文章编号 1672-6634(2023)05-0029-10 DOI 10.19728/j.issn1672－6634.2023020012

基于深度强化学习的细粒度5G RAN切片功能复用映射与路由

蔡晓烽[1],望运武[2],顾家骅[2],朱 敏[2]

(1.东南大学 信息科学与工程学院,江苏 南京 210096,2.东南大学 移动通信国家重点实验室,江苏 南京 210096)

摘 要 在无线接入网(RAN)中,5G网络功能细粒度分割机制可以有效地缓解基带功能集中处理和传输带宽所带来的压力。但这就需要高效的资源管理方法,才能极大地避免细粒度功能单元(FU)部署所导致的网络资源和计算资源的浪费。主要研究了基于功能复用(FR)策略的细粒度功能部署和路由算法。首先使用混合整数线性规划(MILP)来解决上述问题,以此生成的最优解作为基准。在时间复杂度方面,混合整数线性规划模型无法在限定时间内解决大规模问题。由此,提出了一种深度强化学习(DRL)辅助的资源优化分配方法,用于高效地解决基于功能复用机制的FU基带功能部署和路由问题。从MILP仿真结果中可以看出,功能复用机制大大提升了资源利用率,验证其有效性;从DRL的仿真结果可见,DRL方法性能明显优于启发式算法,可以使资源成本显著降低,而且在时间复杂度上,也远远优于MILP方法,具有更高的可扩展性。

关键词 网络功能细粒度分割;5G;无线接入网;深度强化学习

中图分类号 TN915.6

文献标识码 A

开放科学(资源服务)标识码(OSID)

DRL-Assisted Fine-Grained Function Placement and Routing with Reuse Scheme for 5G Network Slicing

CAI Xiaofeng[1], WANG Yunwu[2], GU Jiahua[2], ZHU Min[2]

(1. School of Information Science and Engineering, Southeast University, Nanjing 210096, China; 2. State Key Laboratory of Mobile Communications, Southeast University, Nanjing 210096, China)

Abstract The fine-grained split is an effective way to balance the baseband processing centralization and optical bandwidth saving in radio access network(RAN). However, without efficient resource management measures, fine-grained unit(FU) deployment under fine-grained segmentation can result in a great waste of resources. This paper proposes a deep-reinforcement-learning-assisted resource allocation method for baseband function placement and routing with function reuse(FR) in 5G fine-grained unit architecture. Two novel mixed integer linear programming(MILP) formulations with or without the FR scheme are presented to generate the optimal solution. With respect to the time complexity, the MILP models cannot solve large-scale problems in an acceptable execution time. Therefore, the DRL-assisted method is proposed

收稿日期:2023-02-16

基金项目:国家自然科学基金项目(62271135,62101121)资助

通讯作者:朱敏,男,汉族,博士,副教授,研究方向:光网络与光通信,E-mail: minzhu@seu.edu.cn。

for the issue in the FU-based RAN. The simulation results show that MILP with FR outperforms MILP without FR. So the validity of FR is proved. The proposed DRL-based method with the FR scheme can achieve better performance with the minimized resource cost than the MILP method without the FR scheme and the heuristic method. Also, addressing the large-scale network scenario, the DRL-based method can achieve higher scalability than MILP.

Key words fine-grained; 5G; radio access network, deep-reinforccement-learning

1 引言

随着5G通信设备数量的急剧增加,传统的无线接入网架构已不能满足设备对计算资源、带宽和延迟的严格要求。下一代无线接入网(NG-RAN)是一种很有前景的解决方案,基带单元被分割为分布式单元(DU)和集中式单元(CU)[1],这为5G发展的灵活性奠定了重要的技术基础。受到NG-RAN中基于虚拟网络功能的服务链启发,有研究人员提出了细粒度的功能分割机制[2]。细粒度功能拆分架构参考了3GPP的功能拆分方式,基带处理单元(BBU)被进一步拆分为一组多个细粒度功能单元(FU)。部署在不同网络节点中处理池(PP)里的FU可通过光路路由相互连接,共同实现基带处理功能。与现有的DU-CU架构相比,FU架构可以进一步提高基带处理的灵活性。因此研究了智能FU部署和路由策略,以进一步优化资源分配,提高资源利用率。

在FU部署过程中,我们引入一种策略,即不同切片请求中相同类型的FU功能可以部署到同一个PP,以共享PP上已经激活的实例。这种功能复用(FR)策略可以显著减少PP上所需激活的实例数量,极大提高计算资源利用率。然而,功能复用FR策略不可避免地引入实例竞争现象[3],增加了FU功能的处理延迟。因此,在基于FU的5G-RAN架构中,需要设计一种有效的功能部署和路由优化方案,利用功能复用策略,不仅可以减少PP中实例资源消耗,还能满足每个切片请求的端到端延迟阈值。

以往的工作研究了基带功能部署和路由问题。文献[4]的作者提出了一种最小化功耗的启发式算法,解决了DU-CU部署问题,达到节能的目的。在文献[2]中,作者证明了基于FU的功能部署策略可以缓解基带集中处理的压力,并能降低传输带宽的需求。但上述工作没有考虑FU的功能复用策略,而且BBU处理时延模型较为简单,仅设置为固定值。

近年来,深度强化学习(DRL)方法被应用于解决多维资源管理问题[5-9]。与启发式算法相比,DRL可以根据当前观察到的网络状态采取自适应策略。与整数线性规划方法相比,DRL方法可以在更短的执行时间内获得性能更优的可行解。文献[10]的作者提出了一种DRL算法,该算法为云无线接入网络(C-RAN)中的BBU池虚拟机的处理需求分配计算资源。在文献[11]中,作者同样采用DRL方法为C-RAN和NG-RAN提供网络功能部署策略。上述文献的结果表明,DRL的性能优于传统的启发式算法。但上述文献没有考虑基于细粒度功能拆分FU的RAN应用场景,也没有考虑功能复用策略。

