文章编号:1671-4598(2024)06-0206-07 DOI:10.16526/j.cnki.11-4762/tp.2024.06.030 中图分类号:TP393 文献标识码:A

基于改进 SMDP 的车载任务卸载决策算法

赵振博，付青坤，任雪容
（长安大学 信息工程学院，西安 710064）

摘要：针对边缘服务器的负载过重问题，可以将路边空闲车辆以及移动车辆应用虚拟化技术整合成资源池，为时延敏感类任务提供弹性服务；由此建立了一个分组传输的通信系统模型，为降低二进制指数退避算法中的信道碰撞概率，采用基于网络车辆节点的数量来适当调整最小竞争窗口的方法；结合分配资源的时序决策特点，提出车载边缘计算系统中基于改进的半马尔科夫决策过程的计算卸载策略，在制定系统动作的最优策略时，引入带有余弦项的非线性权重因子，对立即收益和未来期望收益进行动态加权，根据贝尔曼方程进行价值迭代，实现系统长期收益的最大化；仿真结果表明，所提策略能有效降低卸载时延，提高系统吞吐量，同时系统的长期收益也有显著的提升。

关键词：虚拟化技术；分组传输；最小竞争窗口；时序决策；车载边缘计算；非线性权重

Decision Algorithm for Vehicle Task Offloading Based on Improved SMDP

ZHAO Zhenbo, FU Qingkun, REN Xuerong
(School of Information Engineering, Chang'an University, Xi'an 710064, China)

Abstract: Aimed at the overload problem of edge servers, the virtualization technology can be applied to integrate idle vehicles on the roadside and mobile vehicles into a resource pool to provide elastic services for delay-sensitive tasks. Therefore, a packet transmission communication system model is established. In order to reduce the probability of channel collision in the binary exponential backoff algorithm (BEB), the minimum competition window is adjusted appropriately based on the number of vehicle nodes in the network. Considered the timing characteristics of resource allocation, a computational offloading strategy is proposed for vehicle edge computing (VEC) system based on the improved semi-Markov decision process (SMDP). With the optimal strategy of system action developed, the nonlinear weight factor with cosine term is introduced to dynamically weight the immediate reward and future expected reward. The iterative algorithm based on Bellman equation is utilized to achieve the maximum long-term reward. Simulation results show that the proposed strategy can effectively reduce the offloading delay, increase the throughput, and significantly improve the long-term reward of the system.

Keywords: virtualization technology; packet transmission; minimum contention window; timing decision; VEC; nonlinear weight

0 引言

新一轮的产业变革开启了车联网发展的新篇章，通过融合以 5G 技术[1]为支撑的低延时、高带宽网络通信，可以实现车辆的全面智能互联化[2]。在技术更迭的同时，车载终端由于智能软件的使用产生了海量数据[3]，仅凭车辆自身的车载计算单元（OBU，on board unit）来完成计算任务不易实现，由此大量的数据计算与实时传输对网络通信提出了更为严苛的要求[4]。传统移动云计算可以依托丰富的计算和存储资源，向终端用户提供各种计算服务，但偏远的距离会导致较高的数据传输时延[5]，很难满足现实场景的需求。而移动边缘计算（MEC，mobile edge computing)[6]技术可以减轻这种压力，通过在离用户数据源和终端设备更近的路侧单元（RSU，road side unit）部署 MEC 服务器[7]，能提供就近服务，从而显著提高系统的处理速度和响应能力。但是有限的 MEC 计算资源也难以维系不断涌现的数据处理任务[8-9]，在高负载情况下也会出现服务崩溃的现象[10]，又考虑到在车载边缘计算（VEC，vehicle edge computing）环境中有大量闲置的智能网联车辆（ICV，intelligent connected vehicle)，因此车和车通信（V2V，vehicle to vehicle）成为计算卸载的研究热点[11]。

现阶段，有大量关于车辆协作实现计算卸载的理论研究。文献［12］将车辆划分成请求型和服务型两种任务节点，同时把任务调度视为双目标的优化问题，旨在平衡响应时延和系统稳定性两者之间的关系。文献［13］通过招募通信覆盖范围内的志愿者车辆来协同处理计算任务，以改善边缘服务器资源受限以及任务计算密集的困境，最后用改进的遗传算法制定边缘服务器的定价方案，用 Stackelberg 博弈算法得到最佳的任务分配策略。文献［14］中边缘计算节点以反向拍卖机制制定 V2V 卸载策略，将整个处理过程建模成一个整数的非线性规划问题，以期减少服务供应商的成本支出。文献［15］顾及在红绿灯的停留间隙，停留车辆中足够的闲置资源未被使用，因而可以通过 V2V 通信方式实现任务处理，将计算卸载简化成一个 min-max

收稿日期:2023-06-16； **修回日期**:2023-07-16。
作者简介:赵振博(1997-),男,硕士生。
引用格式:赵振博,付青坤,任雪容.基于改进 SMDP 的车载任务卸载决策算法[J].计算机测量与控制,2024,32(6):206-212.

