

文章编号: 2095-4980(2024)11-1193-06

## 面向智慧城轨系统的云边协同计算架构设计与优化

燕增伟<sup>1</sup>, 林森<sup>\*2</sup>, 肖骁<sup>1</sup>

(1. 交控科技股份有限公司, 北京 100071; 2. 北京交通大学 自动化与智能学院, 北京 100071)

**摘要:** 城市轨道交通是城市交通运力的主要承载系统之一, 随着近年来智慧城市发展的需要越来越迫切, 智慧城轨系统的设计、研究与优化也成为许多学者的研究方向和重心。智慧城轨系统要求列车具有智能算力, 进而实现多种智能服务需求。由于列车的车载设备具有诸多限制, 在其上部署高算力的计算机设备并不现实, 故需引入其他设备为其提供算力支撑。本研究面向智慧城轨系统的特殊场景, 基于 5G 和边缘智能, 设计了一种面向智能任务的云边协同计算架构, 并将该架构中资源分配的流程进行了数学建模, 将其转化为最小化任务延迟的最优化问题。针对该最优化问题, 本文采用离散随机逼近算法进行求解, 以最小化智慧城轨系统的任务总处理延迟。仿真结果表明, 该算法能够有效降低智慧城轨系统中的智能任务处理时延。

**关键词:** 智慧城轨; 云边协同; 5G 技术; 边缘智能; 离散随机逼近

中图分类号: U285.4; TP39

文献标志码: A

doi: 10.11805/TKYDA2023044

## Design and optimization of cloud-edge collaborative computing architecture for smart urban rail systems

YAN Zengwei<sup>1</sup>, LIN Sen<sup>\*2</sup>, XIAO Xiao<sup>1</sup>

(1. Traffic Control Technology Co., Ltd, Beijing 100071, China; 2. School of Automation and Intelligence, Beijing Jiaotong University, Beijing 100071, China)

**Abstract:** Urban rail transit is one of the primary systems for urban transportation capacity. With the increasing demand for the development of smart cities in recent years, the design, research, and optimization of smart rail transit systems have become the research direction and focus for many scholars. Smart rail transit systems require trains to have intelligent computing power to meet a variety of intelligent service needs. Due to the numerous limitations of on-board equipment in trains, it is not practical to deploy high-performance computing devices on them, hence the need to introduce other devices to provide computational support. This study, aimed at the special scenarios of smart rail transit systems, designs a cloud-edge collaborative computing architecture for intelligent tasks based on 5G and edge intelligence. The resource allocation process in this architecture is mathematically modeled and transformed into an optimization problem that minimizes task latency. To solve this optimization problem, this paper employs a discrete stochastic approximation algorithm to minimize the total processing delay of tasks in the smart rail transit system. Simulation results indicate that the algorithm can effectively reduce the processing delay of intelligent tasks in smart rail transit systems.

**Keywords:** smart urban rail; cloud-edge collaboration; 5G; edge intelligence; discrete stochastic approximation

城市轨道交通系统可以改善公共交通和城市环境, 并且具有承载力强、速度快、安全便捷等优点, 是很多城市居民的首要出行选择。近年来, 随着智慧城市这一概念逐渐兴起, 为城轨系统智慧赋能也就成为了建设智慧城市的至关重要的一环, 为此, 需要为列车等城轨重要组成单位赋能, 使其满足智能服务的需求<sup>[1-4]</sup>。

收稿日期: 2023-02-24; 修回日期: 2023-04-25

基金项目: 国家自然科学基金资助项目(61973026)

\*通信作者: 林森 email: 21120216@bjtu.edu.cn

智慧城轨系统中的智能服务需求大都具有低时延的要求, 为保证实时性, 需在有限的空间中部署具有足够算力的服务器, 为可靠的低时延的列车智能计算设计提出挑战。针对该问题, 本文结合了 5G 与边缘智能(Edge Intelligence, EI)技术。5G 通信带宽高, 传输时延低, 可提供可靠通信, EI 结合边缘计算与人工智能技术的优势, 将人工智能模型部署于边缘服务器, 为终端设备提供智能算力支持。于轨旁部署 EI 服务器, 利用 5G 通信技术建立可靠的车边通信通道, 使得列车将智能任务发送至 EI 服务器。由于 EI 服务在网络边缘<sup>[5-6]</sup>, 与车之间的传输距离较短, 相比于云计算, 任务数据传输时间更短, 列车能耗更低, 任务实时性更强。

