

# 基于GWO-SA的车联网计算任务卸载策略

彭锐, 李银平, 刘旭

(中南民族大学计算机学院, 湖北武汉430074)

**摘要:** 在车联网中, 车辆终端资源有限, 难以满足计算密集型任务的低时延与低能耗需求。为此, 以降低系统总成本、平衡任务处理时延与能耗为目标, 构建“云—边—端”三层协同计算架构, 并提出一种融合灰狼优化与模拟退火机制的混合任务卸载算法GWO-SA。实验结果表明, GWO-SA算法在系统总成本方面仅为随机卸载算法的57.6%, 为最近边缘服务器计算算法的72.3%; 系统总时延相较于随机卸载算法降低了29.6%。在收敛性能方面, GWO-SA算法比二进制混合灰狼优化算法的收敛速度提升了40%, 较自适应粒子群算法提升了50%。GWO-SA算法在时延控制、能耗优化及负载适应性上均优于对照算法, 为车联网环境中的任务卸载提供了一种高效可行的解决方案。

**关键词:** 车联网; 云边协同; 计算卸载; 灰狼优化算法; 模拟退火算法

DOI: 10.11907/rjdk.251702

扫描二维码阅读全文:



中图分类号: TP301.6

文献标识码: A

文章编号: 1672-7800(2025)012-0025-08

## Computing Task Offloading Strategy for Internet of Vehicles Based on GWO-SA

PENG Rui, LI Yinping, LIU Xu

(College of Computer, South-Central Minzu University, Wuhan 430074, China)

**Abstract:** In the Internet of Vehicles, vehicle terminal resources are limited, making it difficult to meet the low latency and low energy consumption requirements of computationally intensive tasks. To this end, with the goal of reducing the total system cost, balancing task processing latency and energy consumption, a three-layer collaborative computing architecture of "cloud edge end" is constructed, and a hybrid task offloading algorithm GWO-SA that integrates grey wolf optimization and simulated annealing mechanism is proposed. The experimental results show that the GWO-SA algorithm has a total system cost that is only 57.6% of the random offloading algorithm and 72.3% of the nearest edge server computing algorithm; The total system latency has been reduced by 29.6% compared to the random offloading algorithm. In terms of convergence performance, the GWO-SA algorithm has improved the convergence speed by 40% compared to the binary hybrid grey wolf optimization algorithm and by 50% compared to the adaptive particle swarm algorithm. The GWO-SA algorithm outperforms the control algorithm in terms of delay control, energy consumption optimization, and load adaptability, providing an efficient and feasible solution for task offloading in the connected vehicle environment.

**Key Words:** Internet of Vehicles; cloud-edge collaboration; computation offloading; grey wolf optimization algorithm; simulated annealing algorithm

## 0 引言

随着信息技术的迅猛发展和智能交通理念的广泛普及, 车联网成为推动出行环境向高效、安全、智能化转型的

关键力量。其应用场景持续扩展, 已覆盖实时导航、自动驾驶、车载娱乐等多个新兴领域, 显著提升了车辆运行效率与用户驾乘体验<sup>[1-3]</sup>。然而, 该类应用通常生成大量计算密集、延迟敏感且数据吞吐量高的任务。车辆终端受限于自身计算能力、存储空间和通信资源, 导致复杂任务难

收稿日期: 2025-10-14

基金项目: 国家自然科学基金面上项目(62373382); 湖北省自然科学基金一般面上项目(2024AFB704); 中央高校基本科研业务费专项重点项目(CZZ23003)

作者简介: 彭锐(2001-), 女, CCF会员, 中南民族大学计算机学院硕士研究生, 研究方向为云边协同; 李银平(1997-), 男, 中南民族大学计算机学院硕士研究生, 研究方向为自然语言、程序验证; 刘旭(1997-), 男, 中南民族大学计算机学院硕士研究生, 研究方向为云边协同。

以在本地高效执行<sup>[4]</sup>。

为应对上述挑战,云计算技术被引入车联网系统,承担多数计算密集型任务的远程处理工作。然而,完全依赖云端的方式存在明显不足,尤其是数据传输距离较长而引起的高延迟与带宽成本问题,使其难以满足自动驾驶、环境实时感知等对响应时间要求极高的业务场景的需求<sup>[5]</sup>。因此,移动边缘计算(Mobile Edge Computing, MEC)被纳入车联网体系,其核心思想是为将云计算能力自网络核心层下移,部署到更接近车辆与用户的网络边缘侧,从而为车辆提供就近的计算支持<sup>[6]</sup>。

## 1 相关工作

车联网任务卸载是指将车辆上产生的计算密集型或时间敏感型任务,通过无线通信技术,部分或全部转移到边缘计算节点(如基站、其他车辆或云端服务器)进行处理的技术。车联网任务卸载策略研究已围绕不同优化目标展开。早期工作多聚焦于单一性能指标优化。例如,Tran-dang 等<sup>[7]</sup>采用粒子群优化算法实现协同任务卸载,旨在降低系统平均延迟。Cui 等<sup>[8]</sup>设计了多车辆智能协同计算策略,利用双深度 Q 网络来最小化车联网总时延。然而,以上方法在追求低时延的同时往往伴随较高的能耗成本。另一研究方向则关注能耗优化,尤其适用于能量受限的车载场景。例如,Chouikhi 等<sup>[9]</sup>提出基于深度强化学习的计算卸载方法,以降低长期能耗并提升任务在容忍期限前的完成率。Xia 等<sup>[10]</sup>结合李雅普诺夫理论与博弈论,设计分布式卸载策略以减少通信开销。尽管这些方法在节能方面表现良好,但难以满足车辆的实时响应需求。