问题，并基于粒子群算法得到最终的卸载策略。文献［16］以附近智能车辆的移动特性、计算资源以及剩余电量等各种因素为研究内容，将任务处理问题转化成马尔可夫决策过程，使用双层深度 Q 网络制定最佳的卸载决策。

但是，现有的研究大多没有考虑到引起时延的具体信道特性。文献［17］提出二进制指数退避（BEB，binary exponential backoff）机制的数据传输方法，但是在竞争信道时存在占用不公平和信道真实情况难以反映的问题，因此本文使用基于车辆节点数来适当调整最小竞争窗口的方法，从而改善网络性能；同时，文献［17］在应用经典半马尔科夫决策过程（SMDP，semi-Markov decision process）算法进行计算卸载时，未充分权衡立即收益和未来期望收益两者之间的关系，本文引入带有余弦函数的加权因子，对两种收益进行非线性加权，调整动作决策时的最优策略。

1　系统模型

1.1　网络架构

VEC 系统如图 1 所示，该系统由路侧单元 RSU、邻近部署的 MEC 服务器、路边的闲置车辆以及同向行驶的移动车辆组成。当移动车辆产生了计算任务时，将向边缘节点发出请求，由 MEC 服务器做出处理。其中，RSU 与 MEC 服务器之间利用有线链路进行通信，其数据往返传输时间很短。

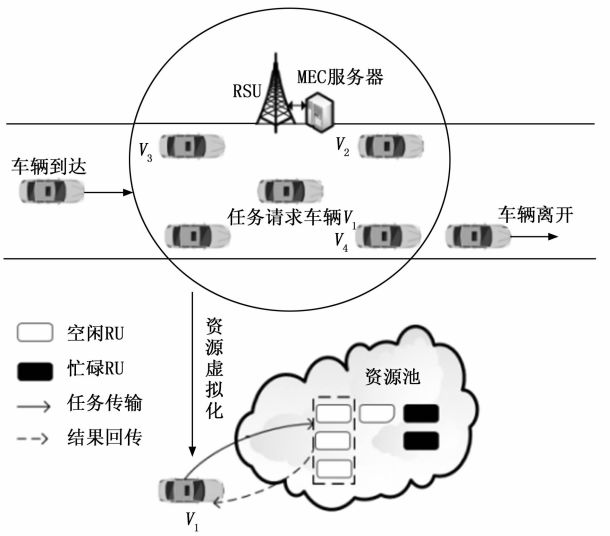

图 1　VEC 系统架构图

在充分考虑到系统可用资源的前提下，通过网络功能虚拟化（NFV，network functions virtualization）技术[18]把 VEC 系统中的闲置车辆以及进入系统的移动车辆虚拟整合成若干个资源单元（RU，resource unit），供任务请求时派遣。进行性能分析时，假定系统处于饱和容量，即车辆都有计算任务需要处理，每辆车都具有相同的计算能力，车辆到达和离开系统的速率分别满足参数为 λf 和 μf 的泊松分布。由于移动车辆不停地更换位置，随时有可能出入系统，因此系统资源数量也相应发生着变动。

1.2　通信模型

针对时延敏感型业务需求，可以将任务数据进行拆包分组，通过以 IEEE 802.11p 协议为基础的车载短程无线通信技术传输至 VEC 系统，进而制定卸载决策。数据传输采取增强型分布式信道访问机制[19]，所有节点使用二进制指数退避算法来竞争信道。

退避算法是以冲突窗口大小为基准的，每个节点有一个退避计数器。具体而言，当有任务数据需要传输时，它将从［0，$W_{min}-1$］中随机选择一个数值初始化计数器，其中 W_{min}是最小竞争窗口。退避计数器的值在每个等待时隙过后相应减少 1，一旦数值变为 0 就发送分组。如果在短帧间隔（SIFS，short interframe space）后接收到确认消息，则传输成功；否则视为传输失败并且需要重传分组。在重传前，竞争窗口将加倍增长，即 $W = 2W_{min}$。接着重新开始下一轮退避过程，此时的计数器数值被初始化并赋值在［0，$2W_{min}-1$］，如此循环。当重传达到最大退避次数时，最大冲突窗口将保持最大值，即最大允许加倍 m 次时竞争窗口大小最终将保持 $2^m W_{min}$。重复所述方法，直到分组被成功传输。成功发送后，竞争窗口重置为 W_{min}，并在分布式帧间间隔（DIFS，DCF interframe space）后开启新的退避过程。

以图 1 系统为例，移动车辆 V_1 发出任务卸载请求，系统根据现有 RU 数目做出相应判断，最终将虚拟化后的 V_2、V_3、V_4 作为执行单元进行分组传输，具体过程见图 2。

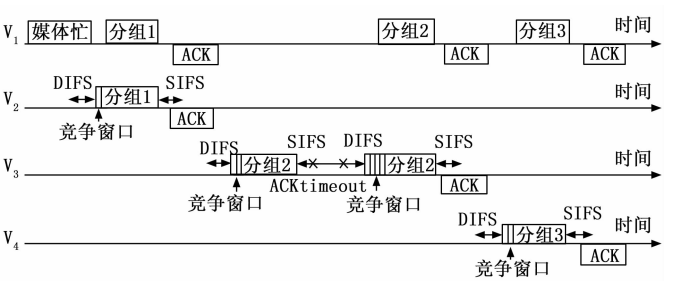

图 2　分组传输过程

1.3　时延模型

当计算任务由 VEC 系统执行卸载时，一般包括任务上传、执行计算和结果反馈 3 个阶段。在任务上传阶段中，车辆首先把任务上传给当前关联的路侧单元，该过程的传输时延取决于信道的数据传输质量。在计算服务阶段，计算时延由系统决定协作卸载的车辆数目以及服务率求得。考虑到最终计算结果的数据量较小，因此结果反馈所产生的回传时延忽略不计。

1.3.1　传输时延

任务数据在传输时，VEC 系统会根据当前的可用 RU 数量对原始数据报文进行分组处理，平均拆分成多个数据包，同时对每个分组增加额外的头部信息。分 i 组后任务传输时延的总计算公式如式（1）所示：

$$D_t(i) = i \cdot E[D_i] = i \cdot T_{slot}(i) \cdot E[N] \qquad (1)$$

其中：$E[D_i]$ 表示在退避时隙中传输分组给一个 RU 的平均时延；$T_{slot}(i)$ 为接入信道的平均时延；$E[N]$ 为成功传输分组所需的竞争窗口均值。

