移动边缘计算场景下双时间尺度在线服务迁移方法

王海艳[1,2]，张家豪[1]，骆健[1,2]
（1. 南京邮电大学计算机学院，江苏 南京 210023；2. 大数据安全与智能处理省高校重点实验室，江苏 南京 210023）

摘　要：在移动边缘计算（MEC）场景中，针对不同任务计算时长的差异性，提出了一种双时间尺度在线服务迁移（TOSM）方法。基于时间尺度将服务迁移问题分层优化，针对长时任务，设计在线拍卖机制，激励系统实时调度资源并结合能耗成本动态调整决策，在宏观尺度上优先确定长时任务的执行时隙与资源分配策略。针对短时任务，在微观尺度上循环求解凸优化模型，确保快速响应。通过理论分析和仿真实验验证TOSM方法在系统效用、平均时延和能耗等方面优于现有方法，且满足拍卖的真实性与个体理性，为MEC场景下服务迁移问题提供了高效、经济的解决方案。

关键词：移动边缘计算；服务迁移；双时间尺度；拍卖算法；动态资源分配
中图分类号：TP393
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025158

Two-timescale online service migration method in mobile edge computing scenarios

WANG Haiyan[1,2], ZHANG Jiahao[1], LUO Jian[1,2]
1. School of Computer Science, Nanjing University of Post and Telecommunications, Nanjing 210023, China
2. Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing 210023, China

Abstract: For the mobile edge computing (MEC) scenario, a two-timescale online service migration (TOSM) method was proposed to address the differences in computing time for different tasks. Service migration problems were optimized in layers according to time scale. For long-duration tasks, an online auction mechanism was designed to motivate the system to schedule resources in real time and dynamically adjust decisions based on energy consumption costs. The execution time slots and resource allocation strategies of long-duration tasks were prioritized on a macro scale. For short-duration tasks, a convex optimization model was solved cyclically on a micro scale to ensure rapid response. Theoretical analysis and simulation experiments verify that the TOSM method is superior to existing methods in terms of system utility, average latency, and energy consumption, and meets the authenticity and individual rationality of auctions, providing an efficient and economical solution to the service migration problem in the MEC scenario.

Keywords: mobile edge computing, service migration, two-timescale, auction algorithm, dynamic resource allocation

0　引言

随着信息技术迅速发展，时延敏感型和计算密集型的新型移动应用不断涌现在网络边缘，对边缘数据处理能力提出了更高要求[1]。移动边缘计算（MEC, mobile edge computing）将边缘服务器与基站（BS, base station）融合形成微服务中心，并部署在网络边缘，在这种架构下，移动设备可以将一些复杂任务卸载至边缘服务器上执行，从而能够运

收稿日期：2025–05–23；修回日期：2025–08–22
通信作者：王海艳，wanghy@njupt.edu.cn
基金项目：国家自然科学基金资助项目（No.62272243）
Foundation Item：The National Natural Science Foundation of China (No.62272243)

行具有严格质量要求的复杂应用程序[2]。由于MEC环境的动态性和用户移动性[3]，服务需要在边缘服务器之间动态迁移，以适应用户位置变化和均衡服务器负载。

现有的服务迁移策略主要根据用户移动性和服务器可用资源等信息制定服务迁移决策[4-6]，但仍然面临多维度挑战，亟须兼顾效率、经济性的解决方案。其一，现实服务迁移问题中不同任务在资源需求和执行时长上存在显著差异，往往需要跨越不同时间尺度处理[7]，如人工智能训练与推理任务以及实时数据处理与持续计算密集型作业的调度安排，传统单一时间尺度的策略难以满足动态的服务需求，易引发资源分配低效与服务质量（QoS, quality of service）波动；其二，用户移动性和任务到达的随机性使资源需求难以预判，易导致边缘服务器负载不均衡，进而因能耗突破阈值影响服务稳定性；其三，边缘服务器运营成本（如能耗费用和硬件异构性）被忽视，易导致资源冗余与高经济成本，从而影响系统整体效用。

针对上述挑战，首先，本文提出了双时间尺度服务迁移框架，基于任务计算时长分层优化，同时兼顾短期与长期的影响。在宏观时间尺度下对长时任务执行拍卖以适应不确定的资源需求变化，在微观时间尺度下快速响应短时任务，实现系统高效决策。其次，设计在线拍卖机制，用户基于资源量和截止时间的需求出价，边缘服务器结合能耗价格动态调整资源定价，从而激励资源分配，制定拍卖决策，同时确保拍卖的真实性与参与者的个体理性。基于此机制，将用户迁移与通信成本纳入框架，进一步构建多目标协同优化模型，致力于实现长期目标的最大化系统效用。最后，通过理论证明和仿真实验验证了本文方法的可行性和鲁棒性，具体工作总结如下。

1) 针对不同任务的异质性，提出了双时间尺度服务迁移框架，在宏观时间尺度下设计在线拍卖机制并结合动态规划技术，制定长时任务跨时隙迁移决策，提升资源分配效率。短时任务通过循环求解凸优化模型，在微观时间尺度下实现快速响应。

2) 将服务迁移抽象为通信和迁移成本、负载均衡及服务器运营成本协同优化的多目标优化问题，构建综合优化模型。引入动态资源分配策略与能耗价格动态调整机制，实现系统长期效用最优化。

3) 在理论层面进行分析与证明，确保本文方法的可行性，同时基于合成数据和真实数据开展仿真实验，证明了本文方法的有效性。

1 相关工作

随着互联网的发展和移动边缘计算的引入，用户对边缘服务的需求也是日趋增长。因此，通过服务迁移为用户提供高质量的服务显得十分必要。

大多数服务迁移方法通常以迁移成本和服务时延的组合为优化目标制定服务迁移决策[8]。文献[9]采用基于马尔可夫决策过程（MDP, Markov decision process）的值迭代方法求解服务迁移决策。由于实际场景中的复杂随机性，难以对MDP进行准确建模，一些工作转向使用无模型的强化学习方法。文献[10]提出了一种强化学习方法Q-learning求解迁移决策。文献[11]进一步提出了使用深度强化学习方法深度Q网络来制定服务迁移决策，综合考虑用户的移动性和服务器的高功耗问题，设计了能量–时延平衡的协同边缘缓存策略和改进的服务迁移策略。文献[12]提出了一种基于车辆行为预测的移动感知深度强化学习服务迁移框架，结合优先经验回放机制设计基于双决斗深度Q网络的强化学习服务迁移算法求解。这些方法通常要求问题的求解空间不能过大，所以大多采用了单用户服务迁移问题的建模。然而，单用户服务迁移方案忽略了多个用户之间存在的任务排队时延以及服务器负载不均衡的问题，当用户数量较多时容易降低QoS、导致服务器负载不均衡。文献[13]提出了一种数字孪生辅助服务迁移方法来最大限度地缩短任务完成时间，而不是将每个用户分配给最近的边缘服务器，通过将任务分类为时延敏感与时延不敏感2种类型来满足不同的QoS要求，并基于隐马尔可夫模型预测用户未来的移动性。然而，该方法没有考虑到任务之间计算时长与资源需求上的差异性。因此，本文构建涵盖通信与迁移成本联合资源分配的多目标优化模型，提出了针对任务差异性的双时间尺度服务迁移方案，通过分层优化，同时兼顾长期和短期的影响，有效制定多用户服务迁移决策。

部分工作也考虑了多用户服务迁移的资源分配问题，文献[14]提出了一种基于多代理邻近策略优化算法的动态服务迁移策略，综合考虑位置、剩余数据量和服务期限约束等信息，建立了多用户随机移动服务迁移模型，将服务迁移决策问题形式化为

迁移效用最大化问题，利用多策略框架实现用户偏好自适应，以获得高效的在线迁移决策。文献[15]针对多个用户盲目竞争资源导致边缘服务器过载的问题，考虑了多个用户之间的干扰，提出了一种基于资源竞争的服务迁移方法，重点解决了服务器资源分配相关服务迁移决策的制定。文献[16]将服务迁移分为时延违反阈值和边缘服务器过度使用2种情况进行考虑，提出了一种基于简洁概率方法的时延感知和移动性感知服务管理方法。文献[17]考虑了不同时间尺度的服务迁移和资源分配问题，针对用户服务请求的动态随机性和制定不同决策的异步性，制定了一个在线优化问题，并提出了一种基于改进 Lyapunov 方法的双时间尺度算法，但难以高效适配服务动态迁移带来的离散资源需求变化与节点切换场景，且对资源波动的适应性和收敛效率较低，本文方法基于拍卖机制巧妙地平衡了即时资源分配的需求和长期系统稳定性的要求。文献[18]提出了一种节能的在线服务迁移机制来同时进行多个服务的迁移。通过制定轻服务共享策略，仅传输顶层容器层，并采用改进的非支配排序遗传算法为容器层和每个迁移服务的时序感知数据迁移生成一条或多条路径。文献[19]通过联合管理计算和无线资源来优化服务迁移和切换策略，并开发了基于松弛舍入的求解方法。然而上述方法都忽视了服务器的运营成本，对此本文设计在线拍卖机制，结合能耗价格动态调整拍卖策略，降低服务迁移过程中资源冗余或负载不均导致的额外经济损耗，提升边缘计算整体系统效用，资源的定价基于服务器的动态拍卖决策，从而激励资源分配和满足用户需求。

