Computer Science and Application 计算机科学与应用, 2025, 15(5), 771-777
Published Online May 2025 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2025.155150

Hans汉斯

车联网中基于改进麻雀搜索算法的任务卸载优化策略

于　勋，严嘉鹏

徐州工程学院信息工程学院(大数据学院)，江苏　徐州

收稿日期：2025年4月25日；录用日期：2025年5月23日；发布日期：2025年5月30日

摘　要

车联网中计算密集型任务对车辆计算能量消耗和网络延迟提出了更高要求。本文提出一种基于改进的麻雀搜索算法(ISSA)的低延迟低能耗的任务卸载优化策略，以应对车联网动态性、异构性和资源受限的挑战。该方法综合考虑任务特性、网络状态和计算资源，建立以最小化延迟、能耗为目标的优化模型，并设计改进的麻雀搜索算法选择最优的卸载策略。仿真结果表明，该方法能有效降低任务处理延迟和能耗，提高资源利用率，为车联网密集型的应用提供支持。

关键词

车联网，任务卸载，延迟感知，改进麻雀搜索算法，资源优化

Optimization Strategy for Task Offloading in Internet of Vehicles Based on Improved Sparrow Search Algorithm

Xun Yu, Jiapeng Yan

School of Information Engineering (School of Big Data), Xuzhou University of Technology, Xuzhou Jiangsu

Received: Apr. 25th, 2025; accepted: May 23rd, 2025; published: May 30th, 2025

Abstract

The computation-intensive tasks in the Internet of Vehicles (IoV) impose higher demands on vehicle computing energy consumption and network latency. This paper proposes a low-latency, low-energy task offloading optimization strategy based on an improved Sparrow Search Algorithm-GA-SSA to address the challenges of dynamicity, heterogeneity, and resource constraints in IoV. The method

文章引用: 于勋, 严嘉鹏. 车联网中基于改进麻雀搜索算法的任务卸载优化策略[J]. 计算机科学与应用, 2025, 15(5): 771-777. DOI: 10.12677/csa.2025.155150

comprehensively considers task characteristics, network state, and computational resources to establish an optimization model aimed at minimizing delay and energy consumption, and designs an improved Sparrow Search Algorithm to select the optimal offloading strategy. Simulation results show that this method can effectively reduce task processing delay and energy consumption, improve resource utilization, and support intensive applications in IoV.

Keywords

IoV, Task Offloading, Delay-Aware, GA-SSA, Resource Optimization

1. 引言

随着无线网络技术和边缘计算的发展，智慧交通系统开始初露头角，如无人驾驶、智慧车载系统、人车对话等。智慧交通系统可通过实时收集和分析交通数据为交管部门提供决策支持，同时也为用户提供安全可靠的智能交通服务[1]。而车联网(Internet of Vehicles, IoV)是智慧交通系统核心，为服务提供商实时收集上传数据，分析车辆当前速度、周围环境等，从而为车辆提供导航和自动驾驶等时延敏感型和计算密集型服务功能提供依据[2] [3]。然而服务过程需要提供大量的实时数据并计算分析。同时车载应用软件的数量规模也海量增长，因此，如何在满足车辆计算服务前提下，快速高效处理海量实时数据，是当前车联网发展继续解决的问题。

在车联网环境下由于车辆自身的计算资源有限，可能会造成无法实时完成计算任务，从而造成车辆导航和自动驾驶等服务延迟过长而导致失败的结果[4]。此外，云服务器距离车辆距离较远，在传输过程中容易造成时延增长，服务质量下降。所以基于边缘服务器的特征，在云服务器和车辆之间部署边缘服务器，能够给车辆用户提供存储和计算资源[5]。车辆用户可以根据当前上下文环境选择将部分的任务卸载到附近的边缘服务器上，从而缓解网络拥塞情况，并且能降低服务延迟[6]。

虽然移动边缘计算提高了车联网的服务响应性能，然而在车联网边缘环境下的任务卸载过程仍存在许多挑战和问题[7]。一是因为边缘服务器的计算能力和存储资源有限，不能在有限时间内处理大量的车辆卸载的任务。二是因为当前车联网环境复杂多变，车辆位置不断变化，而用户服务需求也动态变化，故使得任务卸载问题更加复杂。因此车联网环境下的车辆任务卸载策略问题的研究是当前热点问题。本文主要针对基于多用户车辆的任务卸载场景下的卸载策略问题进行分析研究。

