面向高速行驶车辆的在线任务卸载决策算法

丁爽, 曹沐雨, 何欣

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.221200069

面向高速行驶车辆的在线任务卸载决策算法

丁 爽[1,2] 曹沐雨[1] 何 欣[1,2]
1 河南大学软件学院 河南 开封 475004
2 河南智能网络理论与关键技术国际联合实验室 河南 开封 475004
(ds@henu.edu.cn)

摘 要 车载边缘计算中的任务卸载决策主要解决任务何时卸载,以及卸载至哪里执行的问题。车辆的高速行驶会造成卸载接入设备频繁变化,卸载通信链路随时可能中断,这要求车辆一旦获得卸载机会,就必须立即做出卸载决策。现有的卸载决策研究专注于如何最大化任务卸载执行增益,未充分考虑卸载决策时效对卸载策略的影响,导致提出的卸载决策方法的时间复杂度和空间复杂度高,无法用于高速行驶车辆的在线任务卸载决策。为解决上述问题,首先综合考虑卸载决策时效和卸载增益因素的影响,建立高速行驶车辆的任务卸载决策模型,并将其转化为类秘书问题。然后,提出了一种基于加权二部图匹配的在线车载任务卸载决策算法 OODA,以协助车辆在依次经过多个异构的边缘服务器时,做出实时的任务卸载决策,并最大化整体卸载执行增益。最后,理论分析 OODA 算法的竞争比,并采用仿真实验验证该算法的可行性和有效性。

关键词:车载边缘计算;任务卸载;秘书问题;加权二部图匹配
中图分类号 TN929.5

Online Task Offloading Decision Algorithm for High-speed Vehicles

DING Shuang[1,2],CAO Muyu[1] and HE Xin[1,2]
1 School of Software,Henan University,Kaifeng,Henan 475004,China
2 Henan International Joint Laboratory of Intelligent Network Theory and Key Technology,Kaifeng,Henan 475004,China

Abstract When and where to offloading tasks are the main problems to be solved in the task offloading decision in vehicular edge computing. High speed driving of the vehicle causes frequent changes of offloading access devices,and the offloading communication between the vehicle and the offloading access device may break at any time. This requires that the offloading decision should be made immediately once the vehicle obtains an offloading opportunity. The existing offloading decision research focuses on how to maximize the offloading gain,without fully considering the impact of the timeliness of offloading decision on offloading strategy. As a result,the proposed offloading decision methods have high time and space complexity,and cannot be used for online task offloading decisions of high-speed vehicles. In order to solve the above problems,this paper first comprehensively considers the influence of offloading decision-making timeliness and offloading gain factors,establishes a task offloading decision model for high-speed vehicles,and transforms it into a variation of the secretary problem. Then,an online vehicle task offloading decision algorithm OODA based on weighted bipartite graph matching is proposed to assist the vehicle to make real-time task offloading decisions when passing through multiple heterogeneous edge servers sequentially,and maximize the overall offloading gain. Finally,theoretical analysis shows that the competitive ratio of OODA algorithm is analyzed theoretically. Extensive simulation results show that OODA is feasible and effective.

Keywords Vehicle edge computing,Task offloading,Secretaryproblem,Weighted bipartite graph matching

1 引言

随着智能交通系统的快速发展,车载系统中包含的计算密集型、时延敏感型任务呈爆炸式增长[1]。由于自身的计算和存储资源有限,车辆通常无法独立完成上述车载任务。据此,将车联网与移动边缘计算相结合的车载边缘计算[2]应运

到稿日期:2022-12-09 返修日期:2023-04-06
基金项目:中国博士后科学基金面上资助项目(2020M672217);2022 年度河南省重点研发与推广专项(科技攻关)(222102210133);2020 年度河南省重大科技专项(201300210400)
This work was supported by the General Support from China Postdoctoral Science Foundation(2020M672217),2022 Henan Province Key R&D and Promotion Project(Science and Technology Research)(222102210133) and Major Science and Technology Projects of Henan Province in 2020(201300210400).
通信作者:何欣(hexin@henu.edu.cn)

而生。其主要思想是：车辆会根据任务卸载决策方法，将无法独立执行的任务通过路侧单元、基站等卸载接入设备，选择性地卸载至边缘服务器执行，从而获得卸载执行增益（包括时延或能耗）等[3]。

卸载决策是车载边缘计算的关键技术，也是难点技术[4]，主要解决任务何时卸载以及卸载至哪里执行的问题。图 1 是高速行驶车辆的任务卸载决策场景示意图。与其他的车载任务卸载决策场景相比，高速行驶车辆更注重卸载决策的时效性，原因在于：车辆的高速行驶会造成卸载接入设备的频繁变化，并且负责任务卸载的通信链路随时可能中断[5]。上述情况均要求车辆一旦获得卸载机会，就必须立即做出任务卸载决策，否则有可能永远失去本次卸载机会。对于某些时效性强的车载任务，错失某次卸载机会就意味着执行失败。而当执行失败的任务达到一定量级时，会严重影响车载系统的正常运转。

现有的车载任务卸载决策研究大多以最大化任务卸载执行增益为目标[6-10]，所提卸载决策方法通常大型且复杂，决策的时间复杂度和空间复杂度均较高[11]，无法满足高速行驶车辆对卸载决策的时效性需求。因此，亟需设计一种新的在线车载任务卸载决策方法，协助车辆在获得卸载机会时，做出实时的任务卸载决策，并最大化任务卸载执行增益。

本文认为高速行驶车辆的在线任务卸载决策问题，本质上是对车载边缘计算系统中的边缘服务器资源的在线分配问题。其目标是，在获得实时的任务卸载策略（资源在线分配）的同时，最大化整体任务卸载执行增益（资源分配效益）。该问题的难点在于：在决策之初，车辆携带的车载任务（资源征用方）是已知的，而将经过的边缘服务器（资源方）是未知的。随着车辆的高速行驶，边缘服务器以随机顺序依次出现，并且与其相关的任务卸载可能性（资源分配可能性）以及卸载执行增益（分配效益）会随之变为已知。上述问题属于类秘书问题[12]，如果能将其转化为加权二部图匹配[13]问题，则能采用改进的加权二部图在线匹配算法进行有效求解。

