车辆-边缘协作环境下自适应任务卸载策略

张 超，赵辉，董宇航，唐汉秦，王静，万波，王泉

(西安电子科技大学 计算机科学与技术学院，陕西 西安 710071；

陕西省智能人机交互与可穿戴技术重点实验室，陕西 西安 710071）

摘要：由于边缘服务器的计算资源与覆盖范围有限，当车辆任务卸载请求激增或车辆不在边缘服务器覆盖范围内时，将延迟任务完成时间，影响用户体验。然而，在道路上行驶的部分车辆在满足自身需求的前提下可能存在空闲资源，边缘服务器之间负载也存在负载不均衡情况，造成资源浪费。针对以上问题，本文提出一种车辆-边缘协作环境下的自适应任务卸载策略，以提升任务卸载效率和减少任务完成时间。首先，根据车辆位置与边缘服务器通信范围，考虑车辆间以及边缘服务器间的协作关系，以最小化任务完成时间为目标，构建车辆-边缘协作环境下的任务卸载模型；其次，根据车辆位置提出了一种自适应选择任务卸载策略，当车辆在边缘服务器覆盖范围外时，提出一种最佳服务车辆选择算法，通过车辆之间的协作将任务卸载至最佳服务车辆；当车辆在边缘服务器覆盖范围内时，提出一种基于改进差分进化算法的任务调度算法，通过边缘服务器之间的协作将任务卸载至最优边缘服务器。最后，设计仿真实验并与现有的车联网环境下的任务卸载方法进行对比，验证了本文所提策略的有效性。

关键词：车辆任务卸载；车辆协作；边缘协作；差分进化算法

中图分类号： 文献标识码：A 文章编号：1001-2400(2XXX)0X-0-0

An Adaptive Task Offloading Strategy for Vehicle-Edge Collaborative Environments

## ZHANG Chao, ZHAO Hui, DONG Yuhang, TANG Hanqin, WANG Jing, WAN Bo, WANG Quan

(School of Computer Science and Technology, Xidian University, Xi’an 710071, China;

Key Laboratory of Smart Human-Computer Interaction and Wearable Technology of

Shanxi Province, Xi’an 710071, China)

**Abstract:** Due to the limited computing resources and coverage of the edge server, which will delay the task completion time and affect the user experience when there is a surge of vehicle task offloading requests or the vehicle is not within the coverage of the edge server. However, some of the vehicles traveling on the road may have idle resources under the premise of meeting their own needs, and there are load imbalances between the load of edge servers, resulting in a waste of resources. To address the above issues, we propose an adaptive task offloading strategy for vehicle-edge collaboration environment to improve task offloading efficiency and reduce task completion time. Firstly, according to the communication range of vehicle location and edge server, considering the collaboration relationship between vehicles and between edge servers, the task offloading model in the vehicle-edge collaboration environment is constructed with the goal of minimizing the task completion time. Secondly, an adaptive selection of task offloading strategy is proposed based on the vehicle location. When the vehicles are outside the coverage of edge servers, an optimal auxiliary vehicle selection algorithm is proposed to offload the tasks to the optimal auxiliary vehicles through the collaboration among vehicles. When the vehicle is within the coverage range of the edge server, a task scheduling algorithm based on improved differential evolutionary algorithm is proposed to offload the task to the optimal edge server through the collaboration between edge servers. Finally, simulation experiments are designed and compared with existing task offloading methods in the Telematics environment to verify the effectiveness of the strategy proposed in this paper.

**Key Words:** vehicular task offloading; vehicular collaboration; edge collaboration; differential evolution algorithm

1. **引 言**

随着无线通信和人工智能等技术的快速发展，车辆的智能化程度越来越高，也促进了车联网系统的发展。车联网系统中自动驾驶、实时导航、信息娱乐等智能化服务所需的计算量急速攀升，对时延的要求更高[1]。未来第六代(6G)无线通信技术将为车联网V2X通信提供强有力的保障，这不仅为车联网带来更广泛的高级应用[2]，也对资源受限的智能车辆提出了巨大挑战。

近年来，许多研究致力于车联网系统中的任务卸载技术，旨在将车辆自身难以处理或处理成本过高的任务卸载到云中心，从而提高任务执行效率[2]-[5]。然而，对于车联网系统中延迟敏感型任务而言，传统基于云的任务卸载已难以满足智能车辆应用对实时性的需求。相比之下，在用户附近部署成本较低的边缘服务器是一种有效的计算卸载方案。

在新一代车联网技术中，通过在基站或路边单元部署边缘设备，可以有效减少数据传输的往返时延，确保能够实时响应用户请求[10]-[15]。然而，边缘服务器的资源往往是有限的，当车辆任务卸载请求激增或区域内计算密集型任务较多时，会导致边缘服务器负载过高，影响用户服务质量[9]。此外，由于静态部署的边缘服务器的覆盖范围有限，当车辆超出其覆盖范围内时，任务将无法卸载至边缘服务器。然而，部分在道路上行驶的车辆，在满足自身需求的同时，可能存在部分空闲资源。因此，充分利用道路上行驶车辆的空闲资源，实现车辆-边缘协同的任务卸载，对于提升车联网环境中任务执行效率有着重要的研究意义和价值。

基于上述分析，为充分利用车联网系统中行驶车辆与边缘服务器的整体资源，本文提出了一种车辆-边缘协作环境下的自适应任务卸载策略，以提升任务卸载效率和减少任务完成时间。该策略根据车辆位置与边缘服务器的通信范围，自适应选择任务卸载算法，实现最优化任务卸载。具体来说，当车辆在边缘服务器覆盖范围外时，提出一种最佳车辆选择算法，通过车辆之间的协作实现任务卸载；当车辆在边缘服务器覆盖范围内时，提出一种基于改进差分进化算法的任务调度算法，考虑边缘服务器之间的协作实现任务至边缘服务器的最优卸载，均衡边缘服务器负载，减少任务完成时间。最后，仿真结果证明了本文所提策略可以有效减少任务完成时间。本文的主要贡献如下：

1、为提高任务执行效率，充分利用车联网系统整体资源，考虑车辆间以及边缘服务器间的协作关系，以最小化任务完成时间为目标，建立了车辆-边缘协作环境下的任务卸载模型，并根据车辆位置提出了一种自适应任务卸载策略。