为兼顾时延与能耗,部分学者转向多目标优化研究。例如,杨超等<sup>[11]</sup>构建了以最小化任务计算时延与 MEC 资源服务费为目标的系统模型,并融合层次分析法与麻雀搜索算法,实现了任务优先级划分与服务器负载均衡。Zhu 等<sup>[12]</sup>针对智能交通边缘计算中的多用户任务卸载问题,建立了考虑时延、能耗与负载均衡的多目标优化模型,并提出一种自适应多目标算法 E-NSGA-III。该算法在 NSGA-III 的基础上引入动态交叉与变异策略,结合参考点选择机制,有效提升了搜索效率。Xu 等<sup>[13]</sup>面向 5G 车联网环境,提出混合路侧单元(Road Side Unit, RSU)与宏基站的边缘计算卸载框架。该框架采用基于分解的多目标进化算法生成候选策略,并利用 TOPSIS (Technique for Order Preference by Similarity to an Ideal Solution) 与多准则决策进行最优解选取,可在最小化传输时延的同时最大化边缘系统资源利用率。

与现有工作相比,本文对任务卸载过程中产生的时延、能耗等多目标进行了优化。具体来说,在车联网场景中建立云边协同计算模型,并将车辆任务卸载产生的总时延和总能耗作为目标函数,运用融合灰狼优化算法(Grey

Wolf Optimizer, GWO)与模拟退火机制(Simulated Annealing, SA)的混合任务卸载算法 GWO-SA 进行多目标优化,以实现用户总成本最小化的最终目标。本研究的主要贡献点为:①提出一种车联网场景下边缘云与车载服务终端三层架构的协同计算系统模型,进而提出任务卸载策略问题;②提出 GWO-SA 算法,其具有更强的全局寻优能力,可对云边协同计算系统的卸载时延、能耗进行多目标优化;③提出与任务优先级和计算卸载节点预测相结合的端边协同计算卸载策略,综合考虑了车辆的运动特性和任务时延的敏感性,显著提高了系统的卸载效用,降低了系统计算卸载的时延和能耗。

## 2 系统模型

### 2.1 网络模型

本文针对云边协同架构下的车联网计算卸载问题展开研究,所构建的网络模型如图 1 所示。该模型采用“云—边—端”三级层次化设计,由顶层至底层依次为云计算中心层、边缘服务器层与车辆终端层。云计算中心层作为系统顶层,部署有具备强大存储与计算能力的云计算中心,能够处理全局性、非实时的大规模数据分析与模型训练任务,并为下层系统提供协同调度支持。边缘服务器层作为中间层,由部署于网络边缘的 MEC 服务器和 RSU 组成,两者通过高速有线链路来实现稳定、低时延的连接,负责对来自车辆终端的计算任务进行快速响应、局部处理与资源协调,有效缓解云端回传压力。车辆终端层作为系统底层,由各类智能网联车辆构成,这些车辆搭载多种感知设备与计算模块,能够实时采集行车环境数据并生成各类计算任务,是系统中最基本的数据来源与计算需求发起方。该架构通过合理分布资源与功能,有效支撑起高效、可靠的车载任务卸载与资源协同调度。

在整个架构中,云计算中心作为全局资源调度与复杂

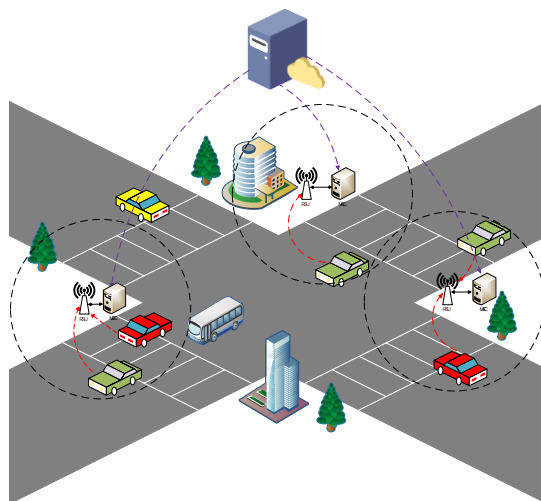


Fig. 1 Network model

图 1 网络模型

计算任务的核心节点,凭借其强大的计算能力,承担着大规模数据的汇聚与深度分析工作<sup>[14]</sup>。在边缘服务器层中,沿道路分布式部署的 RSU 与其连接的 MEC 服务器协同运作,共同构建起具备一定自治能力的边缘服务单元。每个单元能够独立管理其覆盖范围内的通信与计算资源,实现对区域内车辆的低时延、高带宽服务支持。车辆与 RSU 之间依托无线通信技术建立连接,从而实现任务数据的实时上传、卸载决策的执行以及处理结果的快速回传。依托上述 3 层架构在资源、数据与功能上的深度融合与协同运行,系统能有效满足车联网场景中对延迟敏感、计算密集型业务的多样化需求,为实现高效、可靠且可扩展的车载计算卸载奠定了坚实的结构基础。

## 2.2 任务模型

将车辆和 MEC 服务器集合分别定义为  $I = \{1, 2, \dots, n\}$  和  $M = \{1, 2, \dots, m\}$ 。在每个仿真时间步长内,设定每辆车至多产生 1 个计算任务,该任务可以在本地执行、卸载到边缘计算节点,或进一步卸载到云服务器上执行。车辆在每一时间片内生成的任务服从伯努利分布。车辆以一定概率动态生成计算任务  $task_i$  可形式化定义为三元组  $task_i = (D_i, C_i, T_i^{\max})$ ,其中  $D_i$  表示任务  $task_i$  的输入数据量,  $C_i$  表示任务  $task_i$  所需的计算资源,  $T_i^{\max}$  表示任务的最大可容忍延迟。