时间计算：退避算法中，时间被细化成各个时隙，节点只能在时隙开始时刻传输数据。由于所有车辆节点都共享信道，从竞争信道开始到成功把数据包送到目的节点的整个过程中，会有一定概率出现信道空闲、碰撞重传的现象，因此可以由式（2）表示：

$$T_{slot}(i)=P_{idle}\cdot slottime+P_c\cdot T_c(i)+P_s\cdot T_s(i) \tag{2}$$

其中：P_{idle}、P_c、P_s 分别为单位时间内信道空闲、发生碰撞和数据包发送成功的概率，$slottime$ 为信道空闲时间、$T_c(i)$ 和 $T_s(i)$ 分别为发生碰撞和成功传输的时间均值。在基本接入模式[19]下，$T_c(i)$ 和 $T_s(i)$ 可由表达式（3）和（4）给出：

$$T_c(i)=H+E[P]/i+SIFS+\delta+ACKtimeout \tag{3}$$

$$T_s(i)=H+E[P]/i+SIFS+\delta+ACK+\delta+DIFS \tag{4}$$

其中：H 为数据包的包头长度；$E[P]$ 为原始任务数据的平均长度，在均分成 i 组后传输数据包的长度为 $E[P]/i$；$SIFS$、$DIFS$ 分别为协议规定的短帧间间隔和分布协调功能帧间间隔，ACK 和 $ACKtimeout$ 分别为确认帧和超时帧的传输时间，δ 为电磁波的传播延迟。

概率计算：对于整个通信网络，任何节点都能侦听信道和传输数据包，因此每个节点都以等可能的概率来占用信道。定义 τ 为单位时间内节点传输数据的概率，在某一个时隙中只允许一个节点传输，而信道处于空闲状态时就没有任何数据传输。因此，在 M 个节点争用信道过程中，P_{idle}、P_c、P_s 可以由下式计算得到：

$$P_{idle}=(1-\tau)^M \tag{5}$$

$$P_s=M\tau(1-\tau)^{M-1} \tag{6}$$

$$P_c=1-P_{idle}-P_s \tag{7}$$

经典 BEB 算法在数据传送发生碰撞后，竞争窗口将加倍，而传输成功后恢复成最小竞争窗口，这种方法存在两处不合理的地方[20]。首先，信道竞争中不停失败的节点，由于退避计数器数值的反复增大，在抢占信道使用权时一直处于劣势状态，公平性难以得到保证。同时，多个节点传送数据时，一次传输成功的节点会误以为信道比较空闲，而多次发生碰撞而导致退避计数器数值不断增大，此类节点则认定信道一直处于忙碌状态，由此信道的真实状况难以反映出来。

据上所述，有必要对 BEB 算法做出适当的改进，减少不必要的信道碰撞。为了优化网络的整体性能，可以根据传输数据的节点数来适当地调整最小竞争窗口，使得竞争窗口的数值能与网络负载数相匹配，碰撞概率最小化。具体推导公式可以参照文献［20］，最终的最小竞争窗口以及传输概率如表达式（8）（9）所示：

$$W_{min}=\left[\frac{3M^2+7M+4+\sqrt{(M+1)^2(3M+4)^2-16(M-1)^3}}{8(M-1)}\right] \tag{8}$$

$$\tau=\frac{2}{W_{min}+1} \tag{9}$$

由于每次开启退避阶段，竞争窗口将会加倍，而达到最大退避阶数后保持最大值。因此平均时隙数 $E[N]$ 的计算由两部分组成：

$$E[N]=E[N_1]+E[N_2] \tag{10}$$

其中：$E[N_1]$ 和 $E[N_2]$ 分别为重传次数不大于和大于 m 时的平均时隙数。

定义 p 为任务传输过程中的碰撞概率，即单位时间总有至少一个节点在发送数据。

$$p=1-(1-\tau)^{M-1} \tag{11}$$

假定成功发送数据包前需要重传 h 次，即重复传输共 $h+1$ 次，所以传输成功率为 $p^h(1-p)$。在第 s 次重传时，争用窗口为 W_s。重传过程中，随机选择一个退避计数器值的概率为 $1/W_s$，每隔 1 个时隙数值减 1，直到减为 0，故第 k 次重传的平均时隙数为 $\sum_{k=1}^{W_s}\frac{k}{W_s}=\frac{W_s+1}{2}$。因此，$E[N_1]$ 和 $E[N_2]$ 可由式（12）和式（13）分别给出：

$$E[N_1]=\sum_{h=0}^{m}p^h(1-p)\sum_{s=0}^{h}\frac{W_s+1}{2}=\frac{1-(m+2)p^{m+1}+(m+1)p^{m+2}}{2(1-p)}+\frac{(1-p)[1-(2p)^{m+1}]W_{min}}{1-2p}-\frac{(1-p^{m+1})W_{min}}{2} \tag{12}$$

$$E[N_2]=\sum_{h=m+1}^{+\infty}p^h(1-p)\left[\sum_{l=0}^{m-1}\frac{W_l+1}{2}+\sum_{l=m}^{h}\frac{W_m+1}{2}\right]=\frac{p^{m+1}}{2}\left[\frac{(2-p)(2^mW_{min}+1)}{1-p}+m+1+(2^{m+1}-1)W_{min}\right] \tag{13}$$