本文的主要工作有：

1）对高速行驶车辆的任务卸载决策建模，并将其转化为类秘书问题；

2）提出了一种基于二部图匹配的在线任务卸载决策算法 OODA，以获得高速行驶车辆的实时任务卸载策略；

3）采用竞争比分析 OODA 算法的性能，并通过仿真实验验证该算法的可行性和有效性。

2　相关工作

研究者们以最大化任务的卸载执行增益为目标，提出了大量的卸载决策方法。Zhan 等[6]采用马尔可夫决策过程对高速行驶车辆的任务调度过程进行建模，并提出了一种将近端策略优化与卷积神经网络相结合的任务卸载调度算法，在确保时延和能耗权衡的同时，实现车辆长期成本的最小化。Wang 等[7]针对资源有限的车载任务卸载决策问题，综合考虑车辆与卸载接入点之间的距离，以及多个车辆对边缘服务器资源的竞争，提出了一种基于博弈模型的分布式任务卸载决策算法，采用多用户非合作博弈模型调整每个车辆的任务卸载概率，从而实现对 MEC 资源的充分利用，并最大化每个车辆的效用。Dai 等[8]研究了众包场景下由数据驱动的车载任务卸载问题，提出了一种基于异步 Q-learning 算法的卸载决策机制，用于最小化车辆的平均服务时间和平均服务成本。然而，上述方法未充分考虑卸载决策的时效性，造成决策的时间复杂度和空间复杂度较高。

在线车载任务卸载决策是目前的研究热点。Dai 等[9]针对移动车辆的实时任务卸载策略，综合考虑服务时延、服务器异构性和车辆移动性等因素，提出了一种基于强化学习的分布式实时服务调度机制，以最大化车辆的服务比率。Duan 等[10]关注由 UAV 辅助的多车辆协同卸载问题，提出了一种基于内点的迭代算法，根据用户偏好和系统需求调度通信和计算资源，以最大化系统的全局能源效率。Hu 等[14]研究了一个终端设备同时与多个边缘服务器保持通信时的任务卸载决策问题，提出了一种基于最优停止理论的轮询式卸载决策方法。该方法能够实现时间收益约束下的数据迁移平均能耗最小化，但不适用于高速移动设备的在线任务卸载决策。

加权二部图在线匹配问题属于类秘书问题，最早由 Korula 等[13]提出，主要用于解决资源的在线分配问题。其假定二部图的一侧或两侧顶点以随机顺序到达，当新顶点到达时，相应的边和边权重就会显现。其目标是求加权二部图的最佳匹配，即匹配边上的权值和最大。目前，加权二部图在线匹配已被广泛应用于在线组合拍卖和任务分配中。Kesselheim 等[15]针对在线组合拍卖问题，提出了一种适用于一侧顶点随机到达的在线加权匹配算法，该算法的竞争比可以达到 e。Reiffenhauser 等[16]在文献[15]提出的算法的基础上，增加了一个杜绝投标人虚假报价的 VCG 定价机制，使算法更适用于真实场景下的组合拍卖。Tong 等[17]将群智感知中的任务分配问题建模为两侧顶点均随机到达的在线加权匹配问题，并提出了一种具有两个决策阶段的 TGOA 算法，同时采用贪婪策略和匈牙利算法求解最优任务分配策略，有效提高了任务分配总效用。

3　问题建模

如图 1 所示，车辆 L 携带多个待执行任务沿道路高速行驶，途中会与多个异构的路侧单元和边缘服务器组合（RSU_MEC）相遇。当车辆 L 驶入新的 RSU_MEC 服务范围时，会将部分任务卸载至边缘服务器（MEC）执行。其卸载任务的目标是最大化整体任务卸载执行增益。

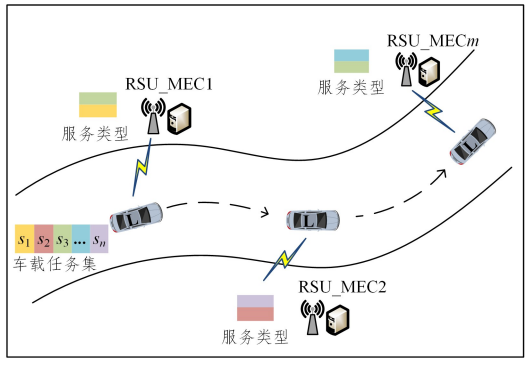


图 1　高速行驶车辆的任务卸载决策场景示意图（电子版为彩图）

Fig. 1　Task offloading decision scenario for high-speed vehicles

假设车辆 L 携带的待执行任务是异构的，即各任务的类型、执行需求不同，并且各 RSU_MEC 提供的卸载服务也不同，例如计算能力[8]和服务类型[18-19]等。那么，对于某个特定的车载任务，只有部分 RSU_MEC 能够为其提供卸载服务；通过不同 RSU_MEC 卸载时，所能获得的卸载执行增益也不同。例如，图 1 采用不同的填充色来标注任务类型。如图 1 所示，任务 s_1 属于橙色类型，那么在视图范围内，只有 RSU_MEC1 可以为其提供卸载服务。

首先，对车辆 L、任务和 RSU_MEC 建模。

1）车辆 L：令 $l=(f_l, e_l, g_l, p_l)$ 表示车辆 L。其中，f_l 是车辆的算力，采用 CPU 计算频率来量化；e_l 是车辆的单位计算能耗，采用 CPU 计算频率和代表芯片架构系数的乘积来量化[20]；g_l 是车辆的数据收发速率；p_l 是车辆的数据收发功率。

2）任务：令 $s=(d_s, b_s, st_s, et_s, c_s)$ 表示车辆 L 携带的待执行任务 s，其中 d_s 是任务的计算量，b_s 是卸载任务需要传输的数据量，st_s 和 et_s 分别是任务可进行卸载决策的开始时间和结束时间，即$[st_s, et_s]$是任务的卸载决策时间窗，c_s 是任务的类型。