假设采用二进制决策变量  $O_i = (a_i, b_i, c_i)$  来表示任务  $task_i$  的卸载位置,且满足  $a_i + b_i + c_i = 1$ 。具体而言,  $a_i = 1$  表示任务在本地执行,  $b_i = 1$  表示任务卸载到边缘服务器执行,  $c_i = 1$  表示任务卸载到云计算中心执行。

## 2.3 通信模型

在车载边缘计算系统中,当车辆终端决定将计算任务卸载至边缘节点时,任务包含的输入数据需首先通过无线信道传输至与之通信的 RSU。在假设该过程不受信道干扰影响的条件下,车辆终端到对应 RSU 的上行传输速率可依据香农定理表述为<sup>[15]</sup>:

$$R_{i,m} = B \cdot \log_2 \left( 1 + \frac{P_i \cdot H_{i,m}}{\sigma^2} \right) \quad (1)$$

式中:  $B$  为信道带宽;  $P_i$  为车辆  $i$  的固定发射功率;  $H_{i,m}$  为车辆  $i$  与 RSU  $m$  之间的信道增益;  $\sigma^2$  为高斯白噪声的功率。

## 2.4 计算模型

### 2.4.1 本地计算模型

当在本地车辆  $i$  上执行计算任务时,处理过程没有经历任何无线数据传输阶段,只需考虑计算过程的时间开销与本地能耗。因此,其能耗开销主要来源于车辆自身嵌入式计算单元的任务处理过程,具体体现在计算时延与计算能耗两大核心指标。计算时延  $T_i^{\text{local}}$  和计算能耗  $E_i^{\text{local}}$  的计算公式分别为:

$$T_i^{\text{local}} = \frac{C_i}{f_i^{\text{local}}} \quad (2)$$

$$E_i^{\text{local}} = \kappa \cdot (f_i^{\text{local}})^2 \cdot C_i \quad (3)$$

式中:  $f_i^{\text{local}}$  为车辆  $i$  的本地计算能力;  $\kappa$  为车辆有效开关电容系数,该系数与芯片结构相关,并由车辆 CPU 硬件工艺决定。

### 2.4.2 边缘计算模型

将任务卸载至边缘节点执行时,每个 RSU 并行处理不同的任务,而与之相关联的 MEC 服务器计算资源有限。假设 MEC 服务器的计算资源总量为  $F_m^{\text{mec}}$ ,任务被卸载到 MEC 服务器  $m$  上执行时,时延包括传输时延与计算时延,能耗则包括 MEC 服务器的计算能耗与任务传输能耗。因此,计算时延  $T_i^{\text{mec}}$  和计算能耗  $E_i^{\text{mec}}$  的计算公式分别为:

$$T_i^{\text{mec}} = \frac{D_i}{R_{i,m}} + \frac{C_i}{f_m^{\text{mec}}} \quad (4)$$

$$E_i^{\text{mec}} = P_i \cdot \frac{D_i}{R_{i,m}} + \lambda \cdot (f_m^{\text{mec}})^2 \cdot C_i \quad (5)$$

式中:  $R_{i,m}$  为车辆到边缘节点的传输速率;  $f_m^{\text{mec}}$  为 MEC 服务器的计算能力;  $P_i$  为车辆传输功率;  $\lambda$  为 MEC 服务器的能耗系数。MEC 服务器需满足能量约束,即分配给该服务器的所有任务总计算能耗不能超过其当前剩余能量  $E_m^{\text{remain}}$ 。

### 2.4.3 云计算卸载模型

在云计算模式下,任务会被卸载至远程云数据中心执行。云端部署位置较远,数据传输需要经过边缘计算节点,这使得任务执行时延包括车辆端传输到边缘节点的时延、边缘节点传输到云服务器的时延以及云端计算时延。从能耗角度看,云端数据中心的能耗通常由服务提供商承担,用户侧主要关注任务传输过程中产生的能耗成本<sup>[16]</sup>。

云任务的总时延  $T_i^{\text{cloud}}$  计算公式为:

$$T_i^{\text{cloud}} = \frac{C_i}{f^{\text{cloud}}} + \frac{D_i}{R_{i,m}} + \frac{D_i}{R_{m,\text{cloud}}} \quad (6)$$

式中:  $f^{\text{cloud}}$  为云服务器的计算能力;  $R_{m,\text{cloud}}$  为边缘节点到云服务器的传输速率。

云任务的总能耗  $E_i^{\text{cloud}}$  计算公式为:

$$E_i^{\text{cloud}} = P_i \cdot \left( \frac{D_i}{R_{i,m}} + \frac{D_i}{R_{m,\text{cloud}}} \right) \quad (7)$$

### 2.4.4 总时延与总能耗

综合上述 3 种计算模型,任务  $task_i$  的总完成时延和总能耗可根据其卸载决策  $O_i = (a_i, b_i, c_i)$  统一表示为:

$$\begin{aligned} T_i^{\text{total}} &= a_i \cdot T_i^{\text{local}} + b_i \cdot T_i^{\text{mec}} + c_i \cdot T_i^{\text{cloud}} \\ E_i^{\text{total}} &= a_i \cdot E_i^{\text{local}} + b_i \cdot E_i^{\text{mec}} + c_i \cdot E_i^{\text{cloud}} \end{aligned} \quad (8)$$

## 2.5 问题定义

在车联网场景中,多车辆产生的计算任务需进行卸载决策。该问题的核心在于对任务处理时延与设备能耗两个相互制约的指标进行有效权衡。为量化这一权衡,本研究以最小化系统总成本为优化目标,将该目标函数定义为所有车辆任务总延迟与总能耗的加权和。该目标函数的形式化表述为:

$$\begin{cases}
Q1: \min \left( \sum_{i \in I} \omega_i \cdot T_i^{total} + \sum_{i \in I} \omega_e \cdot E_i^{total} \right) \\
s.t. \\
C_1: a_i + b_i + c_i = 1, \forall i \in I \\
C_2: T_i \leq T_i^{max}, \forall i \in I \\
C_3: \sum_{i \in \Psi_m} E_i^{comp, mec} \leq E_m^{remain}, \forall m \in M \\
C_4: \sum_{i \in \Psi_m} C_i \leq F_m^{mec}, \forall m \in M \\
C_5: 0 \leq \omega_i, \omega_e \leq 1 \\
C_6: \omega_i + \omega_e = 1
\end{cases} \quad (9)$$