2. 系统建模

2.1. 任务卸载模型

在车辆网场景中，主要包含车辆、路边单元(RSU)、移动边缘计算服务器组成，而一个路边单元和一个 MEC 服务器相连。为了描述方便，后面称为边缘节点。假设在 m 区域内，有多个用户车辆在行驶，该区域内部署 m 个边缘节点，分别记为 $\mathrm{ES}=\{\mathrm{ES}_1,\mathrm{ES}_2,\cdots,\mathrm{ES}_m\}$，在某个时间段内，假设当前有 n 辆车在道路上行驶，记为 $\mathrm{V}=\{\mathrm{V}_1,\mathrm{V}_2,\cdots,\mathrm{V}_n\}$，当前车辆 $\mathrm{V}_i\ (i\in(1,n))$最近的边缘节点为 $\mathrm{ES}_k\ (k\in(1,m))$，假设该边缘节点上的计算能力能够满足周边车辆卸载来的计算任务，每辆车上都有 L 个任务需要计算执行，记为 $T_i=\{T_{i1},T_{i2},\cdots,T_{il}\}$，每辆车的每个任务用 $T_{ij}=\left\{D_{ij},T_{ij}^{\max}\right\}$ 二元组表示，分别表示任务数据大小和任务能

接受的最大时间延迟。

本文研究目标是 n 辆车在某个区域内行驶过程中，找到一个最优的任务卸载策略方案 $X=\{X_1,X_2,\cdots X_n\}$，其中 $X_i=\{x_{i1},x_{i2},\cdots,x_{il}\}$，$i\in(1,n)$，使得所有车辆在当前 m 个区域内的任务延迟和能耗的效应值最小。其中 $x_{ij}\in\{0,1\}$，表示车辆 i 上任务 T_{ij} 卸载的位置。其中 $x_{ij}=0$ 表示任务 T_{ij} 在本地车辆上计算，反之则表示任务 T_{ij} 卸载到边缘服务器上。

2.2. 延迟计算模型

车辆 V_i 在行驶过程中，如果是计算密集型的小型任务直接在本地执行，如果车辆 V_i 上的任务数量较多，且计算量大，则需要把部分任务卸载边缘服务器上 ES_k 进行计算。这里定义一个二进制变量 x_{ij} 表示是否在本地计算，若在本地计算则为 1，否则为 0。

假设边缘节点 ES_k ($k\in(1,m)$)中的各个虚拟机计算性能一致，其中边缘节点的处理能力记为 Ef_k。车辆的处理能力为 Vf_i，D_{ij} 表示数据大小。若任务 T_{ij} 在本地计算，则延迟计算如下：

$$\mathrm{TT}_{ij}=x_{ij}\times\frac{D_{ij}}{Vf_i} \tag{1}$$

若任务卸载到边缘节点 ES_k 上时，任务延迟包含从车辆到边缘节点 ES_k 上的传输延迟、在边缘节点 ES_k 上的计算延迟、计算结果从边缘节点上返回到车辆终端[8]。假设从车辆到边缘节点 ES_k 上的传输速率为 B_{v2e}^k，边缘节点 ES_k 计算能力为 Ef_k，D_{ij}' 表示返回结果数据大小。则任务 T_{ij} 的延迟 TT_{ij} 计算如下：

$$\mathrm{TT}_{ij}=\left(1-x_{ij}\right)\times\left(\frac{D_{ij}}{B_{v2e}^k}+\frac{D_{ij}}{Ef_k}+\frac{D_{ij}'}{B_{v2e}^k}\right) \tag{2}$$

车辆 V_i 上所有任务的延迟时间为

$$\mathrm{TT}_{ij}=\begin{cases}x_{ij}\times\dfrac{D_{ij}}{Vf_i}\\ \left(1-x_{ij}\right)\times\left(\dfrac{D_{ij}}{B_{v2e}^k}+\dfrac{D_{ij}}{Ef_k}+\dfrac{D_{ij}'}{B_{v2e}^k}\right)\end{cases} \tag{3}$$

