一种改善多接入边缘计算网络切片服务质量的双层闭环协同资源分配框架

徐骏涛[①] 范兴刚*[②] 徐常福[③] 沈民扬[①] 梁玉珠[④] 王 田[④]

[①](浙江工业大学计算机科学与技术学院/软件学院 杭州 310023)

[②](浙江工业大学之江学院 绍兴 312030)

[③](江西财经大学软件与物联网工程学院 南昌 330013)

[④](北京师范大学人工智能与未来网络研究院 珠海 519085)

摘 要：在5G/6G驱动的多接入边缘计算(MEC)环境中，提高资源利用率的同时，确保异构网络切片的服务质量(QoS)具有重要意义。然而，当前方法在异构环境下缺乏动态适应性，无法协同优化缓存、带宽和计算资源，存在资源利用率下降、服务成功率下降等问题。该文首先形式化定义了网络切片资源分配问题，并通过将多维0-1背包问题归约到该问题来证明其是NP-难问题。然后提出了一种具有缓存感知的双层闭环协同资源分配框架，上层构建一种群体协同的全局搜索机制，通过候选解演化、历史最优引导与自适应参数调整，在复杂约束条件下生成高质量资源候选分配方案，下层构建一个多维权重评估模型以准确将低延迟、高带宽的服务需求指标转化为QoS约束，通过上下双层闭环协同实现缓存、带宽和计算资源的联合优化，进而改善了异构网络切片的QoS。大量实验表明，与基线方法相比，所提方法使资源利用率提升2.29%～24.50%，用户评分提升4.13%～59.34%，为异构MEC环境提供了一种鲁棒的资源分配解决方案。

关键词：多接入边缘计算；网络切片；资源分配；QoS保障

中图分类号：TN929.5; TP393 **文献标识码**：A **文章编号**：1009-5896(2026)12-0001-11

DOI: 10.11999/JEIT260156 **CSTR**: 32379.14.JEIT260156

1 引言

在5G/6G与多接入边缘计算(Multi-access Edge Computing, MEC)[1,2]深度融合的智能通信时代，网络架构日益趋向柔性化与智能化，面对海量终端接入、多样化服务需求以及严格的服务质量(Quality of Service, QoS)保障要求[3]，传统“集中式-静态化”的网络资源调度方式已难以满足新一代通信系统的弹性服务需求[4]。网络切片(Network Slicing)作为5G/6G时代的关键使能技术，通过网络功能虚拟化(Network Functions Virtualization, NFV)与软件定义网络(Software Defined Network, SDN)技术实现物理资源的逻辑隔离与按需配置[5]，能够为增强型移动宽带(Enhanced Mobile Broadband, eMBB)、低时延高可靠通信(Ultra-Reliable & Low-Latency Communication, URLLC)、大规模机器类通信(Massive Machine-Type Communications, mMTC)等多样化业务提供“定制化”资源保障[6]。网络切片体系的结构图如图1所示。目前，网络切片技术已广泛应用于多媒体内容分发[7]、车联网[8]、工业物联网[9]等多个应用场景中，有效支撑了差异化服务的高效部署与隔离运行。

因此，MEC与网络切片的结合具有重要的研究价值。目前的MEC网络切片资源分配方法主要从3个维度展开研究：在分层式架构优化方面，ETSI MEC标准框架为切片管理提供了基础，基于SDN/NFV的架构实现了资源虚拟化与灵活编排，近期研究则通过引入边缘智能技术提升实时自适应能力；在智能算法优化方面，深度强化学习、博弈论和多种启发式算法被用于提升资源调度效率；在缓存感知资源优化方面，研究多集中于缓存与计算的联合优化以降低延迟。

然而，随着网络业务流量呈现高度波动性与动态性，边缘计算环境中的计算、存储、缓存与带宽等资源的动态竞争加剧，使得网络流量的高动态性与节点资源受限之间的矛盾日益尖锐[4]。传统静态配置或单一启发式算法难以实现资源的最优调度与QoS保障，导致资源利用率下降、服务违约率升高、用户体验劣化等一系列问题[10,11]。

为解决上述问题，本文提出一种面向异构网络切片的具有缓存感知的双层闭环协同资源分配框架，上层构建群体协同的全局搜索机制，通过候选解演化、历史最优引导与自适应参数调整，生成高

收稿日期：2026-02-06；改回日期：2026-06-17；网络出版：2026-06-30
*通信作者： 范兴刚 xgfan@zjut.edu.cn
基金项目：国家自然科学基金(U25A20436)，山东省重点研发计划(2025TSGCCZZB0026)

Foundation Items: The National Natural Science Foundation of China (U25A20436), Shandong Provincial Key R&D Program (2025TSGCCZZB0026)

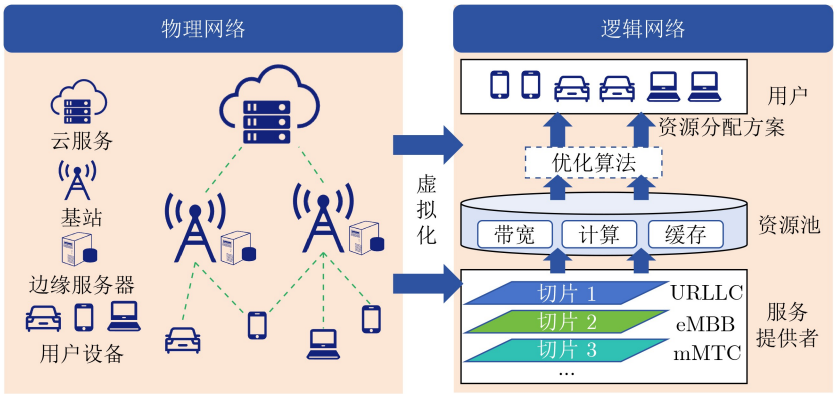


图 1 支持MEC的网络切片体系结构图

质量资源候选分配方案，下层构建多维权重评估模型，考虑不同类型业务的差异化需求，通过加权归一化方法以准确将不同的服务需求指标转化为QoS约束，通过上下双层闭环协同实现缓存、带宽和计算资源的联合优化，进而改善了异构网络切片的资源利用率和QoS。与现有研究相比，本文的主要贡献包括3个方面：

(1)构建了一种新颖的双层闭环协同资源分配框架，在基站侧设计缓存感知模型，统筹缓存容量与用户实时需求。其核心创新点在于，融合全局候选解演化思想与历史最优引导机制，通过基因片段重组增强解空间遍历能力，同时双向引导搜索方向，提升全局探索能力与快速收敛特性，并引入熵权法计算权重，实现缓存感知的端到端资源联合优化。该框架在严格满足带宽、时延、丢包率等多维QoS约束的前提下，显著提升资源利用率，具备动态响应业务波动能力，有效规避局部最优，在求解质量上取得明显提升。

(2)提出一种融合业务速率、时延敏感度等多维指标的量化评估模型，创新性地将静态QoS指标转化为权重约束，通过速率、时延等多维度指标的自适应加权，提升资源分配的精度和稳定性，有效匹配不同时段与业务类型下的切片需求，提升资源分配的精度与稳定性。