然而, 单一网络位置的智能算力支持并不足够, 故需引入协同计算: 在 EI 的架构中保留云计算中心, 进行云边协同计算设计, 云计算强大的计算能力及其丰富的存储资源可以为 EI 服务器提供强力支持<sup>[7-10]</sup>。在云边协同计算框架中, 面对多种不同类型的任务时, EI 服务器应为任务分配相应的云计算资源以及云边通信资源, 将其发送至云计算中心处理, 以实现云边协同计算。如何为 EI 服务器求解最优资源分配策略也是本文需要考虑的关键问题。

本文采用离散随机逼近(Discrete Stochastic Approximation, DSA)算法求解 EI 服务器的最优资源分配策略。DSA 算法是一种求出目标函数最优解的迭代优化算法, 以估计的方式无限逼近最优解<sup>[11]</sup>。所谓离散是指决策变量离散, 这正契合本文所要研究的面向智慧城轨的云边协同架构中 EI 服务器的资源分配动作的离散性。

本文面向智慧城轨系统的特殊场景, 设计了一个针对智能任务的云边协同计算架构。该架构充分利用不同网络位置的性能, 采取 5G 技术建立不同网络位置之间通信信道, 为智慧城轨的智能计算需求提供算力支持, 全面降低智能任务计算时延。同时本文将系统进行数学建模, 将资源分配策略求解问题转化为最小化任务延迟的最优化问题。针对该最优化问题, 采用 DSA 算法进行求解, 以获取最优资源分配策略。

## 1 面向智慧城轨的云边协同计算系统模型与问题公式化

### 1.1 系统模型

本文设计了一个面向智慧城轨的云边协同计算系统, 用于智慧城轨系统中的智能任务资源分配。如图 1 所示, EI 服务器部署在轨道旁, 运行的列车收集智能任务数据, 并通过 5G 无线通道将其转发到最近的 EI 服务器。

设一个 EI 服务器可保存  $Q_{\max}$  个任务,  $\alpha$  表示任务大小,  $A$  表示其计算能力。在  $t$  时间时, EI 服务器上有  $Q_t^1$  个任务等待资源分配。EI 服务器所选择的策略为一个决策链。动作表示为 0、1 或 2, 其中 0 表示搁置, 1 表示本地计算, 2 表示卸载至云。通过计算出决策链中数字 0、1 和 2 的个数与索引, 可知 EI 服务器将在本地计算  $Q_t^1$  个任务, 上传  $Q_t^2$  个任务, 搁置  $Q_t^3$  个任务 ( $Q_t^1 + Q_t^2 + Q_t^3 = Q_t$ )。设在  $t$  时间, 任务的处理时长为  $L_t$ 。

#### 1.1.1 计算模型

设  $L_t^1$  为任务的搁置时长, 设每步时间长度为  $\tau$ , 则有:

$$L_t^1 = Q_t^3 \tau \quad (1)$$

设  $L_t^2$  为 EI 服务器的计算时长, 计算公式为:

$$L_t^2 = \frac{Q_t^1 \sum_{i=1}^{Q_t^1} \alpha_i}{A} \quad (2)$$

类似地, 云服务器计算时长  $L_t^3$  的计算公式为:

$$L_t^3 = \frac{Q_t^2 \sum_{i=1}^{Q_t^2} \alpha_i}{C} \quad (3)$$

式中  $C$  为云服务器的计算能力。

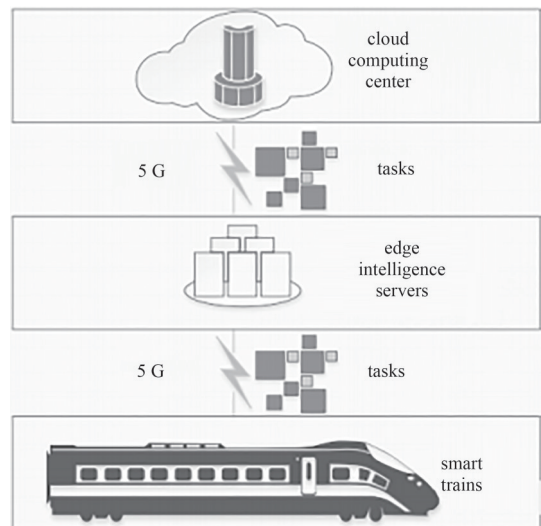


Fig.1 Architecture of cloud-edge collaborative computing system for smart urban rail

图 1 面向智慧城轨的云边协同计算系统架构

### 1.1.2 通信模型

设云边通信链路频道带宽为  $BW$ ，频道增益为  $G$ ，传输功率与噪音功率分别为  $P_T$  和  $P_N$ ，云边数据通信速率计算如下<sup>[12]</sup>：

$$\varphi = BW \log_2 \left( 1 + \frac{P_T |G|^2}{P_N^2} \right) \quad (4)$$

任务从边缘上传到云的通信时长  $L_t^1$  为：

$$L_t^1 = \frac{Q_t^2 \sum_{i=1}^{Q_t^2} \alpha_i}{\varphi} \quad (5)$$

任务处理总时长为：

$$L_t = L_t^1 + L_t^2 + L_t^3 + L_t^4 \quad (6)$$

### 1.1.3 罚函数模型

由于边缘服务器可拒绝接收列车上传的任务，当边缘服务器拒绝所有任务时，任务处理评估结果最优，但这种情况并不应出现，故设计罚函数：

$$\omega = \gamma \max \{ Q_t - Q_{\max}, 0 \} \quad (7)$$

式中  $\gamma$  为常数，大小对应惩罚强弱。

### 1.1.4 风险模型

考虑智慧城轨的特殊性质及 EI 服务器的资源上限，EI 服务器长时间超载运算会带来物理损伤。故本文将超载相关风险考虑为本问题中的一个约束条件。风险模型用于控制本地处理任务数目，平均超载风险  $r$  计算如下：

$$r = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{i=1}^T \frac{Q_t^1 \sum_{i=1}^{Q_t^1} \alpha_i}{A} \quad (8)$$

## 1.2 问题公式化

本资源分配策略求解问题公式化如下：

$$\begin{cases} \min_{Q_t^1, Q_t^2, Q_t^3} P(Q_t^1, Q_t^2, Q_t^3) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{i=1}^T L_t + \omega_t \\ \text{s.t. } C_1: -\varepsilon \leq r \leq 1 + \varepsilon \\ C_2: Q_t^2 \sum_{i=1}^{Q_t^2} \alpha_i \leq \tau \varphi_t \\ C_3: Q_t^1 + Q_t^3 \leq Q_{\max} \end{cases} \quad (9)$$

式中： $C_1$  防止超载运算损害硬件使用寿命； $C_2$  限制卸载任务总大小不可以超过通信速率限制； $C_3$  限制存储任务数目不可超过上限。考虑策略求解空间较大，本文采用 DSA 算法对该问题进行求解，具体算法细节以及概念在下节中具体描述。

## 2 基于离散随机逼近的求解方案

EI 服务器需要对所有任务做出资源分配决策，该决策空间极其庞大，使该优化问题求解困难。在初始阶段，服务器没有具体的策略探索方向，因此可以将  $P(Q_t^1, Q_t^2, Q_t^3)$  的最小化问题看作一个离散随机优化问题。

### 2.1 传统离散随机优化方法

对于给定的离散随机优化的一个解是穷尽地搜索可行集  $\Omega^{[11]}$ 。首先计算目标函数的经验平均值，设采取动作  $(Q^1, Q^2, Q^3)$ ，在很长的一段时间  $T$  内，根据强大数定律：

$$\hat{P}(Q_t^1, Q_t^2, Q_t^3) \triangleq \frac{1}{T} \sum_{i=1}^T P(Q_t^1, Q_t^2, Q_t^3) \rightarrow E[P(Q_t^1, Q_t^2, Q_t^3)] \quad (10)$$