3）RSU_MEC：令 $r=(f_r, smt_r, emt_r, C_r)$ 表示 RSU_MECr，其中，f_r 是 RSU_MEC 的算力，smt_r 和 emt_r 分别是车辆 L 驶入和驶出 RSU_MEC 服务范围的时间，即$[smt_s, emt_s]$是 RSU_MEC 可以为车辆 L 提供卸载服务的时间窗，C_r 是 RSU_MEC 能够服务的任务类型。

然后，给出任务卸载执行增益的计算方法。当任务 s 能够通过 RSU_MECr 卸载时，称卸载服务对(s, r)存在，其对应的卸载执行增益计算函数为：

$$u(s,r)=\theta\frac{T_s^l-T_s^r}{T_s^l}+(1-\theta)\frac{E_s^l-E_s^r}{E_s^l} \tag{1}$$

其中，$T_s^l=d_s/f_l$ 和 $E_s^l=d_s e_l$ 分别表示任务 s 由车辆 L 执行时，车辆 L 需要付出的时延和能耗，$T_s^r=b_s/g_l+d_s/f_r$ 和 $E_s^r=(b_s/g_l)p_l$ 分别表示当任务 s 由 RSU_MECr 卸载执行时，车辆 L 需要付出的时延和能耗。在此，本文假设任务计算结果的数据量很小，因此可忽略车辆在回传阶段的时延和能耗，只计算任务卸载和执行两个阶段。在式(1)中，$(T_s^l-T_s^r)/T_s^l$表示时延增益，$(E_s^l-E_s^r)/E_s^l$表示能耗增益，$\theta\in[0,1]$是时延增益和能耗增益之间的权衡因子。当 $\theta>0.5$ 时，表示车辆 L 更注重获取时延增益；否则更注重获取能耗增益。

综上所述，高速行驶车辆的在线卸载决策问题 P 可表述为：在某个时间段 T 内，车辆 L 携带的待执行任务集合为 $S=\{s_i \mid 1\leqslant i\leqslant n\}$，其中，$n$ 为车载任务的数量；车辆 L 将要经过的 *RSU_MEC* 集合为 $R=\{r_j \mid 1\leqslant j\leqslant m\}$，其中，$m$ 为 RSU_MEC 的数量。则车辆 L 卸载决策目标为：

$$\begin{aligned} &MaxSum(M)=\max\sum_{s_i\in S, r_j\in R}u(s_i,r_j)\\ &\text{s.t. } C1: c_{s_i}\in C_{r_j}, C_{r_j}\subseteq C_{\text{all}}\\ &\qquad C2: et_s\geqslant smt_r \&\& emt_r\geqslant st_s\\ &\qquad C3: u(s_i,r_j)>0 \end{aligned} \tag{2}$$

其中，(s_i, r_j)是卸载服务对；$u(s_i, r_j)$是卸载服务对(s_i, r_j)获得的卸载执行增益计算函数；$M=\{(s_i, r_j) \mid s_i\in S, r_j\in R\}$是任务卸载策略。C1 至 C3 是卸载服务对$(s_i, r_j)$存在的限制条件：C1 是任务类型限制，即任务 s_i 的类型属于 RSU_MECr_j 能够服务的任务类型，C_{all}是所有的任务类型集合；C2 是决策时效限制，即任务 s_i 的卸载决策时间窗和 RSU_MECr_j 的卸载服务时间窗有重叠；C3 是卸载收益限制，即任务 s_i 由 RSU_MECr_j 卸载能够获得有效的卸载收益。

4 在线任务卸载决策算法

本章首先将问题 P 转化为加权二部图匹配问题，然后提出了一种基于加权二部图匹配的在线任务卸载决策算法 OODA(Online Offloading Decision Algorithm)，用于求解该问题。

4.1 转化方法

在问题 P 中，如果将 RSU_MEC 作为资源方，将待执行任务作为资源征用方，将卸载服务对作为可选的资源分配策略，将最大化整体任务卸载执行增益作为资源分配目标，则问题 P 可被视为一个典型的类秘书问题，并能够转化为加权二部图匹配问题进行求解。转化的具体方法如下：将待卸载任务和 RSU_MEC 分别作为二部图左侧子图和右侧子图的顶点，将卸载服务对和卸载执行增益分别作为边和边权重。即可令 $G=(S,R,E)$ 表示问题 P 对应的加权二部图，其中，$S=\{s_i \mid 1\leqslant i\leqslant n\}$表示左子图顶点集；$R=\{r_j \mid 1\leqslant j\leqslant m\}$表示右子图顶点集；$E=\{e(s_i, r_j) \mid s_i\in S, r_j\in R\}$表示边集合，$e(s_i, r_j)$表示卸载服务对$(s_i, r_j)$存在，$w(e(s_i, r_j))=u(s_i, r_j)$表示卸载服务对$(s_i, r_j)$对应的卸载执行增益。那么，问题 P 的最优在线任务卸载策略集合是图 G 的最佳匹配 M_{OPT}，满足条件：

$$\sum_{e(s_i,r_j)\in M_{\text{OPT}}}u(s_i,r_j)=\max\sum_{e(s_i,r_j)\in M}u(s_i,r_j)$$

其中，$M_{\text{OPT}}\subseteq M$，$M=\{e(s_i, r_j) \mid s_i\in S, r_j\in R\}$，$\forall e(s_i, r_j)$，$e(s_i', r_j')\in M$，有 $s_i\neq s_i'$且 $r_j\neq r_j'$。

4.2 基于二部图匹配的在线任务卸载决策算法

本节提出了一种基于二部图匹配的在线任务卸载决策算法 OODA，求解转化后的问题 P。车辆 L 的总行程可以被划分为若干个卸载决策时间段。在每个时间段开始时，车辆 L 都会重新启动 OODA 算法；而在每个卸载决策时间段结束时，车辆 L 都会获得一个可行的最优在线任务卸载策略集合，即加权二部图的最佳匹配，并结束 OODA 算法。