(3)开展大量多场景仿真实验以验证方法有效性。通过在不同业务负载、资源波动与用户密度条件下进行模拟实验，实验结果表明，所提方法使资源利用率提升2.29%～24.50%，用户评分提升4.13%～59.34%，在保障用户QoS的同时，显著降低了资源碎片化并且提高了切片服务质量，表现出较高的全局适应性和运行效率。

2 相关工作

2.1 分层式架构优化

在架构设计级别，ETSI MEC标准框架[12]首先标准化了切片生命周期管理过程，包括切片创建、编排、监控和发布，为资源调度提供了标准化的接口和管理规范。这类工作强调功能模块解耦与接口标准化，但在动态优化能力方面依赖外部策略，灵活性相对有限。随着网络技术的发展，基于SDN/NFV的虚拟化切片架构成为主流研究方向。一些研究[13,14]中提出的基于SDN/NFV的切片编排体系结构通过网络功能虚拟化和软件定义网络实现了资源虚拟化的基础，可以实现网络资源的按需切片与灵活编排，但仍面临跨域协调与资源耦合复杂性等挑战。在此基础上，研究逐步向多层或多域协同编排架构演进。Sindjoung等人[15]提出的多域分层切片架构将系统划分为服务编排层、域级编排层及资源控制层，实现跨域资源的统一管理与协同优化，从而提升切片部署的可扩展性与灵活性。但目前，大多数研究通常依赖于预定义的静态规则或策略，很难自适应地响应实时服务负载的波动。

2.2 智能算法优化

在优化算法领域，近年来，基于智能算法的资源分配研究激增。文献[16]采用深度强化学习(Deep Reinforcement Learning, DRL)来优化切片调度，但很难捕获MEC场景中多维资源之间相互依赖的全部复杂性。博弈论方法将资源分配建模为切片或用户之间的策略性互动，在一定的均衡条件下实现公平[17]。虽然这些方法可以确保均衡分配，但它们通常忽略时变QoS约束的显式建模，并且无法满足高度动态和并发物联网环境中的个性化服务要求。启发式算法由于其轻量级性质和易于实现而仍然具有吸引力[18]。基于先导预测的多目标粒子群优化算法[19]引入了一种先导预测策略，有效地平衡了探索和开发，同时提高了收敛速度和解质量。平均信道带宽资源和计算资源分配算法[20]采用资源预分配机制，在MEC网络服务提供商和用户设备之间均匀分布上/下行信道资源和计算资源。然而，大多数这些研究主要针对核心网络级资源分配或路由优

化，忽略了计算、通信和缓存MEC边缘资源的耦合优化挑战。

2.3 缓存感知资源优化

缓存已成为MEC网络切片中减少延迟和节省带宽的关键促成因素，特别是在以频繁重复访问相似内容为特征的物联网场景中。文献[21,22]研究了缓存和计算的联合优化，表明策略缓存放置可以显著提高端到端QoS。文献[23,24]提出了用于MEC的联合缓存、计算和通信分配模型，优化了异构资源约束下的用户感知延迟。然而，这些方法通常假设静态或准静态缓存可用性，这与真实世界的MEC部署不一致，其中缓存空间可能会因收回策略、系统更新或能量约束而波动。最近的研究考虑了时变缓存动态。在文献[25]中，提出了一种自适应缓存感知调度框架，以基于预测的内容流行度和缓存可用性动态调整资源分配，从而实现更好的服务连续性。尽管如此，大多数缓存感知算法将缓存视为一个独立的子问题，未能将其完全集成到网络切片中计算和带宽资源的联合优化中。

3 网络切片的模型构建

本节主要介绍网络切片中网络模型、传输模型、卸载模型和缓存模型的构建。

3.1 网络模型

考虑一种多边网络切片网络，其中1个基站n包含U个用户设备$\mathcal{U}=\{1,2,\cdots,U\}$，每个用户设备都和基站通过有线或者无线连接。基站配备一个边缘服务器提供计算和缓存资源。不同的用户设备都与基站物理连接。

网络切片提供商将整个网络整合为一个资源池。具体而言，网络切片提供商从网络基础设施提供商处租赁管理域内的资源，这些资源包括总带宽W(Hz)、计算能力F(CPU周期/s)和缓存C(MB)资源并将其抽象为虚拟资源，从而根据服务质量为用户提供不同类型的服务。

由于网络切片是根据服务类型进行分类的，假设存在S种类型的切片$\mathcal{S}=\{1,\cdots,s,\cdots,S\}$。由于不同类型的业务有着不同的需求，不同的切片会占用不同的资源类型。本研究针对3GPP标准所定义的3大典型应用场景进行了差异化建模：eMBB切片以高带宽资源保障为重点，主要面向视频流媒体、高清会议等高吞吐量业务，其资源分配需满足最小带宽约束；URLLC切片则对端到端时延与传输可靠性有严格限制，适用于车联网、工业控制等关键任务，时延和可靠性构成其主要约束条件；mMTC切片侧重于支持高连接密度与大规模设备接入的稳定性，典型应用于智慧城市传感网络，其核心约束为连接密度。

切片s的资源元组表示为$\mathcal{R}_s=\{w_s,f_s,c_s\}$，其中$w_s$,$f_s$和$c_s$分别代表基于切片分配资源的带宽、计算和缓存资源。切片占用系统总资源的相应比例取决于它需要提供的资源服务。

所有切片的资源分配总和应不超过总资源的数量。应满足条件为

$$\sum_{s\in S} w_s \le W \tag{1}$$

$$\sum_{s\in S} f_s \le F \tag{2}$$

$$\sum_{s\in S} c_s \le C \tag{3}$$

对于切片s来说，它需要达到的QoS用q_s表示，$q_s=\{\varphi_s^{\mathrm{com}},\varphi_s^{\mathrm{con}}\}$。其中，$\varphi_s^{\mathrm{com}}$表示切片允许的最大计算卸载延时，$\varphi_s^{\mathrm{con}}$表示切片允许的最大内容传输延时。

3.2 传输模型

传输模型表示网络数据的传输过程。网络切片的实施需要利用虚拟化技术将物理网络划分为多个独立的虚拟网络。这使得用户设备能够与虚拟网络实现逻辑上的关联，同时保持其与物理网络的连接。

在底层物理网络中，采用正交频分复用(Orthogonal Frequency-Division Multiplexing, OFDM)技术进行数据传输。主要考虑小区间的同频干扰。

与基站n关联的用户设备u的信噪比(Signal to Interference plus Noise Ratio, SINR)对于切片s的服务可以计算为

$$\Gamma_{u,n}=\frac{P_{u,n}H_{u,n}}{\sigma^2+\sum\limits_{i=1,i\neq u}^{U}\sum\limits_{j=1,j\neq n}^{N}P_{i,j}H_{i,j}} \tag{4}$$