本文引入功能复用策略,并提出了一种基于DRL算法的FU功能部署和路由解决方案。首先建立了一个以最小化计算资源、带宽资源和切片延迟总成本为目标的MILP模型,以此生成的最优解将作为性能基准,并研究了功能复用FR策略对MILP模型性能的影响。其次,由于MILP模型的不可扩展性,文章提出了一种DRL辅助的功能部署和路由方法,该方法不仅能获得接近最优解方案,而且能在可接受的时间内收敛,得到一种可行解。此外,提出了一种节点驱动策略来帮助DRL进行高效决策,该策略有效缓解了最短路径优先算法中可能出现的局部热点拥塞问题。

2 系统模型与问题建立

2.1 系统模型

如图1所示,弹性光网络(EON)为研究对象。每个节点都配备了EON中带宽可变的光交叉连接(BV-OXC)和可切片带宽可变应答器(S-BVT)。基于EON的5G接入网可以表示为有向图$G(V,E)$,其中V为

节点集，E 为光纤链路集。所有节点都连接到本地的 PP，部分节点连接到射频单元(RU)，图中的一个目的节点连接到 5G 核心(5GC)。每个光纤链路上都有一定数目的频隙(FS)。

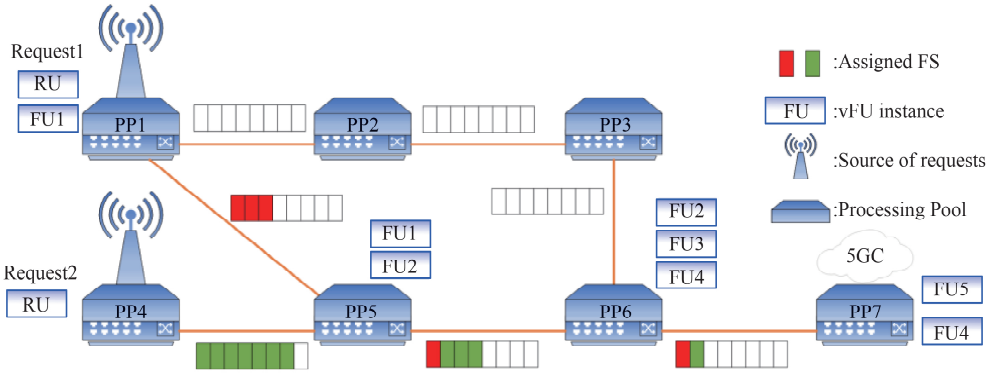
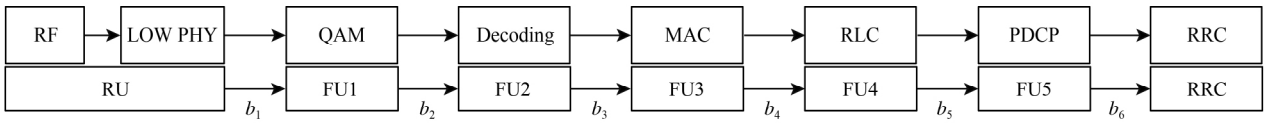


图 1　基于 EON 的 FU 架构图

图 2 展示了 FU 架构的基带处理功能分割方式，其中一个 BBU 被分为五个 FU(FU1 -FU5)[12]。FU1 负责解调；FU2 负责信道解码；FU3 负责媒体访问控制(MAC)，用于对来自不同无线电承载的数据进行多路复用；FU4 负责无线链路控制(RLC)，用来对上一层分割和重组；FU5 负责分组数据汇聚协议(PDCP)协议，用于解决安全问题。系统把包含 5 个 FU 的 RAN 切片请求从 RU 发送到 5GC，这些 FU 可以被部署在相同或不同的 PP 中，并按照规定的顺序进行链接：RU，FU1，FU2，FU3，FU4，FU5，从而生成用于基带处理的服务链。一个 RAN 切片请求可以表示为 $R[s,(b_1,\text{FU1}),(b_2,\text{FU2}),(b_3,\text{FU3}),(b_4,\text{FU4}),(b_5,\text{FU5}),b_6,d]$。其中 s 和 d 分别是切片请求的源节点和目的节点，b_k 是经过 FUk 之前的带宽需求，其中 b_6 是经过 FU5 后的带宽需求。PP 中的 FU 实例在有 FU 到达时被激活，并且消耗计算资源。从 RU 到 5GC 的路由则会导致带宽消耗。简单起见，每一个切片请求都设置了相同的计算资源需求、带宽需求和端到端延迟需求(阈值)，而不考虑服务请求的类型区别。本文假设上述需求是由给定的公式和参数[13]得到的固定值。下文探索一种有效的策略来管理网络切片的 FU 部署和路由，以最小化资源消耗和延迟，而不超过端到端延迟和资源容量的阈值。需要考虑的资源有：计算资源、带宽资源和时延。



注：$b_1=86\ 016$ Mbps；$b_2=32\ 256$ Mbps；$b_3=7\ 140$ Mbps；$b_4=4\ 500$ Mbps；$b_5=3\ 024$ Mbps；$b_6=3\ 000$ Mbps。

图 2　基带处理功能的分割选项

为了量化计算资源，引入 FU 的计算工作量 C_k^{fu}。C_k^{fu} 表示服务器处理 FUk 时所占用的计算能力，单位为每秒千兆运算次数(GOPS)，定义为[2]

$$C_k^{fu}=\alpha_k\left(3A+A^2+\frac{1}{3}M\cdot RT_{\text{coding}}\cdot L\right)\frac{PRB}{5},\tag{1}$$

式中 α_k 为各个 FUk 的因子，A 为天线数，M 为调制比特，RT_{coding} 为编码率，L 为 MIMO 层数，PRB 为物理资源块数。α_k 可通过文献[14]中的参数来计算获得。