假设车辆 L 沿道路匀速行驶，且 RSU_MEC 沿道路均匀铺设。那么，车辆 L 可根据自身的历史行驶信息，预估其在每个卸载决策时间段内将要经过的 RSU_MEC 总数量。例如，如果所有的卸载决策时间段等长，则车辆 L 在某个卸载决策时间段 T 内，将要经过的 RSU_MEC 总数量 $m=\sum_{p=1}^{l}m_p/l$，$l\geqslant 1$，其中 m_p 表示时间段 T 的前 p 个时间段内车辆 L 实际经过的 RSU_MEC 总数量。另外，由于车辆 L 通常能够掌握未来短期内的行驶路线，因此其可预知未来将要经过的若干个 RSU_MEC。综上所述，在 OODA 算法开始时，车辆 L 的已知信息包括待卸载任务集合 S、预估的将要经过的 RSU_MEC 总数量 m，以及能够预知的 RSU_MEC 数量 k。

由于车辆 L 在卸载决策时间段 T 之初，并不知道其会

经过的所有 RSU_MEC，只有当其经过确定的某个 RSU_MEC 时，才会获知该 RSU_MEC 的相关信息，进而判断对应卸载服务对存在的可能性，以及计算卸载执行增益等。因此，OODA 算法选择在车辆 L 经过新的 RSU_MEC 时，进行在线任务卸载决策。另外，为了做出更为准确的卸载决策，车辆 L 在进行实际的卸载决策前，需要掌握一定的先验知识。因此，OODA 算法将整个卸载决策时间段 T 分为决策观察阶段和实际决策阶段。在决策观察阶段，车辆 L 只会收集与卸载决策相关的信息，即不断扩建加权二部图 G，而不做任何的实际卸载决策；在实际决策阶段，车辆 L 不但会扩建加权二部图 G，还会做出实际的卸载决策。

算法 1 是 OODA 算法的伪代码。其中，第 4 行给出了决策观察阶段和实际决策阶段的划分条件，即新经过的 RSU_MECr_j 是否为前$\lfloor m/e \rfloor$个。前$\lfloor m/e \rfloor$个 RSU_MEC 对应决策观察阶段；其他则对应实际决策阶段。第 5－10 行是车辆 L 作出实际卸载决策的方法。首先求扩建后的加权二部图 G 的最佳匹配 M_{r_j}，然后判断 M_{r_j} 中是否存在与 r_j 相关的边，即卸载服务对的存在性。如果边不存在，则在线任务卸载策略为空；否则采用该边对应的任务卸载策略。另外，由于每个任务只需执行一次，因此将任务成功卸载后，需要从图 G 中删除与其相关的所有顶点和边。

算法 1 包含两个较为复杂的功能：1）扩建加权二部图 G；2）求解 G 的最佳匹配 M_{r_j}。上述两个功能分别通过调用 $BigraphEx(r_j,k,G)$算法和 $Greedy(G)$算法来实现。

算法 1　OODA

输入：待卸载任务集合 S，将经过的 RSU_MEC 总数量 m，可预知的 RSU_MEC 数量 k

输出：最优在线任务卸载策略集合 M

1. 初始化：RSU_MEC 集合 R＝∅，边集合 E＝∅，M＝∅，二部图 G＝(S,R,E)
2. for each 新经过的 RSU_MECr_j in 卸载决策时间段 T then
3.　根据 r_j 带来的新信息，调用算法 2 扩建加权二部图 G
4.　if r_j 不属于经过的前⌊m/e⌋个 RSU_MEC then
5.　　调用算法 3，求扩建后的加权二部图 G 的最大权重匹配 M_{r_j}
6.　　if M_{r_j} 存在与 r_j 相关边 e(s_i,r_j) then
7.　　获得在线任务卸载策略(s_i,r_j)，将任务 s_i 卸载至 r_j；
8.　　将 e(s_i,r_j)加入最优在线任务卸载决策集合 M；
9.　　删除图 G 中与任务 s_i 相关的顶点和边；
10.　end if
11.　end if
12. end for
13. return 最优在线任务卸载策略集合 M

算法 2 是 $BigraphEx(r_j,k,G)$的伪代码。其功能是扩建二部图 G，主要是扩建 R 侧顶点和边集合 E。其中，第 1－2 行对应 R 侧顶点的扩建。首先根据可预知的 RSU_MEC 数量，获得 r_j 能够带来的新增已知 RSU_MEC 集合 $R_{r_j}=\{r_j,\cdots,r_{j+k}\}$，其中 r_{j+k}表示 r_j 后的第 k 个将要经过的 RSU_MEC。然后，将 R_{r_j} 加入图 G 的现有 R 侧顶点集合。第 3－15 行对应边集合 E 的扩建，主要思想是为 R_{r_j} 中的每个 RSU_MEC $r_{j'}$ 构建其对应的边集合 $E_{r_{j'}}$，并将 $E_{r_{j'}}$ 加入图 G 的现有边集合 E 中。具体地，对于每个 $r_{j'}$，首先设置其对应边集合 $E_{r_{j'}}$ 为空；然后遍历待卸载任务集合 S，获取与 $r_{j'}$ 对应的卸载服务对；最后根据式(1)和式(2)，判定每个卸载服务对的存在性，并计算对应的卸载执行增益。当卸载服务对同时满足式(2)的 C1 至 C3 约束时，则构建其对应的边，并将该边添加到 $r_{j'}$ 的边集合 $E_{r_{j'}}$ 中。