最优的动作：

$$(Q_t^1, Q_t^2, Q_t^3)^* = \arg \min_Q P(Q_t^1, Q_t^2, Q_t^3) \tag{11}$$

穷举搜索需要大量的计算，计算次数与时间  $T$  成正比，效率低下。另外，智慧城轨场景是一个动态系统，如果任务数目在小于  $T$  的时间段内变化，经验平均值无法获取，穷举搜索的离散随机优化就无法实现。

### 2.2 离散随机逼近方法

DSA 方法可克服穷举法的缺点<sup>[13]</sup>，是一种求出目标函数最优解的迭代优化算法，虽然不能明确计算，但可计算出对最优解极其逼近的估计。解决该资源分配决策问题的 DSA 算法的伪代码如下：

算法 1 基于离散随机逼近的智慧城轨系统资源分配策略求解算法

输入：向量  $\pi^t$ ，策略空间  $\Omega$ ，目标函数  $P$ ，向量  $e$ ，迭代因子  $v_t$

输出： $t$  步的最优资源分配策略

初始化： $t$  置 0，从策略空间  $\Omega$  中随机选出初始动作  $(Q_0^1, Q_0^2, Q_0^3)$ ，设  $\pi^0(Q_0^1, Q_0^2, Q_0^3)$  为 1， $\pi^0$  的其余元素为 0，初始化对最优资源分配策略的估计为  $(Q_0^1, Q_0^2, Q_0^3)^*$

寻找最优策略循环

for  $t=0, \dots$  do

1) 抽样并评估

2) 计算任务处理成本  $P(Q_t^1, Q_t^2, Q_t^3)$

3) 在  $\Omega$  空间中均匀抽样  $(Q_t^1, Q_t^2, Q_t^3).new()$

4) 计算任务处理成本  $P(Q_t^1, Q_t^2, Q_t^3).new()$

5) 接受新策略

6)  $(Q_{t+1}^1, Q_{t+1}^2, Q_{t+1}^3) \leftarrow \arg \max P(Q_t^1, Q_t^2, Q_t^3)$

7) 更新最优策略

$$\text{更新状态选择概率的自适应滤波器 } \pi^{t+1} \leftarrow (1 - v_{t+1})\pi^t + v_{t+1}e_{(Q_{t+1}^1, Q_{t+1}^2, Q_{t+1}^3)}, v_{t+1} = \frac{1}{t+1}$$

$$\text{更新最优资源分配策略的估计值 } (Q_{t+1}^1, Q_{t+1}^2, Q_{t+1}^3)^* \leftarrow \arg \max \pi^{t+1}(Q_t^1, Q_t^2, Q_t^3)$$

end for

输出最优资源分配策略

该算法在随机探索可行集  $\Omega$  的同时，跟踪被访问的资源分配策略的被访问次数。在每一轮迭代中，算法倾向于选择具有较低任务处理成本的资源分配策略。本文使用  $|\Omega| \times 1$  向量  $\pi^t$  来记录在  $t$  时刻之前已经访问过的每个资源分配策略的相对频率：

$$\pi^t(Q_t^1, Q_t^2, Q_t^3) = \frac{\text{selected time of } (Q_t^1, Q_t^2, Q_t^3)}{t} \tag{12}$$

在初始化阶段，向量  $\pi^0$  的所有元素值都被设置为 0，但与初始选择的资源分配策略所对应的元素值除外。

图 2 为该算法的流程，每次迭代包括 4 个步骤。

步骤 1：统一选择一个资源分配策略  $(Q_0^1, Q_0^2, Q_0^3)^*$ ，并计算相应的目标函数  $P$  的值。步骤 2：如果随机选择的目标函数值小于当前状态  $(Q_t^1, Q_t^2, Q_t^3)$  的值，则将其设置为下一个资源分配策略。步骤 3：更新资源分配策略被选中的概率。 $e$  是一个  $|\Omega| \times 1$  向量，其中除了值为 1 的第  $(Q_t^1, Q_t^2, Q_t^3)$  个元素外，其他的所有元素都为 0。新更新的资源分配策略被选中概率由两部分组成。第一部分表示从最后一次迭代计算出的概率，利用一个  $(1 - v_{t+1})$