其中，$P_{u,n}$表示用户设备u到基站n的发射功率，σ^2表示附加白高斯噪声的方差，$H_{u,n}$表示用户设备u和基站n之间的信道增益。

由于处于城市场景中，因此信道增益可以计算为[26]

$$H_{u,n}=\sqrt{G_{\mathrm{tx}}}\sqrt{G_{\mathrm{rx}}}\mathrm{PL}(d) \tag{5}$$

其中，G_{tx}是发送端的天线增益，G_{rx}是接收端的天线增益，$\mathrm{PL}(d)$表示大尺度路径损耗因子。

在城市场景中，使用路径损耗模型进行建模，路径损耗$\mathrm{PL}(d)$表示为

$$\mathrm{PL}(d)=\mathrm{PL}\left(d_0\right)+10m\lg\left(\frac{d}{d_0}\right) \tag{6}$$

其中，d表示用户设备u到基站n的距离，$\mathrm{PL}\left(d_0\right)$是参考距离$d_0$=1 km处的路径损耗因子，$m$=3是路径损耗指数。

定义$w_{u,s}^{\mathrm{up}}$表示切片s为用户设备u分配的上行通信带宽资源，$\Gamma_{u,n}^{\mathrm{up}}$表示接收端为基站$n$的上行链路的信噪比。上行数据速率$r_{u,s}^{\mathrm{up}}$可以表示为

$$r_{u,s}^{\mathrm{up}}=w_{u,s}^{\mathrm{up}}\log_2\left(1+\Gamma_{u,n}^{\mathrm{up}}\right) \tag{7}$$

定义$w_{u,s}^{\mathrm{down}}$表示切片s为用户设备u分配的下行通信带宽资源，$\Gamma_{u,n}^{\mathrm{down}}$表示接收端为用户设备$u$的下行链路的信噪比。下行数据速率$r_{u,s}^{\mathrm{down}}$可以表示为

$$r_{u,s}^{\mathrm{down}}=w_{u,s}^{\mathrm{down}}\log_2\left(1+\Gamma_{u,n}^{\mathrm{down}}\right) \tag{8}$$

3.3 卸载模型

卸载模型表示用户设备向基站的边缘服务器请求卸载任务的过程。

定义$M_{u,s}=\{D_{u,s},C_{u,s}\}$表示用户设备$u$通过切片$s$向边缘服务器传输的计算任务。其中$D_{u,s}$表示用户设备$u$通过切片$s$向边缘服务器传输的任务数据量(单位：bit)，$C_{u,s}$表示边缘服务器处理该用户设备$u$的任务所需的计算量(单位：CPU周期/s)。

定义$T_{u,s}^{\mathrm{up}}$为将任务数据从用户设备u通过切片s传输到边缘服务器所需的时间(上行传输延迟)

$$T_{u,s}^{\mathrm{up}}=\frac{D_{u,s}}{r_{u,s}^{\mathrm{up}}} \tag{9}$$

定义$T_{u,s}^{\mathrm{com}}$为计算处理时延

$$T_{u,s}^{\mathrm{com}}=\frac{C_{u,s}}{f_{u,s}} \tag{10}$$

其中，$f_{u,s}$表示切片s为用户设备u分配的计算资源(单位：CPU周期/s)。

由于任务完成后生成的数据量相对较少，因此下载所需时间相比于任务总体数据量而言可以忽略不计。因此，总卸载延时可以定义为

$$T_{u,s}^{\mathrm{off}}=T_{u,s}^{\mathrm{up}}+T_{u,s}^{\mathrm{com}} \tag{11}$$

为了满足切片的QoS要求，用户设备u的总计算卸载延迟需要满足切片s指定的最大延迟要求，即

$$T_{u,s}^{\mathrm{off}}\le\varphi_s^{\mathrm{com}}. \tag{12}$$

3.4 缓存模型

缓存模型表示用户请求的数据在边缘服务器上的缓存机制，缓存机制使得用户设备请求的内容可以直接从边缘服务器的内容库中获取，从而无需涉及远程数据中心。

定义L为缓存数据的集合$\mathcal{L}=\{1,2,,\cdots,L\}$，其中第$l$个内容的数据量为$D_l$。

定义缓存指示符$x_{s,l,u,t}$表示边缘服务器的第l个缓存内容在时间槽t内是否存储了用户u通过切片s访问的内容。

若$x_{s,l,u,t}$=1，表示在时间槽t内该内容存在，即用户设备u可以请求该内容；若$x_{s,l,u,t}$=0，则在时间槽t内该内容需要从远程数据中心请求。

同时，需要确保切片s存储的内容不超过其存储容量，即

$$\sum_{l\in L,u\in U}x_{s,l,u,t}d_l\le c_s,\forall s,t \tag{13}$$

定义$T_{u,s,l,n,t}^{\mathrm{con}}$为在时间槽$t$内用户$u$请求的数据$l$时的交付时延

$$T_{u,s,l,n,t}^{\mathrm{con}}=\frac{D_l}{r_{u,s}^{\mathrm{down}}}+\frac{D_l}{r_n^{\mathrm{b}}}\left(1-x_{s,l,u,t}\right) \tag{14}$$

其中，$r_{u,s}^{\mathrm{down}}$表示用户设备u与基站n之间的下行通道速率，r_n^b表示基站n与远程数据中心通过回程链路的数据传输速率。

用户切片s中的内容交付时延应满足切片中规定的设备u在切最大内容交付时延要求，即

$$T_{u,s,l,n,t}^{\mathrm{con}}\le\varphi_s^{\mathrm{con}},\forall k,s,l,n,t \tag{15}$$

4 网络切片资源分配问题建模与理论分析

4.1 问题建模

决策变量：每个时间槽内基站会给每个用户分配各种资源。定义缓存指示符$x_{s,l,u,t}$表示在时间槽t内边缘服务器的第l个缓存内容是否存储了用户u通过切片s访问的内容

$$x_{s,l,u,t}=\{0,1\},s\in S,u\in U,t\in T \tag{16}$$

基站会维护$x_{s,l,u,t}$以决定用户访问的数据是否需要缓存在基站。

目标：在满足用户QoS约束的前提下，最小化服务商的总资源成本。

首先分析如何满足用户QoS，用户QoS由对最大计算卸载延时φ_s^{com}和最大内容交付延时φ_s^{con}经过计算得到。

运营商的单位资源成本π^w, π^f, π^c分别为通信、计算和缓存资源的单位成本。单个用户设备k的使用切片s的成本消耗可以列为

$$\begin{aligned}\mathrm{Cost}_{u,s,t}^{w,f,c}&=\mathrm{Cost}_{u,s,t}^{w}+\mathrm{Cost}_{u,s,t}^{f}+\mathrm{Cost}_{u,s,t}^{c}\\&=\pi^w(w_{\mathrm{u},s,t}^{\mathrm{up}}+w_{u,s,t}^{\mathrm{down}})+\pi^f f_{u,s,t}+\pi^c c_{u,s,t}\end{aligned} \tag{17}$$

假设每个用户同一时间只使用1个切片，并且服务商会尽量满足用户的QoS，所以$\mathrm{Cost}_{u,s,t}^{w}$和$\mathrm{Cost}_{u,s,t}^{f}$可以根据切片$s$指定的QoS计算得到。而缓存资源的花费可以列为

$$\mathrm{Cost}_{u,s,t}^{c}=\pi^{c}c_{u,s,t}=\pi^{c}\left(\sum_{l\in L,u\in U}x_{s,l,u,t}d_{l}\right)\tag{18}$$