1.3.2 计算时延

计算时延 $D_p(i)$ 是处理接收到的数据包所消耗的时间，取决于 RU 的计算能力。系统中每个 RU 假定都具有相同的服务率 μ_t，考虑到任务数据是经过拆包分组后传输至多个 RU，因此 i 个 RU 协作处理的计算时延可以表示为：

$$D_p(i)=\frac{1}{i\cdot\mu_t} \tag{14}$$

2 基于改进 SMDP 的策略求解

系统根据当前 RU 的工作状态制定合适的分组策略来卸载任务，这是一个序列决策问题。用传统的数学方法求解不太适合，而根据智能体对环境感知来做出最佳安排，更加贴合研究场景。由于 VEC 系统的下一时刻决策仅与当前决策和网络资源状况密切联系，而与历史时刻的决策记录无关。因此，可以将其抽象为智能体，并基于 SMDP 来制定任务卸载和资源分配方案，达到收益最大化。

SMDP 是一个五元组$<S, A, P, R, \gamma>$，处理此类问题需要分析系统的状态空间、动作空间、奖励函数和转

移概率。在所述 VEC 系统中，将通信网络内的 RU 占用情况定义为系统的状态空间；将特定事件发生时系统所采取的卸载行为定义为动作空间；根据系统制定的卸载决策所产生的收益和代价定义为奖励函数；通过不同动作下的状态得出系统内的状态转移概率。

2.1　状态空间 S

移动车辆随时可能到达或离开系统，被分配的 RU 由于服务进程的更迭也在不停地释放和重新消耗，因此系统中可能发生任务处理、资源释放、车辆到达和车辆离开 4 种不同的离散事件。用 e 表示系统特定事件的发生，则：

$$e=[E,D_1,\cdots,D_N,F_{+1},F_{-1}] \tag{15}$$

其中：E 为任务到达；N 为系统最多可分配给任务卸载的 RU 数目；D_i 表示被 i 个 RU 成功处理的任务完成并释放了其所用 RU；F_{+1}表示行驶车辆的到达；F_{-1}表示行驶车辆的离开。由此，系统的状态空间可以表示如下：

$$S=[s\mid s=(M,n_1,n_2,\cdots,n_N,e)] \tag{16}$$

其中：M 为当前系统中计算单元的总数目，且 $M\leqslant K$，K 为系统中最大车辆数目；n_i 表示正在被 i 个计算单元处理的任务数量；e 表示一种特定事件。系统依据特定事件做出合理分配后，剩余资源量为 $M-\sum_{i=1}^{N}i\cdot n_i$。

2.2　动作空间 A

由系统状态空间可知，在特定事件到达时系统可以采取的离散动作表示为：

$$A=\begin{cases}\{-1\}, & e\in\{D_1,\cdots,D_N,F_{+1},F_{-1}\}\\ \{0,1,2,\cdots,N\}, & e=E\end{cases} \tag{17}$$

其中：在集合 A 中，-1 表示没有采取任何动作，0 表示当计算资源缺乏时，系统拒绝卸载并丢弃该任务，$\{E$，D_1，…，D_i，…，D_N，F_{+1}，$F_{-1}\}$ 表示事件集合，N 表示任务最多可以被 N 个 RU 处理。

2.3　状态转移概率 $P(s'\mid s,a)$

在任务卸载过程中，下一个状态出现的概率仅依赖于前一个状态以及前状态所采取的动作。因此将状态转移概率定义为：在当前状态 s 和采取动作 a 时，转变为状态 s'的概率。设 $\tau(s,a)$ 表示在采取动作 a 时从当前状态 s 转移到下一个状态 s'的服务时间，则 $\sigma(s,a)$ 定义表示状态转移过程中网络的平均事件发生率。

$$\sigma(s,a)=\tau(s,a)^{-1}=$$
$$\begin{cases}M\lambda_t(i)+\lambda_f+\mu_f+(\sum_{j=1}^{N}j\cdot n_j+i)\mu_t, & e=A,a=i(1\leqslant i\leqslant N)\\ M\bar{\lambda}_t+\lambda_f+\mu_f+(\sum_{j=1}^{N}j\cdot n_j-i)\mu_t, & e=D_i,a=-1\\ (M+1)\bar{\lambda}_t+\lambda_f+\mu_f+\sum_{j=1}^{N}j\cdot n_j\cdot\mu_t, & e=F_{+1},a=-1\\ (M-1)\bar{\lambda}_t+\lambda_f+\mu_f+\sum_{j=1}^{N}j\cdot n_j\cdot\mu_t, & e=F_{-1},a=-1\end{cases} \tag{18}$$

其中：$M\lambda_t(i)$ 表示计算任务的到达率；$\sum_{j=1}^{N}j\cdot n_j\cdot\mu_t$，表示计算任务的离去率。由于 SMDP 模型没有记录历史状态和动作，当新任务到达时系统可分配 RU 数量是不确定的，因此用平均任务到达率来衡量，如式（19）所示：

$$\bar{\lambda}_t=\sum_{i=1}^{N}p_i\lambda_t(i) \tag{19}$$

其中：$p_i=n_i/\sum_{i=j}^{N}n_j$。

考虑不同的离散事件，在状态 s 下采取动作 a 转变到状态 s'的转移概率 P（$s'\mid s$，a），计算公式分别如下所示：

1）$s=(M,n_1,n_2,\cdots,n_N,A)$，$a=i$；

$$P(s'\mid s,a)=$$
$$\begin{cases}\frac{M\lambda_t(i)}{\sigma(s,a)},s'=(M,n_1,n_2,\cdots,n_i+1,\cdots,n_N,A)\\ \frac{i(n_i+1)\mu_t}{\sigma(s,a)},s'=(M,n_1,n_2,\cdots,n_i+1,\cdots,n_N,D_i)\\ \frac{jn_j\mu_t}{\sigma(s,a)},s'=(M,n_1,n_2,\cdots,n_i+1,\cdots,n_N,D_j),i\neq j\\ \frac{\lambda f}{\sigma(s,a)},s'=(M,n_1,n_2,\cdots,n_i+1,\cdots,n_N,F_{+1})\\ \frac{\mu f}{\sigma(s,a)},s'=(M,n_1,n_2,\cdots,n_i+1,\cdots,n_N,F_{-1})\end{cases} \tag{20}$$