2、当车辆在边缘服务器覆盖范围外时，提出了一种最佳服务车辆选择算法，通过车辆之间的协作将任务卸载至最佳服务车辆。为保障任务的时延及车辆间通信的稳定性，该算法仅考虑行驶车辆一跳范围内的空闲车辆资源，并构建了服务车辆任务等待队列来确保任务在服务车辆的有序执行。

3、当车辆在边缘服务器覆盖范围内时，提出了一种基于改进差分进化算法的任务调度算法，考虑边缘服务器之间的协作，并根据KKT条件获得任务卸载至边缘服务器的最优位置和比例，实现边缘服务器间负载均衡，减少任务完成时间。

1. **相关工作**

近年来，针对VEC环境下的车辆任务卸载策略已经有了较多的研究，为车辆应用程序的处理创造了新的条件。Liu[10]等人考虑任务依赖性和任务卸载过程中的连续控制问题，提出了一种有效的车辆、边缘、云协同计算的依赖感知任务卸载方案，并提出一种基于深度确定性策略梯度的任务卸载算法以获得最优卸载决策，实现快速收敛。Zhang[11]等人根据当前道路密集程度，提出基于模拟退火机制的车辆用户任务卸载新方法，降低任务计算时间和能耗。Zhu[12]等人考虑任务到达的随机性、多输入多输出非正交多址信道干扰导致的信道条件不确定性以及车辆的移动性，提出了一个基于DRL的分布式功率分配方案，以优化VEC环境下能耗和延迟的长期回报。Huang[13]等人针对任务动态产生并且任务数量未知的场景，研究了一种联合任务卸载和任务调度的卸载算法，得到任务最优卸载位置以及在边缘服务器上的执行顺序，以达到延迟和能耗综合成本最优。Qian[14]等人考虑任务接收率、任务完成率、平均完成时间以及计算资源利用率等多个指标来更好的满足QoS需求，并提出了一种细粒度算法，以实现在边缘服务器内部的任务调度。Hossain[15]等人提出了一种基于非合作博弈技术的动态任务卸载算法，车辆可以制定自己的卸载策略，将任务卸载到边缘或云服务器，并且动态调整任务卸载概率以使效益最大化，降低任务执行时间和执行失败率。Zhu[16]等人针对智慧交通云边端协同计算场景下的任务卸载决策问题，提出了一种基于改进型非支配遗传算法的任务卸载决策方案，能有效降低任务完成时延和能耗。虽然以上研究能够提升VEC环境下的任务卸载效率，但当用户任务卸载请求增多时，由于边缘服务器资源有限，在边缘服务器负载不均时会导致任务执行效率下降。此外，边缘服务器覆盖范围有限，不在覆盖范围内的车辆任务无法卸载至边缘服务器，影响用户体验，因此平衡边缘服务器负载和探寻其他可用资源对提升车联网系统任务执行性能有重要意义。

针对边缘服务器之间的协作，在之前的工作[17]中已经做出了充分的研究。在道路模型中部分车辆在满足自身需求的前提下会有剩余空闲资源，如何利用这些空闲资源协助边缘服务器进行任务卸载对车辆边缘计算的发展有重要作用。由于车辆的移动性，车辆在RSU覆盖范围内的停留时间是有限的，当车辆驶出RSU通信范围时任务无法直接卸载到边缘服务器中，容易导致较低的服务质量。Ming[18]等人将汽车和边缘服务器同时作为计算服务节点，提出了一个动态适应的分布式和集中式深度强化学习以响应不同任务的需求，降低系统延迟。Chen[19]等人采用附近RSU和车辆提供边缘计算服务，采用博弈论的概念激励附近车辆共享计算资源，保证车辆安全性和任务延迟最小化。Gong[20]等人提出了一种分步计算卸载算法以形成一个车辆云，并将应用程序分解具有依赖性的子任务，最小化任务卸载和任务执行之间的时间间隔。Li[21]等人提出了一种新的可扩展调制的任务卸载策略，根据当前拓扑网络选择最优的移动车辆作为中继节点协助将任务卸载到边缘服务器，在保证延迟和传输质量的情况下最小化能源消耗。Pang[22]等人为了减少智能车辆网络中的时延，提出了一种中继车辆选择和信誉管理算法，选择高可靠性的中继车辆，同时管理车辆的可信度。Zhang[23]等人针对V2V及V2I通信过程中可能产生的车辆信任问题,提出了基于改进的三值主观逻辑算法的车辆间声誉管理系统，利用区块链技术保证了车辆互联网络的信息安全性。以上研究利用车辆计算资源或通信资源辅助进行任务卸载，但未充分考虑车辆之间的位置、速度、方向等信息，并且在动态环境下多跳传输也存在一定风险，如何在保证通信稳定的同时提升任务卸载效率仍需要进一步研究。

与上述研究不同，本文根据车辆位置，在考虑边缘协作的同时利用车辆之间的协作来提升任务执行效率。当车辆在边缘服务器覆盖范围内时，为了提高边缘服务器资源利用率，本文利用边缘服务器之间的协作来保证任务的执行效率；当车辆在边缘服务器覆盖范围外时，本文仅考虑在车辆一跳通信范围内具有空闲资源的车辆，并考虑车辆之间的行驶方向、相对位置、速度等信息来选取服务车辆以保证通信的稳定性。针对以上情况，本文提出了一种车辆-边缘协作环境下自适应任务卸载的算法，根据车辆所处位置与周围可用资源状况做出最优决策。在仿真实验中，实验结果证明了本文所提方法的有效性。

1. **车辆-边缘协作任务卸载模型**

本文考虑车辆行驶在快速公路的场景，如图1所示。个RSU均匀分布在路边，每个RSU通过有线链路配备一个MEC服务器，RSU集合表示为，路边相邻RSU之间的距离表示为。车辆自由行驶在拥有双向车道的道路上，行驶车辆可以分为请求车辆（RV）和服务车辆（SV），假设请求车辆会产生计算密集型任务，服务车辆当前时刻自身没有额外任务产生，即在满足自身需求下还有空闲资源未使用。请求车辆集合表示为，车辆的计算资源为。每个车辆生成任务遵循参数为的泊松分布，计算任务表示为为，其中表示计算每比特数据需要的CPU周期数，表示生成任务的数据大小。定义为卸载至其他节点执行的比例，任务的部分在本地处理，剩余部分可以卸载至边缘服务器或服务车辆执行，但同一个任务不能同时卸载至边缘服务器或服务车辆。