所以最终的优化目标可以列为

$$\sum_{t\in\mathcal{T}}\sum_{u\in\mathcal{U}}\sum_{s\in\mathcal{S}}\mathrm{Cost}_{u,s,t}^{w,f,c}\tag{19}$$

约束条件为：(1), (2), (3), (12), (13), (15), $x_{s,l,u,t}\in\{0,1\}$, $\forall s,l,u,t$。

4.2 理论分析

定理1　网络切片资源分配问题是NP难问题。

证明　通过把著名的NP-难多维0-1背包问题(Multiple Knapsack Problem, MKP)通过多项式时间归约到网络切片资源分配问题来证明它是NP-难问题[27]。

给定一组n个项目$\mathcal{J}=\{1,2,\cdots,n\}$，其中每个项目$i$具有利润$p_i$，并为第$m$个资源维度消耗$r_{i,j}$个资源单元$j$，并且给定资源容量向量$(R_1,R_2,\cdots,R_m)$，MKP旨在选择项目的子集以在不违反任何资源约束的情况下最大化总利润

$$\max\sum_{i=1}^{n}p_{i}x_{i}\tag{20}$$

$$\text{s.t.}\sum_{i=1}^{n}r_{i,j}x_{i}\le R_{j},\ \forall j=1,2,\cdots,m\tag{21}$$

$$x_{i}\in\{0,1\},\ \forall i\in\mathcal{J}\tag{22}$$

对于任何MKP实例，本文构造网络切片资源分配问题的实例，如下所示：每个项目$i\in\mathcal{J}$对应于可行的缓存决策$x_{s,l,u,t}$，其唯一地标识用户u在时间t通过切片s访问内容l的潜在缓存和服务操作。

当$x_{s,l,u,t}$=1时，利润p_i映射到网络切片资源分配问题中总服务成本的相应减少。由于MKP使利润最大化，而网络切片资源分配问题使成本最小，因此目标通过2元性进行调整。每个资源消耗$r_{i,j}$映射到特定类型的资源使用：

缓存大小d_l作为缓存负载；传输带宽$w_{u,s}^{\mathrm{down}}$和$w_{\mathrm{u,s}}^{\mathrm{up}}$作为通信资源使用；计算要求$f_{u,s}$作为计算资源消耗。

背包容量约束R_j映射到网络切片资源分配问题中的资源约束

$$\sum_{l,u}x_{s,l,u,t}d_{l}\le c_{s}\tag{23}$$

$$T_{k,s,l,\mathrm{n},t}^{\mathrm{con}}\le\varphi_{s}^{\mathrm{con}}\tag{24}$$

$$x_{s,l,u,t}\in\{0,1\}\tag{25}$$

注意，网络切片资源分配问题中的目标函数和约束保持了MKP的结构：这两个问题都寻求在可加资源约束下选择2元决策的子集。网络切片资源分配问题还包括QoS保证，这些QoS保障要么被建模为额外的背包约束，要么集成到资源参数中。

对于$|\mathcal{J}|$和m，这种缩减可以在多项式时间内执行，因为每个项和资源维度对应于元组(s,l,u,t)和一个物理约束。

由于MKP是NP难的，并且本文为网络切片资源分配问题提供了多项式时间归约，因此网络切片资源分配问题也是NP难问题。　　证毕

5　双层闭环协同优化框架

上一节定义了在满足用户QoS约束下最小化总资源成本的问题。由于高维变量使传统数学规划难以求解，本文提出一种名为QCache的双层闭环协同优化框架，由全局探索层与多维权重评估层构成，采用混合启发式算法导航解空间、避免局部最优并处理非线性约束，如图2所示，上层全局探索层融合全局候选解演化思想与基于历史最优引导的粒子更新机制来解决复杂的、NP难的网络切片资源分配优化问题。下层是多维权重评估层，通过使用熵权方法实时调整不同切片类型的优先级来动态平衡冲突目标。

5.1 全局探索层

全局探索层作为优化流程的第1阶段，负责粗粒度的全局搜索以发现潜在的优质解，为后续局部优化提供初始种群。该层采用群体演化策略与粒子更新机制相结合的混合式启发策略，平衡搜索精度与效率。

初始化阶段通过随机或分布式策略生成种群个体，每个个体对应计算、存储、带宽等资源的分配方案。在群体演化部分采用二进制或实数编码，通过选择、交叉和变异操作更新种群以增加多样性，避免局部最优。同时引入粒子群优化的速度-位置更新机制，使个体向当前最优解移动以增强局部搜索能力、加快收敛。更新公式为

$$\begin{aligned}v_{i,d}^{k+1}=&\omega v_{i,d}^{k}+c_{1}r_{1}\left(p_{i,d,\mathrm{pbest}}^{k}-x_{i,d}^{k}\right)\\&+c_{2}r_{2}\left(p_{d,\mathrm{gbest}}^{k}-x_{i,d}^{k}\right)\end{aligned}\tag{26}$$

$$x_{i,d}^{k+1}=x_{i,d}^{k}+v_{i,d}^{k+1}\tag{27}$$

为了避免过早收敛和陷入局部最优，设计了动态权重更新机制和精英保留策略：前者通过监控迭代次数与适应度变化分阶段调整搜索行为；后者保留当前最佳适应度个体进入下一代，防止丢失全局最优解。同时结合自适应参数调整技术，根据搜索状态动态调整交叉率、变异率和惯性权重，增