2.3. 能耗计算模型

车辆 V_i 在行驶过程中，若任务 T_{ij} 在当前边缘服务器上运行，则在本地消耗的能耗计算如下：

$$TE_{ij}=\frac{D_{ij}}{Vf_i}\times PVC_i \tag{4}$$

若任务计算量较大，需要卸载到边缘服务 V_i 上进行计算，则该任务能耗计算如下：

$$\mathrm{TE}_{ij}=\left(\frac{D_{ij}}{B_{v2e}^k}\times PVT_i+\frac{D_{ij}}{Ef_k}\times PEC_k+\frac{D_{ij}'}{B_{v2e}^k}\times PER_k\right) \tag{5}$$

其中 PVC_i、PVT_i、PEC_k 和 PER_k 分别表示车辆 i 的计算功率，车辆 i 的传输功率，边缘服务器 k 的计算功率和传输功率。则

车辆 V_i 上每个任务计算能耗公式如下：

$$\mathrm{TE}_{ij}=\begin{cases}x_{ij}\times\dfrac{D_{ij}}{Vf_i}\times PVC_i\\ \left(1-x_{ij}\right)\times\left(\dfrac{D_{ij}}{B_{v2e}^k}\times PVT_i+\dfrac{D_{ij}}{Ef_k}\times PEC_k+\dfrac{D_{ij}'}{B_{v2e}^k}\times PER_k\right)\end{cases} \tag{6}$$

当前区域内所有车辆任务计算时延和消耗的能耗分别是：

$$T=\sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{l}TT_{ij} \tag{7}$$

$$E=\sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{l}TE_{ij} \tag{8}$$

本文将任务卸载过程抽象为一个多目标优化问题，同时考虑任务完成的时间和能耗。最小化系统中的时间延迟和能耗的多目标优化函数为：

$$\min\ T,\ \min E \tag{9}$$

$$s.t\ TT_{ij}<T_{ij}^{\max}\quad \forall i=1,2,...,n,\ \forall j=1,2,...,l \tag{10}$$

2.4. 适应度函数

任务卸载过程需要考虑当前场景中的延迟和能耗，为了简化计算过程，将公式 9 转换成单目标优化函数，具体如下。

$$\min f=\left(w_1*T+w_2*E\right) \tag{11}$$

当前场景中的问题就转换成一个单目标优化问题，即在时间最大约束下，每个任务执行的延迟和能耗的效应总和最小。针对该问题，本课题采用了改进的麻雀搜索算法进行问题的求解。其中 w_1 和 w_2 分别表示延迟和能耗在效应函数值中所占权重比例。

3. 算法设计

麻雀搜索算法(SSA)是一种新型的群智能优化算法，其参数少、收敛速度块，寻优能力较强。但 SSA 容易陷入局部最优解，具有早熟收敛、参数敏感、动态适应性不足问题。而遗传算法(GA)全局搜索能力较强，本文将 GA 算法和麻雀搜索算法的优点结合起来，提出了一种改进的麻雀算法(GA-SSA)来解决车联网中的任务卸载策略优化问题。GA-SSA 算法首先在初始化阶段用 GA 生成高质量初始种群(迭代次数为 5~10 次)，并根据种群的多样性来自适应调整发现者数量。然后在麻雀位置更新过程中，引入自适应权重因子，动态调整麻雀的搜索步长，提高算法的全局搜索能力。其次 ISSA 算法还加入变异操作：在麻雀位置更新后，对部分麻雀进行变异操作，增加种群的多样性，避免算法陷入局部最优。

基于 GA-SSA 延迟感知任务卸载策略具体步骤如下：

1) 初始化：首先随机初始化麻雀种群，每个麻雀的位置代表一个任务卸载决策方案，利用 GA 的交叉和变异生成多样性高的初始种群，每个个体(染色体)表示一个卸载决策方案，二进制编码表示卸载策略(0 表示本地计算，1 表示卸载到边缘节点上)，避免 SSA 陷入局部最优。