2　系统模型和问题构建

2.1　系统模型

本文考虑了在地铁运营的场景[15]中进行服务迁移，双时间尺度在线服务迁移（TOSM, two-timescale online service migration）场景如图 1 所示，本文讨论的系统由一个分布式 MEC 服务器集 S 组成，该服务器集为高移动性的用户集提供计算资源。每个边缘服务器与特定地理位置的基站位于同一位置，以托管服务实例并处理用户卸载的任务请求，部署的基站位置集合表示为 L。有线回程链路将部署在不同地理位置的服务器连接起来，从而支持通信和服务迁移。尽管 MEC 服务器在加速任务处理、提高服务性能方面发挥了关键作用，但它也存在一个关键的瓶颈，即每个 MEC 服务器拥有的最大资源容量，由于可用资源有限，它在时刻 t 只能服务于有限数量的任务请求，设定系统中边缘服务器提供 K 种类资源，如中央处理器（CPU, central processing unit）、随机存取存储器（RAM, random access memory）和内存，由此组成不同类型的虚拟机（VM, virtual machine）。此外，边缘服务器集群往往存在异构性，因为在边缘网络中，服务器通常分布在不同的位置，如偏远郊区、工厂或城市办公楼等地方，为了满足不同地区和应用的需求，边缘服务器会采用不同类型和规格的硬件设备，从而导致服务器的能耗水平高低不一。

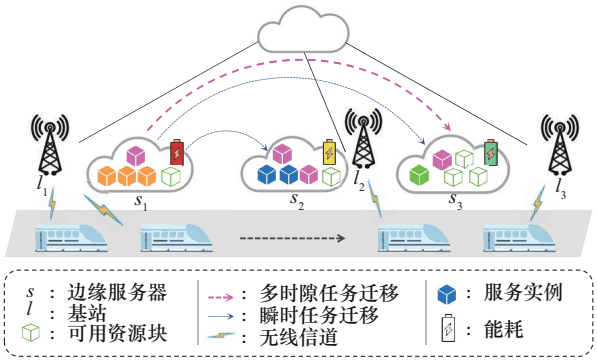

图1　TOSM场景

整个系统的生命周期被划分为一系列时隙（宏观时间尺度），构成集合 $T=\{1,2,\cdots,t\}$，每个时隙又细分为一系列时间槽（微观时间尺度），记为集合 $D=\{1,2,\cdots,d\}$，其中用户的位置在一个时隙持续时间内保持不变。定义至少执行一个时隙的任务为长时任务，定义任务时长不足一个时隙的任务为短时任务，在 t 时隙的 d 时间槽执行，双时间尺度关系如图2所示。

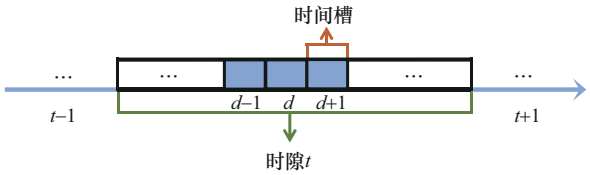

图2　双时间尺度关系

2.2　问题表述

本文从三方面建模优化服务迁移问题：基于拍卖理论构建资源拍卖模型，实现资源高效配置；综

合数据量、带宽等因素建立迁移与通信成本函数，量化迁移损耗；结合资源消耗、电价等要素构建服务器运营成本评估模型。通过子模型协同分析与参数耦合，提出总体最优化模型，平衡资源、成本与开销，实现服务迁移效用最大化。

2.2.1 资源拍卖模型

针对资源需求量较大的长时任务，设计在线拍卖机制激励系统资源的动态分配，核心网络充当拍卖商，通过在线拍卖将虚拟机租赁给用户。设定系统中边缘服务器提供K种类资源，如CPU、RAM和内存。由此组成不同类型的虚拟机。设$[X]$表示整数集$\{1,2,\cdots,X\}$，假设边缘服务器$s\in[S]$中可用的资源类型k的数量为集合$[c_{ks}]$。对于系统中的I个长时用户，在一个大的时间跨度$\{1,2,\cdots,T\}$中，每一个用户i都提供了一个执行其任务的出价，用户的出价随机到达，t_i为用户i出价到达的时隙。每个用户根据各种类型的资源需求量都请求一组定制的VM来执行作业。B_i为用户i在t_i时隙提交的出价，它包含：① 所需k类型的资源数量r_i^k；② 完成任务所需的时隙数量（不一定是连续的）ω_i；③ 任务完成的预期截止日期d_i；④ 任务超过截止日期的惩罚函数$g_i(\tau_i)$。

$$g_i(\tau_i)=\begin{cases}g_{c_i}(\tau_i), & \tau_i\in[0,T-d_i]\\ +\infty, & \text{其他}\end{cases} \tag{1}$$

定义b_i为用户i的任务在截止日期d_i之前完成的出价，那么$b_i-g_i(\tau_i)$则是任务完成时间为$d_i+\tau_i$时的出价。$g_{c_i}(\tau_i)$是一个非递减函数且$g_{c_i}(0)=0$。综上，用户i的出价可以描述为$B_i=\{t_i,\{r_i^k\},k\in[K],\omega_i,d_i,b_i,g_i(\tau_i)\}$。在每一个出价到达后，通过计算资源分配并宣布拍卖结果。①$x_{is}\in\{0,1\}$，其中当$x_{is}=1$时，长时用户i的任务被接受并分配在服务器s上；否则$x_{is}=0$。②$y_{is}(t)\in\{0,1\}$，当用户i的任务计划在t时隙运行在服务器s上，则$y_{is}(t)=1$；否则$y_{is}(t)=0$。同理，$z_{jsd}^t\in\{0,1\}$，当短时用户j的任务计划在t时隙的d时间槽运行在服务器s上，则$z_{jsd}^t=1$；否则$z_{jsd}^t=0$，用户j的资源需求量为r_j^k。③用户i的支付p_i，当工作在截止日期d_i前完成，定义γ_i和$g_i'(\tau_i)$分别为用户i的真实估值和真罚函数，因此$\gamma_i-g_i'(\tau_i)$为用户i的真实出价。用户i的竞标价格$b_i-g_i(\tau_i)$的效用可以表示为$u_i(b_i-g_i(\tau_i))=\sum_{s\in[S]}\gamma_i x_{is}-g_i'(\tau_i)-p_i$。每个用户都被认为是自私和理性的，都以最大化用户的个人效用为目标，因此用户可能会选择在真实估值上撒谎，从而获得更高的效用。在本文的在线拍卖机制设计中，为了实现系统总体效用最大化，选择真实投标很重要。

定义1 真实拍卖。当用户的出价是真实估值时，本文认为该拍卖是真实拍卖，那么对于所有的$b_i-g_i(\tau_i)\neq\gamma_i-g_i'(\tau_i),u_i(\gamma_i-g_i'(\tau_i))\geqslant u_i(b_i-g_i(\tau_i))$，始终最大化用户效用。

2.2.2 迁移与通信成本

将任务从源服务器迁移至目标服务器，本文考虑迁移过程中的迁移与通信成本。设定长时任务的源服务器地址为L_{ori}，目标服务器地址为L_s，终端用户地址为L。短时任务的源服务器地址为l_{ori}，目标服务器地址为l_s，终端用户地址为l。与欧式距离不同的是，本节采用2个蜂窝网络之间的跳数来衡量2台服务器之间的距离，分别记为$\delta_i=\|L_{\text{ori}}-L_s\|$和$\delta_j=\|l_{\text{ori}}-l_s\|$，定义执行完一次服务迁移后用户和目标服务器之间的距离分别为$\partial_i=\|L-L_s\|$和$\partial_j=\|l-l_s\|$。

定义迁移成本函数$m(\delta)$为

$$m(\delta)=\begin{cases}\omega_0+\omega_d\theta^{\delta}, & \delta>0\\ 0, & \delta=0\end{cases} \tag{2}$$

定义通信成本函数$n(\partial)$为

$$n(\partial)=\begin{cases}\mu_0+\mu_d\varepsilon^{\partial}, & \varphi>0\\ 0, & \varphi=0\end{cases} \tag{3}$$

其中，ω_0、ω_d、μ_0、μ_d、θ和ε都是真实值，而且$0\leqslant\varepsilon\leqslant1$，$\theta\geqslant1$。

故长时任务的迁移与通信总成本函数表示为

$$C_{is}^t=m(\delta_i)+n(\partial_i) \tag{4}$$

短时任务的迁移与通信总成本函数表示为

$$G_{jsd}^t=m(\delta_j)+n(\partial_j) \tag{5}$$

2.2.3 服务器运营成本

根据边缘需求响应机制[20]，在移动边缘计算场景下，电网在系统初始状态向核心网络发送一个信号，制定每个时隙的能耗价格用于模型计算迁移优化策略，以调度任务执行。基于长时任务资源需求量大的特性，当优化边缘服务器运营成本时，设定其主要由系统能耗组成，服务器的能耗随着资源的增加而增加，将其建模为$\sum_{k\in[K]}\beta_k\mu_k$，其中$\mu_k$表示$k$种类资源的应用量，而$\beta_k$表示$k$种类资源完全使用的能耗量。定义$e^t$为长时任务在时隙$t$的实际