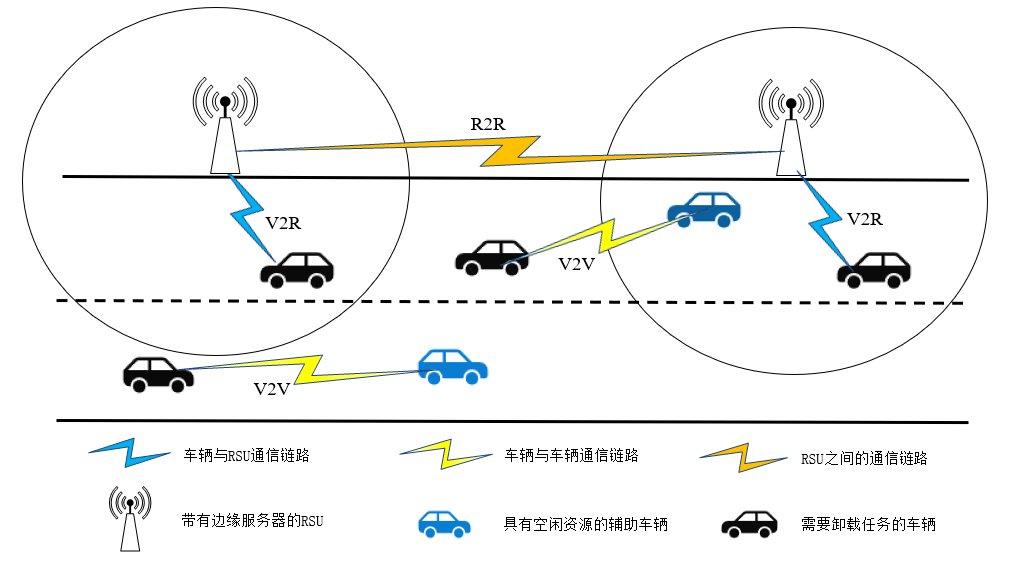


图1 车辆-边缘协作任务卸载架构

* 1. **通信模型**

V2R通信模型：当车辆行驶在RSU覆盖范围内时，车辆信息会被RSU自动收集，可以通过车辆到基础设施（V2R）通信链路将任务卸载至RSU。假设为RSU为车辆分配的带宽资源，那么车辆与RSU通信链路的传输速率可以表示为：



其中，表示车辆的发射功率，为车辆和RSU之间的信道增益，为车辆和RSU之间的距离，代表路径损耗指数，代表高斯白噪声功率。

假设在时刻请求车辆的位置为（），边缘服务器覆盖半径为，当时表示车辆已经进入RSU的通信范围，可以建立V2R通信连接。可以表示为：



V2V通信模型：当请求车辆需要将任务卸载至服务车辆时，在任务数据传输前存在一个连接延迟用于完成信息广播和建立连接，通常为是一个常数。假设服务车辆的总带宽为，并且请求车辆在传输数据时平分带宽。请求车辆与服务车辆之间的传输速率可以表示为：



其中为选择将任务卸载至服务车辆的请求车辆数量，为请求车辆与服务车辆之间的距离。请求车辆的通信范围为，当时，表示车辆之间可以建立V2V通信连接。表示为：



请求车辆与服务车辆之间的通信连接保持时间可以表示为：



其中和分别代表请求车辆与服务车辆在当前时刻的行驶速度，根据车辆位置和速度信息可以分为四种情况。代表车辆与行驶方向相同并且前车速度大于后车；表示车辆与行驶方向相同并且前车速度小于后车；表示车辆与已经相遇过且往相反方向驶去；表示车辆与行驶方向相反且还未相遇。

* 1. **计算模型**

本地计算时间：假设车辆的任务卸载比例为，本地执行速度为，因此任务的本地执行时间为：



服务车辆计算时间：考虑到车载计算资源是有限的，无法并行处理计算任务，因此卸载至服务车辆的任务需要进入等待队列排队执行，只有在前一个任务执行完以后才能处理当前任务。任务卸载至服务车辆的执行时间包括任务传输时间、任务等待时间以及任务执行时间，可以表示为：



其中为服务车辆可以提供的空闲资源，表示需要在队列中等待的时间，代表任务传输时间。



任务在传输过程中可以先执行本地计算部分，因此本地和卸载至服务车辆是并行执行的过程，总耗费时间为二者最大值，本文假设执行结果较小，数据返回时间可以忽略不计。因此任务卸载至服务车辆的总执行时间为：



由于车辆连接时间有限，因此还需保证，否则通信连接中断导致任务执行失败后会重新在本地执行计算任务产生额外成本。

边缘服务器计算时间：当用户处于边缘服务器覆盖范围内时，可以将任务卸载至边缘服务器执行，边缘服务器之间存在有线链路，可以通过任务转发来降低边缘服务器负载，提升任务执行效率。因此任务卸载至边缘服务器时执行时间包括任务上传时间、边缘服务器之间转发时间以及任务执行时间，边缘服务器分配给卸载任务的计算资源为，任务卸载至边缘服务器执行总时间为：



当时表示任务在RSU覆盖范围内，可以将任务卸载至边缘服务器。任务上传时间为：



任务转发时间为边缘服务器之间通过有线链路转发，为边缘服务器之间转发任务的跳数，传输速率固定为，转发时间可以表示为：



与卸载至服务车辆相同，在本地和边缘服务器也可以并行执行任务，因此任务执行时间为：



* 1. **问题定义**

综上所述，对于请求车辆的任务的处理延迟主要包括本地执行时间和卸载部分的执行时间的最大值。本文假设任务只能选择一个节点进行任务卸载，表示任务卸载至服务车辆执行，表示任务卸载至边缘服务器执行，。任务的执行时间可以表示为：