2）$s=(M,n_1,n_2,\cdots,n_N,D_i)$，$a=-1$；

$$P(s'\mid s,a)=$$
$$\begin{cases}\frac{M\bar{\lambda}_t}{\sigma(s,a)},s'=(M,n_1,n_2,\cdots,n_i-1,\cdots,n_N,A)\\ \frac{i(n_i-1)\mu_t}{\sigma(s,a)},s'=(M,n_1,n_2,\cdots,n_i-1,\cdots,n_N,D_i)\\ \frac{jn_j\mu_t}{\sigma(s,a)},s'=(M,n_1,n_2,\cdots,n_i-1,\cdots,n_N,D_j),i\neq j\\ \frac{\lambda_f}{\sigma(s,a)},s'=(M,n_1,n_2,\cdots,n_i-1,\cdots,n_N,F_{+1})\\ \frac{\mu_f}{\sigma(s,a)},s'=(M,n_1,n_2,\cdots,n_i-1,\cdots,n_N,F_{-1})\end{cases} \tag{21}$$

3）$s=(M,n_1,n_2,\cdots,n_N,F_{+1})$，$a=-1$；

$$P(s'\mid s,a)=$$
$$\begin{cases}\frac{(M+1)\bar{\lambda}_t}{\sigma(s,a)},s'=(M+1,n_1,n_2,\cdots,n_N,A)\\ \frac{in_i\mu_t}{\sigma(s,a)},s'=(M+1,n_1,n_2\cdots,n_N,D_i)\\ \frac{\lambda_f}{\sigma(s,a)},s'=(M+1,n_1,n_2\cdots,n_N,F_{+1})\\ \frac{\mu_f}{\sigma(s,a)},s'=(M+1,n_1,n_2\cdots,n_N,F_{-1})\end{cases} \tag{22}$$

4）$s=(M,n_1,n_2,\cdots,n_N,F_{-1})$，$a=-1$；

$$P(s'\mid s,a)=\begin{cases}\frac{(M-1)\bar{\lambda}_t}{\sigma(s,a)},s'=(M-1,n_1,n_2,\cdots,n_N,A)\\ \frac{in_i\mu_t}{\sigma(s,a)},s'=(M-1,n_1,n_2,\cdots,n_N,D_i)\\ \frac{\lambda_f}{\sigma(s,a)},s'=(M-1,n_1,n_2,\cdots,n_N,F_{+1})\\ \frac{\mu_f}{\sigma(s,a)},s'=(M-1,n_1,n_2,\cdots,n_N,F_{-1})\end{cases} \tag{23}$$

2.4 奖励函数 $R(s,a)$

在状态 s 采取动作 a 的系统收益可以表示为立即收益和系统成本的差值，表示如下：

$$R(s,a) = I(s,a) - C(s,a) \tag{24}$$

立即收益 $I(s,a)$ 包括分组传输的节省时延收益、资源不足时丢包处理的惩罚以及执行计算的移动车辆未能及时完成就提前离开系统的惩罚。其计算公式如下：

$$I(s,a) = \begin{cases} \beta[T - D_t(i) - D_p(i)], & a = i, e = A(i > 0) \\ -\xi, & a = 0, e = A \\ 0, & a = -1, e \in \{D_1, D_2, \cdots, D_N, F_{+1}\} \\ 0, & a = -1, e = F-1, \sum_{j=1}^{N} j \cdot n_j < M \\ -\eta, & a = -1, e = F-1, \sum_{j=1}^{N} j \cdot n_j = M \end{cases} \tag{25}$$

其中：β 是节省时延的单位价格；T 为计算任务在本地处理的最大时延容忍；ξ 表示资源匮乏时系统丢包的惩罚；η 表示正在提供计算服务的移动车辆离开系统的惩罚。

$C(s,a)$ 表示在一段连续时间内，从状态 s 转变到状态 s' 前处理服务请求所需的代价。当前状态下，VEC 系统从资源池中获得相应 RU 信息，选择合适动作执行，决策之间的持续时间服从指数分布。由于系统状态在状态转变期间之间不发生改变[21]，为满足实际情况，考虑连续时间的折扣因子 $\alpha \in [0,1]$，由式(18) 可知 $\sigma(s,a) = \tau(s,a)^{-1}$，则 $C(s,a)$ 计算公式如下：

$$C(s,a) = b(s,a)E_s^a\left(\int_o^\tau e^{-\alpha t}\mathrm{d}t\right) = b(s,a)E_s^a\left(\frac{1-e^{-\alpha\tau}}{\alpha}\right) = \frac{b(s,a)}{\alpha+\sigma(s,a)} \tag{26}$$

其中：$b(s,a)$ 表示已分配的资源量。

2.5 改进的最优策略求解

VEC 系统中的 RU 分配方案就是要找到最优策略使得长期收益达到理想最大值。策略定义为：π：$S \to a$，即 $\pi(S) = a$。可描述为：当系统处于特定状态下，采取某种动作得到一定的收益值。因此，当事件发生时，当前状态下网络获得的最大收益值如式 (27) 所示：

$$V_a^{\pi^*}(s) = \max_{\pi} E_s^{\pi}\left[\sum_{m=1}^{\infty} e^{-\alpha\sigma m} R(s_m, a_m) \mid s_0 = s\right] \tag{27}$$