能源消耗量，h_t为t时的能耗价格，t时的运营成本可以描述为

$$f_t(e^t) = h_t e^t \tag{6}$$

此外，对于资源使用量较少的短时任务，采用常规的能耗公式来优化。首先，定义φ_{jsd}^t参数为短时任务j在d时间槽从源服务器转移到s边缘服务器所能节省的能耗，$z_{jsd}^t \varphi_{jsd}^t$即表示短时任务在该时隙完成迁移后所节省的能耗。由于能耗的计算涉及多个方面，这里采用常规的CPU能耗关系式来处理此问题。服务器的动态能耗源于CPU内部逻辑门的活动。当逻辑门切换时，能量会随着内部电容器的充电和放电而流动。CPU消耗的动态功率与CPU频率和CPU电压的平方大致成正比。同时，在移动边缘计算场景中，由于边缘服务器存在异构性，不同服务器存在不同的运行频率[21]，定义源服务器的频率为f_{ori}，目标服务器的频率为f_s，因此，短时任务j在d时间槽从源服务器转移到s边缘服务器所能节省的能耗φ_{jsd}^t可以描述为

$$\varphi_{jsd}^t = CV(f_s - f_{\text{ori}}) \tag{7}$$

其中，C是开关负载电容，f是频率，V是电压。本文参数设置如表1所示。

2.3　总体效用最大化模型P

总体效用最大化模型由两部分组成，即用户效用的总体与边缘服务器的效用相加的最优化。用户的效用与出价b_i、惩罚函数$g_i(\tau_i)$、迁移与通信成本C_{is}^t和G_{jsd}^t有关，边缘服务器的效用主要与运营成本$f_t(e^t)$和φ_{jsd}^d有关。由于二者相加会自动抵消支付p_i，因此，在真实投标的假设下，总体效用最大化问题可以公式化为问题P1。

P1:

$$\begin{aligned}\text{Maximize: } & \sum_{i\in[I]}\Big(\sum_{s\in[S]} b_i x_{is} - g_i(\tau_i)\Big) - \\ & \sum_{i\in[I]}\sum_{t\in[T]}\sum_{s\in[S]} x_{is} C_{is}^t - \sum_{t\in[T]} f_t\left(e^t\right) + \\ & \sum_{j\in[J]}\sum_{s\in[S]}\sum_{d\in[D]}\sum_{t\in[T]}\left(z_{jsd}^t \varphi_{jsd}^t - z_{jsd}^t G_{jsd}^t\right)\end{aligned} \tag{8}$$

$$\text{s.t. } C_1: \sum_{s\in[S]} x_{is} \leqslant 1, \forall i \in [I]$$

$$C_2: y_{is}^t t \leqslant d_i + \tau_i,\ \forall t \in [T],\ \forall s \in [S],\ \forall i \in [I]$$

$$C_3: \omega_i x_{is} \leqslant \sum_{\forall t\in[T]} y_{is}^t, \forall s \in [S], \forall i \in [I]$$

$$C_4: \sum_{\forall i\in[I]} r_i^k y_{is}^t + \sum_{\forall j\in[J]}\sum_{\forall d\in[D]} r_j^k z_{jsd}^t \leqslant c_{ks},$$
$$\forall t \in [T], \forall s \in [S], \forall k \in [K]$$

$$C_5: \sum_{\forall s\in[S]}\sum_{\forall k\in[K]} \beta_{ks}\left(\frac{\sum_{\forall i\in[I]} r_i^k y_{is}^t}{c_{ks}}\right) \leqslant$$
$$e^t, \forall t \in [T]$$

表1　　**参数设置**

参数	含义	参数	含义
I	长时任务用户	p_i	用户i的支付
J	短时任务用户	c_{ks}	服务器s上的类型k资源的容量
T	时隙	x_{is}	用户i在服务器s上是否出价
D	时间槽	$y_{is}(t)$	是否在t时将用户i的作业分配到服务器s
S	边缘服务器	z_{jsd}^t	是否在t时隙的d时间槽将用户j的作业分配到服务器s上
K	资源种类	L	终端用户地址
b_i	用户i的投标价格	L_{ori}	源服务器地址
r_i^k	用户i所需k类型的资源数量。	L_s	目标服务器地址
t_i	用户i到达的时隙	μ_k	k种类资源的应用量
ω_i	用户i完成任务所需的时隙数量	β_k	k种类资源完全使用的能耗量
d_i	用户i任务完成的预期截止日期	h_t	t时的能耗价格
g	惩罚函数	e^t	时隙t的功耗量
τ_i	用户i超出截止日期的时隙	φ_{jsd}^t	短时任务j在d时间槽从源服务器转移到s服务器所节省的能耗
γ_i	用户i的任务真实估值		

$$C_6: \sum_{\forall s\in[S]} z_{jsd}^t \leqslant \Omega_j^d,\ \forall t\in[T],\forall j\in[J],\forall d\in[D]$$

$$C_7: \tau_i\in[0,T-d_i],e^t\geqslant 0,x_{is},y_{is}^t\in\{0,1\},z_{jsd}^t\in\{0,1\},$$
$$\forall t\in[T],\ \forall s\in[S],\ \forall j\in[J],\forall i\in[I]$$

其中，由于约束C_1表明每个用户的任务最多只能在一台服务器上执行，且$y_{is}(t)\in\{0,1\}$，因此约束$y_{is}^t\leqslant x_{is},\forall i\in[I],\forall s\in[S],\forall t\in[T]$是冗余的，未包含在上述问题中，表明任务仅会在接受用户出价的边缘服务器上执行，保证拍卖的有效性。定义Ω_j^d为d时间槽内任务j所在服务器执行任务的数量。约束C_2确保调度任务只在其到达时间之后运行。约束C_3保证分配给被接受投标的时隙数量足够客户完成工作。每种资源的容量限制在约束C_4中建模，约束C_5将每个时隙的总功耗记录为e^t。约束C_6控制每个边缘服务器执行短时任务的数量上限。

3 基于拍卖算法的双时间尺度服务迁移方法

拍卖算法能够通过竞价机制有效地反映参与者的偏好，对于资源的动态分配方面体现出良好性能，特别适用于信息不完全的实时分配问题，为了满足用户需求以及优化服务器的运营成本，本文设计TOSM方法，重点研究了均衡边缘网络的资源负载以及优化边缘服务器的能耗成本。TOSM方法主要分为长时任务迁移和短时任务迁移2个模块，由于2种任务的特性和状态各不相同，对于资源分配和能耗优化的方案也采用了适合的方式。

3.1 问题拆分及算法概览

TOSM方法总体架构如图3所示，原多目标优化问题P被分解为2个部分。① 宏观时间尺度上的问题：长时任务迁移模型P_1；② 一系列微观时间尺度上的问题：短时任务迁移模型$\{P_2^{(t)},\forall t\}$。问题P_1主要关注投标选择和支付决策的连续拍卖过程，而问题$P_2^{(t)}$则专注于时隙t中竞拍结束后的短时工作负载分配策略优化。首先在线解决基础问题P_1，随后利用该问题得到的解作为基础，针对每个时隙t内的时间槽，继续在线解决子问题$P_2^{(t)}$，以此逐步攻克整体问题P。

图3中(b)模块为本文在宏观尺度下设计的在线拍卖机制，基于用户对资源及任务完成时间的需求生成用户出价，同时根据服务器资源数量和拍卖过程中资源的变化动态调整资源定价，确保系统资源负载均衡。在边缘服务器消耗资源的过程中，会结合电网提供的能耗价格以自适应调整资源分配策略，