FU 实例是一种可复用的资源，相同类型的 FU 可以共享一个实例。例如，要将一个 FU2 部署到一个无已激活实例的 PP 中，就需要在 PP 中激活一个 FU2 实例，从而产生相应的计算资源开销。之后有新的 FU2 部署到这个 PP 中时，由于 FU2 实例复用，就不会产生新的计算资源成本。

每当部署一个 FU 之后，请求的带宽需求将会改变。所需要的 FS 数量可以用$[b_k/(ML\cdot b_{fs})]$表示[2]，其中 b_k 是通过 FUk 前的带宽需求，ML 是调制格式的阶数，b_{fs} 是每个 FS 的带宽[3]。在 EON 中设置光路时，应考虑光谱的连续性、一致性和容量限制。

对于时延，可从五个方面来计算。

处理时延：一个 FU 在 PP 中的停留时间定义为 FU 的处理延迟。假设经过每个 FU 实例的 FU 流是一

个输入队列,每个实例的计算资源可视为服务于 FU 的服务器。因此,对应的 M/M/1 模型为[15]

$$T_{i,k}^{\text{proc}}=\frac{1}{RT_{k}^{\text{proc}}-RT_{\text{reach}}}, \tag{2}$$

式中 $T_{i,k}^{\text{proc}}$ 是 FUk 在节点 i 的停留时间,RT_{k}^{proc} 是服务器处理 FUk 的速率,RT_{reach}是请求的到达率。

虚拟化时延:假设启动一个 FU 实例需要的虚拟化延迟为 T_v。由于 FU 实例是一种可复用的资源,当 PP 中已经启动了相同类型的 FU 实例时,系统不会启动新实例,因此 FU 实例复用方案可以节省相应的虚拟化时延。

传播时延:从节点 i 到节点 j 的光路传播时延为 $T_{i,j}^{\text{path}}=lp/(c/n)$,其中 lp 为光路长度,c 为光速,n 为光纤折射率。

封装时延:细粒度分割采用了通用接口(GI)。GI 时延包括处理 PP 中最后一个 FU 后的封装时延和 PP 中第一个 FU 前的解封装时延。

光电切换时延:切片请求的光信号在 PP 中进行 FU 处理之前需要切换到电子域,这就造成了 T_{oeo}时延。请求离开此 PP 时,信号需要进行反向切换。

因此,请求 r 的端到端总时延为

$$T_r=\sum_{i,j,k}X_{i,j,r,k}\cdot T_{i,j}^{\text{path}}+\sum_{i}H_{r,i}\cdot T_{\text{oeo}}+\sum_{k,i}Z_{r,k,i}\cdot 2\cdot T_{k}^{\text{encap}}+\sum_{k,i}O_{r,k,i}\cdot T_{k,i}^{\text{node}}+\sum_{k}SV_{r,k}\cdot T_v, \tag{3}$$

式中 $X_{i,j,r,k}$是二进制变量,表示前 k 个 FU 处理完毕后的请求 r 是否在链路$e(i,j)$上传输。$H_{r,i}$是二进制变量,表示请求 r 是否在节点 i 上处理。$Z_{r,k,i}$是二进制变量,表示 FUk 是否为请求 r 在节点 i 上处理的最后一个 FU。$O_{r,k,i}$是二进制变量,表示请求 r 的 FUk 是否在节点 i 上处理。$SV_{r,k}$是二进制变量,表示请求 r 的 FUk 是否激活了新的实例。

2.2 MILP 模型

本文提出了一个 MILP 模型来管理的 FU 的部署和路由,目标函数设计为使计算资源、带宽容量和时延三种资源的总成本最小化。

MILP 的符号及符号含义如:S_{node}表示 EON 的节点集合;K 表示 FU 的集合,1 表示 RU,2 表示 FU1,6 表示 FU5,7 表示虚拟 FU;$F_{r,i}$表示请求 r 是否以 PPi 为起点;E_i 表示节点 i 是否为 5G 核心网的所在地;NB_{link}表示每个链路上的 FS 个数;C_{cp}表示每个 PP 的计算资源容量;T_k^{fu} 表示 FUk 的处理时延;$T_{\max}$表示每个请求的时延阈值;N 表示一个大数值正整数;NB_k^{fu} 表示 FUk 的 FS 需求;RD 表示请求的到达速率。

MILP 的变量及变量含义如下:二进制变量 D_i,表示 PPi 是否被使用;整数变量 M_{FSI},表示已使用的 FS 索引最大值;二进制变量 $X_{i,j,r,k}$,表示请求 r 是否在处理完 k 个 FU 后在 $e(i,j)$上传输;二进制变量 $XF_{i,j,f,r,k}$,表示请求 r 是否在处理完 k 个 FU 后在 e(i,j)的 FSf 上传输;二进制变量 $UF_{i,j,f}$,表示$e(i,j)$的 FSf 是否被使用;连续变量 $T_{r,k}^{\text{req}}$,表示请求 r 中 FUk 的处理时延;连续变量 $T_{k,i}^{\text{node}}$,表示 FUk 在 PPi 中的处理时延;二进制变量 $Q_{k,i}$,表示 FUk 是否在 PPi 中处理;整数变量 $U_{k,i}$,表示 FUk 在 PPi 中的复用次数;二进制变量 $a_{r,k,i}$,是 $T_{r,k}^{req}$的辅助变量;连续变量 $G_{k,i}$,是 $T_{k,i}^{\text{node}}$的辅助变量;连续变量 $P_{r,k,i}$,是 $G_{k,i}$的辅助变量。