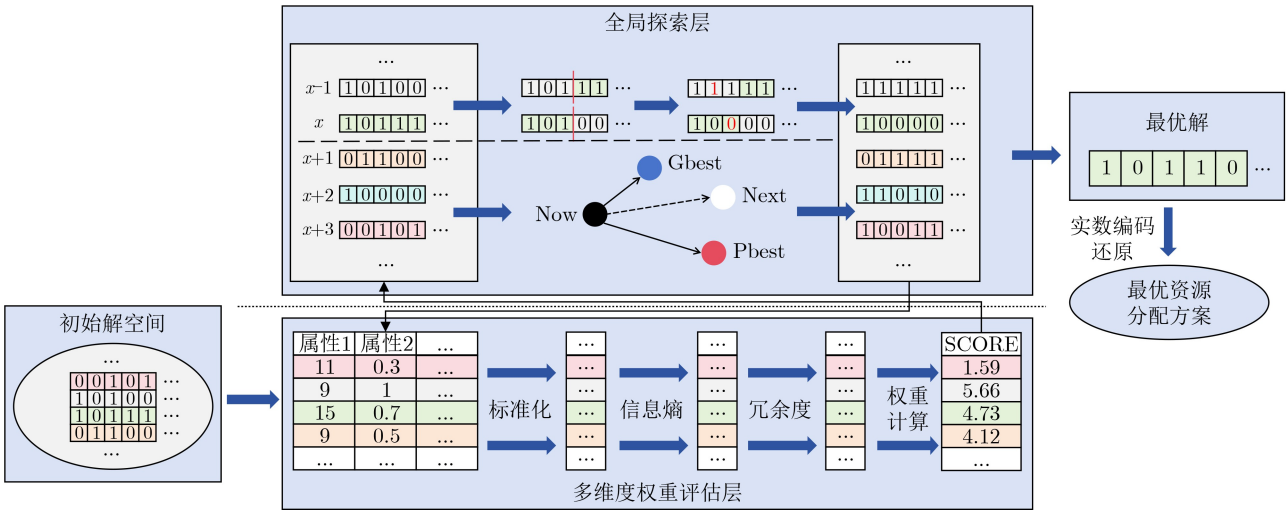


图 2 双层闭环协同优化框架示意图

强搜索过程的灵活性和鲁棒性。多次迭代后，输出高质量候选解集，并从中缓存需要缓存的用户的数据。

该层不仅提供了解空间结构的信息，而且通过多算法协作增强了解的多样性和初始质量，显著提高了整个优化系统的性能和稳定性。

具体算法如算法1所示。

5.2 多维度权重评估层

多维权重评估层作为双层协同优化框架的局部细化阶段，基于全局探索层的候选解，从用户需求、QoS和资源利用效率等多个维度构建综合目标函数，通过多指标加权方法对候选解进行评价和排序。首先对每个评价维度定量建模：用户满意度通过需求与资源的匹配度计算，资源利用率用实际资源分配与资源上限的比值来表示，网络延迟基于当前方案估计，采用熵权法分配各维度的权重以融合多目标评估。实际应用中可根据业务场景灵活调整权重配置，以满足特定的优化目标。

该层输出每个候选解的综合得分，作为局部搜索迭代的依据。在评价结果的驱动下，上层混合启发式算法可以再次调整个体解，以获得更准确的资源分配方案。

具体算法如算法2所示。

5.3 算法复杂度与收敛性分析

对于所提两层协调优化框架，其核心算法的时间复杂性分析如下：全局探索层(算法1)的时间复杂性为$O\left(T\cdot P\cdot\left(N+D+\log_2 P\right)\right)$，其中$T$表示迭代次数，$P$表示种群大小，$N$是用户请求数，$D$对应于解向量的维数。算法2的时间复杂度为$O\left(N\right)$，主要成本来自每个个体需要$O\left(1\right)$的时间复杂度计算信号干扰加噪声比、传输速率、延迟和综合得分。

收敛性分析如下，全局探索层中，个体更新遵循粒子群优化机制，其速度与位置更新如式(26)

算法 1 缓存资源分配计算算法

输入：用户需求集REQ，基站缓存集CACHE，切片资源配置集SLICE；

输出：基站缓存资源最优分配集UPDATE。

(1) 使用随机资源分配策略初始化种群pop，初始化种群fitness、个体最优pbest和种群最优gbest的适应度以及基站的缓存策略UPDATE；

(2) for 种群的每次迭代 do

(3) 基于多维权重模型计算个体的fitness；

(4) 更新pbest和gbest；

(5) 基于fitness从大到小对pop进行排序；

(6) 选择pop小于精英比例的部分为pop1，其余部分为pop2；

(7) 对pop1执行锦标赛选择、单点交叉和高斯变异操作，对pop2执行更新位置和速度操作；

(8) end for

(9) 基于多维权重模型计算每个个体fitness；

(10) 输出缓存分配最适合的策略UPDATE。

和式(27)所示。在精英保留策略作用下，每一代均保留当前最优个体，使得全局最优适应度序列满足单调非增性，即$f(g^{t+1})\le f(g^t)$，且由于目标函数存在下界，根据单调有界收敛定理可知算法在适应度意义下必然收敛。同时，在期望意义下对更新过程分析，当参数满足$0<\omega<1$且$c_1+c_2<4$时，粒子系统满足均方稳定性，其搜索轨迹将收敛至由个体最优与全局最优构成的吸引域。结合自适应惯性权重策略实现由全局探索向局部开发的平滑过渡，最终保证算法在复杂约束条件下具备良好的渐进收敛性与稳定性。

6 实验评估

6.1 实验设计及评估指标

为了验证所提出的双层闭环协同优化框架在网

络切片资源分配问题中的有效性与优越性，本文在PyCharm 2024.3.4 环境下进行了一系列仿真实验。实验平台为Windows 10操作系统，使用Python语言进行算法实现与仿真。参比方案覆盖了进化计算、群智能、图神经网络和随机基线四种主流范式，具体如下：

(1)改进的遗传算法方案(Genetic Algorithm, GA)：代表进化计算范式，是资源分配领域的经典基准算法，被广泛应用于切片调度优化，与本文同属启发式优化范畴；

(2)基于个体与群体最优经验引导的自适应优化算法(Particle Swarm Optimization-Leader, PSO-Leader)[19]：代表群智能优化范式，引入先导预测策略平衡探索与开发，与本文问题设定(NP难优化、多约束)一致；

(3)基于图神经网络的资源分配方法(Graph SAmple and aggreGatE, GraphSAGE)[28]：代表图神经网络(Graph Neural Network, GNN)驱动的资源优化范式，通过对网络拓扑结构进行嵌入表示学习，利用邻域聚合机制捕获节点间关联关系，实现对资源分配策略的智能优化；

(4)不考虑服务质量的随机资源选择方案(Randomly Select Slices Without Considering Service Quality, NCSQ)[17]：作为最差情况基线，在不考虑任何服务质量约束的条件下随机分配资源，反映了缺乏优化指导时系统性能的下界。

实验从资源利用率与用户服务满意度两大核心维度衡量算法性能。资源利用率反映了系统总体资源的利用情况，通过实际使用的资源总量与系统提供的资源总量之比计算得到；平均服务得分反映了用户的服务满意度，由用户在各个切片的资源分配比(获得/需求)加权求和得到。

算法 2　多维度权重评估计算算法

输入：用户需求UREQ，用户与基站的距离DIS，用户数据包PACK，基站缓存状态STATE，切片给用户分配的资源配置集RES；

输出：用户的综合评分SCORE。

(1) 根据式(4)计算SINR，根据式(7)和式(8)计算rate，根据式(11)计算delay；

(2) if 所需数据没有在基站中缓存

(3) delay需根据式(14)加上额外的时延；

(4) end if

(5) 归一化速率rate和时延delay等属性；

(6) 计算各属性的信息熵；

(7) 计算各属性的冗余度；

(8) 输出每个指标加权后的分数SCORE。

实验参数如表1所示。仿真参数设计基于实际边缘计算与网络切片应用场景，参照了3GPP对5G网络切片的标准分类逻辑[17,29]，符合实际网络部署中多类切片共存的应用场景。仿真中设置4个切片的具体对应关系如下：切片1(高带宽需求型，类eMBB场景)，切片2(低时延高可靠型，类URLLC场景)，切片3(大规模接入型，类mMTC场景)，切片4(混合需求型，同时承载多种业务类型的混合流量)。

6.2　总体对比分析

图3是各基线方案和本文提出的QCache在相同的资源数量和相同的用户需求的情况下所得到的资源利用率和平均服务得分。实验表明，QCache以80.30%的资源利用率和1.109的平均服务得分位居首位。相较其他方案，QCache的资源利用率提升2.29%～24.50%，平均服务得分提升4.13%～59.34%。这一对比结果显示，QCache算法通过双层闭环机制，在资源分配优化与用户体验保障方面实现了更好的平衡，验证了其在网络切片资源分配场景中的有效性与先进性。