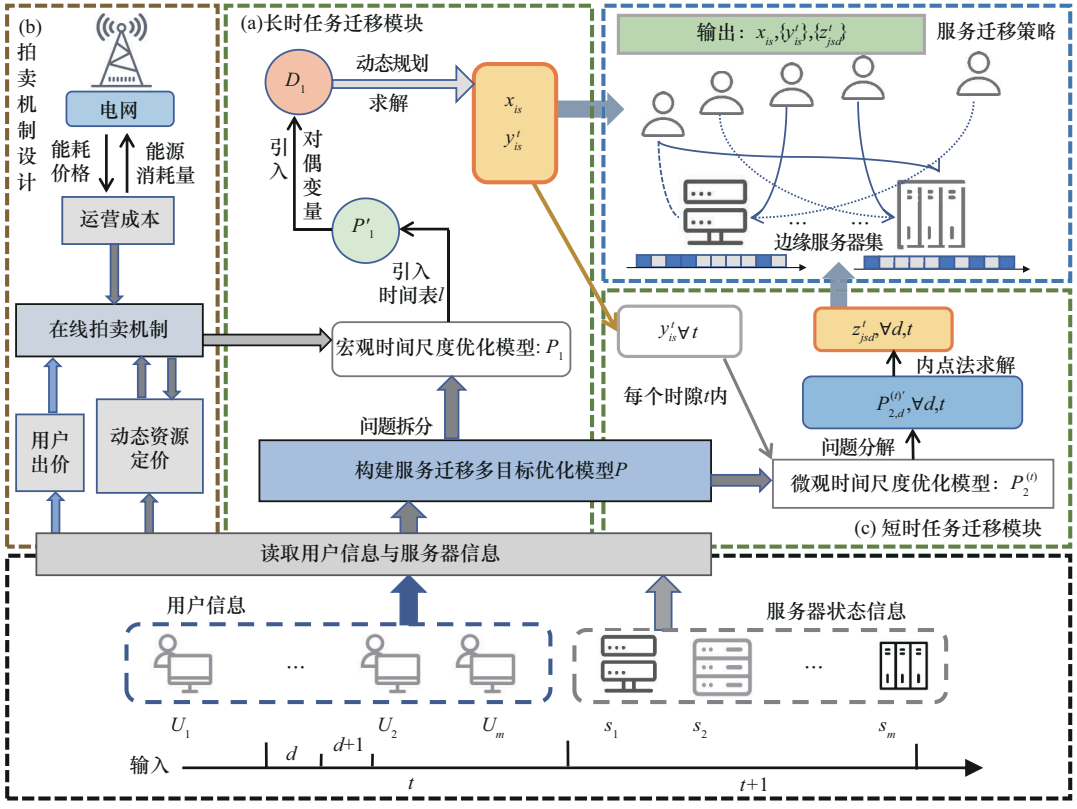

图3 TOSM方法总体架构

降低自身的运营成本，提高拍卖机制的系统效用。

本文提出A_{control}作为总体控制算法，如算法1所示，并针对每个用户运行。在每个时隙开始时，算法A_{control}调用算法A_{core}以确定当前时隙拍卖的中标投标。在当前时隙的每个时间槽内，算法A_{control}调用算法A_{solve}确定基于算法A_{core}的拍卖结果的短时任务迁移策略。由算法1可知，A_{control}首先初始化所有原始变量和对偶变量，然后分别调用算法A_{core}和算法A_{solve}制定实时资源动态分配和迁移策略。在3.2节和3.3节分别对长时任务迁移和短时任务迁移方法的设计理念、关键步骤进行详细阐述。

算法1　总体控制算法A_{control}

输入　$\{B_i\},\{c_{ks}\},\{\beta_k\},\{h_t\}$

输出　$x_{is},\{y_{is}^t\},\{z_{jsd}^t\}$

1) 根据式(4)定义迁移与通信成本函数C_{is}^t，根据式(6)定义运营成本函数$f_t(e^t)$

2) 根据式(15)定义函数p_{ks}^t

3) 初始化 $x_{is}=0,y_{is}^t=0,w_{ks}^t=0,\tau_i=0,x_{il}=0,\mu_i=0,p_{ks}^t=0,m_t=h_t,e^t=0,\forall i\in[I],l\in[\xi_i],k\in[K],s\in[S],t\in[T]$

4) while第i个用户任务到达时

5)$(x_{is},\{y_{is}^t\},p_i,\{p_{ks}^t\},\{w_{ks}^t\},\{e^t\})=A_{\text{core}}(B_i,\{c_{ks}\},\{p_{ks}^t\},\{w_{ks}^t\},\{m_t\},\{e^t\})$

6) if $\exists s\in[S],x_{is}=1$ then

7)　　接受用户i的出价，并根据y_{is}^t将服务器资源分配给任务；向用户i收取p_i

8) else

9)　　拒绝用户i

10) end if

11)　for all $t\in[T]$ do

12)　$(\{z_{jsd}^t\})=A_{\text{solve}}(\{y_{is}^t\},\{r_i^k\},\{c_{ks}\},\{\Omega_j^d\},\{r_j^k\})$

13)　end for

14)end while

3.2　长时任务迁移模型P_1

本节针对原始多目标优化问题拆分后的长时任务迁移模型P_1，由于模型在宏观时间尺度下涉及长时任务的计算时延和运营成本，引入时间表l和对偶变量处理模型，最后通过设计动态规划算法来求解，得出长时任务的竞拍结果x_{is}和迁移决策y_{is}^t。

P2:

$$\text{Maximize: }\sum_{i\in[I]}\Big(\sum_{s\in[S]}b_i x_{is}-g_i(\tau_i)\Big)-\sum_{i\in[I]}\sum_{t\in[T]}\sum_{s\in[S]}x_{is}C_{is}^t-\sum_{t\in[T]}f_t(e^t)\tag{9}$$

s.t.

$$C_1:\sum_{s\in[S]}x_{is}\leqslant 1,\forall i\in[I]$$

$$C_2:y_{is}^t t\leqslant d_i+\tau_i,\forall t\in[T],\forall s\in[S],\forall i\in[I]$$

$$C_3:\omega_i x_{is}\leqslant\sum_{\forall t\in[T]}y_{is}^t,\forall s\in[S],\forall i\in[I]$$

$$C_4:\sum_{\forall i\in[I]}r_i^k y_{is}^t\leqslant c_{ks},\forall t\in[T],\forall s\in[S],\forall k\in[K]$$

$$C_5:\sum_{\forall s\in[S]}\sum_{\forall k\in[K]}\beta_{ks}\left(\frac{\sum_{\forall i\in[I]}r_i^k y_{is}^t}{c_{ks}}\right)\leqslant e^t,\forall t\in[T]$$

$$C_6:\tau_i\in[0,T-d_i],e^t\geqslant 0,x_{is},y_{is}^t\in\{0,1\},\ \forall t\in[T],\forall s\in[S],\forall i\in[I]$$

如果令$g_i(\tau_i)=f_t(e^t)=0$，使场景在离线的设置下，没有约束C_2、C_3和C_5的问题P2仍然是一个NP-hard组合优化问题，且当涉及任务的软期限和运营成本时，挑战进一步升级。因为它涉及建模工作截止日期的非常规约束，本文提出了一个处理非常规约束的新框架P_1'，通过引入时间表l来吸收约束，变成一个新的变量x_{il}。

P3:

$$\text{Maximize: }\sum_{i\in[I]}\sum_{l\in[\xi_i]}b_{il}x_{il}-\sum_{t\in[T]}C_{is}^t x_{is}-\sum_{t\in[T]}f_t(e^t)\tag{10}$$

s.t.

$$C_1:\sum_{l\in[\xi_i]}x_{il}\leqslant 1,\forall i\in[I]$$

$$C_2:\sum_{i\in[I]}\sum_{l:t\in T(l),s\in S(l)}r_i^k x_{il}\leqslant c_{ks},\ \forall k\in[K],\forall s\in[S],\forall t\in[T]$$

$$C_3:\sum_{\forall s\in[S]}\sum_{\forall k\in[K]}\beta_{ks}\left(\frac{\sum_{i\in[I]}\sum_{l:t\in T(l),s\in S(l)}r_i^k x_{il}}{c_{ks}}\right)\leqslant e^t,\forall t\in[T]$$

$$C_4:e^t\geqslant 0,x_{il}\in\{0,1\},\forall t\in[T],\forall i\in[I],\forall l\in[\xi_i]$$

其中，ξ_i是用户工作的一组可行的时间表。一个可行的时间表是满足约束C_1、C_2和C_3的向量$l=(\{x_{is}\}_{\forall s\in[S]},\{y_{is}^t\}_{\forall s\in[S],\forall t\in[T]},\tau_i)$。$x_{il}$是二进制决策变量，根据时间表$l\in[\xi_i]$执行，其中如果用户$i$的作业被接受，则$x_{il}=1$，否则为0。$b_{il}$是基于时间表$l$的值，$b_{il}=b_i-g_i(\tau_i)$，$\tau_i$是根据超出截止日期

的持续时间。$T(l)$和$S(l)$分别表示在时间表l中何时何地执行用户i的任务的时隙集和服务器集。约束C_2和C_3等效于问题P2中的约束C_4和C_5。约束C_1保证作业只能根据一个时间表被接受。式(9)的可行解与式(10)的可行解互相对应，因此这2个问题的最优目标值相等。

虽然处理了非常规约束的麻烦，但是想要用原始对偶算法来解决，仍涉及指数数量的决策变量，还需要设计一个多项式时间内运行的算法。本文将$x_{il} \in \{0,1\}$松弛到$x_{il} \geqslant 0$，并引入对偶变量u_i、p_{ks}^t和m_t到约束C_1、C_2和C_3中。松弛问题式(10)的对偶问题P4如式(11)所示。

P4:

$$\text{Minimize: } \sum_{i \in [I]} u_i + \sum_{\forall k \in [K]} \sum_{\forall s \in [S]} \sum_{\forall t \in [T]} c_{ks} p_{ks}^t + \sum_{\forall t \in [T]} f_t^*(m_t) \tag{11}$$

s.t.

$$C_1: u_i \geqslant b_{il} - \sum_{\forall k \in [K]} \sum_{\forall t \in T(l)} \sum_{\forall s \in S(l)} r_i^k \left(p_{ks}^t + m_t \frac{\beta_{ks}}{c_{ks}} \right) - \sum_{\forall t \in T(l)} \sum_{\forall s \in S(l)} C_{is}^t, \forall i \in [I], \forall l \in [\xi_i]$$

$$C_2: p_{ks}^t, u_i, m_t \geqslant 0, \forall i \in [I], \forall t \in [T], \forall k \in [K], \forall s \in [S]$$

其中，$f_t^*(m(t))$是成本函数$f_t(\cdot)$的凸共轭函数，表示为

$$f_t^*(m(t)) = \sup_{e^t \geqslant 0} \left\{ m_t e^t - f_t\left(e^t\right) \right\} \tag{12}$$

$f_t^*(m(t))$的显式表达式为

$$f_t^*(m(t)) = \begin{cases} 0, & m_t \leqslant h_t \\ (m_t - h_t)e^t, & m_t > h_t \end{cases} \tag{13}$$