目标函数 Minimize:

$$\begin{aligned}\alpha\cdot\sum_{i}D_i+\beta\cdot M_{\text{FSI}}+\gamma\cdot\Big[&\sum_{i,j,r,k}X_{i,j,r,k}\cdot T_{i,j}^{\text{path}}+\sum_{r,i\in[1,|S_{\text{node}}|)}H_{r,i}\cdot T_{\text{oeo}}\\&+\sum_{r,k,i\in[1,|S_{\text{node}}|)}Z_{r,k,i}\cdot 2\cdot TE_k+\sum_{r,k}T_{r,k}^{\text{req}}+\sum_{k,i}Q_{k,i}\cdot T_v\Big]。\end{aligned} \tag{4}$$

MILP 的目标是同时最小化计算资源、带宽资源和请求的端到端延迟。目标函数中第一项是所需 PP 的数量,第二项是 M_{FSI},第三项是请求的端到端延迟。权重 α,β,γ 用于平衡每个资源对目标函数的贡献。

限制条件

FU 部署限制

$$O_{r,k,i}=\begin{cases}F_{r,i},k=1\\E_i,k=|K|\end{cases},\forall r,i \tag{5}$$

$$\sum_i O_{r,k,i}=1,\forall r,k \tag{6}$$

$$D_i\leqslant\sum_{r,1<k<|K|}O_{r,k,i}\leqslant N\cdot D_i,\forall i \tag{7}$$

式(5)确保请求从起点出发，到终点 5GC 结束。式(6)确保每个请求中的每个 FU 选择且仅选择一个 PP 进行部署。式(7)保证了 D_i 值为 1 时，有 FU 在 PPi 上处理，值为 0 时则无。

$$Q_{k,i}\leqslant\sum_r O_{r,k,i}\leqslant N\cdot Q_{k,i},\forall k,i \tag{8}$$

$$\sum_{1\leqslant k<|K|}Q_{k,i}\cdot C_k^{fu}\leqslant C_{cp},\forall i \tag{9}$$

式(8)保证了 $Q_{k,i}$ 值为 1 时，有 FUk 在 PPi 上处理，值为 0 时则无。式(9)保证了 PP 的计算工作量不超过其计算能力限制。

路由限制

$$\sum_{i\neq j,k<|K|}X_{i,j,r,k}=\begin{cases}0,F_{r,j}=1\\1,E_j=1\end{cases},\forall r,j \tag{10}$$

$$\sum_{i\neq r,k<|K|}X_{i,r,b,k}-\sum_{j\neq r,k<|K|}X_{i,j,r,k}=0,F_{r,j}=0,E_j=0,\forall r,j \tag{11}$$

$$\sum_{k<|K|}X_{i,j,r,k}+\sum_{k<|K|}X_{j,i,r,k}\leqslant 1,\forall i,j(i\neq j),r \tag{12}$$

式(10)确保了请求在部署过程中不会回到起点，同时保证最终到达 5GC。式(11)确保建立了从起点到 5GC 的光路。式(12)保证了一个链路不能被一个请求重复使用。

$$O_{r,k,i}=\sum_{j,k\leqslant l<|K|}X_{i,j,r,l},if\ F_{r,i}=1,\forall r,k,i \tag{13}$$

$$2\cdot O_{r,k,i}\leqslant\sum_{i,1\leqslant l1\leqslant k-1}X_{i,nd,r,l1}+\sum_{j,k\leqslant l2<|K|}X_{nd,j,r,l2}\leqslant O_{r,k,i}+1,if\ F_{r,i}=0,\forall r,k,nd \tag{14}$$

式(13)表示如果节点 i 是请求 r 的起点，并且请求 r 在这里处理了 FUk，请求 r 将会以 FUk 已处理的状态离开节点 i。式(14)表示如果请求 b 的 FUk 在节点 nd 被处理，请求 r 将以 FUk 未处理的状态到达这个节点，并以 FUk 已处理的状态离开节点 nd。

$$XF_{i,j,0,r,k}=0,\forall i,j(j\neq i),r,k \tag{15}$$

$$\sum_{f\leqslant f'\leqslant\min(f+NB_k^{fu},NB_{\text{link}})}XF_{i,j,f',r,k}\geqslant NB_k^{fu}+N\cdot(XF_{i,j,f,r,k}-XF_{i,j,f-1,r,k}-1)-(1-X_{i,j,r,k})\cdot N,$$
$$\forall i,j(j\neq i),r,k,f \tag{16}$$

式(15)确保 FS 的索引从 1 开始，而不是从 0。式(16)确保处理完 FUk 后，为请求 r 选择一组连续的 FS。

$$UF_{i,j,f}\geqslant\sum_{r,k}XF_{i,j,f,r,k},\forall i,j(j\neq i),f \tag{17}$$

$$M_{\text{FSI}}\geqslant f\cdot UF_{i,j,f},\forall i,j(j\neq i),f \tag{18}$$

$$M_{\text{FSI}}\leqslant NB_{\text{link}} \tag{19}$$

式(17)保证了频谱不重叠。式(18)计算网络的 M_{FSI}，式(19)保证 M_{FSI} 不能超过每条链路上的 FS 数量。

时延限制

$$-H_{b,nd}\leqslant\sum_{i\neq nd}XF_{i,nd,f,r,k}-\sum_{j\neq nd}XF_{nd,j,f,r,k}\leqslant H_{r,nd},\forall nd,f,r,k \tag{20}$$

式(20)为频谱的连续性约束。

$$U_{k,i}=\sum_r O_{r,k,i},\forall i,1\leqslant k<|K| \tag{21}$$

$$H_{r,i}\leqslant\sum_k O_{r,k,i}\leqslant N\cdot H_{r,i},\forall r,i\neq|S_{\text{node}}| \tag{22}$$

$$2 \cdot Z_{r,k,i} \leqslant O_{r,k,i} - O_{r,k+1,i} + 1 \leqslant Z_{r,k,i} + 1, \forall r, 2 < k < |K|, i, \tag{23}$$