因此，本文在车辆辅助的VEC系统下的任务最小化问题定义为：



 (15a)

 (15b)

 (15c)

 (15d)

 (15e)

 (15f)

约束(15a)表示任务卸载比例限制，约束(15b)和(15d)表示边缘服务器给卸载任务分配的资源不应超过自身计算资源限制, 约束(15c)和(15e)表示边缘服务器给任务传输信道分配的通信资源不应超过最大带宽限制，约束(15f)表示同一时刻一个任务只能卸载至服务车辆或边缘服务器其中一个节点执行。

1. **自适应任务卸载策略**

首先根据约束条件(15f)可以将问题(14)转换为：



问题(16)为任务选择执行延迟最小的节点进行卸载，可以转换为问题(9)和问题(13)分别进行求解。对于前者，即当车辆在边缘服务器覆盖范围外时，本文提出了一种服务车辆选择算法来保证通信稳定的同时最小化任务执行时间。对于后者，本文提出了一种基于改进差分进化算法的任务调度策略获得任务卸载至边缘服务器的比例与卸载位置，并通过KKT条件求得边缘服务器分配给卸载任务的资源比例，两者交替迭代求解，获得任务卸载的最优决策。

* 1. **最佳服务车辆选择算法**

通过分析可知，当=时，任务执行时间是最短的，可以据此得到任务卸载比例。如下所示：



其中为在当前服务车辆上任务执行前的等待时间。

求得任务卸载比例为：



所有车辆可以被划分请求车辆和服务车辆，后者除了处理本地计算任务以外存在部分空闲资源可以利用，因此请求车辆可以将任务卸载至服务车辆执行。由于车载计算资源有限，无法做资源分配功能，因此任务卸载至服务车辆执行时首先需要进入等待队列，等待先到达的任务执行完成，如图2所示。



图2 任务在服务车辆执行等待队列

此外，车辆在道路上是高速移动的并且可能行驶方向也会相反，车辆通信范围也非常有限，V2V通信过程中连接的稳定性是很重要的。因此，本文只考虑在请求车辆一跳范围内的服务车辆，根据公式(18)求得任务卸载比例可以得到任务在本地执行时间和服务车辆的执行时间。然后，根据请求车辆和服务车辆之间的距离估计两者之间的最大连接时间，在连接时间约束的条件下选择执行延迟最低的服务车辆。本文提出的基于队列的服务车辆选择算法如下所示。

|  |
| --- |
| **算法1：**服务车辆选择算法 |
| 初始化:车辆集合  1: for 每个车辆 do  2: 搜索自身通信范围内的服务车辆集合  3: for 每个服务车辆 do  4: 根据公式(5)得到通信保持时间  5: 根据公式(18)得到任务卸载比例  6: 初始化任务等待时间  7: for 每个任务 in  8: + =  9: 根据公式(7)计算服务车辆执行时间  10: if and  11:  12: end if  13: end for  14: end for |
| 15: 返回服务车辆选择决策 |

* 1. **基于改进差分进化算法的任务调度算法**

当请求车辆在RSU覆盖范围内时，可以将任务卸载至该边缘服务器，若本地边缘服务器资源负载过高，还可以将任务转发至负载较低的边缘服务器执行。此时任务的执行位置决策、任务卸载比例与资源分配策略都是不确定的，并且相互耦合。因此本文通过将其分解为资源分配子问题与任务卸载子问题交替迭代求解。

**4.2.1 资源分配子问题**

资源分配子问题可以表示为：





其中。通过观察可知，任务在服务器之间的转发延迟只与任务卸载子问题相关，此时先假设任务卸载位置和卸载比例为定值。因此问题(19)可以写为：





可以观察到约束(15b)-约束(15e)都为凸约束，因此只需要关注目标函数。问题(20)的海森矩阵如下所示：



因此，问题(20)的海森矩阵为正定矩阵，证明该问题为一个凸优化问题，定义其拉格朗日函数为：



根据KKT(Karush-Kuhn-Tucker)条件，求得边缘服务器为每个任务分配的计算资源和带宽资源分别为：





因此，当任务卸载比例与任务卸载位置固定时，通过公式(23)对任务执行分配计算资源，通过公式(24)为任务传输分配带宽资源。

**4.2.2 任务卸载子问题**

本文应用改进的差分进化算法求解任务执行位置和卸载比例搜索问题，并与资源分配子问题结合，不断交替迭代求解。差分进化算法是一种随机搜索全局优化算法[25]，通过群体内个体间的合作与竞争而产生的全局搜索策略，具有原理简单、优化性能高、扩展性良好等特点。然而，传统的差分进化算法在求解复杂高维、多峰优化问题时，容易出现收敛速度慢等情况。教学优化算法[26]模拟了教师和学生学习的过程，是一种基于种群的随机搜索算法，具有概念简单、调控参数少、收敛速度快等特点，能够较好的弥补差分进化算法的缺陷。因此，本文结合差分进化算法和教学优化算法提出一种混合差分教学优化算法（HDTO）, 将教学优化算法的教学阶段加入到差分进化算法迭代过程，帮助实现快速收敛，并将其应用于任务卸载决策子问题求解过程中。不同的变异策略对差分进化算法性能也会有不同影响，在算法执行前期，DE/rand/1具有较大搜索范围，有利于增加种群多样性，而到算法中后期，DE/ best/1寻优速度较快，有利于局部精确搜索。针对此问题，本文提出了一种自适应控制策略来对差分进化算法的变异策略进行改进，通过动态调整权重来保证算法执行效率。此外，为了降低随机产生初始解对算法效率的影响，本文在产生初始种群的过程中引入反向学习策略，提升算法收敛速度。算法主要流程如下所示。

编码：为了将所提算法应用于任务卸载子问题求解，假设种群大小为，由具有维决策变量的个体组成，其中，其中，和分别对应任务卸载位置与任务卸载比例。在任务卸载子问题中得到当前任务卸载位置和任务卸载比例，然后根据资源分配子问题中分配的计算资源和通信资源，可以求得任务执行时间。因此，可以将适应度函数设计为：