2) 通过预警阈值公式 $ST(t)=0.6-0.3*(t/T_{\max})$ 动态调整发现者的比例。同时设置最大迭代次数。

3) 适应度计算：根据任务特性、网络状态和计算资源，计算每个麻雀的适应度值，即对应任务卸载决策方案的总执行延迟。

4) 更新发现者位置：根据 GA-SSA 的位置更新公式，更新发现者的位置。

5) 更新跟随者位置；根据 GA-SSA 的位置更新公式，更新跟随者的位置。

6) 变异操作：对部分麻雀进行变异操作，增加种群多样性。

7) 判断终止条件：如果达到最大迭代次数，则算法终止，输出最优解；否则，返回步骤 2。

4. 实验结果与分析

本文通过仿真实验对 GA-SSA 算法的总延迟、总能耗的效应值进行评估，并与本地计算(LC)、粒子

群优化算法(PSO)和遗传算法(GA)进行实验对比，实验数据来源于仿真数据。实验仿真环境是基于matlab2023b 软件，Intel(R) Core(TM) i7-13700H，2.40 GHz，内存为 32.00 GB，Windows 11 操作系统。

(1) LC：该方法是将每个车辆上所有任务均放在本地车辆上进行计算处理，不将其卸载到任务边缘服务器上。

(2) PSO：该方法是基于粒子群搜索算法的计算卸载方式，寻找到最优的卸载策略，使得当前时刻函数效应值最小。假设惯性权重为 0.8，学习因子 c_1 和 c_2 的值分别为 1.5 和 1.5，最大的迭代次数为 100。

(3) GA：该方法是基于遗传算法的计算卸载方式，寻找到最优的卸载策略，使得当前时刻函数效应值最小。假设当前交叉概率为 0.8，选择方式是以锦标赛方式。变异概率为 0.05，最大的迭代次数为 100。

(4) GA-SSA：该方法是基于改进的麻雀算法计算卸载方式，寻找到最优的卸载策略，使得当前时刻函数效应值最小。假设最大迭代次数为 100。

假设当前使用的是 5G 网络，传输速率为 100 Mbps，车辆数量 n 取值 10、20、30 三种情况进行实验比较，其中每辆车上任务数量为 10 个，边缘节点数量为 6。公式 11 中的 w_1 和 w_2 分别取 0.6 和 0.4。为了方便计算，我们将实验过程中其他参数进行设置如下表 1 所示。

Table 1. Parameter settings
表 1. 参数设置

参数	取值
车辆数量	10、20、30
边缘节点	8
任务数量	10
任务数据量	15 KB~20 KB
任务最大容忍延迟	100 mS
车辆计算功率	10 w
车辆传输功率	2 W
边缘服务器的计算功率	300 w
边缘服务器的传输功率	2 W
车辆计算能力	500 MIPS
边缘服务器计算能力	3000 MIPS

通过仿真数据，对比四种方法进行实验，通过对多个边缘服务器节点环境中的多车辆多任务的计算卸载决策进行实验分析，旨在寻找良好的卸载决策，使得这些车辆上的任务部分卸载在边缘节点上，从而使得车辆在行驶过程中能获取最低的效应值，这里的效应值是延迟和能耗的加权和。实验计算结果如下图 1 所示和表 2 所示：

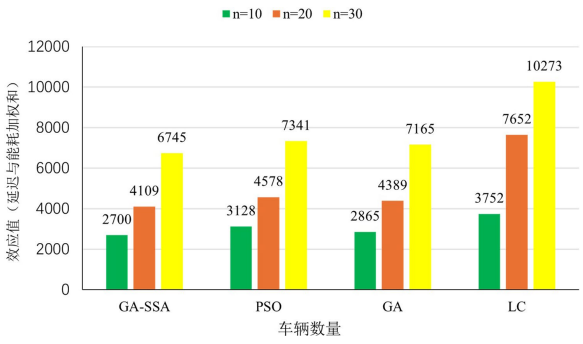

Figure 1. Comparison of effect values of different algorithms
图 1. 不同算法的效应值比较

Table 2. Unloading proportions of different algorithms
表 2. 不同算法的卸载比例

算法	卸载比例	本地计算比例
GA-SSA	45%	55%
PSO	58%	42%
GA	37%	63%
LC	0%	100%

由图 1 可知，LC 算法的效应值最大，即任务均在本地车辆进行处理时延迟和所消耗的时间总效应值最大，而 GA-SSA 算法效应值最小。GA 算法的效应值仅次于 GA-SSA 算法。GA-SSA 算法在任务总执行延迟方面优于传统的 SSA、PSO 和 GA 算法。随着任务数量的增加，GA-SSA 算法的优势更加明显。这是因为 GA-SSA 算法通过引入自适应权重因子和变异操作，提高了算法的全局搜索能力和局部搜索能力，能够更好地找到最优的任务卸载决策方案。