3.2.1 *在线拍卖机制设计*

在线拍卖机制设计中的一个关键问题是决定是否接受用户i的工作，以及如何安排其工作以最大限度地提高其效用，同时每个时隙的能耗是受限的。如果云提供商接受用户i在引入时间表l的服务器s上的作业，则$x_{is} = 1$，τ_i根据完成时间分配，y_{is}^t被更新，并且对于$T(l)$中的时隙，e^t是持续增加的。为了求解原凸问题式(9)，根据紧致指数凸问题式(10)及其对偶方程式(11)，对于每个原始变量x_{il}，问题P4中都有一个对偶约束C_1与之相关。原始对偶技术中的互补松弛表明x_{il}是基于其对偶约束更新的。x_{il}保持为0，除非其相关联的对偶约束变得严格。因为对偶变量u_i是非负的，所以问题P4中令u_i是0和约束C_1右侧之间的最大值，即

$$u_i = \max \left\{ 0, \max_{l \in [\xi_i]} \left\{ b_{il} - \sum_{\forall k \in [K]} \sum_{\forall t \in T(l)} \sum_{\forall s \in S(l)} r_i^k \left(p_{ks}^t + m_t \frac{\beta_{ks}}{c_{ks}} \right) - \sum_{\forall t \in T(l)} \sum_{\forall s \in S(l)} C_{is}^t \right\} \right\} \tag{14}$$

因此，中标者是基于u_i确定的。如果$u_i > 0$，边缘服务器接受用户i的工作，并根据时间表为其提供服务，且该时间表能使问题P4中约束C_1的右侧最大化。如果是$u_i \leqslant 0$，则边缘服务器将拒绝用户i的工作。如果将p_{ks}^t解释为时间t时边缘服务器s上的k型资源的单位资本价格，并且将m_t理解为时间t的单位能耗价格，则$\sum_{\forall k \in [K]} \sum_{\forall t \in T(l)} \sum_{\forall s \in S(l)} r_i^k \left(p_{ks}^t + m_t \frac{\beta_{ks}}{c_{ks}} \right) - \sum_{\forall t \in T(l)} \sum_{\forall s \in S(l)} C_{is}^t$是用户$i$的工作被时间表$l$接受和调度的总成本。

此外，问题P4中约束C_1的右侧是时间表l下用户i的效用。如果将u_i解释为用户i的效用，则式(14)中u_i的分配则保证了用户i的工作调度是最佳的，因为它是在当前价格的基础上有效地使其效用最大化，从而实现总体效用最大化以及拍卖的真实性。尽管u_i的计算涉及指数数量的对偶约束，但通过引入时间表，大多数约束可以通过基于动态规划的对偶算法进行过滤。通过算法2（第1~14行）完成固定多项式数量的调度，并使u_i取零值或取具有这些调度的公式（问题P4中约束C_1右侧）的最大值。将作业完成时间设定为$t_c, t_c \epsilon [t_i + \omega_i - 1, T)$，则最佳调度是在相同完成时间的所有调度中成本最低的一个。对于每个边缘服务器$s \in [S]$，都构造一组最佳调度，因此对偶算法输出S组最佳调度。最佳调度的构建可以通过动态规划方法来完成。基本案例是调度l_0，时隙槽为$[t_i, t_i + \omega_i - 1]$。通过每次都将完成时间向后推一个时隙槽，并计算用户i的任务在时间t运行的价格$c(t)$，即$c(t) = \sum_{k \epsilon [K]} r_i^k \left(p_{ks}^t + m_t \frac{\beta_{ks}}{c_{ks}} \right) - C_{is}^t$。如果完成时间超过最后期限$d_i$，价格将增加相应的惩罚，

即 $c(t)=\sum_{k\epsilon[K]} r_i^k\left(p_{ks}^t+m_t\frac{\beta_{ks}}{c_{ks}}\right)-C_{is}^t+g_i(t-d_i)$。在替换完成时间的过程中，只需要比较旧完成时间和旧完成时间之前的 ω_i-1 时隙的价格。接下来，用一个简单的例子来说明算法2是如何工作的。如果用户 i 在时间3到达并且 $\omega_i=4$，则基本调度 $l_0=\{3,4,5,6\}$。下一步是构建完成时间为7的最佳调度。假设 $\arg\max_{t\in\{3,4,5\}}c(t)=3$，如果 $c(6)<c(3)$，则最佳调度为{6,4,5,7}，否则为{3,4,5,7}。重复该过程，直到完成时间达到 T。

3.2.2　关于价格的设计

单位资本价格 p_{ks}^t 和单位电价 m_t。h_t 是电网在时间 t 收取的单位电价；在对偶变量 $m(t)$ 的解释基础上，设 $m_t=h_t$。对于 p_{ks}^t 的设计，为了激励资源分配并均衡负载，考虑指数函数的形式，能够反映价格的变动趋势。引入了一个新的变量 w_{ks}^t，表示在时隙 t 时服务器 s 上所分配的 k 类型资源的使用量。设 U_k 和 L_k 分别是每单位时间内每单位 k 类型资源的最大值和最小值。因此，不失一般性，设 $L_k>h_t,\forall t\in[T]$，并提出了一个价格函数，使总单价 $p_{ks}^t+m_t$ 在开始时等于 L_k，并最终达到 U_k。因为 $m_t=h_t$，让 p_{ks}^t 从 L_k-h_t 开始，并随着当前使用量 w_{ks}^t 的增长而呈指数增长。当 w_{ks}^t 超过 c_{ks} 时，p_{ks}^t 等于 U_k-h_t，边缘服务器将不再接受任何任务。本文依据稀缺性原理，资源越稀缺价格增长越快以抑制过度需求，且契合边际成本递增特性，负载高时价格反映高边际成本。同时保障拍卖真实性，价格与用户出价无关，并能动态适应边缘计算场景的负载变化，实现资源高效配置，使用指数形式动态定价。综上所述，$p_{ks}(t)$ 和 $m(t)$ 定义分别如式(15)和式(16)所示。

$$p_{ks}^t=(L_k-h_t)\left(\frac{U_k-h_t}{L_k-h_t}\right)^{\frac{w_{ks}^t}{c_{ks}}} \tag{15}$$

$$m_t=h_t,\forall t\in[T] \tag{16}$$

其中，U_k 和 L_k 具有真实性，分别表示为

$$U_k=\max_{i\in[I]:r_i^{k>0}}\left\{\frac{b_i}{r_i^k}\right\} \tag{17}$$

$$L_k=\min_{i\in[I]:r_i^{k>0}}\left\{\frac{b_i-g_i(T-d_i)}{\omega_i\sum_{k\in[K]}r_i^k}\right\} \tag{18}$$

算法2　调度算法 A_{core}

输入　$B_i,\{c_{ks}\},\{p_{ks}^t\},\{w_{ks}^t\},\{m_t\}\{e^t\}$

输出　$x_{is},\{y_{is}^t\},p_i,\{p_{ks}^t\},\{w_{ks}^t\},\{e^t\}$

1) for all $s\in[S]$ do
2) 将时隙 t（$t\in[t_i,T]$）添加到时间集合 $\mathcal{T}$ 上
3)　令时间表 l_0 包括 $\mathcal{T}$ 中的前 ω_i 个时隙 $(t_1,t_2,\cdots,t_{\omega_i})$；定义 $q=1$
4)　while $\omega_i+q\leqslant|\mathcal{T}|$ do
5)　　$l_q=l_{q-1}$
6)　　令 t_c 为 $\mathcal{T}$ 中第 ω_i+q 个时隙
7)　　$c(t)=\sum_{k\epsilon[K]}r_i^k\left(p_{ks}^t+m_t\frac{\beta_{ks}}{c_{ks}}\right)-C_{is}^t,$
　　$\forall t\in\{t_1,t_2,\cdots,t_{\omega_i},t_c\}$
8)　　if $t_c>d_i, c(t_c)=c(t_c)+g_i(t-d_i)$
9)　　$t_m=\arg\max_{t\in\{t_1,\cdots,t_{\omega_i-1}\}}c(t)$
10)　end if
11)　if $c(t_{\omega_i})<c(t_m)$，则对于时间表 l_q，将时隙 t_m 替换为 t_{ω_i}，将 t_c 存入 t_{ω_i} 中
12)　end if
13)　$\mathcal{P}_q=\sum_{t\in T(l_q)}c(t);\ q=q+1$
14)　end while
15)　$s^*=\arg\min_j\{\mathcal{P}_j\};\ \mathcal{P}_s^*=\mathcal{P}_{s^*};\ l_s^*=l_{s^*}$
16) end for
17) $\hat{s}=\arg\min_s\{\mathcal{P}_s^*\};\ \mathcal{P}\wedge=\mathcal{P}_{\hat{s}}^*, l\wedge=l_{\hat{s}}^*$
18) if $b_i-\mathcal{P}\wedge>0$ then
19)　$x_{i\hat{s}}=1;\ y_{i\hat{s}}^t=1;\ \forall t\in T(l\wedge);\ x_{il\wedge}=1$
20)　$u_i=b_i-\mathcal{P}\wedge;\ p_i=\sum_{k\epsilon[K]}\sum_{t\epsilon[T(l\wedge)]}r_i^k\left(p_{k\hat{s}}^t+m_t\frac{\beta_{k\hat{s}}}{c_{k\hat{s}}}\right)$
21)　$w_{k\hat{s}}^t=w_{k\hat{s}}^t+r_i^k,\forall k\in[K],t\in T(l\wedge);$
　　$e^t=e^t+\sum_{k\epsilon[K]}r_i^k\frac{\beta_{k\hat{s}}}{c_{k\hat{s}}},\forall t\in T(l\wedge)$
22)　$p_{k\hat{s}}^t=(L_k-h_t)\left(\frac{U_k-h_t}{L_k-h_t}\right)^{\frac{w_{k\hat{s}}^t}{c_{k\hat{s}}}},\forall k\in[K],$
　　$t\in T(l\wedge)$
23) end if
24) return $x_{is},\{y_{is}^t\},p_i,\{p_{ks}^t\},\{w_{ks}^t\},\{e^t\}$