初始化：本文引入反向学习策略，通过公式(26)对初始种群解产生反向解，然后根据公式(27)从初始解和反向解中挑选新的解作为种群初始解。



其中为新产生的解，和为编码取值的上界和下界，为(0,1)之间的随机数。



变异：通过加入自适应权重因子，根据迭代次数来动态调整DE/rand/1和DE/ best/1的主导作用，充分发挥两种策略在不断阶段的优势。





其中为控制变异策略权重，为最大迭代次数，为当前迭代次数，为缩放因子,，，分别为随机挑选的个体。

教学：首先选取当前种群中最优的个体作为老师，然后获得种群所有个体的平均水平，并通过公式(30)对每个个体进行教学获得下一代。



其中为(0,1)之间的随机数，随机取1或2。从上可以看出，差分进化算法的变异策略中受个体影响较大，而忽略了全局信息对算法执行效率的影响，容易出现收敛速度慢的情况，而结合教学优化算法的教学阶段可以根据种群最优个体和平均水平来产生下一代，可以保证算法收敛速度。在算法执行前期，DE/rand/1主导变异策略，通过差分进化算法的随机搜索策略来扩大种群多样性，个体之间差异较大，每个个体可以基于教学阶段来获取更多的知识，到了算法执行的中后期，个体之间普遍较为相同，教学优化算法往往无法获得更多知识，差分进化算法中DE/ best/1占主导策略，加强了搜索的集中性，从而提高算法的收敛精度。

交叉：差分进化算法通过公式(31)所示的二项式交叉对和进行交叉操作以得到新解。



其中为(0,1)之间的随机数，为交叉概率，。

选择：执行完上述操作后，根据公式(27)来选择新一代个体。若新个体的适应度值小于原始个体则保留新个体，否则原始个体进入下一代。

用于解决任务卸载子问题的改进的混合差分教学优化算法流程如算法2所示：

|  |
| --- |
| **算法2：** 改进的混合差分教学优化算法 |
| 1: 初始化: 种群集合，最大迭代次数，权重因子，缩放因子，交叉概率  2: 根据公式(26)获得反向解  3: 根据公式(27)重新初始化种群  4: while  5: for in do  6: 根据公式(28)获得权重因子  7: 根据公式(29)执行变异策略  8: 选择种群最优个体作为教师，计算种群平均水平  9: 根据公式(30)执行教学阶段  10: for in  11: if  12:  13: end if  14: end for  15: 求解资源分配子问题获得计算资源和带宽资源分配策略  16: if  17: 更新  18: end for  19:  20: end while  21:返回最优任务卸载决策 |

* 1. **算法复杂度分析**

本文提出了基于车辆辅助的任务卸载策略主要分为两部分：服务车辆的选取、计算资源和带宽资源分配和任务卸载决策。服务车辆选取算法如算法1所示，假设道路行驶车辆数量为，服务车辆的任务等待数量为，那么算法1的总时间复杂度为。在资源分配子问题中只需对所有任务遍历一遍即可完成，因此资源分配子问题时间复杂度为。本文提出的基于改进的差分教学优化算法的任务调度策略主要可分为：初始化阶段、变异阶段、教学阶段和选择阶段，在初始化阶段和选择阶段会应用资源分配策略。设种群数量为，最大循环次数为，由于边缘服务器数量较少，因此可以忽略不计。每个阶段的时间复杂度如下：初始化阶段，通过反向学习策略获得种群初始解，总时间复杂度为，变异阶段和教学阶段的时间复杂度为，选择阶段时间复杂度为，因此算法2的时间复杂度为。由于每个车辆任务产生服从泊松分布也较小，因而可当作常数忽略不计，所以最终时间复杂度为。

1. **实验结果与分析**

本文模拟一个由2km长的双向车道组成的道路拓扑，3个携带边缘服务器的RSU均匀分布在路边，相邻RSU之间距离为500m，RSU覆盖范围为200m，车辆速度从NGSIM的高速公路数据集里随机选取，每辆车的通信范围为100m。其他仿真参数设置如表1所示：

表1 仿真参数

|  |  |
| --- | --- |
| 参数 | 值 |
| 车辆数量 | 50-200 |
| 信道衰落系数 | 4.2 |
| 边缘服务器无线信号覆盖半径/m | 200m |
| 信道背景噪声功率/dBm | -110 |
| 车辆上传功率/dBm | 30 |
| RSU通信带宽/MHz  车辆之间通信带宽/MHz | 30  10 |
| 车辆处理速度/GHz | [5,10] |
| 边缘服务器处理速度/GHz | 30 |
| 边缘服务器间有线链路带宽/Mbps | 500 |
| 任务输入数据大小/MB | [0.5, 1] |
| 任务所需计算量大小cycle/bit | [500，1000] |