式(21)计算了 FUk 在 PPi 上的复用次数。式(22)确保了请求 r 在节点 i 上处理时 $H_{r,i}$ 值为 1,否则为 0。式(23)确保了 FUk 为请求 r 在节点 i 上处理的最后一个 FU 时 $Z_{r,k,i}$ 值为 1,否则值为 0。

$$T_{k,i}^{\text{node}} = \frac{T_k^{fu}}{1 - RD \cdot U_{k,i} / C_k^{fu}}, \forall k, i, \tag{24}$$

式(24)计算了节点 i 上每个 FUk 的处理延迟,属于非线性约束。这里引入一个辅助变量 $G_{k,i}$ 来线性化。

$$G_{k,i} = T_{k,i}^{\text{node}} \cdot U_{k,i}, \forall k, i, \tag{25}$$

$$N \cdot (Q_{k,i} - 1) \leqslant U_{k,i} - \sum_r r \cdot L_{r,k,i} \leqslant - N \cdot (Q_{k,i} - 1), \forall k, i, \tag{26}$$

$$T_{k,i}^{\text{node}} = \sum_r \frac{1}{r} \cdot G_{k,i} \cdot L_{r,k,i}, \forall r, k, \tag{27}$$

$$\sum_r L_{r,k,i} = 1, \forall k, i, \tag{28}$$

式(25~28)列出了变量 $G_{k,i}$ 的约束条件,其中有一个新的非线性约束条件(27),该约束条件也需要线性化。式(26)确保 FU 处理延迟的计算只与实际参与处理的节点有关。

$$P_{r,k,i} = G_{k,i} \cdot L_{r,k,i}, \forall r, k, i, \tag{29}$$

$$P_{r,k,i} \leqslant G_{k,i}, \forall r, k, i, \tag{30}$$

$$P_{r,k,i} \geqslant G_{k,i} + N \cdot (L_{r,k,i} - 1), \forall r, k, i, \tag{31}$$

$$P_{r,k,i} \leqslant N \cdot L_{r,k,i}, \forall r, k, i, \tag{32}$$

$$N \cdot (Q_{k,i} - 1) \leqslant T_{k,i}^{\text{node}} - \sum_r \frac{1}{r} \cdot P_{r,k,i} \leqslant - N \cdot (Q_{k,i} - 1), \forall k, i, \tag{33}$$

式(29~33)列出了变量 $P_{r,k,i}$ 的约束条件。

$$T_{r,k}^{\text{req}} = \sum_i O_{r,k,i} \cdot T_{k,i}^{\text{node}}, \forall r, 1 \leqslant k < |K|, \tag{34}$$

式(34)计算了请求 r 中每个 FUk 的处理时延,属于非线性约束。这里引入一个辅助变量 $a_{b,k,r}$ 来线性化。

$$a_{r,k,i} = O_{r,k,i} \cdot T_{k,i}^{\text{node}}, \forall r, 1 \leqslant k < |K|, i, \tag{35}$$

$$a_{r,k,i} \leqslant T_{k,i}^{\text{proc}}, \forall r, 1 \leqslant k < |K|, i, \tag{36}$$

$$a_{r,k,i} \geqslant T_{k,i}^{\text{node}} + N \cdot (O_{r,k,i} - 1), \forall r, 1 \leqslant k < |K|, i, \tag{37}$$

$$a_{r,k,i} \leqslant N \cdot O_{r,k,i}, \forall r, 1 \leqslant k < |K|, i, \tag{38}$$

$$T_{r,k}^{req} = \sum_i a_{r,k,i}, \forall r, 1 \leqslant k < |K|, \tag{39}$$

式(35)为 $a_{r,k,i}$ 的定义式,其约束条件为式(36~38)。式(39)为线性化的结果。

$$\sum_{i,j,k} X_{i,j,r,k} \cdot T_{i,j} + \sum_i H_{r,i} \cdot T_{\text{oeo}} + \sum_{k,i \in [1, |S_{\text{node}}|]} Z_{r,k,i} \cdot 2 \cdot T_k^{\text{encap}} + \sum_k T_{r,k}^{\text{req}} + 5 \cdot T_v \leqslant T_{\max}, \forall r, \tag{40}$$

式(40)确保满足延迟要求。简单起见,所有的限制都加入了虚拟化延迟。

3 深度强化学习模型

值得一提的是,上述提出的混合整数线性规划 MLIP 模型仅仅适用于规模网络小,请求较少的情况,虽然可求得最优解,但不适用于大网络场景,缺乏可扩展性。因此,本文设计一种深度强化学习 DRL 模型,它可以适用于网络规模大,请求较多的情况,而且从后面仿真中可以看出,性能优于启发式算法,可求得近似最优解,具有较好的可扩展性。

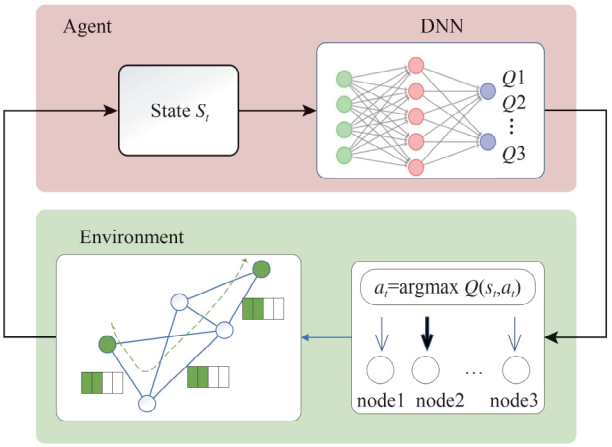


图 3 基于 DRL 的资源分配

图 3 展示了 DRL 如何通过与基于 FU 的 RAN 环境进行迭代交互来学习和做出决策。在每个决策时刻，DRL 从环境接收更新的状态。深度神经网络（DNN）将状态输入和切片请求的需求映射为多个 Q 值。每个 Q 值分别与一个动作相关。DRL 会选择执行 Q 值最大的动作。在动作执行后，DRL 从 RAN 环境中获得奖励。至此，DRL 执行了一个循环，并且从这次经历中学习如何行动，以有效地解决 BBP&R 问题。