算法 2　$BigraphEx(r_j,k,G)$

输入：RSU_MECr_j，现有的二部图 G＝(S,R,E)，可预知的 RSU_MEC 数量 k

输出：扩建后的二部图 G

1. 根据可预知的 RSU_MEC 数量 k，获得 r_j 带来的新增已知 RSU_MEC 集合 $R_{r_j}=\{r_j,\cdots,r_{j+k}\}$；
2. 将 R_{r_j} 加入顶点集合 R；
3. for each $r_{j'}$ in R_{r_j} do
4.　初始化 $r_{j'}$ 对应的边集合 $E_{r_{j'}}=\emptyset$；
5.　for each s_i in S do
6.　　if 卸载服务对(s_i,$r_{j'}$)满足式(2)中 C1 和 C2 约束 then
7.　　根据式(1)计算卸载服务对(s_i,$r_{j'}$)的卸载执行增益 u(s_i,$r_{j'}$)；
8.　　if u(s_i,$r_{j'}$)满足式(2)中 C3 约束 then
9.　　构建边 e(s_i,$r_{j'}$)，并将 u(s_i,$r_{j'}$)作为权重；
10.　　将边 e(s_i,$r_{j'}$)添加到 $E_{r_{j'}}$；
11.　　end if
12.　end if
13. end for
14. 将 $E_{r_{j'}}$ 添加到边集合 E
15. end for
16. return 扩建后的二部图 G

算法 3 是 $Greedy(G)$的伪代码，其主要功能是采用贪婪策略求加权二部图 G 的最佳匹配。该算法总是先判断权重最大的边是否符合匹配条件，若符合，则将其加入 M_{OPT} 中。

算法 3　$Greedy(G)$

输入：加权二部图 G＝(S,R,E)

输出：最大权匹配 M_{OPT}

1. 初始化 $M_{OPT}=\emptyset$；
2. 根据边权重降序排列 E；
3. for each e(s_i,r_j)in E，按照降序遍历 do
4.　if $M_{OPT}\cup$e(s_i,r_j)是一个匹配 then
5.　　将边 e(s_i,r_j)加入匹配 M_{OPT}；
6.　end if
7. end for
8. return 最大权匹配 M_{OPT}

4.3　竞争比分析

本文采用竞争比来评估 OODA 算法的性能。竞争比指最优离线方案获得的权重与在线算法实现的期望权重之间的比值[21]。对于 OODA 算法，其竞争比为：

$$CR=\frac{MaxSum(OPT)}{E[MaxSum(M)]} \tag{3}$$

其中，$MaxSum(OPT)$是最优离线卸载决策方案获得的卸载总增益，$E[MaxSum(M)]$是 OODA 算法获得的期望卸载总增益。

下文将通过 4 个引理和 1 个定理推导 OODA 算法的竞争比。

令 $R'_{r_j}=R^* \cup R_{r_j}$ 表示车辆 L 与 RSU_MECr_j 相遇时已知的 RSU_MEC 集合，其中，R^* 为已经经过的 RSU_MEC 集合，$R_{r_j}=\{r_j,\cdots,r_{j+k}\}$ 为 r_j 带来的新增已知 RSU_MEC 集合。令 M_{r_j} 表示车辆 L 与 RSU_MEC r_j 相遇时，二部图 G 的局部最优匹配，$MaxSum(M_{r_j})$表示 M_{r_j} 获得的卸载总增益。

引理 1 $E[MaxSum(M_{r_j})]\geqslant\frac{|R'_{r_j}|}{m}MaxSum(OPT)$。

证明：令 $OPT_{(n,|R'_{r_j}|)}$ 表示由 n 个任务和 $|R'_{r_j}|$ 个 RSU_MEC 得到的最优匹配，则 $MaxSum(OPT_{(n,|R'_{r_j}|)})$ 为 $OPT_{(n,|R'_{r_j}|)}$ 获得的卸载总增益。因此，有：

$$E[MaxSum(M_{r_j})]\geqslant MaxSum(OPT_{(n,|R'_{r_j}|)}) \tag{4}$$

另外，由于车辆 L 经过 RSU_MEC 的次序符合随机顺序模型的随机性，因此 R'_{r_j} 可被看作从 RSU_MEC 集中随机选择的 $|R'_{r_j}|$ 个 RSU_MEC。则

$$MaxSum(OPT_{(n,|R'_{r_j}|)})\geqslant\frac{|R'_{r_j}|}{m}MaxSum(OPT) \tag{5}$$

那么，推导得：

$$E[MaxSum(M_{r_j})]\geqslant\frac{|R'_{r_j}|}{m}MaxSum(OPT) \tag{6}$$

引理 2 若边 $e(s_i,r_j)\in M_{r_j}$，则其期望卸载增益 $E[u(s_i,r_j)]\geqslant\frac{1}{m}MaxSum(OPT)$。

证明：当车辆 L 到达 RSU_MEC r_j 时，r_j 可被视为从 R'_{r_j} 中随机选择的 RSU_MEC，则卸载服务对 $e(s_i,r_j)\in M_{r_j}$ 的期望增益为：

$$E[u(s_i,r_j)]=\frac{1}{|R'_{r_j}|}MaxSum(M_{r_j}) \tag{7}$$

联合引理 1 有：

$$E[u(s_i,r_j)]=\frac{1}{|R'_{r_j}|}MaxSum(M_{r_j})\geqslant\frac{1}{m}MaxSum(OPT) \tag{8}$$

引理 1 和引理 2 给出了 $e(s_i,r_j)\in M_{r_j}$ 期望增益的界。下文采用引理 3 和引理 4 给出 $e(s_i,r_j)\in M$ 的期望卸载增益。

引理 3 令 $P[s_i|(\lfloor m/e\rfloor+1,j-1)]$表示任务 s_i 从第 $\lfloor m/e\rfloor+1$ 个到第 $j-1$ 个 RSU_MEC 均未卸载执行的概率，则 $P[s_i|(\lfloor m/e\rfloor+1,j-1)]=\frac{\lfloor m/e\rfloor}{j-1}$。

证明：令 R' 表示从第 $\lfloor m/e\rfloor+1$ 个到第 $j-1$ 个 RSU_MEC 的集合，r_l 表示 R' 中的第 l 个 RSU_MEC。由于 r_l 可以看作是从 R' 中的前 l 个 RSU_MEC 中随机选择的 RSU_MEC，且任务 s_i 至少在前 $l-1$ 个 RSU_MEC 不能接入卸载，那么任务 s_i 通过 r_l 卸载执行的最大概率为 $1/l$。进而任务 s_i 不能通过 RSU_MEC r_l 卸载执行的概率为 $(l-1)/l$，因此有：

$$P[s_i|(\lfloor m/e\rfloor+1,j-1)]=\prod_{l=\lfloor m/e\rfloor+1}^{j-1}\frac{l-1}{l}=\frac{\lfloor m/e\rfloor}{j-1} \tag{9}$$