式中:  $C_1$  表示每个任务执行位置唯一;  $C_2$  表示任务必须在规定时间内完成;  $C_3$  表示 MEC 服务器的能量约束;  $C_4$  表示 MEC 服务器的计算资源约束;  $C_5$  和  $C_6$  表示延迟权重  $\omega_i$  和能耗权重  $\omega_e$  的值在 0~1 之间, 且二者之和为 1。

### 3 算法设计

#### 3.1 标准 GWO

GWO 是一种基于灰狼社会层级与协作狩猎行为的群体智能优化算法<sup>[17]</sup>。该算法将狼群中的  $\alpha$ 、 $\beta$  和  $\delta$  狼分别定义为当前种群中的最优解、次优解与第三优解, 通过模拟其引导群体搜索的机制, 展现出良好的全局探索特性。然而, 标准 GWO 算法在迭代后期因种群多样性下降而易陷入局部最优, 且局部开采能力存在不足。

GWO 算法的核心在于个体位置根据  $\alpha$ 、 $\beta$  和  $\delta$  狼的位置进行协同更新。首先, 每个个体 ( $\omega$  狼) 与 3 头领头狼的相对距离可通过公式 (10) 计算:

$$\begin{cases}
\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \\
\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \\
\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|
\end{cases} \quad (10)$$

式中:  $\vec{X}$  为当前个体位置向量;  $\vec{X}_\alpha$ 、 $\vec{X}_\beta$  和  $\vec{X}_\delta$  分别为当前  $\alpha$ 、 $\beta$  和  $\delta$  狼的位置向量; 系数向量  $\vec{C} = 2 \cdot \vec{r}_2$ ,  $\vec{r}_2$  为  $[0, 1]$  范围内的随机向量。

基于上述距离, 可确定个体朝向各领头狼的候选移动方向:

$$\begin{cases}
\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \\
\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \\
\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta
\end{cases} \quad (11)$$

式中: 收敛因子向量  $\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}$ ,  $\vec{r}_1$  为  $[0, 1]$  范围内的随机向量; 核心参数  $\vec{a}$  作为控制全局探索与局部开发的关键因子, 其值随迭代次数线性递减。其更新公式为:

$$a = 2 \times \left( 1 - \frac{t}{T_{max}} \right) \quad (12)$$

式中:  $t$  为当前迭代次数;  $T_{max}$  为最大迭代次数。参数  $a$  的递减控制着算法的搜索行为: 当  $|\vec{A}| > 1$  时, 算法倾向于全局探索; 当  $|\vec{A}| < 1$  时, 算法倾向于局部开发。

最终, 个体通过综合 3 类候选方向来更新其位置。更新公式为:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (13)$$

虽然 GWO 算法结构简单、参数较少, 但其依赖线性递减的收敛因子, 在优化后期难以维持种群多样性, 易导致早熟收敛。为此, 本文引入 SA 机制对其进行增强, 以提升其跳出局部最优的能力与收敛精度。

#### 3.2 GWO-SA 混合算法

##### 3.2.1 算法框架设计

GWO-SA 混合算法框架集成了 GWO 的群体智能与 SA 的局部搜索能力, 通过多策略协同机制来保持种群多样性、提高收敛精度, 并平衡全局探索与局部开发。该算法的核心思想为利用 GWO 的  $\alpha$ 、 $\beta$ 、 $\delta$  狼引导机制进行全局方向指导; 引入 SA 的 Metropolis 接受准则增强算法跳出局部最优的能力; 引入 Lévy 飞行策略来提升全局探索效率, 并设计动态参数调整策略, 使算法更好地适应车联网环境特性。

##### 3.2.2 适应度函数

算法的优化目标为最小化任务总时延与总能耗加权构成的系统总成本。灰狼种群中的每个个体代表一个卸载方案, 其适应度值定义为:

$$f(\vec{X}) = \sum_{i \in I} \omega_i \cdot T_i^{total} + \sum_{i \in I} \omega_e \cdot E_i^{total} \quad (14)$$

##### 3.2.3 Metropolis 接受准则

Metropolis 接受准则以一定的概率接受一个比当前解更差的解, 从而帮助算法跳出局部最优解<sup>[18]</sup>。用公式表示为:

$$\Delta E = f(\vec{X}_{new}) - f(\vec{X}_{current}) \quad (15)$$

$$P_{accept} = \begin{cases} 1 & \text{if } \Delta E < 0 \\ \exp\left(-\frac{\Delta E}{T}\right) & \text{if } \Delta E \geq 0 \end{cases} \quad (16)$$

$$T_{current} = T_0 \times (\alpha_T + 0.06 \times \frac{t}{T_{max}})^t \quad (17)$$

式中:  $\Delta E$  为新旧解的能量差。当  $\Delta E < 0$ , 即新解更优时, 系统总是接受新解, 此时  $P_{accept} = 1$ ; 当  $\Delta E \geq 0$  时, 系统以概率  $\exp(-\Delta E/T)$  接受新解。该机制允许解空间出现暂时性退化。温度参数  $T$  随迭代过程逐步递减, 在早期高温阶段鼓励全局探索, 在后期低温阶段侧重局部开发。该温度调度机制可有效避免算法陷入局部最优状态。