算法 1　DRL 辅助的节点驱动 FU 部署

```
for 每个回合：
    初始化状态 s_t；
        for 每个 RAN 请求 R：
            把 R 分解为 5 个部分
            获取当前状态 s_t
            for 每个部分：
                产生随机数 c ∈[0,1]
                if c > ε then 选取一个随机动作 a_t
                else 选择 a_t = argmax_a Q(s_t, a)
                if a_t 可行 then
                    部署到 FU 实例
                    从上一次选择的 PP 到当前 PP 之间获取 K 条最短路径
                    for k ∈ 1,2,…,K：
                        计算 I_k
                    end
                    在 I_k：最小的路径上部署光路
                    获取奖励 r_t
                else
                    获取惩罚 r_t
                存储经验数据 (s_t, a_t, r_t, s_{t+1})
                如果经验数据超过阈值，则使用损失函数更新 DNN 网络
            end
        end
end
```

与传统的最短路径优先映射方案不同，算法 1 提出了 DRL 辅助的节点优先映射方案。算法的第 1，2 行初始化要部署的切片请求的状态。在第 3，4 行中，为方便 DRL 进行决策，算法将一个请求 R 分解为 5 个单元，即$(b_1, \mathrm{FU1}),(b_2, \mathrm{FU2}),(b_3, \mathrm{FU3}),(b_4, \mathrm{FU4}),(b_5, \mathrm{FU5}, b_6)$，分 5 步来部署一个请求。FU$k$ 与第 k $(k \neq 5)$个频谱资源需求 b_k 组合为 R 中的第 k 个元素，其中 R 中的第 5 个元素$(b_5, \mathrm{FU5}, b_6)$包括第 5 个计算资源需求 FU5，第 5 个和第 6 个频谱资源需求（即 b_5 和 b_6）。该方案主要由节点映射来驱动。首先 DRL 选择一个 PP 节点来部署 FU。然后在所选节点和上一个已选节点之间路由，以此来选择光路。

在第 5 行，DRL 在第 t 个时间步中获得将要部署第 r 个切片请求的第 k 个 FU 时的环境状态，状态信息由第 k 个 FU 的信息和当前基于 FU 的 RAN 信息组成。状态定义为

$$s_t^{r,k-1}(\boldsymbol{M}_{r,k-1}, D_{r,k-1}^{\mathrm{used}}, C_{r,k-1}^{\mathrm{avail}}, T_{r,k-1}^{\mathrm{total}}, ACT_{r,k-1}, \boldsymbol{M}_{\mathrm{FSI}_{r,k-1}}),$$

式中 $\boldsymbol{M}_{r,k-1}$ 是一个矩阵，描述了需要部署哪种 FU，以及每种 FU 在每个节点上已经部署了多少个。$D_{r,k-1}^{\mathrm{used}}$、$C_{r,k-1}^{\mathrm{avail}}$、$T_{r,k-1}^{\mathrm{total}}$和 $ACT_{r,k-1}$为一维数组。$D_{r,k-1}^{\mathrm{used}}$表示 RAN 中哪些节点已经部署了 FU。$C_{r,k-1}^{\mathrm{avail}}$为每个 PP 节

点的可用计算资源。$T_{r,k-1}^{\text{total}}$是目前已部署第 r 个请求的前 $k-1$ 个 FU 的累计延迟。$ACT_{r,k-1}$是部署请求 r 的前面 $k-1$ 个 FU 的所采取的动作记录。$\boldsymbol{M}_{\text{FSI}_{r,k-1}}$是部署了第 r 个请求的 $k-1$ 个 FU 后的 MFSI。

在第 6～9 行中,DRL 通过 ε-greedy 策略[11]的来选择动作。动作空间包括 $|S_{\text{node}}|$ 个动作,其中 $|S_{\text{node}}|$ 是 RAN 中的节点总数。每个动作都与一个候选 PP 节点对应。在第 10～12 行中,如果选择了可行的动作,FU 将被部署到指定的节点中。然后,计算所选节点和上一个所选节点之间的 K 个最短路径。在第 13～16 行中,I_k 定义为部署所需 FS 的最小起始索引。选择具有最小 I_k 的路径,并在光路中的光纤上分配所需的 FS。在部署切片请求的最后一个 FU 时,应该在所选节点和上一个所选节点以及 5GC 节点之间分别分配两条光路。

如果发生以下三种情况,动作选择了同一个请求的前几个 FU(上一个 FU 除外)已经选择的某个 PP,可用资源不足,或者超过请求的延迟阈值,动作就不可行。在第 18～19 行中,如果这个动作是不可行的,则给 DRL 一个较大的惩罚。在第 17 行中,当行动可行时,根据式(3)设置负值作为每一步的奖励。

在第 20～24 行中,经验数据以(s_t, a_t, r_t, s_{t+1})的方式记录下来。为了训练 DRL 模型,DNN 的参数通过 $loss = E\{[r_t + \zeta \max Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]^2\}$进行更新,其中 $r_t + \zeta \max Q(s_{t+1}, a_{t+1})$为理想 Q 值,$Q(s_t, a_t)$表示更新前的 Q 值。当损失函数收敛到一个较小的固定值时,DNN 的输出很好地拟合了动作的最优 Q 值。