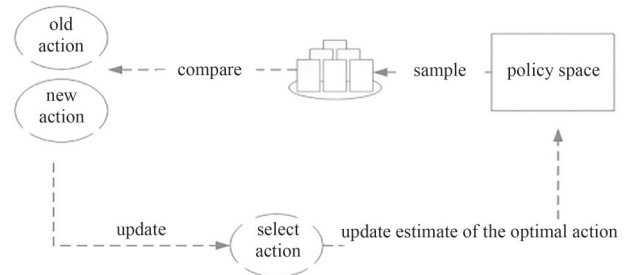


Fig.2 Flow of discrete stochastic approximation algorithm  
图 2 离散随机逼近算法流程

因子进行收缩。第二部分是步骤 2 中新选中的资源分配策略的影响，其中  $e$  是一个向量，除了  $(Q_{t+1}^1, Q_{t+1}^2, Q_{t+1}^3)$  元素外，其余都为 0。迭代的步长为  $1/t$ ，即随着时间的推移，算法在探索资源分配策略新的可能性方面越来越保

守<sup>[14]</sup>。步骤 4：如果新资源分配策略被选中的频率高于最后一次迭代中选择的资源分配策略的被选中频率，则将其选择为新的资源分配策略。该算法的每次迭代都对应了一批新任务的到达以及旧任务的资源分配，在每次迭代结束时，将  $(Q_i^1, Q_i^2, Q_i^3)$  选择作为资源分配策略。

本算法的状态序列是状态空间  $\Omega$  上的一个马尔可夫链，即使访问的状态不能保证收敛，序列  $(Q_i^1, Q_i^2, Q_i^3)$  几乎肯定可以收敛到全局最优，使用与文献[13]中类似的证明，本文验证了该算法在经过足够数量的迭代后能够收敛于全局最优。假设目标的平均值和方差分别为  $\mu$  和  $\sigma^2$ 。考虑目标函数的观测值构成的经验累积分布函数，其中随机变量表示这些观测值。可得目标函数值的经验分布。图 3 是目标函数值的一个示例，其分布可以近似为高斯分布，进而验证了其收敛性。

### 3 仿真

通过计算机仿真验证面向智慧城轨系统的云边协同计算架构的性能。首先设置仿真环境，包括仿真参数以及智能任务和模型，然后展示仿真结果，同时根据仿真数据评估所提方法的性能。如表 1 所示，本研究将智能任务的性质，包括任务的类型、每个时间段完成的任务的数量，以及在 EI 服务器上每个任务所需的 CPU 周期进行了总结分析。

表 1 智能任务的特性

Table1 Characteristics of artificial intelligence tasks

categories	number of tasks arriving per time period	CPU cycles required for each task processing/ $10^7$
text-based fault diagnosis	60	20
image-based intrusion detection	30	60
system health management based on hybrid data	50	32

仿真过程中，无线频道带宽设为 10 MHz，单个任务平均大小为 400 kB，信道传输功率为 500 mW，噪音功率为 -50 dBm，EI 服务器计算能力为 5 GHz，云计算能力为 100 GHz。

基于仿真结果，本文评估了风险模型对算法性能的提升效果，测试了一个没有风险模型的 DSA 算法，用来对比当该算法不考虑 EI 服务器的计算能力约束时，在 EI 服务器上的计算过载概率和持续时间。图 4 为两种算法下的延迟性能，以及在改变风险常数值的情况下，2 个算法导致的 EI 服务器计算超载情况。随着算法逐渐收敛，有风险约束的算法与一般算法的任务处理延时几乎一致，但后者的策略导致了 EI 服务器的计算过载。

