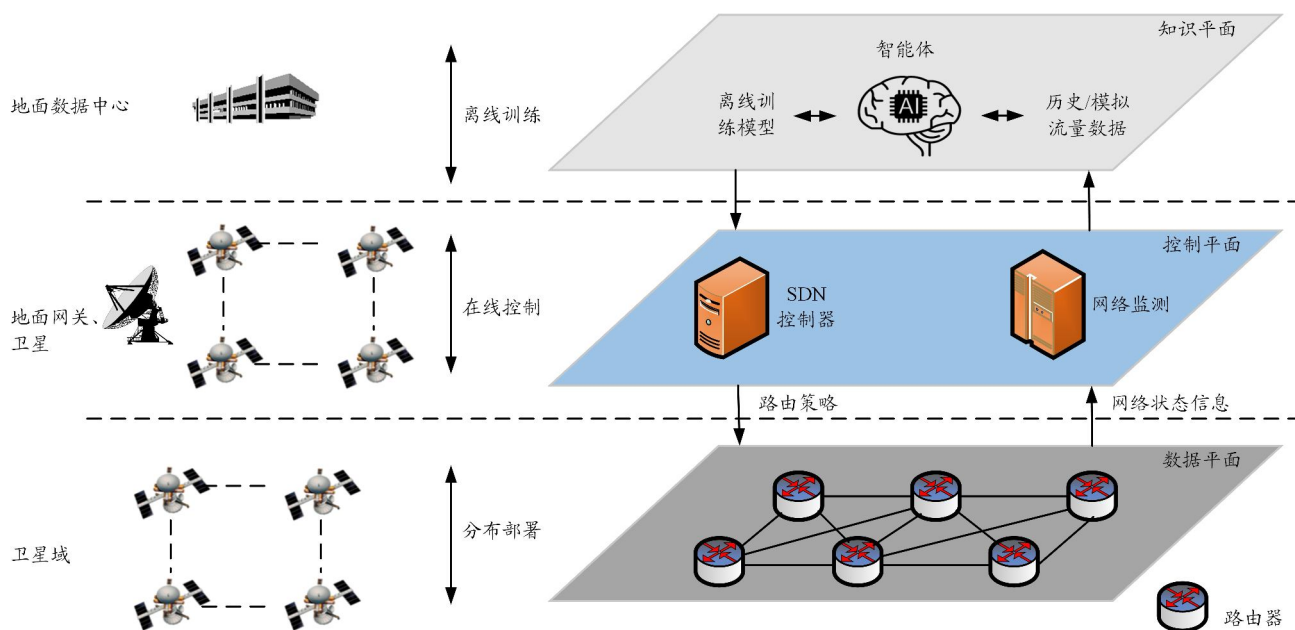


图2 基于KDN的卫星光网络架构

络时延、丢包率、链路吞吐量等网络状态信息,并实时地将其同步给知识平面。

3)数据平面是由多个卫星节点和多条激光链路构成的卫星光网络,它负责接收来自控制平面下发的业务路由信息。

由于智能路由决策模型训练需要大量的数据,而卫星光网络已公开的真实流量数据较少,这使得难以使用真实数据对模型进行训练。此外,卫星光网络的动态特性会导致网络信息维度的变化,进而增加了特征提取的难度。文献[11]分析总结了多种生成真实流量矩阵的方法,流量矩阵是新型网络路由算法和工程任务的重要输入。然而,由于流量矩阵通常被视为重要的商业机密,网络运营商不会公开这些数据,而且真实的网络流量矩阵往往无法满足不同应用场景和节点规模的网络仿真需求。ROUGHAN M 等人^[12-13]对文献[11]中采用的对数正态分布模型生成流量矩阵的方法进行了分析,认为该方法仅适用于已知的北美和欧洲网络业务流量,并没有说明是否适用于其它地区(例如亚洲),因此不具备足够的通用性。相比之下,重力模型具有以下优点:

1)通用性。重力模型更易于在全球范围内推广,因为它考虑到了业务流量的普遍规律。

2)简单性。重力模型采用更简单的指数分布方法生成随机变量,只需要一个参数。对于包含 N 个节点的网络,重力模型只需生成 $2N$ 个随机变量,而对数正态分布模型则需要生成 N^2 个随机变量。