如算法2所示，算法 A_{core} 在第 i 个用户到达时，通过对偶算法选择最大化用户 i 效用的最佳调度方案（第1~17行）。如果用户 i 可以获得正效用，则

原始变量x_{is}、y_{is}^t和x_{il}将相应更新（第19行）。然后第20行计算效用和支付。第21行增加指定服务器上的K个资源的使用量，并记录当前的功耗水平。最后，在第22行中更新单位资源价格。

3.3 短时任务模型$P_2^{(t)}$

根据3.2节求解模型P_1，得到长时任务的迁移决策参数y_{is}^t，在本模块引入y_{is}^t，从而能够证明短时任务$P_2^{(t)}$问题是凸优化问题，可以求解短时任务迁移决策z_{jsd}^t。

P5:

Maximize:

$$\sum_{j\in[J]}\sum_{s\in[S]}\sum_{d\in[D]}\sum_{t\in[T]}(z_{jsd}^t\varphi_{jsd}^t - z_{jsd}^t G_{jsd}^t) \tag{19}$$

s.t.

$$\mathrm{C}_1:\sum_{\forall j\in[J]}\sum_{\forall d\in[D]} r_j^k z_{jsd}^t \leqslant c_{ks} - \sum_{\forall i\in[I]} r_i^k y_{is}^t,$$
$$\forall t\in[T],\forall s\in[S],\forall k\in[K]$$

$$\mathrm{C}_2:\sum_{\forall s\in[S]} z_{jsd}^t \leqslant \Omega_j,\ \forall t\in[T],\forall j\in[J],\forall d\in[D]$$

$$\mathrm{C}_3: z_{jsd}^t \geqslant 0,\forall t\in[T],\forall s\in[S],\forall d\in[D],\forall j\in[J]$$

对于凸优化模型$P_2^{(t)}$，根据内点法求解器来求解模型，循环每个时隙求解得出所到达系统的短时任务的迁移决策z_{jsd}^t。

算法3 短时任务求解算法A_{solve}

输入 $\{y_{is}^t\},\{r_i^k\},\{c_{ks}\},\{\Omega_j^d\},\{r_j^k\}$

输出 $\{z_{jsd}^t\}$

1) 根据式(5)定义迁移成本函数G_{jsd}^t

2) 根据式(7)定义运营成本函数φ_{jsd}^t

3) for all $d\in[D]$ do

4) 在每个时间槽d，采用优化求解器（如内点法）求解问题$P_2^{(t)}$

5) end for

6) return $\{z_{jsd}^t\}$

3.4 理论证明

3.4.1 证明总体效用模型P是NP-hard问题

证明 在探讨本文模型构建与优化过程中，不可避免地面对了计算复杂性与求解效率之间的权衡。在仅考虑长时隙任务并保留关键约束（问题P1中约束C_4）的情境下，问题P1的求解难度显著上升，其本质特性与经典的背包问题呈现出高度的相似性[22]。背包问题作为组合优化领域的基石之一，其NP-hard性质早已被学术界广泛认可，这为分析该问题的复杂度提供了坚实的理论基础。具体而言，P模型在构建时纳入了更为丰富的变量集合与更为错综复杂的限制条件体系，这一设计旨在更精确地模拟现实世界的复杂场景，但同时也显著增加了问题的求解难度。通过对比分析不难发现，即使是在简化的背景下（即聚焦于长时隙任务与特定约束），问题P1也依然保留了背包问题那般的计算挑战，进而可以合理推断，本文构建的完整模型在本质上同样属于NP-hard范畴。证毕。

3.4.2 证明模型$P_2^{(t)}$为凸优化问题

证明 在目标函数中，z_{jsd}^t是决策变量，而φ_{jsd}^t和G_{jsd}^t是给定的参数（不依赖于决策变量）。由于目标函数是决策变量的线性函数（即每一项都是z_{jsd}^t与某个常数的乘积），因此目标函数是凸的（线性函数既是凸的也是凹的）。问题P5中约束C_1是线性的，线性不等式约束满足凸性要求，约束C_2同样是线性的，因此也满足凸性要求[23]。由于目标函数是凸的（实际上是线性的），且所有约束条件是线性的，因此模型$P_2^{(t)}$为凸优化问题，可以使用凸优化技术来求解。证毕。

3.4.3 证明算法在多项式时间内完成并求得可行解

证明 分析算法A_{core}的计算复杂度时，主要关注其内部循环和条件判断的效率。算法首先遍历所有服务器，复杂度为$O(S)$，对于每个服务器，它执行一系列复杂的操作来构建并优化时间表。这些操作包括检查资源限制，复杂度为$O(K)$，初始化时间集合（受限于$[t_i,T]$内的实际时隙数，复杂度小于$O(T)$），以及一个嵌套循环，该循环在最坏情况下可达到$O(T^2)$的复杂度，因为它涉及在多个潜在时间点上评估成本。尽管算法A_{core}中步骤13)中的选择操作复杂度理论上为$O(S)$，但使用适当的数据结构可优化为接近$O(1)$。最终，算法通过再次遍历服务器来选择成本最低的调度方案，并执行一系列更新操作，这些操作的复杂度通常受限于$O(KT)$。综合所有部分，算法2的总体时间复杂度主要由外层服务器遍历和内层时间优化决定，因此总体复杂度为$O(KST^2)$，算法A_{control}与A_{core}的复杂度为$O(IKST^2)$。此外，模型$P_2^{(t)}$为凸优化问题，采用内点法在多项式时间内求得最优解。综上，本文算法能够在多项式时间内完成。证毕。

3.4.4 证明TOSM满足拍卖的经济属性：个人理性与真实性

证明 本文的拍卖机制符合发布定价机制[24]，其中获胜者的确定过程和支付计算仅取决于当前资

源的价格。即 $p_i=\sum_{k\in[K]}\sum_{t\in[T(l^\wedge)]}r_i^k\left(p_{k\hat{s}}^t+m_t\frac{\beta_{k\hat{s}}}{c_{k\hat{s}}}\right)$，获胜的用户 i 为其任务执行付出的代价取决于分配的资源量和用户 i 的需求。它与用户 i 的出价 b_i 无关。因此，用户无法通过谎报投标价格来提高效用，因为效用是其估值与价格之间的差额。因此本文设计的在线拍卖机制是一个真实的拍卖，保证了通过真实的出价实现最大效用。

个人理性。即出价者无论竞拍是否成功，其利益非负。当用户竞拍失败时，利益 u_i 为 0，当用户竞拍成功时，$u_i=\left(v_i-g_i(\tau_i)\right)-p_i\geqslant\left(b_i-g_i(\tau_i)\right)-p_i=\left(b_i-g_i(\tau_i)\right)-\sum_{k\in[K]}\sum_{t\in[T(l^\wedge)]}r_i^k\left(p_{k\hat{s}}^t+m_t\frac{\beta_{k\hat{s}}}{c_{k\hat{s}}}\right)>0$，因此本文算法满足个体理性。综上，本文算法满足拍卖的经济属性：个人理性与真实性。证毕。

4　实验及结果分析

4.1　仿真参数设置和数据集

本文实验数据均由真实世界数据集和合理的随机化产生，模拟了用户乘坐地铁从始发站到终点站服务迁移全过程，地铁线路长度设置为 30 km。任务相关信息的数据集来自 Google Cluster Data，包含了有关在 Google 计算单元上运行的任务信息，包括任务开始时间、执行时间、持续时间和规范化资源需求（CPU 和 RAM）。本文将每一份任务转化为一份报价，根据需求从数据集中提取 2 种类型的资源。每个任务的截止日期是在其到达时间和系统结束时间之间随机生成的。通过将总资源需求乘以在 $[L_k,U_k]$ 范围内随机选择的单价来设定每个任务的投标价格，故参数单位为元/时隙/单位资源单位，默认的 L_k 值为 5，U_k 值为 50。

服务器相关信息基于墨尔本 CBD 地区用户数量最多的前 40 台边缘服务器，覆盖半径为 200 m，本文设定系统中边缘服务器提供资源由 CPU、RAM 和内存组成不同类型的虚拟机，因此服务器可用资源量抽象为[2 000,6 000]。此外，对于服务器的功耗，参数频率 f 设置为[1,4]（单位为 GHz），参数 β_{ks} 对于 CPU 设置在 [20, 60]（单位为 W），RAM 设置在[0.2,2][25]（单位为 W），服务器运营成本相关信息基于 ComEd 的实时电价数据。