式中，s_0 为系统的初始状态；s_m 和 a_m 分别为第 m 个状态和对应的行为，σ_m 表示在状态 s_m 下单位时间内的平均事件发生率。考虑到有限离散的状态空间和动作空间，而在一段连续时间内，系统中的总车辆数目存在一定的波动，因此采取值迭代的方式来解决此类 SMDP 问题。

VEC 系统需要做出一系列决策，以获得尽可能高的累计收益。然而，由于随机事件的发生，未来奖励往往比当前奖励更加不明确，因此需要谨慎地权衡眼前的和未来的奖励。经典 SMDP 算法[17]将两者赋予同等权重，难以挖掘出将来的潜在收益。例如，一个状态可能总是得到一个低的直接奖赏但仍有较高的值，因为其后续状态有极高概率能获得更高的奖赏，或者反过来也是有可能的。

对此，提出基于余弦函数对两种收益采取非线性加权的方法，动态调整两者之间的权重比例，制定规则如式 (28) 所示：

$$\omega = r\cos\left(\frac{\pi}{2}\sqrt{1 - \frac{k}{iter\_max}}\right) \tag{28}$$

式中，r 为最大权重系数；k 为迭代次数，且 k 不超过最大次数 $iter\_max$。

根据贝尔曼方程[21]，考虑状态转移概率和折扣因子，改进后的最优策略见式 (29)：

$$V_{k+1}(s) = \max_{a \in A_c}[\omega R(s,a) + (1-\omega)\gamma \sum_{s' \in S} P(s' \mid s,a) V_k(s')] \tag{29}$$

此外，引入了参数 y 进一步均衡连续时间 SMDP，且 $y = K \cdot \lambda_t + K \cdot N \cdot \mu_t + \lambda_f + \mu_f$。$\hat{P}(s' \mid s,a)$、$\hat{R}(s,a)$、$\hat{\gamma}$ 分别表示归一化后的转移概率、系统奖励和折扣因子，表达式如下所示：

$$\hat{R}(s,a) = R(s,a)\frac{\alpha + \sigma(s,a)}{\alpha + y} \tag{30}$$

$$\hat{\gamma} = \frac{y}{y+\alpha} \tag{31}$$

$$\hat{P}(s' \mid s,a) = \begin{cases} 1 - \dfrac{[1 - P(s' \mid s,a)]\sigma(s,a)}{y}, & s' = s \\ \dfrac{P(s' \mid s,a)\sigma(s,a)}{y}, & s' \neq s \end{cases} \tag{32}$$

迭代过程的收敛率为 $\varepsilon = \dfrac{\varepsilon(1-\hat{\gamma})}{2\hat{\gamma}}$。由此，式 (29) 可重新定义为：

$$\hat{V}k+1(s) = \max_{a \in A_c}[\omega\hat{R}(s,a) + (1-\omega)\hat{\gamma}\sum_{s' \in S}\hat{P}(s' \mid s,a)\hat{V}k(s')] \tag{33}$$

最后，所得最优策略表达式为：

$$\pi^*(s) = \operatorname*{argmax}_{a \in A}[\omega\hat{R}(s,a) + (1-\omega)\hat{\gamma}\sum_{s' \in S}\hat{P}(s' \mid s,a)\hat{V}k(s')] \tag{34}$$

改进最优策略后的 SMDP 求解步骤如下：

1) 对于状态 s，初始化 $V(s) = 0$，设置 $k = 1, r = 1, \varepsilon = 10$；

2) 计算 ω，利用前一个策略价值函数 $V_k(s)$，根据式(33) 计算当前的价值函数 $V_{k+1}(s)$；

3) 如果 $\|\hat{V}_{k+1} - \hat{V}_k\| < \dfrac{\varepsilon(1-\hat{\gamma})}{2\hat{\gamma}}$，转入 4)，否则 $k = k+1$，转入 2)；

4) 对任意时刻 $s \in S$ 下，最优策略为 $\pi^*(s) = \operatorname*{argmax}_{a \in A}[\omega\hat{R}(s,a) + (1-\omega)\hat{\gamma}\sum_{s' \in S}\hat{P}(s' \mid s,a)\hat{V}k(s')]$。

3 实验仿真验证

3.1 参数设置

本节利用 Matlab 2018a 对所提卸载算法进行仿真实验，

结合实际场景需求，假定任务数据最多允许 3 个资源单元协作计算，其相应的分组传输策略分别记为动作 1、2、3，即整包发送、两组和三组传输。实验参数依据文献［17］进行设置，部分变量做出适应性调整。

通信参数中，最大退避阶数 m 设置为 1；$SIFS$ 和 $DIFS$ 分别为 10 μs 和 50 μs；包头长度 H 为 229 μs；ACK、$ACKtimeout$ 分别为 304 μs 和 356 μs；时隙长度 $slottime$ 为 20 μs；电磁波传输时延 δ 为 2 μs；数据平均长度 $E[P]$ 为 1 920 bytes；本地计算时延 T 为 100 ms。SMDP 模型参数如表 1 所示。

表 1　改进最优策略后的 SMDP 模型参数

参数	数值	参数	数值
车辆到达率 λ_f	10	车辆离去率 μ_f	10
最大迭代次数 $iter\_max$	100	最大车辆数目 K	[7,21]
节省时延奖励单价 β	5	服务率 μ_t	[20,50]
采取丢包策略惩罚 ξ	10	收敛判别值 ε	10
服务车辆提前离开惩罚 η	18	折扣因子 α	0.1