##### 3.2.4 Lévy 飞行机制

Lévy 飞行是一种随机游走模式, 其特征是偶尔出现长距离跳跃的步长分布, 能有效增强优化算法的全局探索能力<sup>[19]</sup>。用公式表示为:

$$\vec{X}_i^{new} = \vec{X}_i^{old} + \vartheta \cdot \text{Levy}(\tau) \cdot r \quad (18)$$

式中:  $\vartheta$  为步长缩放因子;  $\text{Levy}(\tau)$  为随机数。

##### 3.2.5 边界修复策略

在 GWO-SA 算法的迭代优化过程中, 边界修复策略通

过反射机制与随机重置,确保灰狼位置更新后始终处于解空间可行域内<sup>[20]</sup>。当个体越过边界时,首先尝试反射回区间,若仍越界则在该维度随机重置,以避免产生无效解,提升算法的收敛效率与稳定性。该策略用公式表示为:

$$\vec{X}_i^{new}(d) = \begin{cases} LB(d) + \text{rand} \cdot (UB(d) - LB(d)) & \text{if } \vec{X}_i^{new}(d) < LB(d) \\ UB(d) - \text{rand} \cdot (UB(d) - LB(d)) & \text{if } \vec{X}_i^{new}(d) > UB(d) \\ \vec{X}_i^{new}(d) & \text{otherwise} \end{cases} \quad (19)$$

式中:rand 为在[0,1]内均匀分布的随机数;LB(d)和UB(d)分别为第d维变量的上界与下界。该策略通过随机重置方式确保解向量的每个维度均满足边界约束,从而保证了解的有效性,提升了算法在复杂解空间中的稳健寻优能力。

### 3.2.6 动态参数调整策略

动态参数调整策略通过线性递减收敛因子 $\alpha$ 与控制温度T指数,实现算法从全局探索到局部开发的自适应过渡;同时基于种群多样性动态调节变异概率,平衡搜索效率与收敛精度<sup>[21]</sup>。收敛因子的更新公式为:

$$a = a_{\text{initial}} - (a_{\text{initial}} - a_{\text{final}}) \cdot \frac{\text{gen}}{G} \quad (20)$$

式中:gen 为当前迭代次数;G 为最大迭代次数。

综上所述,GWO-SA 算法的伪代码为:

**输入:** 车辆数量  $I$ , 最大迭代次数  $G$ , 种群规模  $NP$

**输出:** 计算卸载决策  $O^*$ , 最优适应度值  $F^*$

1 初始灰狼位置  $X = \{\vec{X}_1, \vec{X}_2, \dots, \vec{X}_{NP}\}$

2 初始化领袖狼群  $\alpha, \beta, \delta$ , 温度  $T_0$ , 收敛因子  $a_0$

3 FOR gen = 1 TO G DO

4 根据公式(20)更新收敛因子  $a$

5 根据公式(15)更新退火温度  $T$

6 FOR  $i = 1$  TO  $NP$  DO

7 IF  $\text{rand}() < 0.3$  THEN

8 根据公式(17)使用 Lévy 飞行更新位置  $\vec{X}_i^{new}$

9 ELSE

10 根据公式(10)-(13)使用标准 GWO 机制更新位置  $\vec{X}_i^{new}$

11 END IF

12 根据公式(18)对  $\vec{X}_i^{new}$  进行边界修复

13 根据公式(14)计算新位置适应度  $f(\vec{X}_i^{new})$

14 根据模拟退火准则公式(15)使用模拟退火准则接受新解  $\vec{X}_i^{new}$

15 END FOR

16  $X^{elite} \leftarrow \text{select\_top\_10\%}(X, F)$

17  $X \leftarrow X^{elite} + \text{random\_new\_10\%}(NP)$

18 更新领袖狼群  $\alpha, \beta, \delta$

19 记录当前最优适应度  $F(\text{gen})$

20 END FOR

21 将领袖狼  $\alpha$  位置解码为最优卸载决策  $O^*$

22  $F^* = f(\vec{X}_\alpha)$

## 4 实验方法与结果分析

### 4.1 实验环境与参数设置

本文实验在一台配备有 AMD Ryzen 7 7840HS 处理器与 32 GB 内存的计算机上进行,在 Windows11 操作系统以及 MATLAB R2023b 平台下完成了仿真环境的搭建与测试。实验参数配置如表 1 所示。

Table 1 Experimental parameter setting

表 1 实验参数配置

| 参数名称                                     | 参数值     |
|------------------------------------------|---------|
| RSU 的通信覆盖范围 $R/\text{m}$                 | 800     |
| 车辆速度 $v/(\text{km} \cdot \text{h}^{-1})$ | 30      |
| 本地车辆计算能力 $f^{local}/\text{GHz}$          | 1       |
| MEC 服务器的计算能力 $f^{mec}/\text{GHz}$        | 30      |
| 云服务器的计算能力 $f^{cloud}/\text{GHz}$         | 100     |
| MEC 服务器计算总资源量 $f^{MEC}/\text{GHz}$       | 40      |
| 车辆传输功率/W                                 | 0.5     |
| 信道带宽/MHz                                 | 40      |
| 高斯信道白噪声/W                                | 2.0e-13 |
| 最大计算时延 $T^{MAX}/\text{ms}$               | 200~500 |

### 4.2 对照方法

采用以下对照方法与本文方法进行比较:①最近边缘服务器计算(Nearby Mobile Edge Computing, NMEC)。该方法将计算任务卸载至地理位置上最近的边缘服务器进行处理,依据距离最优原则选取计算节点,充分借助边缘计算的低延迟特性;②随机卸载计算(Random Offloading Algorithm, Random)。该方法将所有计算任务随机分配至本地车辆、MEC 服务器的任一节点执行,不考虑各节点的实际负载与性能差异;③二进制混合灰狼优化算法(Binary Hybrid Grey Wolf Optimizer, BHGWO)<sup>[22]</sup>。该方法结合粒子群的全局搜索与 GWO 算法的局部开发能力,采用二进制位置更新机制,适用于混合整数优化问题,在 MEC 卸载中能有效降低系统开销并提升效率;④自适应粒子群算法(Adaptive Particle Swarm Optimization, APSO)<sup>[23]</sup>。该方法在标准粒子群算法中引入自适应惯性权重,通过在迭代前期增强全局探索、后期加强局部开发来避免早熟收敛并提升优化精度,主要适用于连续优化问题。