为展示 DSA 算法的收敛性能，采用强化学习中的贪婪算法和深度强化学习进行仿真对比。图 5 为 3 种算法的收敛对比，以及在任务数成倍增加时，任务处理延时性能的对比。从图 5(a)可以看出，DSA 算法在该架构中，收敛速度更快，能够较快地达到一个令人满意的效果。而另外 2 个强化学习算法则需要较长的时间才能达到收敛，体现了 DSA 算法在大策略空间下收敛速度优势。

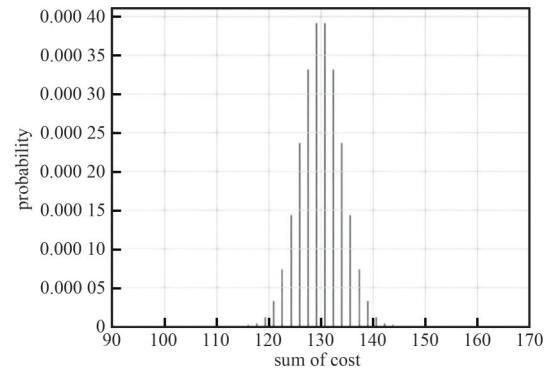


Fig.3 Statistical distribution map of the sum of task processing costs

图 3 任务处理成本总和的统计分布图

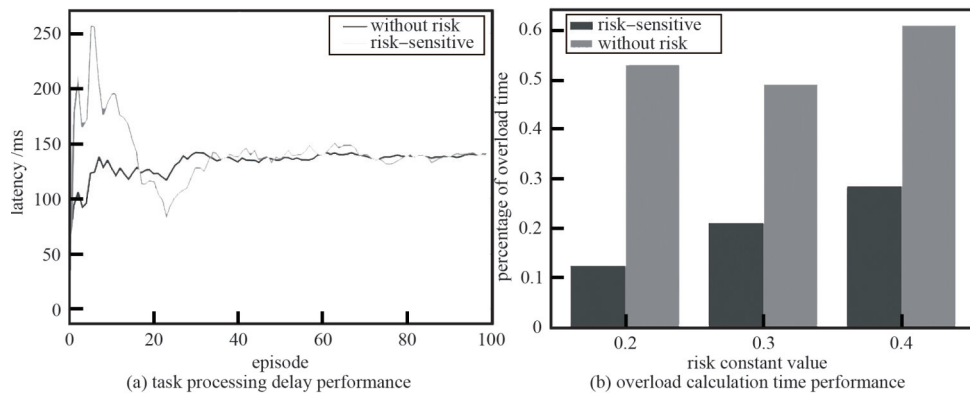


Fig.4 Risk-sensitive algorithm versus algorithm without-risk

图 4 风险敏感算法与无风险敏感算法对比

从图 5(b)中可看出, DSA 算法、贪婪算法以及深度强化学习算法在任务总数不断提升的情况下, 仍能保证一定的任务处理效果, 即保证较优的资源分配决策, 但不采用该云边协同计算架构, 即完全在列车本地处理任务的情况下, 任务处理的平均成本有明显的上升, 这与智慧城轨系统中列车的车载设备算力不足有密切关系。同样地, 这也证明了本文所采用的 DSA 算法能够有效降低智能任务的平均处理时延, 为智慧城轨系统的列车提供智能算力支持。

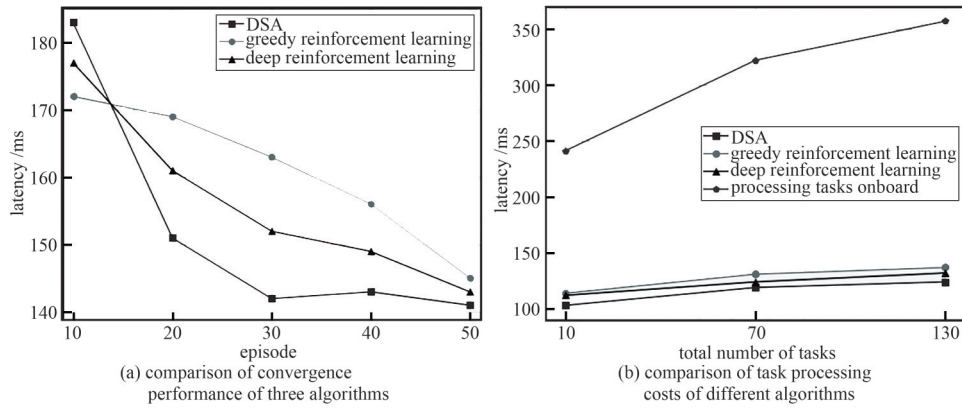


Fig.5 Comparison of convergence performance and task processing cost of different algorithms