3)准确性。重力模型生成的流量矩阵的统计特性更接近真实的业务流量分布,尤其是在流量分布的尾部。

因此,本文选择采用重力模型来生成用于训练和测试的卫星网络用户流量,以便进行仿真测试。

3.1 星座仿真

本文基于 STK 和 OMNeT++ 设计了一个卫星光网络仿真平台。该平台由 1 152 颗卫星组成的 Walker 低地球轨道星座构成,这些卫星被均匀分布在 24 个轨道上,每个轨道上有 48 颗卫星,以实现全球无缝覆盖。星座的相关参数如表 3 所示。根据信关站部署的“大三角”原则,即任意 3 个信关站可以构成一个三角形,本文为 1 152 颗卫星设计了 50 个信关站,这些站点遍布全国各大城市及重要地区,具体包括北京、上海、天津、重庆 4 个直辖市,以及山东的青岛、江苏的苏州、浙江的宁波、河南的洛阳等省级行政单位的代表性城市,同时还覆盖了河北、辽宁、甘肃、宁夏、山西、陕西、安徽、江西、海南、广西、贵州、湖南、湖北、青海、四川、福建、广东、云南、西藏、新疆、内蒙古、黑龙江、吉林等省区的重要城市。此外,还有深圳作为经济特区的代表,以及我国港澳台地区的台湾、香港、澳

表3 星座相关参数

星座类型	倾斜角/ $^{\circ}$	轨道高度/km	轨道数/个	相位因子	升交点赤经(RAAN)范围/ $^{\circ}$
Walker	89	1 050	24	9	180

门。国际方面,信关站还延伸到了莫斯科(俄罗斯)、雅库茨克(俄罗斯)、巴基斯坦、沙特阿拉伯、吉隆坡(马来西亚)、雅加达(印度尼西亚)、西班牙、匈牙利、埃及、尼日利亚等国家或城市。对于集中式拓扑结构,控制器被放置在单个信关站内,在所使用的网络切片中,控制器与馈电卫星相连。

由于每个信关站与卫星连接的时间各不相同,导致馈电时长也会有所差异,因此需要通过时间切片的方式将卫星网络划分为多个相对静止的子网络,以便进行路径规划和计算。针对低轨星座,采用了基于时间窗口的网络切片方案,并从切片数量、切片平均持续时长、平均馈电链路数量以及信关站平均连接馈电卫星的数量等方面进行了测试。馈电链路在不同时间窗口的切片统计如表 4 所示。可以看出,在低轨星座中采用时间窗口的切片方案有效降低了切片数量并消除了 1 s 切片($\Delta \geq 4.6$ s)。由于切片生成时刻不考虑剩余时长小于时间窗口 T 的链路,故随着 T 的增加切片内馈电链路数量减少,个别信关站平均连接馈电卫星数量也有小幅的降低。此外,由于国内信关站部署密集,结合星地连接的约束,部分密集部署的信关站同时连接的馈电卫星数量较少。因此,采用基于时间窗口的网络切片方案符合国内低轨星座真实网络运行状态。

表 4 不同时间窗口的切片统计

时间窗口 T/s	切片平均数量/个	切片平均持续时长 Δ/s	平均馈电链路数量/个
0	2 231	4.6	259
6	804	10.8	245
60	135	78.6	219

3.2 路由规划仿真

1 152 颗卫星节点均作为算力节点和传输节点,且卫星节点设定为业务的入口,都支持 OSU 作为业务最小封装单元的转发能力。仿真实验中卫星算力节点参数如表 5 所示,OISL-DDPG 算法的主要超参数设置如表 6 所示。将 1 152 颗星网络仿真中的“星间链路带宽”设置为 10 Gb/s,为了简化仿真实验,假定每个星间链路只包含单个波长,波长信道容量为 10 Gb/s,每个 OSU 的业务容量为 2.6 Mb/s,OSU 的灵活颗粒度调整方式为 2.6 Mb/s~10 Gb/s,卫星节点交换转发最小粒度为 ODU2(10 Gb/s)。综合考虑网络切片数量及时间窗口大小对于网络理论容量的影响,本文使用时间窗口 $T=6$ s 的 1 152 颗星网络切片,分别测试在使用 OISL-

表 5 卫星算力节点仿真参数配置

配置项	取值范围
异构计算容量/unit	CPU:[5,15] $\times 10^3$,GPU:[10,50] $\times 10^4$, 内存:[10,20] $\times 10^3$,存储:[20,30] $\times 10^5$
异构计算单位成本 ^[15]	CPU:[5,10],GPU:[100,500], 内存:[1,2],存储:[0.20,0.30]
带宽单位成本	[5,15]
业务计算需求量/unit	CPU:[5,15],GPU:[1,10], 内存:[10,20],存储:[2,10]
带宽需求/Gb/s	[0.01,10]
时延需求/ms	[10,100]