为了验证本文所提车辆选择算法与基于混合差分教学优化算法的任务卸载算法的有效性，将与以下方案进行比较，并在相同条件下对其性能进行评估。

1、DE：原始差分进化算法，没有添加自适应控制策略，并且缺少了教学阶段，结合本文所提的服务车辆选取算法。

2、HDTO-MR：任务卸载问题使用本文所提的混合差分教学优化算法，服务车辆算法为选择车辆剩余资源最多的服务车辆。

3、JOS-QPSO[27]：将移动车辆根据位置、速度和方向划分为簇，每次在簇中选择服务车辆，并采用量子粒子群算法来解决任务分配和调度问题。

4、HDTO-AVG：使用本文提出的HDTO作为任务调度算法，但任务平均分配带宽资源和计算资源。

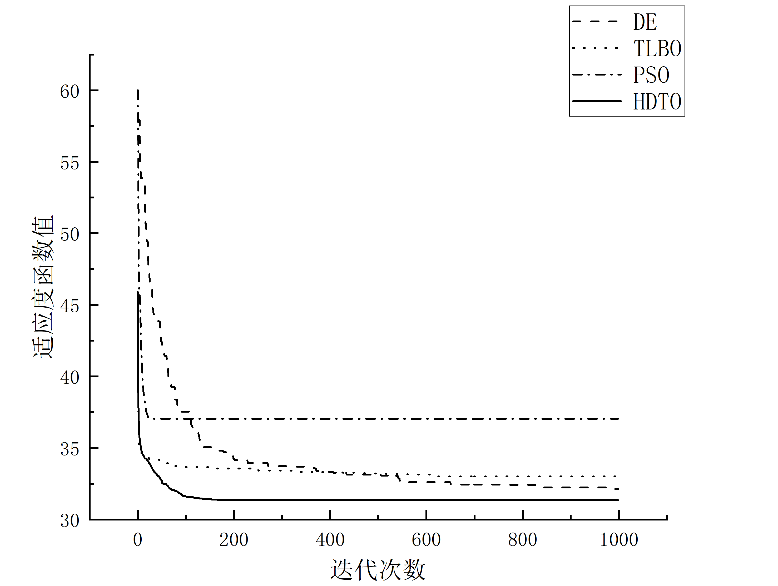


图3 不同算法适应度函数值的变化

图3展示了在相同迭代次数情况下不同算法的适应度函数值。对比基础的DE算法与PSO算法，可以观察到在算法执行前期HDTO算法收敛速度较快，原始的DE算法前期具有较强的搜索能力，但收敛性较低。本文结合反向学习策略初始化种群能扩大搜索范围，增强种群多样性，教学机制可以保证算法快速收敛，因此在算法执行前期能保证快速收敛到较优的水平。并且在算法执行后期，通过自适应控制策略，也能在保证算法稳定的同时尽可能地搜索最优解。

图4展示了采用不同车辆选择算法时对任务平均执行时间的影响。其中对比算法有：Random，随机选择服务车辆；WaitTime，选择排队时间最短的车辆；Resource，选择资源最多的车辆；K-means，根据车辆位置信息划分为簇，从簇中选择服务车辆；VAS，本文所提服务车辆选择算法。可以看到，由于前三种算法没有考虑通信中断带来的影响，因此任务传输失败时会产生额外的执行延迟，K-means虽然在选择车辆时考虑了通信中断的情况，但划分簇的过程中会导致服务车辆分布不均，部分簇中服务车辆资源不够用。而本文提出的车辆选择算法会搜寻自身周围所有的可用节点，在保证通信稳定的情况下最小化任务执行时间，因此效果会由于其他算法。

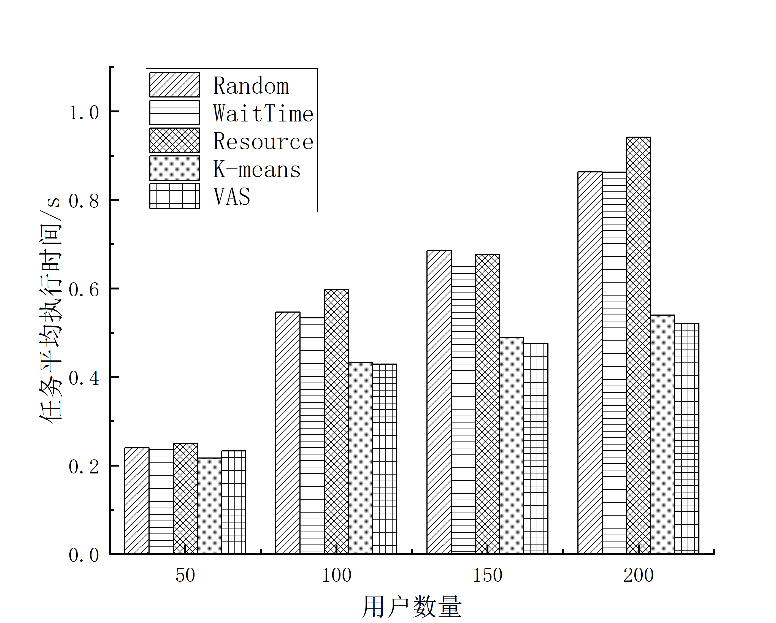


图4 不同车辆选择算法的影响

图5展示了在车辆数量逐渐增多时任务的平均执行时间。当车辆数量较少时，不同算法之间的差别不大是因为此时计算资源足够，每一个请求车辆都能分配到足够的计算资源执行卸载任务。而随着车辆逐渐增多，HDTO算法逐渐与其他算法的差距逐渐增大，其中HDTO-MR由于选择资源最多的服务车辆会导致资源利用不合理，任务延迟增高较为明显。HDTO算法能够最大程度的利用服务车辆与边缘服务器资源，因此在车辆增多，任务数量较大时也能保持良好的执行效率。

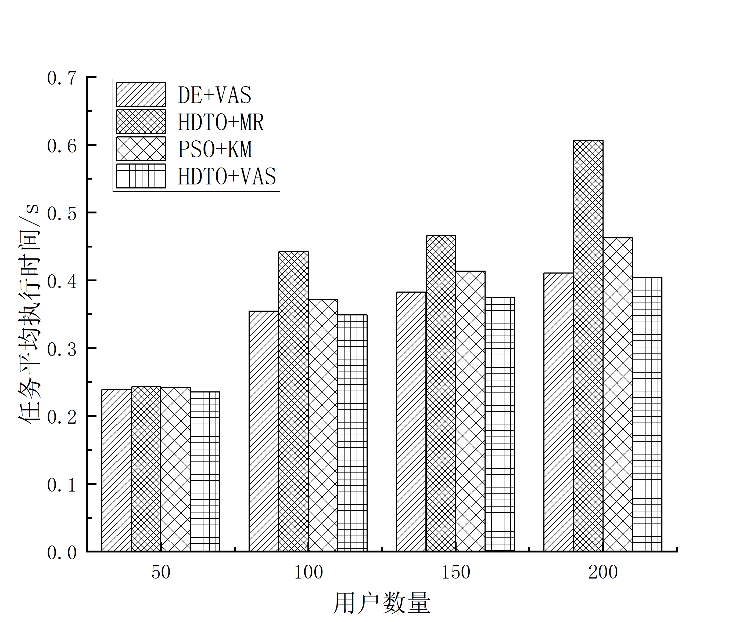


图5 车辆数量不同时的任务平均执行延迟

图6展示了在采取不同算法时每个边缘服务器的任务计算量，在每个边缘服务器计算能力都相同的情况下可以看到本文所提的HDTO算法能有效平衡边缘服务器之间负载。HDTO算法结合反向学习策略初始化种群能扩大搜索范围，增强种群多样性，并且在算法执行后期，通过自适应控制策略，也能在保证算法稳定的同时尽可能地搜索最优解。因此，HDTO算法能够更有效地在边缘服务器之间协作进行任务调度，保证每个边缘服务器的计算量相近，提升边缘服务器计算效率，降低任务执行延迟。