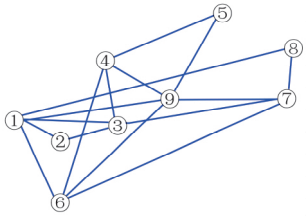


图 4 拓扑图

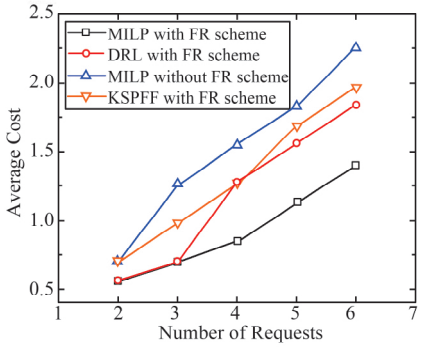


图 5 总成本

4 实验结果

本节通过仿真来评估 MILP 和 DRL 辅助算法的性能,这两种算法将和启发式算法 K 最短路径首次匹配(KSPFF)相比较,同时也比较了 FR 方案的优劣。仿真采用了 9 节点网络,如图 4 所示。节点 1 设置为请求 1～4 的起点,节点 2 设置为请求 5～7 的起点。节点 9 作为 5GC 的所在点,是所有请求的目的地。本仿真中 MILP 和 DRL 均运行在的 Intel i7-10750H CPU 2.60 GHz 16G RAM 环境,其中 MILP 在 GUROBI 9.1.2(https://www.gurobi.com/)上编程并执行,DRL 神经网络的搭建和训练使用了 Theano 优化编译器。详细仿真参数如下:天线个数为 32,PRB 个数为 500,调制比特为 6 bit,MIMO 层数为 2 层,光路存在 100 个 FS,学习率为 10^{-4},衰减因子为 0.9,训练批次大小为 32,经验回放池容量为 4×10^4,探索率为 0.8。

图 5 给出了代表不同算法消耗的总资源的平均成本。随着请求数量的增加,这几种算法都会消耗更多的资源,因为更多的请求会带来更多的资源需求。需要注意的是,三种使用 FR 方案的算法比不使用 FR 方案的 MILP 消耗的资源更少,这是由于多个切片请求之间实现了 PP 上的 FU 实例资源共享。除此以外,DRL 与 KSPFF 相比,节省了更多的平均资源成本。

图 6 的(a)和(b)中,使用的 PP 和 MFSI 数量随着请求数量的增加而增加。有 FR 方案的算法比没有 FR 方案的算法使用更少的 PP 和 MFSI。这是因为一个 FU 实例可以由来自不同请求的相同类型的 FU 共享,从而开启更少的 FU 实例,使用更少的 PP。较低的 MFSI 是由于 FU 在地理位置上的集中节省了 FU 之间的带宽。而 KSPFF 虽然所选择的节点较为集中,需要最少的 PP,但不同请求路径的选择也由于首次匹

配的规则而过于集中，导致了高 MFSI。从图 6(c)中可以看出，三种算法都能满足时延要求，时延性能相近。FR 方案虽然会带来排队延迟，但通过复用 FU 实例，避免了虚拟化延迟和光电切换延迟。因此，使用 FR 方案不会带来太大的额外延迟。

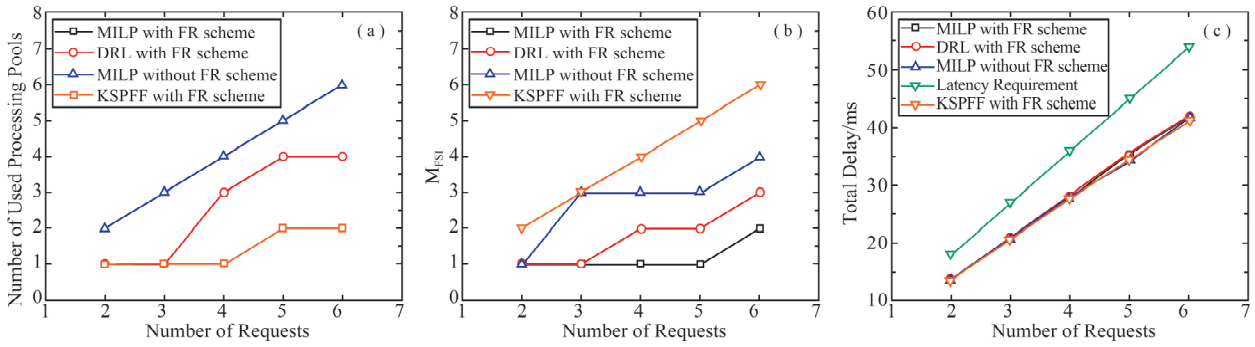


图 6　(a) PP 的使用个数；(b) MFSI；(c) 总时延

图 7 为 DRL 的训练过程。图 7 显示，DRL 的损失在 500 回合内迅速下降，之后收敛到 0。在 1 300 个回合内，总奖励会因为探索过程而随机变化。在 1 300 个回合之后，总奖励收敛到一个较高的值，约为 −0.7。结果表明，本文基于 DRL 的算法具有较快的收敛速度，从而达到近似最优解。