表 6 OISL-DDPG 算法超参数设置

参数	取值
mini-batch 数据	512
经验回放缓存	1 600
Critic 学习率 α_{critic}	0.001
Actor 学习率 α_{actor}	0.000 1
UO 噪声:常数 θ 、均值 μ 、方差 σ	0.0,0.2,0.4
$\gamma_1, \gamma_2, \gamma_3$	-0.33,-0.33,-0.33
训练周期/episode	100
每个训练周期步数/step	1 000

DDPG 算法和最小跳数算法^[14]时的链路利用率、平均业务时延和链路占用率等技术指标,结果如表 7 所示。可以看出,OISL-DDPG 算法相比于最小跳数算法,以增加时延为代价,获得了更高的链路利用率和链路占用率。

表 7 OISL-DDPG 算法与最小跳数算法性能对比

算法	业务数量/个	链路利用率/%	平均业务时延/ms	链路占用率/%
OISL-DDPG	5 000	7.43	89.43	96.55
	10 000	9.13	89.29	98.65
最小跳数	5 000	7.34	82.12	58.67
	10 000	8.98	79.98	60.75

3.3 算法性能

首先,本文分别测试了多约束最短路由优先(MC-SPF)算法在 10 000 个业务和 OISL-DDPG 算法在业务数 N_{ss} 分别为 10 000、5 000、1 000 时的性能,结果如图 3 所示。可以看出,与 MCSPF 算法相比,本文提出的方法性能在 3 种优化目标函数下,标准化性能值均低于 MCSPF 算法。

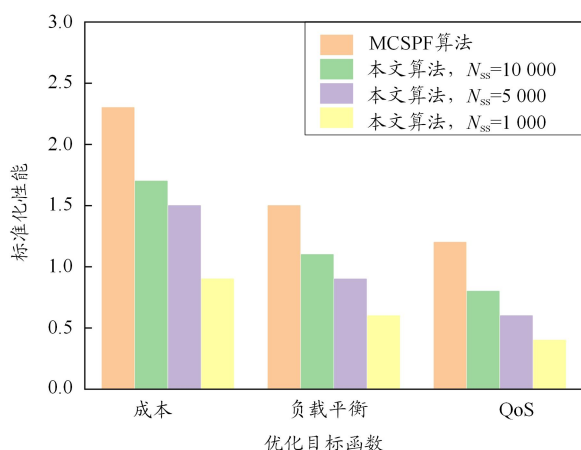
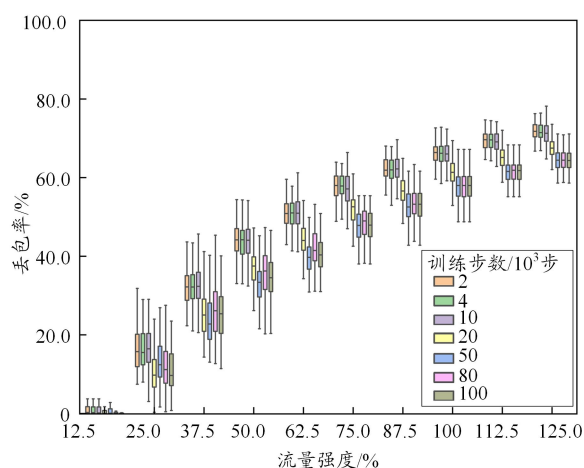


图3 OISL-DDPG 算法与 MCSPF 算法标准化性能比较

其次,在 QoS 性能优化方面,OISL-DDPG 算法的有效性和收敛性结果如图 4 所示。箱线图的底部和顶部分别表示数据的第一和第三四分位数,而须线则代表下/上四分位数(四分位数区间)内的最低/最高观测值。可以看出,随着训练步数的增加(从 2×10^3 步到