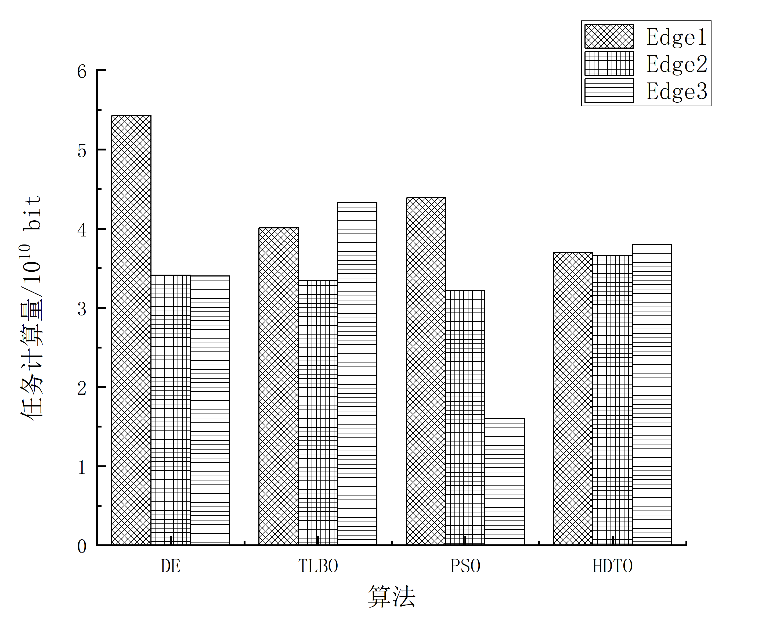


图6 不同边缘服务器的计算量

图7展示了在不同任务大小下每种算法的任务平均执行时间。其中LOC为任务在本地执行的时间，可以看到HDTO+MR和PSO+MR在延迟都远高于HDTO+VAS算法，是因为前两种算法在选取服务车辆时都采用了基于贪心的思想，一个是选择资源最多的车辆，一个是选择通信最稳定的，然而VAS考虑的是选取最优卸载决策，不会盲目进行卸载，并且考虑了边缘服务器之间的协作，因此在任务数量增大时依旧能保证任务执行延迟较低。

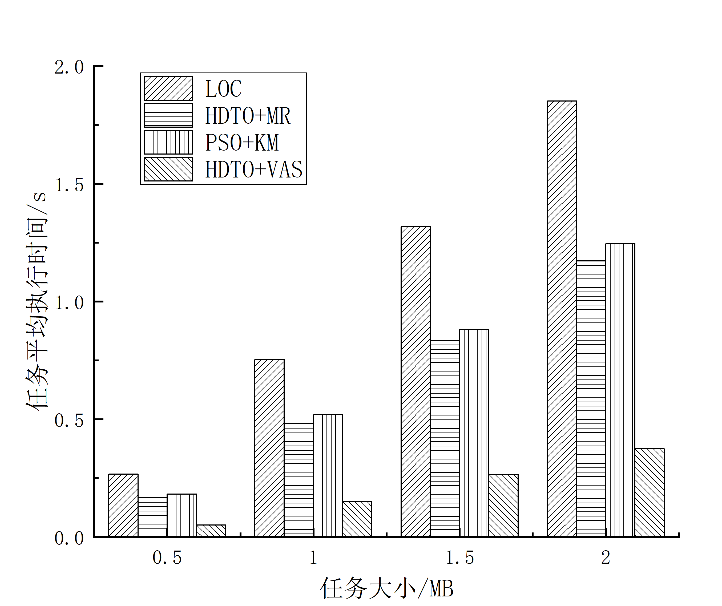


图7 不同任务大小下的任务平均执行延迟

1. **结论**

本文针对车辆边缘计算环境下因为车辆与边缘服务器通信覆盖范围、计算资源受限而导致任务卸载不及时以及任务完成时延较长等问题，提出了一种车辆-边缘协作环境下自适应任务卸载策略来获得任务调度的最优决策，以提升任务卸载效率和减少任务完成时间。首先，根据车辆位置与边缘服务器通信范围，考虑车辆间以及边缘服务器间的协作关系，以最小化任务完成时间为目标，构建车辆-边缘协作环境下的任务卸载模型；其次，根据车辆位置提出了一种自适应选择任务卸载策略，当车辆在边缘服务器覆盖范围外时，提出一种最佳服务车辆选择算法，通过车辆之间的协作将任务卸载至最佳服务车辆；当车辆在边缘服务器覆盖范围内时，提出一种基于改进差分进化算法的任务调度算法，通过边缘服务器之间的协作将任务卸载至最优边缘服务器。仿真结果证明，本文所提的自适应任务卸载策略可以有效降低任务执行时间。在未来的研究中，我们将考虑任务不同特征对任务卸载问题的影响，并研究如何使用激励机制来促进更多计算资源的共享。