引理 4 $E[e(s_i,r_j)\in M]\geqslant\frac{1}{m}\frac{\lfloor m/e\rfloor}{j-1}MaxSum(OPT)$。

证明：当且仅当任务 s_i 在路过第 $\lfloor m/e\rfloor+1$ 个到第 $j-1$ 个 RSU_MEC 时未卸载成功，$e(s_i,r_j)\in M_{r_j}$ 才会被添加到匹配集 M 中。那么，联合引理 2 和引理 3，可得 $e(s_i,r_j)\in M$ 的期望卸载增益为：

$$E[e(s_i,r_j)\in M]\geqslant\frac{1}{m}\frac{\lfloor m/e\rfloor}{j-1}MaxSum(OPT) \tag{10}$$

定理 1 OODA 算法的期望竞争比 CR 为 e。

证明：令 $u(s_i,r_j)$表示 RSU_MECr_j 在最终匹配 M 中贡献的卸载增益，则 $E[u(s_i,r_j)]=E[e(s_i,r_j)\in M]$。联合引理 4 可得：

$$\begin{aligned}E[MaxSum(M)]&=E[\sum_{j=1}^{m}u(s_i,r_j)]\\&\geqslant\sum_{j=\lfloor m/e\rfloor+1}^{m}\frac{\lfloor m/e\rfloor}{j-1}\frac{1}{m}MaxSum(OPT)\\&=\frac{\lfloor m/e\rfloor}{m}\sum_{j=\lfloor m/e\rfloor+1}^{m}\frac{1}{j-1}MaxSum(OPT)\end{aligned} \tag{11}$$

由 $\frac{m/e-1}{m}\leqslant\frac{\lfloor m/e\rfloor}{m}\leqslant\frac{m/e}{m}$，可得 $\frac{1}{e}-\frac{1}{m}\leqslant\frac{\lfloor m/e\rfloor}{m}\leqslant\frac{1}{e}$；由调和数列的求和公式，可得 $\sum_{j=\lfloor m/e\rfloor+1}^{m}\frac{1}{j-1}=\ln\frac{m-1}{\lfloor m/e\rfloor-1}\geqslant\ln\frac{m}{\lfloor m/e\rfloor}\geqslant 1$。将 $\frac{\lfloor m/e\rfloor}{m}$ 和 $\sum_{j=\lfloor m/e\rfloor+1}^{m}\frac{1}{j-1}$ 均取下限值，有 $\frac{\lfloor m/e\rfloor}{m}\sum_{j=\lfloor m/e\rfloor+1}^{m}\frac{1}{j-1}\geqslant\frac{1}{e}-\frac{1}{m}$。联合式(11)，有 $\frac{MaxSum(OPT)}{E[MaxSum(M)]}\leqslant e$，即 $CR\leqslant e$。

5 实验分析与比较

5.1 实验环境设置

本节采用 Python 搭建高速移动车辆的任务卸载决策场景。在每个卸载决策周期 T 内，车辆 L 携带的待执行任务数量 $n\in(1,36)$，任务类型 $|C_{all}|=10$。车辆 L 沿道路匀速行驶，经过的 RSU_MEC 的数量 $m\in(5,35)$，每个 RSU_MEC 的服务半径为 500 m。其他参数设置如下：

1)车辆 L：$f_l=10$ GCPUcycles/s，$e_l=0.3$ J/GCPUcycle，$g_l=150$ M/s，$p_l=10$ dBM。

2)任务：$d_s\in[35,50]$GCPUcycles，$b_s\in[45,75]$MB，st_s 和 et_s 是时间段 T 内的随机时间，间隔为[500,1 000]s，c_s 从 C_{all} 中随机抽取。

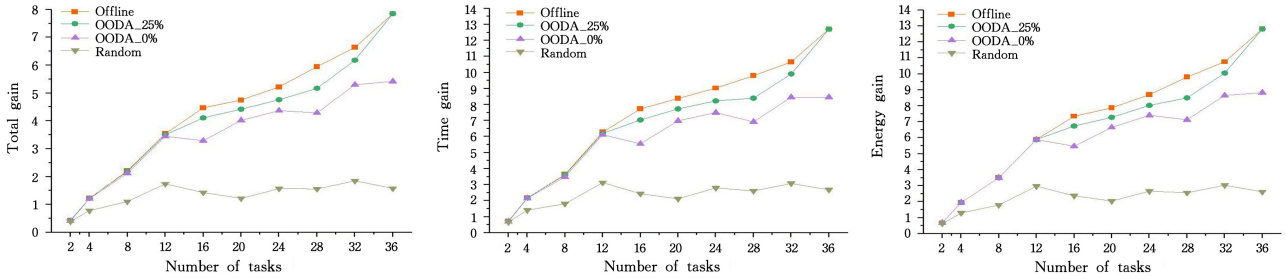
3)RSU_MEC：$f_r\in[40,60]$GCPUcycles/s，$[st_s,et_s]$由 RSU_MEC 的服务范围、车辆 L 的速度以及 RSU_MEC 的出现顺序综合计算得到，C_r 为从 C_{all} 随机抽取的子集。

5.2 结果与分析

本节将 OODA 算法和离线卸载决策算法 Offline、随机卸载决策算法 Random 进行比较。其中，Offline 算法指在卸载决策时间段 T 之初，车辆就已知问题 P 对应的完整加权二部图 G，并采用算法 3 求图 G 的最大权匹配，其结果理论上对应于最优离线卸载决策方案；Random 算法指当车辆 L 与某个 RSU_MEC 相遇时，会从待执行任务中随机选择一个卸载至该 RSU_MEC。实验采用的所有数据为 100 次实验的平均结果。另外，由于 OODA 算法在决策观察阶段不做任务卸载决策，因此所有算法均采用卸载决策阶段的结果。在结果图中，OODA_x%表示车辆 L 采用 OODA 算法，预知的 RSU_MEC 数量 k 与 m 的比值为 x%。另外，在计算总增益时，对能耗增益与时延增益进行 Min-Max 归一化处理[22]。