6.3　参数影响分析

6.3.1　总缓存资源变化实验

实验模拟边缘节点缓存容量因系统更新或能耗限制发生波动。通过设置多种缓存大小，对比不同状态下资源利用率与用户服务满意度的变化。实验结果如图4所示。相较其他方案，QCache的资源利用率提升了2.43%到27.53%，平均服务得分提升了10.81%到119.56%。结果表明，所提出框架在缓存受限时仍能保持较高的服务满意度，在缓存资源充足时进一步提升资源利用效率，得益于群体演化策略和历史最优引导的耦合使得资源能被更好的分配。

表 1　仿真参数[17,29]

参数名称	参数取值
切片数量	4
总带宽资源	75 MHz
总计算资源	150 GHz
总缓存资源	75 GB
每个切片分配的带宽资源	[0,75] MHz
每个切片分配的计算资源	[0,150] GHz
每个切片分配的缓存资源	[0,75] GB
数据包大小	[5,125] MB
数据缓存标识$x_{s,l,u,t}$	0,1
最大计算卸载延时	[0.02,0.2] s
最大内容传输延时	[0.5,5] s
回程链路传输速率	[50,200] Mbit/(s·Hz)

6.3.2 总用户数量变化实验

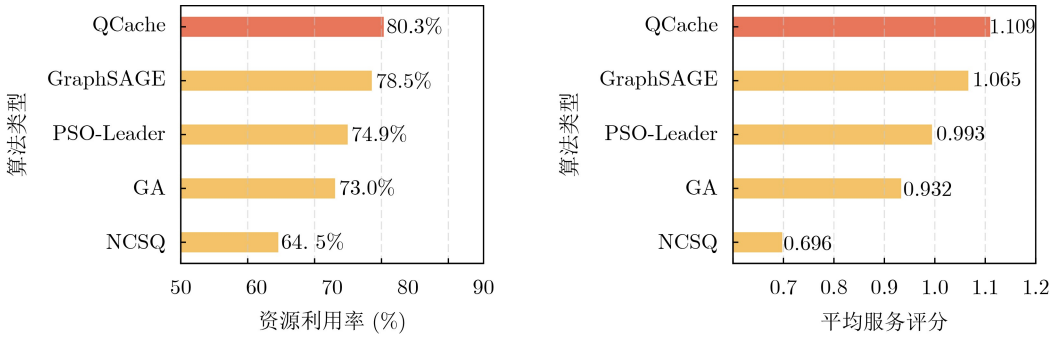
模拟接入不同的总用户数量，评估系统在用户密集接入情况下的可扩展性与调度稳定性。实验结果如图5所示。结果表明，QCache的资源利用率最高达到85.83%，平均服务得分最高达到3.06。本框架利用多维度评估框架有效缓解了资源冲突与拥塞问题，使得资源利用率和用户满意度保持在可接受范围内。

6.3.3 用户需求随时刻变化实验

模拟用户对切片资源的需求随时变化模式。采用滑动窗口统计方法对资源分配策略进行分析。实验结果如图6所示。相较基线方案，QCache的资源利用率平均提升了9.20%，平均服务得分提升了23.45%。实验表明，QCache不仅降低了资源浪费率，还提升了QoS保障能力，验证了该框架的双层闭环协同机制在动态环境中的适应性与鲁棒性。

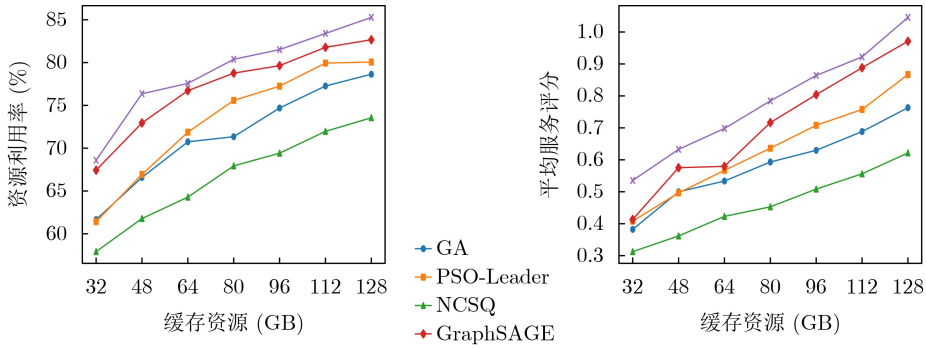
7 结论

本文提出了一种感知缓存的双层闭环协同框架QCache，在显式QoS约束下联合优化缓存、带宽和计算资源。上层全局探索层将群体演化策略和基于历史最优引导的粒子更新机制相耦合以生成高质量的候选解，下层多维权重评估层准确地量化了各种QoS需求，并引导上层的求解。通过缓存资源变



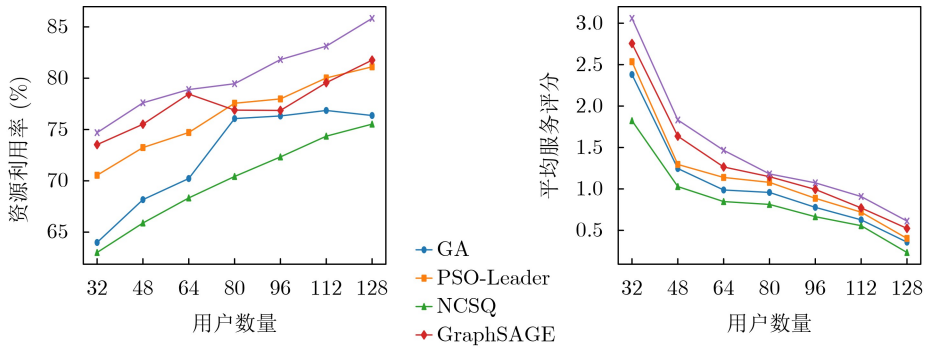
(a) 资源利用率 (b) 平均服务评分

图 3 相同场景下各方案总体比较



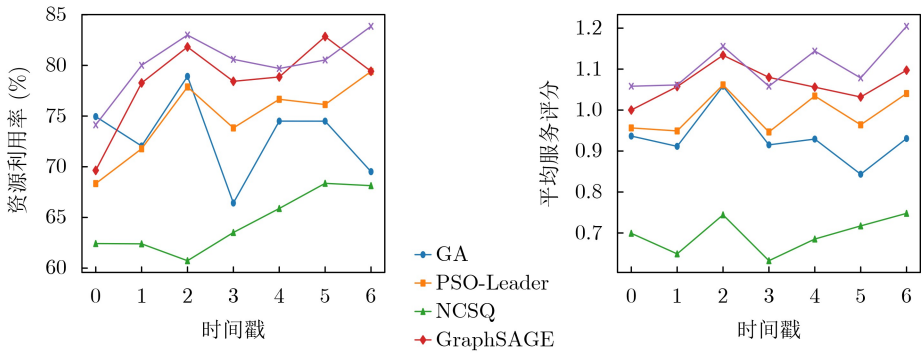
(a) 资源利用率 (b) 平均服务评分

图 4 总缓存资源变化下各方案比较



(a) 资源利用率 (b) 平均服务评分

图 5 总用户数量变化下各方案比较



(a) 资源利用率　　(b) 平均服务评分

图 6 用户需求变化下各方案比较

化、用户增长和波动需求等广泛实验表明，QCache有效改善了多接入边缘计算网络切片的QoS。

参 考 文 献

[1] ZHUANSUN Chenlu, LI Pengdeng, LIU Yuan, *et al*. Generative AI-assisted mobile-edge computation offloading in digital-twin-enabled IIoT[J]. *IEEE Internet of Things Journal*, 2025, 12(10): 13248–13258. doi: 10.1109/JIOT.2025.3547370.