### 4.3 实验结果与分析

图 2 展示了 GWO-SA、BHGWO 和 APSO 3 种算法的收敛性能比较情况。三者采用相同的适应度函数与编码方式,最大迭代次数均设定为 200 次。随着迭代次数的增加,各算法系统总成本均呈下降趋势,并在 50 次迭代后普遍收敛至较优解。在收敛效率方面,GWO-SA 算法表现最佳,仅需 30 次迭代即可稳定在最优解附近;BHGWO 与 APSO 算法则分别需要 50 次与 70 次迭代。APSO 算法在收敛过程中目标函数值波动较大,表明其更易陷入局部最优。

相较之下,GWO-SA算法的收敛速度比BHGWO和APSO算法分别提升了40%和50%。

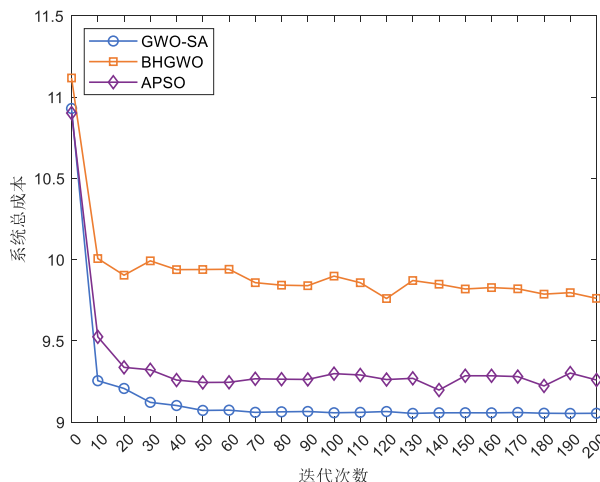


Fig. 2 Comparison of convergence performance of different algorithms

图2 不同算法收敛性能比较

图3反映了任务输入数据量在[400,1 200]KB区间变化时对系统总成本的影响情况。可以看出,随着任务输入数据量的增大,各算法成本因传输延迟与能耗上升而增加。具体来说,GWO-SA算法始终维持着最优的成本控制性能,其系统总成本远低于其他算法。当任务输入数据量达到1 200 KB时,其系统总成本仅为Random算法与NMEC算法的57.6%与72.3%;系统总成本增长速率也仅为Random算法的69.3%,显示出对任务负载变化的强适应能力。相比之下,Random算法成本最高且增长最快,稳定性最差;BHGWO与NMEC算法性能居中,且BHGWO性能优于NMEC;APSO算法性能虽优于前3种算法,但仍次于GWO-SA。

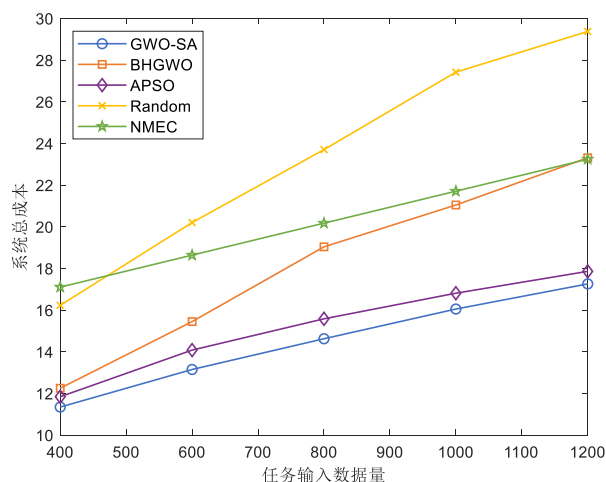


Fig. 3 Influence of the amount of task input data on total system cost

图3 任务输入数据量对系统总成本的影响

图4展示了车辆数量在[10,90]区间内变化时对系统总时延的影响。可以看出,随着车辆数量的增加,所有算法的系统总时延均呈现上升趋势,这主要源于两方面因

素:一是车辆密度增加导致无线信道竞争加剧,上行传输速率下降,显著提高了任务传输时延;二是边缘服务器面临的队列长度随着请求增加而增长,直接延长了任务排队等待时间。GWO-SA算法在该过程中将系统总时延控制在最低水平。当车辆数为90时,其系统总时延分别为Random、NMEC、BHGWO和APSO算法的70.4%、41.2%、84.0%与87.0%。NMEC算法对车辆数量变化最为敏感,时延上升最为显著;Random算法次之;BHGWO与APSO算法的上升趋势相对平缓。综合来看,GWO-SA算法能更有效地控制系统总时延,在车辆密集场景下具备明显优势。