3.2　性能评估

图 3 描述了 3 种不同服务率下数据整包传输的时延变化。随着任务量的不断增加，总体时延呈现上升趋势。一方面是因为任务处理量的增加，另一方面是信道资源的竞争。但在系统最高 21 辆车时，卸载时延也尚未超过本地的最大计算时延 100 ms。同时，也清楚地看到 3 种服务率下，50 tasks/s 服务率下的卸载时延明显低于其他两组。服务率的提升意味着单位时间内占用计算单元的任务会更快地将其释放出来，因此其他业务需求能及时得到满足，系统的资源利用率有了一定程度的提升。

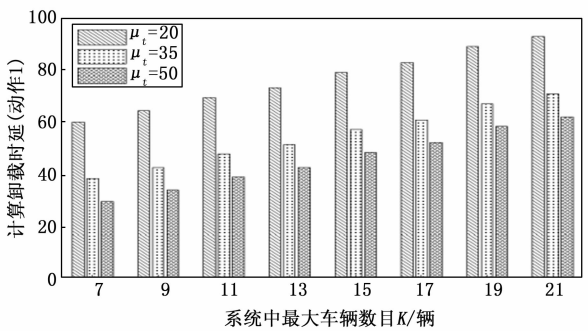

图 3　整包传输的卸载时延

图 4 和图 5 集中反映的是拆包发送数据时两组和 3 组不同策略下的时延变化。相比于整包传输，两种方案的卸载时延仍在不断增加，但相比都有一定程度地降低。由于拆分后的数据包会一段一段地断续占用通信资源，单个分组长度小于整个报文长度，因而数据传输效率高。此外，3 组卸载时延比起两组稍微高些。究其原因，原始数据在拆分重组时会加上报头信息，添加额外数据，使得传送的信息量变大。

图 6 可以看出分 3 组传输的数据包到达率明显低于其他

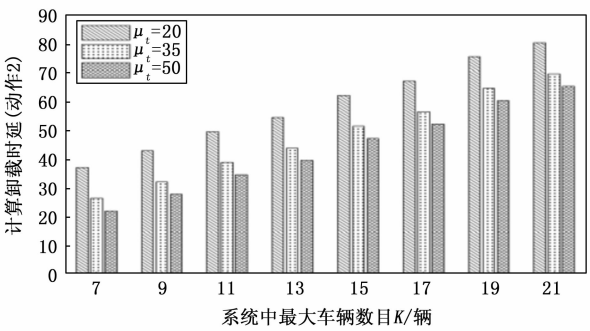

图 4　分两组传输的卸载时延

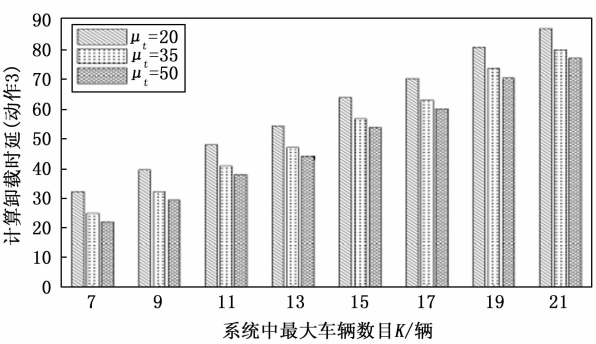

图 5　分 3 组传输的卸载时延

两种方式，因为当多组传送时，分组本身的数据量较小，计算资源的占用时间也相应减少。被使用的资源随着服务的终止被释放到资源池中，新的分组到达时能得到实时有效的处理，因此能更好地适应资源动态变化。

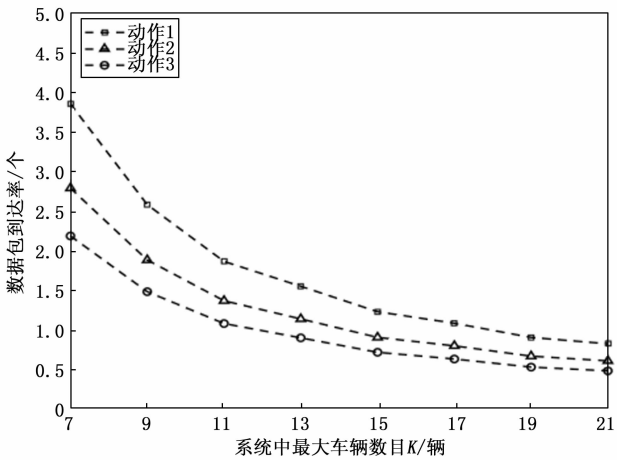

图 6　数据包到达率与最大车辆数目的关系

图 7 描述的是两种服务率下的改进 SMDP 和经典 SMDP 算法的长期收益状况（对比实验时，经典 SMDP 算法的立即收益和未来期望收益的比例权重均设置为 0.5）。改进的 SMDP 算法能大幅提升系统的长期收益，这是因为在决策过程中，通过不断调整立即收益和未来期望收益的比重，系统能以一定概率选择使将来收益更大化的动作；