[2] ZOU Haodong, GUO Jianxiong, SUN Suping, *et al*. Adaptive image batching and slicing in edge networks for delay-critical small object detection[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2026, 37(7): 1657–1670. doi: 10.1109/TPDS.2026.3694562.

[3] MALEKI E F, MA Weibin, MASHAYEKHY L, *et al*. QoS-aware content delivery in 5G-enabled edge computing: Learning-based approaches[J]. *IEEE Transactions on Mobile Computing*, 2024, 23(10): 9324–9336. doi: 10.1109/TMC.2024.3363143.

[4] WANG Tian, LIANG Yuzhu, SHEN Xuewei, *et al*. Edge computing and sensor-cloud: Overview, solutions, and directions[J]. *ACM Computing Surveys*, 2023, 55(13s): 1–37. doi: 10.1145/3582270.

[5] 林粤伟, 张奇勋, 尉志青, 等. 通信感知一体化硬件设计——现状与展望[J]. 电子与信息学报, 2025, 47(1): 1–21. doi: 10.11999/JEIT240012.

LIN Yuewei, ZHANG Qixun, WEI Zhiqing, *et al*. Status and prospect of hardware design on integrated sensing and communication[J]. *Journal of Electronics & Information Technology*, 2025, 47(1): 1–21. doi: 10.11999/JEIT240012.

[6] LIANG Yuzhu, YIN Mujun, WANG Wenhua, *et al*. Collaborative edge server placement for maximizing QoS with distributed data cleaning[J]. *IEEE Transactions on Services Computing*, 2025, 18(3): 1321–1335. doi: 10.1109/TSC.2025.3552337.

[7] YANG Hui, YU Ao, ZHANG Jie, *et al*. Data-driven network slicing from core to RAN for 5G broadcasting services[J]. *IEEE Transactions on Broadcasting*, 2021, 67(1): 23–32. doi: 10.1109/TBC.2020.3031742.

[8] HAMDI W, KSOURI C, BULUT H, *et al*. Network slicing-based learning techniques for IoV in 5G and beyond networks[J]. *IEEE Communications Surveys & Tutorials*, 2024, 26(3): 1989–2047. doi: 10.1109/COMST.2024.3372083.

[9] WANG Tian, LIANG Yuzhu, JIA Weijia, *et al*. Coupling resource management based on fog computing in smart city systems[J]. *Journal of Network and Computer Applications*, 2019, 135: 11–19. doi: 10.1016/j.jnca.2019.02.021.

[10] 徐冬竹, 周安福, 马华东, 等. 基于连续学习的视频物联网任务需求理解与调度方法[J]. 计算机研究与发展, 2024, 61(11): 2793–2805. doi: 10.7544/ISSN1000-1239.202440403.

XU Dongzhu, ZHOU Anfu, MA Huadong, *et al*. Continuous learning-based task demand understanding and scheduling method for video internet of things[J]. *Journal of Computer Research and Development*, 2024, 61(11): 2793–2805. doi: 10.7544/ISSN1000-1239.202440403.

[11] LIANG Yuzhu, XU Changfu, MEI Yaxin, *et al*. A comprehensive survey on large language model compression for artificial intelligence applications in edge systems[J]. *IEEE Internet of Things Journal*, 2026, 13(10): 20583–20599. doi: 10.1109/JIOT.2026.3675866.

[12] European Telecommunications Standards Institute. ETSI GS MEC 003 V2.1. 1 (2019-01) Multi-access edge computing (MEC); framework and reference architecture[S]. Alpes Maritimes: ETSI, 2019.

[13] YUE Yi, CHENG Bo, SUN Shiding, *et al*. Availability guaranteed and resource efficient VNF placement in SDN/NFV-Enabled network through traffic forecasting[C]. 2025 IEEE International Conference on Web Services, Helsinki, Finland, 2025: 983–992. doi: 10.1109/ICWS67624.2025.00139.

[14] MAHDI S S and ABDULLAH A A. Survey on enabling network slicing based on SDN/NFV[C]. International Conference on Information Systems and Intelligent Applications, ICISIA 2022, Cham, Germany, 2022: 733–758. doi: 10.1007/978-3-031-16865-9_59.

[15] SINDJOUNG M L F, VELEMPINI M, and BOMGNI A B.

A MEC architecture for a better quality of service in an Autonomous Vehicular Network[J]. *Computer Networks*, 2022, 219: 109454. doi: 10.1016/j.comnet.2022.109454.

[16] ZHANG Zhengming, HUANG Yongming, ZHANG Cheng, *et al*. Digital twin-enhanced deep reinforcement learning for resource management in networks slicing[J]. *IEEE Transactions on Communications*, 2024, 72(10): 6209–6224. doi: 10.1109/TCOMM.2024.3395698.

[17] FAN Wenhao, LI Xuewei, TANG Bihua, *et al*. MEC network slicing: Stackelberg-game-based slice pricing and resource allocation with QoS guarantee[J]. *IEEE Transactions on Network and Service Management*, 2024, 21(4): 4494–4509. doi: 10.1109/TNSM.2024.3409277.

[18] LIANG Yuzhu, YIN Mujun, ZHANG Yilin, *et al*. Grouping reduces energy cost in directionally rechargeable wireless vehicular and sensor networks[J]. *IEEE Transactions on Vehicular Technology*, 2023, 72(8): 10840–10851. doi: 10.1109/TVT.2023.3259683.

[19] WANG Shuai and ZHOU Aimin. Leader prediction for multiobjective particle swarm optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2025, 29(4): 1356–1370. doi: 10.1109/TEVC.2024.3417978.

[20] HWANG J, NKENYEREYE L, SUNG N, *et al*. IoT service slicing and task offloading for edge computing[J]. *IEEE Internet of Things Journal*, 2021, 8(14): 11526–11547. doi: 10.1109/JIOT.2021.3052498.

[21] ZHANG Tiankui, WANG Yi, YI Wenqiang, *et al*. Joint optimization of caching placement and trajectory for UAV-D2D networks[J]. *IEEE Transactions on Communications*, 2022, 70(8): 5514–5527. doi: 10.1109/TCOMM.2022.3182033.