4.2　基线方法

首先，将 TOSM 与下列 5 种基线方法对比，分别是 Offline 算法、OASTR 算法[17]、JMH 算法[19]、价格贪婪算法（PGA, price greedy algorithm）和期限贪婪算法（DGA, deadline greedy algorithm）[26]。由于实验环境不同，以上 5 种算法都在本文场景中进行了再现。同时为了对比算法的性能，将离线算法的最优解[27]作为基准进行对比，每个对比方法的简单描述如下。

1) Offline[27]。假设能够预知整个系统生命周期内的所有输入参数，通过 Gurobi 求解器解决原始问题 P 的离线最优解。

2) OASTR[17]。联合优化服务迁移和资源管理，基于 Lyapunov 的双时间尺度在线优化算法。

3) JMH[19]。基于最优 JMH 决策的迁移方法，在单时间尺度下联合管理计算和无线资源的服务迁移优化方法，利用整数松弛迭代算法求解决策。

4) PGA[26]。优先考虑服务器能耗成本，选择电价低的时隙进行迁移。

5) DGA[26]。优先考虑任务完成的时间，在尽量短的时间内完成任务。

实验研究不同用户数量对系统效用、平均时延以及平均能耗的影响；不同线性惩罚函数（系数各异）、不同服务器容量限制以及不同时延权重系数对系统效用的影响。

4.3　实验结果分析

4.3.1　用户数量对时延、能耗及系统效用的影响

首先比较了上述几种服务迁移方法中不同的用户数量对系统归一化平均时延、平均能耗和总体效用的影响。

如图 4 和图 5 所示，用户数量从 1 000 增加到 5 000 的过程中，每种方法的归一化平均时延和归一化平均能耗都有不同幅度的增长，这是由于用户数量的增加会成比例地提升节点间通信次数和数据量，占用更多带宽并增加时延。同时资源需求线性增长，边缘服务器需要处理和分配更多的资源量，从而导致能耗上升。从图 4 中可以看出，PGA 的归一化平均时延最高，这是由于 PGA 优先考虑服务器能耗成本，不考虑任务完成的时间，而 DGA 由于优先考虑了任务的完成时间，时延相比 PGA 较低，但与其他算法相比仍然较高，这是因为服务器的资源有限，DGA 虽然优先考虑任务完成的时间，但未进行合理的资源分配，负载不均导致时延增加甚至服务中断，而 TOSM、OASTR 和 JMH 算法综合优化

时延与能耗，在控制时延的同时合理分配资源。本文提出的TOSM算法在归一化平均时延、归一化平均能耗维度都最接近离线最优解，且波动水平较为稳定，体现了TOSM算法优化能耗和降低任务时延的优势。

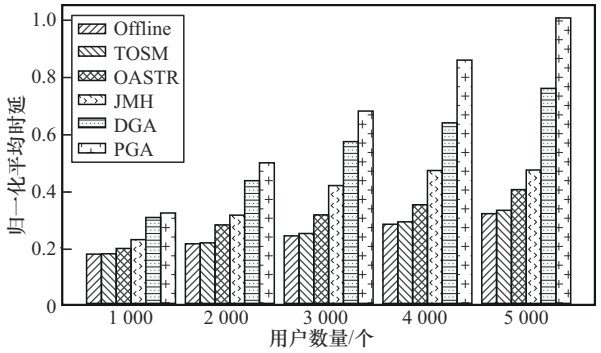

图4　不同用户数量对平均时延的影响

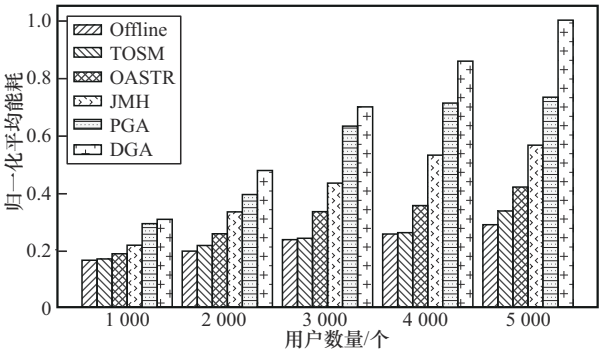

图5　不同用户数量对平均能耗的影响

如图6所示，用户数量从1 000增加到5 000的过程中，TOSM在归一化系统总体效用上提升了20%~50%，且最接近离线算法。这是因为随着用户数量的增加，拍卖机制能够调度更多不同的竞标来最大化整体效用。因此，TOSM在不同用户数量下的迁移效果都具有明显优势。

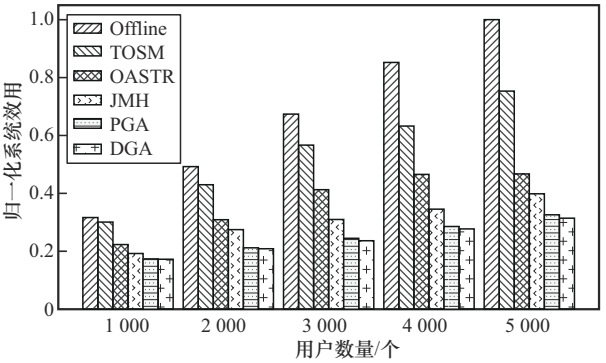

图6　不同用户数量对系统效用的影响

4.3.2　不同容量限制对系统效用的影响

在用户数量设置为1 000的条件下，逐步增加边缘服务器的系统容量。图7展示了不同容量限制下，TOSM与其他算法的归一化系统效用比较。随着边缘服务器资源容量的增加，系统效用逐渐上升，这是因为系统能够同时接受更多的竞标请求，因此能够创造更多的效用。实验结果表示，TOSM在不同边缘服务器容量限制条件下，均能实现较好的系统效用，展现了本文方法的鲁棒性。

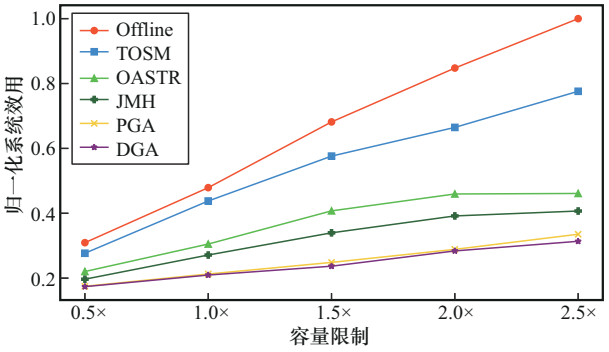

图7　不同容量限制对系统效用的影响

4.3.3　U/L比值对竞争比的影响

在线算法中的系统效用与离线算法（假设所有信息已知）的系统效用的比值为竞争比，在其他条件固定的情况下，将用户数量从1 000增加到5 000，同时设置U/L比值为80%、100%和120%。如图8所示，TOSM算法在不同U/L比值下的竞争比为1.1~1.2，且随U/L比值变化无明显波动，体现了本文算法的优越性，同时竞标用户数量对竞争比的影响不足10%，由此也体现了TOSM的稳定性。

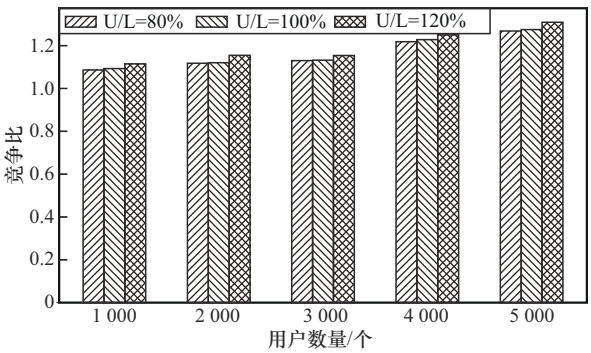

图8　不同U/L比值对竞争比的影响

4.3.4　不同时延权重系数对系统效用的影响

不同权重系数也是影响边缘服务迁移性能的重

要因素之一。按照相同的服务迁移场景设置进行模拟实验，将用户服务请求数量设置为 1 000，对目标函数中的相关时延系数进行调整，不同的时延权重系数对系统效用影响结果如图 9 所示。首先，随着时延权重系数的不断增加，每个方法的归一化系统效用逐步下降，这是由于优先选择通信成本低且能够在期限内完成任务的方案，调度空间受限，降低了拍卖机制所产生的效益，从而减少了系统效用的提升空间。从图 9 中可以看出，即使是在不同的时延权重系数下，TOSM 算法的归一化系统效用都具有明显优势。此外，随着时延权重系数增大，DGA 因考虑了任务完成时间而逐渐表现优于未考虑任务完成时间的 PGA，后者在高时延权重系数下的表现更糟糕。