立即收益不是很好的动作策略，由于潜在收益相对较高而被挖掘出来，在决策后使计算资源更优地服务于业务需求。

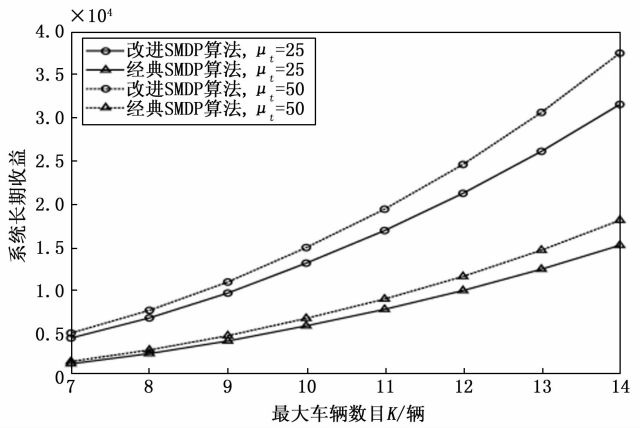

图 7 改进与经典 SMDP 算法对长期收益的影响

4 结束语

在 VEC 系统中，利用 NFV 技术将闲置和移动车辆虚拟化成资源池。将时延敏感类任务数据进行拆包重组，由系统制定分组方案。利用车辆节点数调整最小竞争窗口，降低信道碰撞概率。结合分配资源的时序决策特点，基于改进的 SMDP 算法制定不同状态下的动作决策，在最优策略中提出带有余弦项的非线性权重因子，对立即收益和未来期望收益动态加权，使收益最大化。实验结果表明所提策略能有效提高 VEC 系统的服务能力。在高速公路上，如何在不同的运行时速下确保任务调度的及时性，从而提高交通效率，是未来研究的主要内容。

参考文献：

[1] TEZERGIL B, ONUR E. Wireless backhaul in 5G and beyond: Issues, challenges and opportunities [J]. IEEE Communications Surveys & Tutorials, 2022, 24 (4): 2579 - 2632.

[2] LAI C Z, LU R X, DONG Z, et al. Security and privacy challenges in 5G-enabled vehicular networks [J]. IEEE Network, 2020, 34 (2): 37 - 45.

[3] SIRIWARDHANA Y, PORAMBAGE P, LIYANAGE M. A survey on mobile augmented reality with 5G mobile edge computing: architectures, applications, and technical aspects [J]. IEEE Communications Surveys & Tutorials, 2021, 23 (2): 1160 - 1192.

[4] MENEGUETTE R, DEGRANDE R, UEYAMA J, et al. Vehicular edge computing: architecture, resource management, security, and challenges [J]. ACM Computing Surveys, 2021, 55: 1 - 46.

[5] TANG C G, ZHU C S, ZHANG N, et al. SDN-assisted mobile edge computing for collaborative computation offloading in industrial internet of things [J]. IEEE Internet of Things Journal, 2022, 9 (23): 24253 - 24263.

[6] LIU Y Q, PENG M G, SHOU G C, et al. Toward edge intelligence: multiaccess edge computing for 5G and internet of things [J]. IEEE Internet of Things Journal, 2020, 7 (8): 6722 - 6747.

[7] AVASALCAI C, TSIGKANOS C, DUSTDAR S. Resource management for latency-sensitive IoT applications with satisfiability [J]. IEEE Transactions on Services Computing, 2022, 15 (5): 2982 - 2993.

[8] 王 练，闫润搏，徐 静. 车载边缘计算中多任务部分卸载方案研究 [J]. 电子与信息学报，2023，45 (3)：1094 - 1101.

[9] 孙麒惠，朱金奇，花季伟，等. 停车辅助的车辆边缘计算中的双重任务卸载设计 [J]. 计算机工程与设计，2023，44 (2)：364 - 371.

[10] WANG H S, LI X, JI H, et al. Federated offloading scheme to minimize latency in MEC-enabled vehicular networks [C] // 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 2018: 1 - 6.

[11] SUN Y X, GUO X Y, SONG J H, et al. Adaptive learning-based task offloading for vehicular edge computing systems [J]. IEEE Transactions on Vehicular Technology, 2019, 68 (4): 3061 - 3074.

[12] ZHU C, TAO J, PASTOR G, et al. Folo: Latency and quality optimized task allocation in vehicular fog computing [J]. IEEE Internet of Things Journal, 2019, 6 (3): 4150 - 4161.

[13] ZENG F, CHEN Q, MENG L, et al. Volunteer assisted collaborative offloading and resource allocation in vehicular edge computing [J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22 (6): 3247 - 3257.

[14] ZHOU H, WU T, ZHANG H J, et al. Incentive-driven deep reinforcement learning for content caching and D2D offloading [J]. IEEE Journal on Selected Areas in Communications, 2021, 39 (8): 2445 - 2460.

[15] CHEN C, CHEN L L, LIU L, et al. Delay-optimized V2V-based computation offloading in urban vehicular edge computing and networks [J]. IEEE Access, 2020, 8: 18863 - 18873.

[16] 范艳芳，袁 爽，蔡 英，等. 车载边缘计算中基于深度强化学习的协同计算卸载方案 [J]. 计算机科学，2021，48 (5)：270 - 276.

[17] WU Q, LIU H X, FAN P Y, et al. Delay-sensitive task offloading in the 802.11p-based vehicular fog computing systems [J]. IEEE Internet of Things Journal, 2020, 7 (1): 773 - 785.

[18] 孟 芸，牛永豪，刘鑫一，等. 基于可靠性保障的车联网中服务功能链映射算法 [J]. 南京大学学报 (自然科学版)，2023，59 (1)：173 - 182.

[19] BIANCHI G. Performance analysis of the IEEE 802.11 distributed coordination function [J]. IEEE Journal on Selected Areas in Communications, 2000, 18 (3): 535 - 547.

[20] 段志荣. IEEE 802.11 DCF 协议退避算法的研究 [D]. 秦皇岛：燕山大学，2018.

[21] ZHANG K, MENG H L, CHATZIMISIOS P, et al. An SMDP-based resource allocation in vehicular cloud computing systems [J]. IEEE Transactions on Industrial Electronics, 2015, 62 (12): 7920 - 7928.