**参考文献：**

1. L. Wang, X. Zhu, N. Lit, et al. "Dynamic Vehicle Aware Task Offloading Based on Reinforcement Learning in a Vehicular Edge Computing Network," *2022 18th International Conference on Mobility, Sensing and Networking (MSN)*, Guangzhou, China, 2022, pp. 263-270.
2. M. Noor-A-Rahim, et al. "6G for Vehicle-to-Everything (V2X) Communications: Enabling Technologies, Challenges, and Opportunities," in *Proceedings of the IEEE*, vol. 110, no. 6, pp. 712-734, June 2022.
3. Y. Li, B. Yang, H. Wu, et al. "Joint Offloading Decision and Resource Allocation for Vehicular Fog-Edge Computing Networks: A Contract-Stackelberg Approach," in IEEE Internet of Things Journal, vol. 9, no. 17, pp. 15969-15982.
4. P. Dong, C. Nie, Y. Li, et al."Joint Latency and Energy Optimization of Online Task Offloading Algorithms for Vehicular Edge Computing," 2022 IEEE 24th Int Conf on High Performance Computing & Communications, Hainan, China, 2022, pp. 1180-1187.
5. Y. Zhou et al. "A Novel Approach to Applications Deployment with Multiple Interdependent Tasks in a Hybrid Three-Layer Vehicular Computing Environment," 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia, 2021, pp. 251-256.
6. Lu Y, Han D, Wang X, et al. Enhancing vehicular edge computing system through cooperative computation offloading. Cluster Comput 26, 771–788 (2023).
7. S. Pang, N. Wang, M. Wang, et al. "A Smart Network Resource Management System for High Mobility Edge Computing in 5G Internet of Vehicles," in *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 3179-3191.
8. B. Gu and Z. Zhou, Task Offloading in Vehicular Mobile Edge Computing: A Matching-Theoretic Framework, in IEEE Vehicular Technology Magazine, vol. 14, no. 3, pp. 100-106, Sept. 2019.
9. C. Tang, C. Zhu, H. Wu, et al. "Toward Response Time Minimization Considering Energy Consumption in Caching-Assisted Vehicular Edge Computing," in IEEE Internet of Things Journal, vol. 9, no. 7, pp. 5051-5064, 1 April1, 2022.
10. Liu G, Dai F, Huang B, et al. A collaborative computation and dependency-aware task offloading method for vehicular edge computing: a reinforcement learning approach. J Cloud Comp 11, 68 (2022).
11. 张德干, 李霞, 张捷, 等. 基于模拟退火机制的车辆用户移动边缘计算任务卸载新方法[J]. 电子与信息学报, 2022, 44(9): 3220-3230.

ZHANG Degan, LI Xia, ZHANG Jie, et al. New Method of Task Offloading in Mobile Edge Computing for Vehicles Based on Simulated Annealing Mechanism[J]. Journal of Electronics & Information Technology, 2022, 44(9): 3220-3230.

1. H. Zhu, Q. Wu, X. -J. Wu, Q. Fan, P. Fan and J. Wang, "Decentralized Power Allocation for MIMO-NOMA Vehicular Edge Computing Based on Deep Reinforcement Learning," in IEEE Internet of Things Journal, vol. 9, no. 14, pp. 12770-12782, 2022.
2. C. Huang, Q. Fu, C. Wang, et al. "Joint Task Offloading and Scheduling Algorithm in Vehicular Edge Computing Networks," 2023 IEEE 10th International Conference on Cyber Security and Cloud Computing (CSCloud), Xiangtan, Hunan, China, 2023.
3. L. Qian, P. Sun, K. Yang, et al. "A Novel Multi-Factor Aware Online Scheduling Method for Improving Vehicular Edge Computing Efficiency," ICC 2023 - IEEE International Conference on Communications, Rome, Italy, 2023, pp. 3357-3362.
4. Hossain M.D, Sultana T, Hossain, M.A, et al. Dynamic Task Offloading for Cloud-Assisted Vehicular Edge Computing Networks: A Non-Cooperative Game Theoretic Approach. Sensors 2022, 22, 3678.
5. 朱思峰, 宋兆威, 陈昊, 等. 智慧交通场景下云边端协同的多目标优化卸载决策[J]. 西安电子科技大学学报, 2024, 51(3): 63-75.

ZHU Sifeng, SONG Zhaowei, CHEN Hao, et al. Multi-objective optimization offloading decision with cloud-side-end collaboration in smart transportation scenarios[J]. Journal of Xidian University, 2024, 51(3): 63-75.

1. 张超, 赵辉, 张智峰, 等. 边缘协作环境下最小化完工时间任务调度方法[J/OL].西安电子科技大学学报,2024, 1-16.

ZHANG Chao, ZHAO Hui, ZHANG Zhifeng, et al. A task scheduling method for minimizing completion time in edge collaborative environment[J/OL]. Journal of Xidian University, 2024, 1-16.

1. Ming, Yujun and Supeng Leng. “Collaborative Task Offloading for Vehicular Edge Computing.” 2022 IEEE 22nd International Conference on Communication Technology (ICCT) (2022): 798-803.
2. X. Chen, H. Guo and J. Liu. "Efficient and Trusted Task Offloading in Vehicular Edge Computing Networks," GLOBECOM 2022 - 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022, pp. 5201-5206.
3. M. Gong, Y. Yoo and S. Ahn. "Vehicular Cloud Forming and Task Scheduling for Energy-Efficient Cooperative Computing," in IEEE Access, vol. 11, pp. 3858-3871, 2023.
4. W. Li, N. Zhang, Q. Liu, et al. "Scalable Modulation based Computation Offloading in Vehicular Edge Computing System," 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), Victoria, BC, Canada, 2020, pp. 1-5.
5. S. Pang, N. Wang, M. Wang, et al. "A Smart Network Resource Management System for High Mobility Edge Computing in 5G Internet of Vehicles," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 4, pp. 3179-3191, 1 Oct.-Dec. 2021.
6. 张海波, 卞霞, 徐勇军, 等. 区块链辅助的VANET中车辆声誉管理方法[J]. 西安电子科技大学学报, 2022, 49(4): 49-59.

ZHANG Haibo, BIAN Xia, XU Yongjun, et al. Blockchain-assisted vehicle reputation management method for VANET[J]. Journal of Xidian University, 2022, 49(4): 49-59.

1. BUSHRA JAMIL, HUMAIRA IJAZ, MOHAMMAD SHOJAFAR, et al. 2022. Resource Allocation and Task Scheduling in Fog Computing and Internet of Everything Environments: A Taxonomy, Review, and Future Directions. ACM Comput. Surv. 54, 11s, Article 233 (January 2022), 38 pages.
2. R. Storn, "System design by constraint adaptation and differential evolution", *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 22-34, 1999.
3. RAO, RAVIPUDI VENKATA, et al. “Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems.” *Comput. Aided Des.* 43 (2011): 303-315.
4. SHU,Wanneng ,and LI Yan. “Joint offloading strategy based on quantum particle swarm optimization for MEC-enabled vehicular networks”. Digital Communications and Networks,vol.9, 2023, pp.56-66.