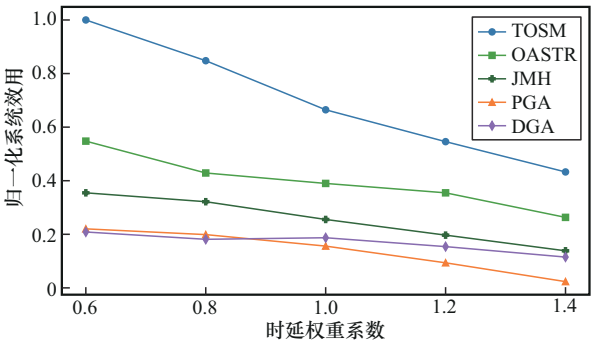

图 9　不同时延权重系数对系统效用的影响

4.3.5　拍卖的真实性

本节深入分析了竞价对拍卖结果和竞拍者个人利益的影响。通过仿真实验揭示了两者之间的内在联系和规律，如图 10 所示。随着竞价的逐步提高，拍卖结果逐渐从失败转向成功，并在达到关键出价点后，竞拍成功率保持在高位。这一发现不仅验证了本文算法严格遵循 Myerson 定理，确保了拍卖机制的真实性和公正性，还突显了算法在提升拍卖效率和透明度方面的优势，进一步揭示了竞拍者出价策略对其个人利益的优化作用。

4.3.6　算法运行时间

图 11 展现了本文算法的运行时间随用户数量变化的关系，实验结果表明，随着用户数量的增加，本文算法的运行时间逐渐延长，反映出在处理更复杂的竞拍场景时，该算法计算量的相应增加。这一趋势说明了本文算法在扩展性方面的特点，即在面对大规模竞拍时，虽然计算负荷增加，但算法仍能保持合理的运行效率。

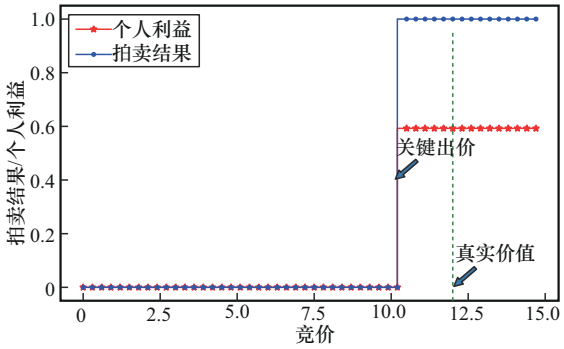

图 10　拍卖真实性

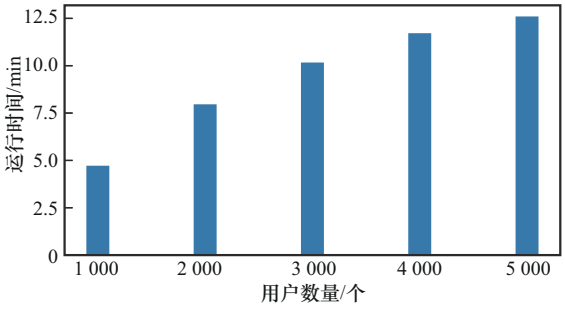

图 11　不同用户数量对运行时间的影响

5　结束语

本文针对移动边缘计算场景下的服务迁移问题，提出了一种双时间尺度的设计方案，通过分层优化宏观与微观时间尺度下的迁移决策过程，有效兼顾了长期约束和短期影响，同时设计在线拍卖机制，应对了动态资源分配和系统未来输入未知的挑战。基于真实数据集的仿真结果验证了 TOSM 方法优异的长期性能，与基线方法相比，不仅减少了平均时延和能耗，优化了系统效用，而且展现了本文方法更好的稳定性和鲁棒性。未来工作可将拍卖机制与服务器资源感知预测技术结合，进一步提高边缘网络的资源利用率和动态分配效率。

参考文献：

[1] LIN J C, RAO H L, LIANG S S, et al. Aphto: a task offloading strategy for autonomous driving under mobile edge[J]. The Journal of Supercomputing, 2024, 80(11): 16013-16045.

[2] LI C L, ZHANG Y, LUO Y L. Flexible heterogeneous data fusion strategy for object positioning applications in edge computing environment[J]. Computer Networks, 2022, 212: 109083.

[3] WANG J, HU J, MIN G Y, et al. Online service migration in mobile edge with incomplete system information: a deep recurrent actor-critic learning approach[J]. IEEE Transactions on Mobile Computing, 2023, 22(11): 6663-6675.

[4] WANG S Q, URGAONKAR R, ZAFER M, et al. Dynamic service mi-

gration in mobile edge computing based on Markov decision process[J]. IEEE/ACM Transactions on Networking, 2019, 27(3): 1272-1288.

[5] LIU Z J, XU X L. Latency-aware service migration with decision theory for Internet of vehicles in mobile edge computing[J]. Wireless Networks, 2024, 30(5): 4261-4273.

[6] MWASINGA L J, LE D T, RAZA S M, et al. RASM: resource-aware service migration in edge computing based on deep reinforcement learning[J]. Journal of Parallel and Distributed Computing, 2023, 182: 104745.

[7] QI Y X, PAN L, LIU S J. Service provisioning based on edge-cloud collaboration: a two-timescale online scheduling algorithm[J]. IEEE Internet of Things Journal, 2024, 11(19): 31999-32011.

[8] TOUMI N, BAGAA M, KSENTINI A. Machine learning for service migration: a survey[J]. IEEE Communications Surveys & Tutorials, 2023, 25(3): 1991-2020.

[9] SHAMSADINI A, ENTEZARI-MALEKI R. Time-aware MDP-based service migration in 5G mobile edge computing[C]//Proceedings of the 2022 27th International Computer Conference, Computer Society of Iran (CSICC). Piscataway: IEEE Press, 2022: 1-5.

[10] WANG S G, GUO Y, ZHANG N, et al. Delay-aware microservice coordination in mobile edge computing: a reinforcement learning approach[J]. IEEE Transactions on Mobile Computing, 2021, 20(3): 939-951.

[11] LI C L, ZHANG Y, GAO X, et al. Energy-latency tradeoffs for edge caching and dynamic service migration based on DQN in mobile edge computing[J]. Journal of Parallel and Distributed Computing, 2022, 166: 15-31.

[12] HUA K F, SU S C, WANG Y W. Intelligent service migration for the Internet of vehicles in edge computing: a mobility-aware deep reinforcement learning framework[J]. Computer Networks, 2025, 257: 111021.

[13] BOZKAYA E. Digital twin-assisted and mobility-aware service migration in mobile edge computing[J]. Computer Networks, 2023, 231: 109798.

[14] CHEN S Y, RUI L L, GAO Z P, et al. Service migration with edge collaboration: multi-agent deep reinforcement learning approach combined with user preference adaptation[J]. Future Generation Computer Systems, 2025, 165: 107612.

[15] DUAN J R, REN K, ZHOU W, et al. A service migration method for resource competition in mobile edge computing[C]//Proceedings of the 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC). Piscataway: IEEE Press, 2021: 1-8.

[16] XU M X, ZHOU Q H, WU H M, et al. PDMA: probabilistic service migration approach for delay-aware and mobility-aware mobile edge computing[J]. Software: Practice and Experience, 2022, 52(2): 394-414.

[17] SHI Y, YI C Y, WANG R, et al. Service migration or task rerouting: a two-timescale online resource optimization for MEC[J]. IEEE Transactions on Wireless Communications, 2024, 23(2): 1503-1519.

[18] LI J W, ZHAO D, SHI Z S, et al. Energy-efficient online service migration in edge networks[J]. IEEE Internet of Things Journal, 2024, 11(18): 29689-29708.

[19] LIANG Z Z, LIU Y, LOK T M, et al. Multi-cell mobile edge computing: joint service migration and resource allocation[J]. IEEE Transactions on Wireless Communications, 2021, 20(9): 5898-5912.

[20] CHEN S T, JIAO L, LIU F M, et al. EdgeDR: an online mechanism design for demand response in edge clouds[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(2): 343-358.

[21] ZENG Q S, DU Y Q, HUANG K B, et al. Energy-efficient resource management for federated edge learning with CPU-GPU heterogeneous computing[J]. IEEE Transactions on Wireless Communications, 2021, 20(12): 7947-7962.

[22] WANG L N, HE Y C, WANG X Z, et al. A novel discrete differential evolution algorithm combining transfer function with modulo operation for solving the multiple knapsack problem[J]. Information Sciences, 2024, 680: 121170.

[23] LIU J P, FU Z M, TAO F Z, et al. Energy management strategy based on convex optimization for fuel cell/battery/ultracapacitor hybrid vehicle[J]. Engineering Optimization, 2024, 56(3): 447-467.

[24] HUANG Z Y, KIM A. Welfare maximization with production costs: a primal dual approach[J]. Games and Economic Behavior, 2019, 118: 648-667.

[25] CAO H, CHEN L, ZHANG J Q, et al. An energy-aware IoT functions offloading strategy in solar-powered edge environment for smart agriculture[C]//CCF Conference on Computer Supported Cooperative Work and Social Computing. Singapore: Springer, 2025: 179-196.

[26] BORODIN A, LUCIER B. on the limitations of greedy mechanism design for truthful combinatorial auctions[C]//International Colloquium on Automata, Languages, and Programming. Berlin: Springer, 2010: 90-101.

[27] Gurobi Optimization. Gurobi optimizer reference manual, version 11.0[R]. Houston, USA: Gurobi Optimization, LLC, 2025.

[作者简介]

王海艳（1974– ），女，江苏东台人，博士，南京邮电大学教授、博士生导师，主要研究方向为服务计算、可信计算、大数据应用与云计算技术、隐私保护技术等。

张家豪（1999– ），男，江苏徐州人，南京邮电大学硕士生，主要研究方向为边缘计算。

骆健（1976– ），女，江西赣州人，南京邮电大学副教授，主要研究方向为服务计算、可信计算、服务推荐等。