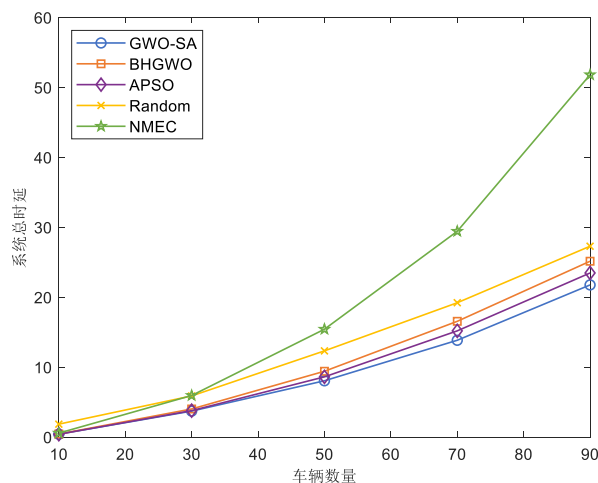


Fig. 4 Influence of vehicle numbers on total system delay

图4 车辆数量对系统总时延的影响

图5展示了车辆数量在[10,90]区间内变化时对系统总成本的影响。可以看出,随着车辆数量的增加,任务处理时延和能耗相应上升,导致各算法的系统总成本持续增长。其中,GWO-SA算法始终保持最低的系统总成本,且增长幅度最为平缓。以车辆数为90的场景为例,GWO-SA算法的系统总成本为24,仅为BHGWO算法的35.8%,NMEC算法的43.6%,相较于Random算法和APSO算法同样展现出显著优势。各算法性能差异主要源于资源调度机制的不同。BHGWO算法在车辆增多时适应性不足,导致成本急剧上升;Random算法由于缺乏规则化决策逻辑,在多车环境下资源竞争矛盾突出;NMEC算法过度依赖边缘侧资源,容易在高负载时出现资源拥堵;APSO算法虽优于前述3种算法,但仍未达到GWO-SA的性能水平。当车辆数超过30后,多数算法的成本曲线斜率明显增大,而GWO-SA算法通过高效的资源协调与负载均衡机制,维持了轻微的成本增长趋势,显示出其在多车辆高负载场景下的出色调度能力。

图6展示了任务计算数据量在[500,1 300]Megacycles范围内变化时,各算法总成本的变化情况。可以看出,随着任务计算量的增加,所有算法的系统总成本均呈现上升趋势,这主要源于计算量增加导致的处理时延延长与资源消耗上升。然而,各算法对计算量增长的响应特性存在显著差异。具体而言,Random算法的总成本增长幅度最为

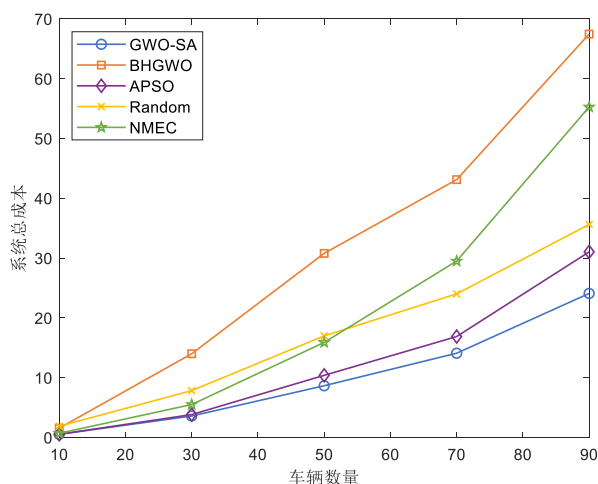


Fig. 5 Influence of vehicle numbers on total system cost

图 5 车辆数量对系统总成本的影响

突出,始终维持在最高水平;APSO 算法的成本增长幅度仅次于前者,整体高于 BHGWO 算法与 GWO-SA 算法;BHGWO 算法虽然成本增长相对平缓,但仍高于 GWO-SA 算法。值得注意的是,GWO-SA 算法在整个任务量区间内始终保持着最低的系统总成本,且增长幅度最为平缓,展现出最优的综合性能。

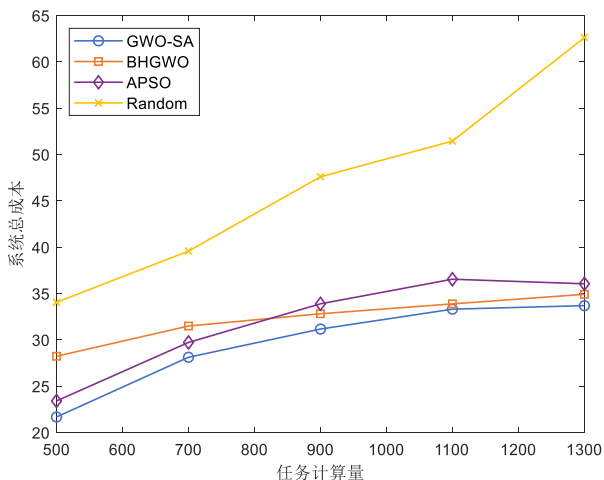


Fig. 6 Influence of the amount of task calculation data on total system cost

图 6 任务计算量对系统总成本的影响

图 7 展示了延迟权重在  $[0.1, 0.9]$  区间内变化时各算法的系统总成本。可以看出,随着延迟权重的增加,所有算法的系统总成本均呈现上升趋势,这反映了时延优化与能耗节约之间的内在权衡关系。在整个权重变化区间内,GWO-SA 算法在大多数配置下均表现出最优的成本控制能力。当延迟权重设置为 0.1 时,系统以能耗优化为主要目标,GWO-SA 在满足基本时延约束的前提下实现了最优能效。随着延迟权重的逐步增加,各算法成本均有所上升,其中 GWO-SA 的增长幅度最小,相较于 NMEC 和 Random 等算法优势显著。在延迟权重达到 0.9 的时延敏感场景下,系统重点优化时延性能。GWO-SA 通过有效的资源

分配与任务调度机制,在严格时延约束下仍保持着最低系统总成本。该算法能够依据不同优化目标的权重配置自适应地平衡时延与能耗,展现出卓越的多目标协调能力。在时延敏感型与能耗敏感型等多样化应用场景中,该算法均能提供接近最优的系统解决方案,展现出适应车联网复杂服务质量需求的强大能力。