图5 不同算法收敛性能、任务处理成本对比

## 4 结论

本文提出了一个面向智慧城轨系统的云边协同计算架构以及资源分配方案, 为智慧城轨系统提供可靠和实时的智能计算服务。在协同计算系统中, 设计了面向智慧城轨系统的云边协同计算的架构和流程。考虑到EI服务器计算资源容量有限, 智慧城轨场景特殊, 将资源分配问题表述为一个约束马尔可夫决策过程。为最小化长期平均延迟, 提高收敛速度, 采用 DSA 算法进行资源分配策略的求解, 并对协同计算系统进行优化。综合仿真结果表明, 本文设计的基于 DSA 算法的面向智慧城轨系统的云边协同计算架构在满足 EI 服务器的计算资源约束的同时, 也可显著减少任务处理延迟。通过与贪婪算法、深度强化学习算法的对比, 更加证明了 DSA 算法的收敛性能以及策略的最优性。与不采用该云边协同计算架构相比, 本文所采用的云边协同计算架构以及资源分配方案可以显著降低智慧城轨系统中列车收集的智能任务的平均处理时延, 为智慧城轨系统提供低延迟的智能计算服务, 为智慧城轨系统发展助力, 加速智慧城市的建设。

### 参考文献:

- [1] 路向阳,李东林,李雷,等. 智慧城轨下智慧车辆装备技术的研究与展望[J]. 机车电传动, 2018(6):1-8. (LU Xiangyang,LI Donglin,LI Lei,et al. Study and prospect of equipment technology for smart rolling stock in smart urban rail transit system[J]. Electric Drive for Locomotives, 2018(6):1-8.) doi:10.13890/j.issn.1000-128x.2018.06.001.
- [2] 雷成健,张超,吕浩炯,等. 智慧城轨下智能列车运行技术的研究与展望[J]. 现代城市轨道交通, 2020(8):38-44. (LEI Chengjian,ZHANG Chao,LYU Haojiong,et al. Study and prospect of operation technology for smart train in transit system[J]. Modern Urban Transit, 2020(8):38-44.)
- [3] 张祥杰,杨廷志,徐世东,等. 智慧城轨车辆技术标准体系构建研究[J]. 中国标准化(英文版), 2022,117(5):60-64. (ZHANG Xiangjie,YANG Tingzhi,XU Shidong,et al. Research on the construction of smart city rail vehicle technical standards system[J]. China Standardization, 2022,117(5):60-64.)
- [4] 丁叁叁. 智能列车助力智慧城轨发展[J]. 国外铁道车辆, 2021,58(5):1-8. (DING Sansan. Smart train contributes to the development of smart urban rail transit[J]. Smart Rail Transit, 2021,58(5):1-8.) doi:10.3969/j.issn.2097-0366.2021.05.001.
- [5] SHI Weisong,CAO Jie,ZHANG Quan,et al. Edge computing: vision and challenges[J]. IEEE Internet of Things Journal, 2016, 3(5):637-646. doi:10.1109/JIOT.2016.2579198.
- [6] 陆鹏威,颜子彦,张伟,等. 一种面向电磁识别模型的分散计算方法[J]. 太赫兹科学与电子信息学报, 2022,20(1):22-28. (LU Pengwei,YAN Ziyang,ZHANG Wei,et al. Decentralized calculation of neural network model for electromagnetic object detection[J]. Journal of Terahertz Science and Electronic Information Technology, 2022,20(1):22-28.) doi:10.11805/TKYDA2021153.