图 2 是当 $m=30$，$\theta=0.5$ 时，卸载增益与任务数量的变化趋势图。如图 2 所示，所有卸载策略算法的总增益、时延增

益和能耗增益，均会随待执行任务数量的增加而增大。其中，Random 最差，OODA 次优，Offline 最优。对于 OODA 算法，当车辆 L 对未来路径一无所知时，即 OODA_0% 算法，其获得的各项卸载增益明显优于 Random 算法；当车辆能够预知 25% 的 RSU_MEC 时，即 OODA_25% 算法，其获得的各项卸载增益已接近于 Offline 算法。对于 Random 算法，当任务数量 $n\geqslant 12$ 时，其取得的卸载增益出现不增反降的现象。这是因为车辆经过的 RSU_MEC 的总资源有限，当任务密度过大时，任务间的卸载竞争造成的。

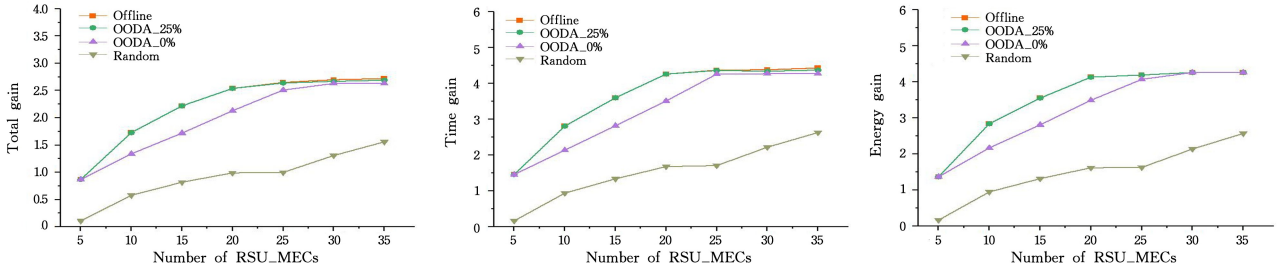


(a)卸载总增益 vs. 任务数量　(b)卸载时延增益 vs. 任务数量　(c)卸载能耗增益 vs. 任务数量

图 2　卸载增益 vs. 任务数量

Fig. 2　Offloading gain vs. the number of tasks

图 3 是当 $n=10$，$\theta=0.5$ 时，卸载增益与 RSU_MEC 数量的变化趋势图。与图 2 类似，OODA_0%算法获得的各项卸载增益明显优于 Random 算法；OODA_25%算法获得的各项卸载增益已接近于 Offline 算法。图 3(c)中，当 RSU_MEC 的数量为 30 或 35 时，OODA_25%算法和 Offline 算法的能耗增益指标线发生重叠，表明两种算法取得的增益相同。如图 3 所示，所有算法的总增益、时延增益和能耗增益，均会随着 RSU_MEC 数量的增加而增大。其中，优劣次序依次为 Offline，OODA 和 Random。当 $m\leqslant 20$ 时，各算法的卸载增益随着 RSU_MEC 数量的增加迅速增长。而当 $m>20$ 时，各项卸载增益的增长速度放缓，这是因为此时大部分的任务都获得了成功卸载。

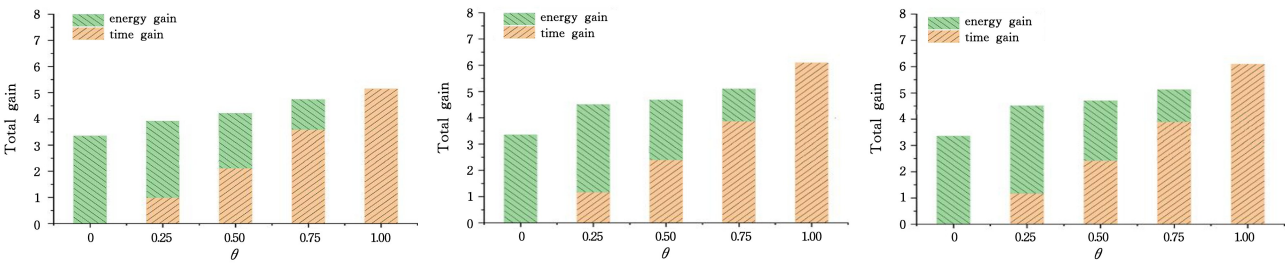


(a)卸载总增益 vs. RSU_MEC 数量　(b)卸载时延增益 vs. RSU_MEC 数量　(c)卸载能耗增益 vs. RSU_MEC 数量

图 3　卸载增益 vs. RSU_MEC 数量

Fig. 3　Offloading gain v. s. the number of RSU_MECs

图 4 是当 $m=30$，$n=20$ 时，OODA 算法获得的卸载增益与权衡因子 θ 的变化趋势图。如图 4 所示，卸载增益由时延增益与能耗增益共同构成，权衡因子 θ 的取值范围是[0，1]。随着 θ 值从 0 至 1 增加，时延增益占总增益的比值不断增加，能耗增益则不断减少。这是因为车辆 L 将重视能耗增益逐步转变为重视时延增益。当 $\theta=0$ 时，表示车辆只考虑能耗增益，此时图中所有的增益都是由能耗增益构成的；当 $\theta=1$ 时，表示车辆只考虑时延增益，此时图中所有的增益都是由时延增益构成。这完全符合 OODA 算法的预期。



(a)OODA_0%算法　(b)OODA_25%算法　(c)OODA_100%算法

图 4　卸载增益 vs. 权衡因子 θ

Fig. 4　Offloading gain vs. tradeoff factor θ

结束语　本文对高速行驶车辆的任务卸载决策建模，将其转化为类秘书问题，并提出了一个竞争比为 e 的在线任务卸载决策算法 OODA。该算法基于求解加权二部图在线匹配的思想，协助车辆做出实时的任务卸载决策，并最大化整体