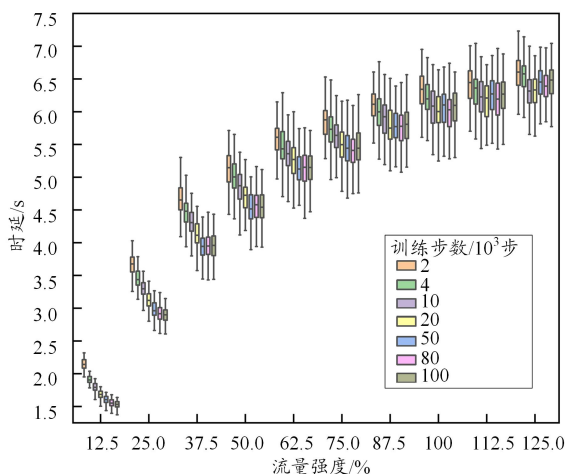


(c) 吞吐量

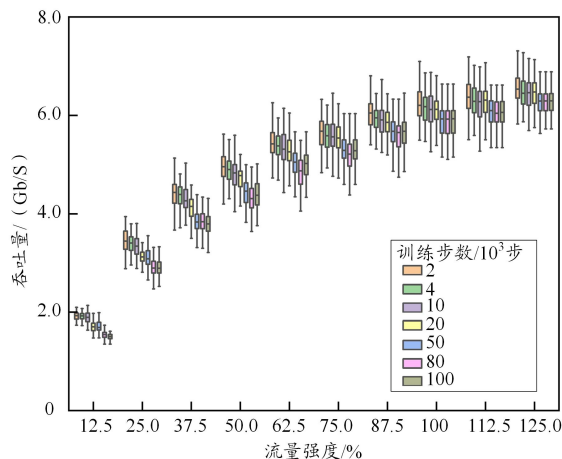
图4 OISL-DDPG 算法 QoS 性能优化

100×10^3 步),OISL-DDPG 算法能够有效降低网络时延和丢包率(如图 4 箱线图的中值所示)。具体而言,当流量强度从 12.5% 增加至 125% 时,时延和丢包率的中值分别下降了 48.4%~29.6% 和 99%~13.2%,同时链路吞吐量趋于稳定。

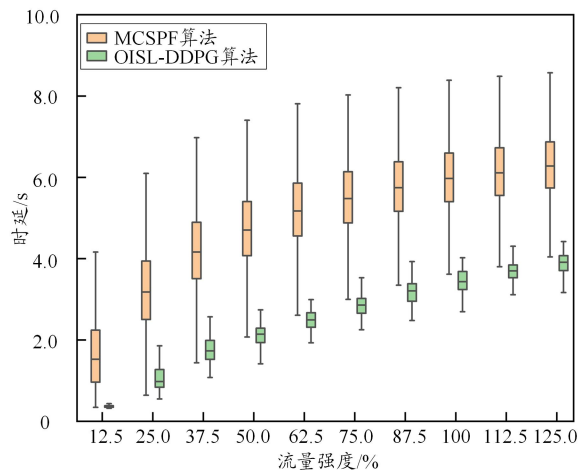
最后,将 OISL-DDPG 算法与 MCSPF 算法在 QoS 性能优化方面进行了详细对比,结果如图 5 所示。OISL-DDPG 算法经过了 100×10^3 步的训练,而 MCSPF 算法则生成了 100×10^3 条路由。在图 5(a)和图 5(b)中,由 OISL-DDPG 算法生成的路由产生的网络时延和丢包率的中值,在 12.5%~125% 的流量强度下均接近 MCSPF 算法的最小值。在图 5(c)中,当流量强度分别为 100%、112.5% 和 125% 时,OISL-DDPG 算法的吞吐量接近甚至高于 MCSPF 算法。这些仿真结果表明,与传统 MCSPF 算法相比,OISL-DDPG 算法的性能有明显提升。