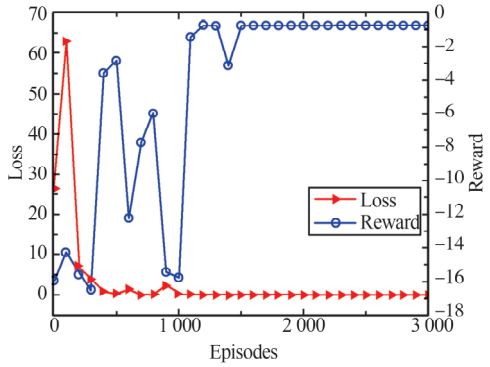


图 7　DRL 算法的训练过程

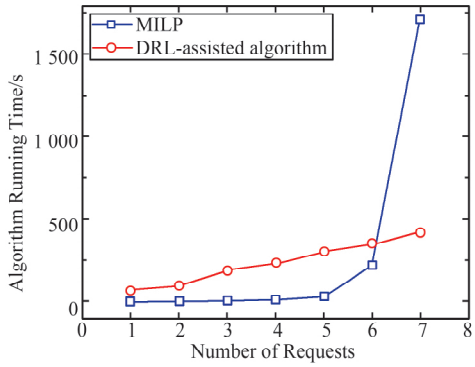


图 8　DRL 算法和 MILP 算法的处理时间

图 8 对比了 MILP 和 DRL 算法的运算时间。在小于 6 个请求的情况下，MILP 比 DRL 算法消耗的运行时间更短，因为小规模的情况下，GUROBI 中的数学运算只需几步就能精确求解，而 DRL 算法需要足够的迭代来进行训练。但是，在超过 6 个请求的情况下，MILP 的运行时间明显多于 DRL 算法的运行时间，并且 MILP 的运行时间随着请求数目增加而呈指数级增长，而 DRL 算法运算时间呈线性增长趋势。因此，在大规模场景下，基于 DRL 的细粒度功能部署和路由算法在运行时间方面优于 MILP 模型。在请求较少的情况下，例如切片业务请求的低谷期，使用 MILP 能够快速地给出最优 FU 部署方案，而在请求较多的场景下，例如切片业务请求的高峰期，使用 DRL 是更好的选择。

5　结论

本文提出了一种 DRL 辅助的资源优化分配方法，高效地解决了基于功能复用机制的 FU 基带功能部署和路由的问题。首先采用 MILP 模型，对问题进行建模分析。从 MILP 仿真结果中可以看出，功能复用策略大大提升的资源利用率，验证其有效性；在 DRL 仿真中，DRL 算法比不采用功能复用策略的 MILP 具有更好的性能。可见，DRL 算法能够充分发挥资源复用的潜力，有效提高系统性能。而且，DRL 算法在总成本上优于 KSPFF 启发性算法，这体现了 DRL 算法在不同网络状态下能够灵活地做出有效决策。另一方面，在大规模请求情况下，DRL 算法比 MILP 花费的时间要少得多，更具扩展性。因此，我们提出的 DRL 辅助方法，有望为 5G RAN 架构中细粒度功能布局和路由算法提供出色的性能。

参 考 文 献

[1] GIANNONE F. Impact of virtualization technologies on virtualized RAN midhaul latency budget: a quantitative experimental evaluation [J]. IEEE Communications Letters, 2019, 23(4): 604-607.

[2] XIAO Y M, ZHANG J W, JI Y F. Can fine-grained functional split benefit to the converged optical-wireless access networks in 5G and beyond[J]. IEEE Transactions on Network and Service Management, 2020, 17(3): 1774-1787.

[3] GU J H, ZHU M, SHEN T Y, et al. Delay-aware and resource-efficient VNF-service chain deployment in inter-datacenter elastic optical networks[C]// 2021 Optical Fiber Communications Conference and Exhibition(OFC), 2021: 1-3.

[4] XIAO Y M, ZHANG J W, JI Y F. Energy-efficient DU-CU deployment and lightpath provisioning for service-oriented 5G metro access/aggregation networks[J]. Journal of Lightwave Technology, 2021,39(17): 5347-5361.

[5] **陈斌,顾家骅,朱敏,等. 基于深度强化学习的** OFDMA-PON **三维资源分配研究与性能分析**[J]. **聊城大学学报(自然科学版)**, 2020, 33(6): 40-46.

[6] CHEN X L, LI B J, PROIETTI R, et al. DeepRMSA: a deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks[J]. Journal of Lightwave Technology, 2019, 37(16): 4155-4163.

[7] ZHU R J, LI G, Wang P, et al. DRL-based deadline-driven advance reservation allocation in EONs for cloud-edge computing[J]. IEEE Internet of Things Journal, 2022, 9(21): 21444-21457.

[8] ZHU M, CHEN Q, GU J H, et al. Deep reinforcement learning for provisioning virtualized network function in inter-datacenter elastic optical networks[J]. IEEE Transactions on Network and Service Management, 2022, 19(3): 3341-3351.

[9] XU L F, HUANG Y C, XUE Y, et al. Deep reinforcement learning-based routing and spectrum assignment of EONs by exploiting GCN and RNN for feature extraction[J]. Journal of Lightwave Technology, 2022, 40(15):4945-4955.

[10] RODOSHI R T, KIM T, CHOI W. Deep reinforcement learning based dynamic resource allocation in cloud radio access networks[C]// 2020 International Conference on Information and Communication Technology Convergence(ICTC), 2020:618-623.

[11] GAO Z G, YAN S Y, ZHANG J W, et al. Deep reinforcement learning-based policy for baseband function placement and routing of RAN in 5G and beyond[J]. Journal of Lightwave Technology, 2022, 40(2): 470-480.

[12] ASKARI L, MUSUMECI F, SALERNO L, et al. Dynamic DU/CU placement for 3-layer C-RANs in optical metro-access networks[C]// 2020 22nd International Conference on Transparent Optical Networks(ICTON), 2020:1-4.

[13] NTT DOCOMO, R3-162102: CU-DUSplit: Refinement for Annex A[S]. 3GPP, France 2016.

[14] SHEHATA M, ELBANNA A, MUSUMECI F, et al. Multiplexing gain and processing savings of 5G radio-access-network functional splits[J]. IEEE Transactions on Green Communications and Networking, 2018, 2(4): 982-991.

[15] CHEN X L, ZHU Z Q, GUO J N, et al. Leveraging mixed-strategy gaming to realize incentive-driven VNF service chain provisioning in broker-based elastic optical inter-datacenter networks[J]. Journal of Optical Communications and Networking, 2018, 10(2): A232-A240.

(责任编辑:赵海军)