[22] LIU Xing, LV Jianhui, KIM B G, *et al*. Cooperative digital healthcare task scheduling and resource management in edge intelligence systems[J]. *Tsinghua Science and Technology*, 2025, 30(2): 926–945. doi: 10.26599/TST.2024.9010140.

[23] HELMY M, ABDELLATIF A A, MHAISEN N, *et al*. Slicing for AI: An online learning framework for network slicing supporting AI services[J]. *IEEE Transactions on Network and Service Management*, 2025, 22(6): 5239–5254. doi: 10.1109/TNSM2025.3603391.

[24] NASSAR A and YILMAZ Y. Deep reinforcement learning for adaptive network slicing in 5G for intelligent vehicular systems and smart cities[J]. *IEEE Internet of Things Journal*, 2022, 9(1): 222–235. doi: 10.1109/JIOT.2021.3091674.

[25] QIAO Kai, WANG Hongchao, ZHANG Weiting, *et al*. Resource allocation for network slicing in open RAN: A hierarchical learning approach[J]. *IEEE Transactions on Cognitive Communications and Networking*, 2025, 11(4): 2584–2600. doi: 10.1109/TCCN.2024.3524641.

[26] PEREIRA P M R, DE BAIRROS T A M, DE SOUZA R A A, *et al*. Mobility, path loss, and composite fading: Performance of a conventional and of a non-conventional system with a robust autoencoder[J]. *IEEE Transactions on Vehicular Technology*, 2023, 72(12): 16725–16730. doi: 10.1109/TVT.2023.3294756.

[27] ANGELELLI E, MANSINI R, and SPERANZA M G. Kernel search: A general heuristic for the multi-dimensional knapsack problem[J]. *Computers & Operations Research*, 2010, 37(11): 2017–2026. doi: 10.1016/j.cor.2010.02.002.

[28] LIU Tao, JIANG Aimin, ZHOU Jia, *et al*. GraphSAGE-based dynamic spatial-temporal graph convolutional network for traffic prediction[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(10): 11210–11224. doi: 10.1109/TITS.2023.3279929.

[29] WANG Zhaoying, WEI Yifei, YU F R, *et al*. Utility optimization for resource allocation in multi-access edge network slicing: A twin-actor deep deterministic policy gradient approach[J]. *IEEE Transactions on Wireless Communications*, 2022, 21(8): 5842–5856. doi: 10.1109/TWC.2022.3143949.

徐骏涛：男，硕士生，研究方向为5G，移动计算和边缘计算.
范兴刚：男，教授，研究方向为物联网，边缘计算.
徐常福：男，博士，研究方向为边缘计算，AIGC.
沈民扬：男，硕士生，研究方向为物联网，边缘计算.
梁玉珠：男，博士，研究方向为边缘计算，移动计算和人工智能.
王 田：男，教授，研究方向为物联网，边缘计算和移动计算.

责任编辑：余 蓉

A Two-layer Closed-loop Cooperative Resource Allocation Framework for Improving QoS of MEC Network Slicing

XU Juntao[①] FAN Xinggang[②] XU Changfu[③] SHEN Minyang[①]
LIANG Yuzhu[④] WANG Tian[④]

[①](*College of Computer Science and Technology Zhejiang University of Technology*, *Hangzhou* 310023, *China*)
[②](*Zhijiang College*, *Zhejiang University of Technology*, *Shaoxing* 312030, *China*)

③(*School of Software and IoT Engineering Jiangxi University of Finance and Economics*, *Nanchang* 330013, *China*)

④(*Institute of AI and Future Networks Beijing Normal University*, *Zhuhai* 519085, *China*)

Abstract:

Objective Driven by 5G/6G networks, Multi-access Edge Computing (MEC) environments face major challenges in ensuring Quality of Service (QoS) for heterogeneous network slices while improving resource utilization. Existing resource allocation methods often lack dynamic adaptability in heterogeneous settings. They also fail to jointly optimize caching, bandwidth, and computing resources, which reduces resource utilization and service success rates. This paper addresses these limitations by proposing a robust framework for joint multi-dimensional resource optimization under strict QoS constraints. The framework provides a tailored solution for heterogeneous MEC environments.

Methods The network slicing resource allocation problem is first formally defined. Its NP-hardness is proved by reducing the NP-hard multidimensional 0-1 knapsack problem to this problem. To solve this complex optimization problem, a cache-aware two-layer closed-loop cooperative framework, termed QCache, is proposed. The framework uses a synergistic "generate-evaluate-feedback" loop. In the upper Global Exploration Layer, a hybrid heuristic algorithm is designed by combining a population evolution strategy, including selection, crossover, and mutation, with a particle update mechanism guided by historical individual and global best positions. This layer broadly explores the solution space and generates high-quality candidate resource allocation schemes under complex constraints. Adaptive parameter adjustment and elite retention strategies are used to avoid local optima. In the lower Multi-Dimensional Weight Evaluation Layer, a quantitative assessment model is constructed. This model converts low-latency and high-bandwidth service demands into explicit QoS constraints by normalizing key performance indicators, including delay and rate, and by dynamically assigning weights through the entropy weight method. The weighted score reflects different slice priorities. The evaluated score (SCORE) from this layer is fed back to the upper layer as the fitness value, guiding the iterative evolution of candidate solutions until convergence.

Results and Discussions Extensive simulations are conducted to validate the effectiveness of QCache against several baseline methods, including Genetic Algorithm (GA), PSO-Leader, GraphSAGE, and the No-Consideration-of-Service-Quality (NCSQ) scheme. Under identical resource and user demand scenarios, the overall comparison (Fig. 3) shows that QCache achieves the highest resource utilization rate of 80.30% and the best average service score of 1.109. Compared with the baseline methods, QCache improves resource utilization by 2.29% to 24.50% and increases user service scores by 4.13% to 59.34%. Experiments with varying total cache resources (Fig. 4) show that QCache maintains superior performance across different cache states. It improves resource utilization by 2.43% to 27.53% and service scores by 10.81% to 119.56%, confirming its cache-aware adaptability. Tests with increasing user numbers (Fig. 5) show that QCache scales effectively, achieving up to 85.83% resource utilization and a service score of 3.06. These results demonstrate its ability to handle dense access scenarios. Experiments with time-varying user demands (Fig. 6) further confirm the dynamic robustness of the framework. In these tests, QCache achieves average improvements of 9.20% in resource utilization and 23.45% in service score over the baselines.

Conclusions This paper studies the NP-hard resource allocation problem in dynamic MEC environments with heterogeneous network slices. The proposed cache-aware two-layer closed-loop cooperative framework, QCache, jointly optimizes caching, bandwidth, and computing resources under explicit QoS constraints. The upper-layer hybrid heuristic provides strong global search capability. The lower-layer multi-dimensional weight model supports accurate QoS quantification and dynamic feedback. Comprehensive experimental results show that QCache outperforms existing methods in both resource utilization efficiency and user QoS satisfaction. Future work will explore reinforcement learning and traffic prediction mechanisms to further improve the response of the framework to bursty traffic and anomalous demands. This may support more intelligent and autonomous MEC network slice resource management.

Key words: Multi-access Edge Computing(MEC); Network slicing; Resource allocation; Quality of Service(QoS) guarantee