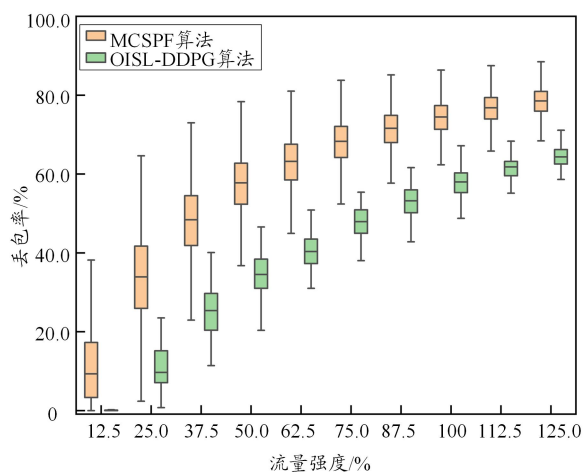
(a) 时延



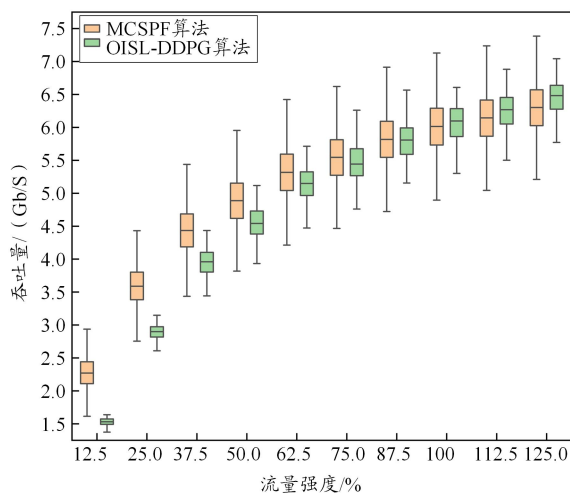
(b) 丢包率



(a) 时延



(b) 丢包率



(c) 吞吐量

图5 OISL-DDPG与MCSPF QoS性能比较

4 结束语

本文针对卫星光网络中的网络拓扑高动态性、业务承载效率与流量均衡等实际问题,根据空间环境特性提出了基于 OISL-OSU 和 DDPG 的路由优化算法——OISL-DDPG 算法。通过仿真结果验证了采用基于时间窗口的网络切片方案符合国内低轨星座的真实网络运行状态。其次,与最小跳数算法相比,OISL-DDPG 算法以增加微小时延为代价,获得了更高的链路利用率和链路占用率。此外,与 MCSPF 算法相比,OISL-DDPG 算法能够快速生成接近最优的路由,实现

业务服务资源利用量、计算资源负载均衡和传输资源负载均衡等多个优化目标。在 QoS 性能优化方面,OISL-DDPG 算法的优化路由配置所得的最小时延、最小丢包率都与 MCSPF 算法相近,而在高流量强度时吞吐量则优于 MCSPF 算法。

参考文献:

- [1] 景毅,姜春晓,詹亚锋. 面向卫星通信的 6G 通感算融合架构、技术与挑战[J]. 无线电通信技术, 2023, 49(1): 12–20.
- [2] SUN X, CAO S. A routing and wavelength assignment algorithm based on two types of LEO constellations in optical satellite networks[J]. Journal of Lightwave Technology, 2020, 38(8): 2106–2113.
- [3] 华为技术有限公司. 智能电网下一代承载技术 Liquid OTN 白皮书[S]. 广州: 华为技术有限公司, 2020.
- [4] 工业和信息化部. 光业务单元(OSU)技术要求: 2020-0587T-YD[S]. 北京: 中国通信标准化协会, 2020.
- [5] 谭艳霞,段致岩,满祥银,等. 基于 OSU 的 OTN 智能管控技术研究[J]. 电信科学, 2023, 39(10): 147–155.
- [6] MA H, ZHANG J, JI Y. Graph sequence attention network-enabled reinforcement learning for time-aware robust routing in OSU-based OTN[EB/OL]. (2022-03-06) [2024-04-01]. <https://opg.optica.org/abstract.cfm?uri=OFC-2022-Th2A.18>.
- [7] JING R, ZHANG C, ZHOU H, et al. Innovation and demonstration of optical service unit-based metro-optimized OTN technologies[J]. Journal of Optical Communications and Networking, 2022, 14(4): 236–247.
- [8] 周伟. 卫星光网络组网协议与路由算法研究[D]. 桂林: 桂林电子科技大学, 2023.
- [9] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[EB/OL]. (2015-09-09) [2019-07-05]. <https://arxiv.org/abs/1509.02971>.
- [10] MESTRES A, RODRIGUEZ-NATAL A, CARNER J, et al. Knowledge-defined networking[J]. ACM SIGCOMM Computer Communication Review, 2017, 47(3): 2–10.
- [11] NUCCI A, SRIDHARAN A, TAFT N. The problem of synthetically generating IP traffic matrices: Initial recommendations[J]. ACM Computer Communication Review, 2005, 35(3): 19–32.
- [12] ROUGHAN M. Simplifying the synthesis of internet traffic matrices[J]. ACM SIGCOMM Computer Communication Review, 2005, 35(5): 93–96.
- [13] TUNE P, ROUGHAN M. Spatiotemporal traffic matrix synthesis[J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 579–592.
- [14] 许向阳,张俊强,黄昊波. 基于 SDN 的最小跳数卫星网络路由算法[J]. 长江信息通信, 2023, 36(7): 35–39.
- [15] Google. Google cloud pricing[EB/OL]. (2023-02-14) [2024-04-08]. <https://cloud.google.com/pricing>.