卸载执行增益。仿真实验将 OODA 与 Offline、Random 算法进行了充分比较，结果表明 OODA 算法获得卸载执行增益的能力远优于 Random 算法，并且只需较少的预知信息，就能获得趋近于最优 Offline 算法的性能。

本文在进行任务卸载时，将待卸载任务作为整体考虑，实现了粗粒度层面上的在线任务卸载决策。而在真实场景下，单个智能车载任务通常由多个相互依赖的子任务构成。因此，在未来的工作中，需要进一步考虑各子任务之间的相互依赖关系，以及它们对卸载资源的竞争，从而探讨如何实现细粒度层面上的在线任务卸载决策。

参 考 文 献

[1] ESKANDARIAN A, WU C X, SUN C Y. Research Advances and Challenges of Autonomous and Connected Ground Vehicles [J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(2): 683-711.

[2] SHI W S, SUN H, CAO J, et al. Edge Computing-An Emerging Computing Model for the Internet of Everything Era [J]. Journal of Computer Research and Development, 2017, 54(5): 907-924.

[3] FAN Y F, YUAN S, CAI Y, et al. Deep Reinforcement Learning-based Collaborative Computation Offloading Scheme in Vehicular Edge Computing [J]. Computer Science, 2021, 48(5): 270-276.

[4] GU B, ZHOU Z Y. Task Offloading in Vehicular Mobile Edge Computing: A Matching-Theoretic Framework [J]. IEEE Vehicular Technology Magazine, 2019, 14(3): 100-106.

[5] LI C, LIU F G, WANG B, et al. Dependency-Aware Vehicular Task Scheduling Policy for Tracking Service VEC Networks [J]. IEEE Transactions on Intelligent Vehicles, 2023, 8(3): 2400-2414.

[6] ZHAN W H, LUO C B, WANG J, et al. Deep-Reinforcement-Learning-Based Offloading Scheduling for Vehicular Edge Computing [J]. IEEE Internet of Things Journal, 2020, 7(6): 5449-5465.

[7] WANG Y P, LANG P, TIAN D X, et al. A Game-Based Computation Offloading Method in Vehicular Multiaccess Edge Computing Networks [J]. IEEE Internet of Things Journal, 2020, 7(6): 4987-4996.

[8] DAI P L, HU K W, WU X, et al. Asynchronous Deep Reinforcement Learning for Data-Driven Task Offloading in MEC-Empowered Vehicular Networks [C] // IEEE INFOCOM 2021 — IEEE Conference on Computer Communications. Vancouver, BC, Ca-nada: IEEE, 2021: 1-10.

[9] DAI P L, LIU K, WU X, et al. A Learning Algorithm for Real-Time Service in Vehicular Networks with Mobile-Edge Computing [C] // 2019 IEEE International Conference on Communications (ICC). Shanghai, China: IEEE, 2019: 1-6.

[10] DUAN X T, ZHOU Y K, TIAN D X, et al. Weighted Energy-Efficiency Maximization for a UAV-Assisted Multiplatoon Mobile-Edge Computing System [J]. IEEE Internet of Things Journal, 2022, 9(19): 18208-18220.

[11] NING Z L, DONG P R, WANG X J, et al. Deep Reinforcement Learning for Intelligent Internet of Vehicles: An Energy-Efficient Computational Offloading Scheme [J]. IEEE Transactions on Cognitive Communications and Networking, 2019, 5(4): 1060-1072.

[12] PREATER J. On Multiple Choice Secretary Problems [J]. Mathematics of Operations Research, 1994, 19(3): 597-602.

[13] KORULA N, PAL M. Algorithms for Secretary Problems on Graphs and Hypergraphs [C] // International Colloquium on Automata, Languages and Programming. Berlin, Heidelberg: Springer, 2008: 508-520.

[14] HU J T, WANG G C, XU X T. Task Migration Strategy with Energy Optimization in Mobile Edge Computing [J]. Computer Science, 2020, 47(6): 260-265.

[15] KESSELHEIM T, RADKE K, TÖNNIS A, et al. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions [C] // European symposium on algorithms. Berlin, Heidelberg: Springer, 2013: 589-600.

[16] REIFFENHAUSER R. An optimal truthful mechanism for the online weighted bipartite matching problem [C] // Proceedings of the Thirtieth Annual ACM—SIAM Symposium on Discrete Algorithms. San Diego California: Society for Industrial and Applied Mathematics, 2019: 1982-1993.

[17] TONG Y X, SHE J Y, DING B, et al. Online mobile micro-task allocation in spatial crowdsourcing [C] // 2016 IEEE 32nd International Conference on Data Engineering (ICDE). Helsinki, Finland: IEEE, 2016: 49-60.

[18] ZHAO G M, XU H L, ZHAO Y M, et al. Offloading tasks with dependency and service caching in mobile edge computing [J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(11): 2777-2792.

[19] CAI L F, WEI X L, XING C Y, et al. Failure-resilient DAG Task Rescheduling in Edge Computing [J]. Computer Science, 2021, 48(10): 334-342.

[20] XU C, ZHENG G Y, ZHAO X W. Energy-minimization task offloading and resource allocation for mobile edge computing in NOMA heterogeneous networks [J]. IEEE Transactions on Vehicular Technology, 2020, 69(12): 16001-16016.

[21] BORODIN A, EL-YANIV R. Online computation and competitive analysis [M]. Cambridge University Press, 2005: 23-29.

[22] PATRO S, SAHU K. Normalization: A preprocessing stage [J]. International Advanced Research Journal in Science, Engineering and Technology, 2015, 2(3): 20-22.

DING Shuang, born in 1982, associate professor, Ph.D, is a senior member of CCF (No. 91019M). Her main research interests include mobile crowd sensing and mobile edge computing.

HE Xin, born in 1974, professor, Ph.D supervisor, is a senior member of CCF (No. 16041S). His main research interests include computer network and mobile computing.

（责任编辑：喻藜）