从表 2 可知，本文提出的 GA-SSA 算法卸载比例是 45%，比例适中，能够防止边缘节点过载情况。而在 GA 算法中，卸载比例最低，但是效应值比本文提出的 GA-SSA 算法大。PSO 算法卸载比例最大，边缘节点的负载过高，造成部分任务延迟超标。所以，GA-SSA 算法最适合该问题，因其全局搜索能力强，适应度最低。GA 适用于实时性要求高的场景，如计算速度较快，但适应度稍高。PSO 容易陷入局部最优，需调整参数(如增加多样性)以提升性能。所以本文提出的 GA-SSA 算法比较适合在实时性要求比较高的自动驾驶场景中，延迟较低。并且能够快速响应用户的服务请求。

5. 总结

本文通过改进的 GA-SSA 算法解决车联网中计算密集型任务的能量消耗与网络延迟低的高要求问题。通过构建最小化延迟与能耗加权和的效应值为目标的优化模型，将车联网中的任务卸载策略优化问题转换成一个单目标优化问题。GA-SSA 算法在麻雀位置更新过程中，引入自适应权重因子，动态调整麻雀的搜索步长，提高算法的全局搜索能力。通过实验表明 GA-SSA 算法在解决该问题时具有一定的优势，比 GA、PSO 算法获得的策略更优。车辆网中多边缘节点的任务卸载问题的研究对智能驾驶技术的发展具有实际的指导意义。所以后期课题组还会对车辆网中计算卸载的安全性问题进行研究，这是后期开展的研究工作的主要方向。

基金项目

2024 年江苏省大学生创新训练计划项目 xcx2024184。

参考文献

[1] 武斌, 刘鹏程, 丛佳, 等. 基于缓存模型改进 NSGA-III算法车联网卸载策略[J]. 物联网技术, 2025, 15(1): 108-112, 117.

[2] 崔萌萌, 施静燕, 项昊龙. 基于空地协同的动态车载边缘任务卸载方法[J/OL]. 计算机工程, 1-15. https://doi.org/10.19678/j.issn.1000-3428.69836, 2025-03-17.

[3] 刘建华, 魏金城, 涂晓光. 基于深度强化学习的多方式协同车联网边缘计算任务卸载[J/OL]. 南京信息工程大学学报, 1-20. https://doi.org/10.13878/j.cnki.jnuist.20240523001, 2025-03-17.

[4] 陈琳, 董甲东, 潘凯, 等. 云边端和 D2D 边缘架构下的计算卸载策略综述[J]. 计算机工程与应用, 2023, 59(15): 55-67.

[5] 李文旺, 周浩浩, 邓苏, 等. 面向车辆边缘计算任务卸载的延迟与能耗联合优化方法[J]. 计算机科学, 2024, 51(S2): 658-664.

[6] 王诗，曹大焱，朱笑莹，等. 车联网业务特性模型下卸载反馈策略的设计与评估[J/OL]. 重庆大学学报, 1-13. http://kns.cnki.net/kcms/detail/50.1044.N.20250227.1438.002.html, 2025-03-17.

[7] 唐朝刚，李召，肖硕，等. 一种面向车载边缘计算基于服务缓存的任务协同卸载算法[J/OL]. 计算机学报, 1-13. http://kns.cnki.net/kcms/detail/11.1826.tp.20250211.1316.002.html, 2025-03-17.

[8] 陈学硕，毛玉星，徐宜航，等. 资源受限环境下的智能终端任务卸载方法[J/OL]. 计算机工程与应用, 1-10. http://kns.cnki.net/kcms/detail/11.2127.TP.20241225.1035.004.html, 2025-03-17.