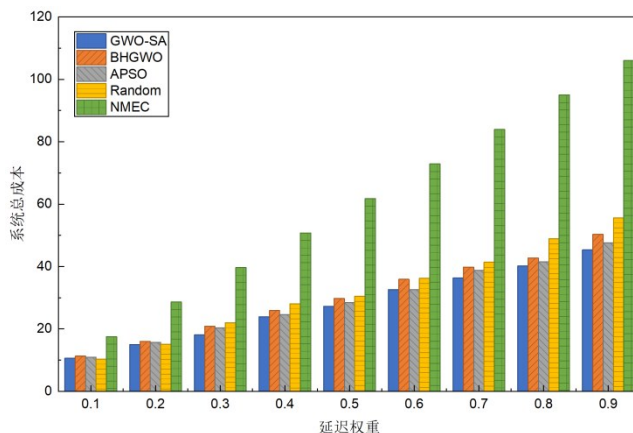


Fig. 7 Influence of latency preference coefficient on total system cost

图 7 延迟权重对系统总成本的影响

## 5 结语

本文针对车联网中计算任务的高效卸载问题,提出 GWO-SA 混合卸载策略,并通过构建“云—边—端”三级系统模型,综合考虑时延、能耗与负载均衡等多重约束条件,建立了以系统总成本最小化为目标的优化问题。该算法引入多策略位置更新机制、模拟退火接受准则与精英保留策略,有效提高了求解质量与算法稳定性。仿真实验结果表明,该算法在系统总成本与任务处理时延方面均优于对照方法,展现出良好的综合性能。

本研究仍存在若干局限性。例如,算法参数设置主要依赖经验性调整,缺乏完全自适应的优化机制;仿真环境尚未考虑极端网络波动或恶意攻击等复杂场景;模型对任务间的依赖关系与多用户的优先级调度描述仍不够充分等。针对上述问题,未来工作可从多个维度展开,包括完善参数自适应调节机制以增强算法在动态环境下的鲁棒性,拓展模型以支持任务依赖关系及满足差异化服务质量需求,推进在真实边缘计算平台中的部署验证工作,以及研究在安全威胁下的可靠卸载机制,从而更加有效地应对车联网环境中的潜在安全挑战。

## 参考文献:

- [1] XING L, JIA X, GAO J, et al. A location privacy protection algorithm based on double K-anonymity in the social Internet of Vehicles [J]. IEEE Communications Letters, 2021, 25(10): 3199–3203.
- [2] LIU C, CHEN Y, LIU M, et al. Using eye gaze to enhance generalization of imitation networks to unseen environments [J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(5): 2066–2074.

- [3] LI Z T. Research and analysis of automotive Ethernet[J]. Auto Electric Parts, 2018(3): 9-12.  
李志涛. 车载以太网的研究与分析[J]. 汽车电器, 2018(3): 9-12.
- [4] LIU F, CUI Y, MASOUROS C, et al. Integrated sensing and communications: toward dual-functional wireless networks for 6G and beyond[J]. IEEE Journal on Selected Areas in Communications, 2022, 40(6): 1728-1767.
- [5] HONG J L, GAO B Z, DONG S Y, et al. Key issues and research advances in energy-saving optimization for intelligent and connected vehicles[J]. China Journal of Highway and Transport, 2021, 34(11): 306-334.  
洪金龙, 高炳钊, 董世营, 等. 智能网联汽车节能优化关键问题与研究进展[J]. 中国公路学报, 2021, 34(11): 306-334.
- [6] VAN TRUONG T, NAYYAR A, BILAL M, et al. System design and optimization of mobile edge computing in the NOMA wireless tactile Internet of Things network[J]. Alexandria Engineering Journal, 2023, 73: 737-749.
- [7] TRAN-DANG H, KIM D. FRATO: fog resource based adaptive task offloading for delay-minimizing IoT service provisioning[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(10): 2491-2508.
- [8] CUI Y, DU L, HE P, et al. Multi-vehicle intelligent collaborative computing strategy for Internet of Vehicles[C]//2022 IEEE Wireless Communications and Networking Conference, 2022: 1647-1652.
- [9] CHOUIKHI S, ESSEGHIR M, MERGHEM-BOULAHIA L. Energy-efficient computation offloading based on multiagent deep reinforcement learning for industrial Internet of Things systems[J]. IEEE Internet of Things Journal, 2024, 11(7): 12228-12239.
- [10] XIA S, YAO Z, LI Y, et al. Online distributed offloading and computing resource management with energy harvesting for heterogeneous MEC-enabled IoT[J]. IEEE Transactions on Wireless Communications, 2021, 20(10): 6743-6757.
- [11] YANG C, WANG Z S, NIE R C, et al. Computation offloading strategy based on sparrow search algorithm in Internet of Vehicles[J]. Computer Engineering and Design, 2023, 44(1): 1-7.  
杨超, 王宗山, 聂仁灿, 等. 车联网中基于麻雀搜索算法的计算卸载策略[J]. 计算机工程与设计, 2023, 44(1): 1-7.
- [12] ZHU S, ZHAO M, ZHANG Q. Multi-objective optimal offloading decision for multi-user structured tasks in intelligent transportation edge computing scenario[J]. The Journal of Supercomputing, 2022, 78(16): 17797-17825.
- [13] XU X, ZHANG X, LIU X, et al. Adaptive computation offloading with edge for 5G-envisioned Internet of connected vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(8): 5213-5222.
- [14] PHAM Q, FANG F, HA V N, et al. A survey of multi-access edge computing in 5G and beyond: fundamentals, technology integration, and state-of-the-art[J]. IEEE Access, 2020, 8: 116974-117017.
- [15] WANG K, WANG X, LIU X, et al. Task offloading strategy based on reinforcement learning computing in edge computing architecture of Internet of Vehicles[J]. IEEE Access, 2020, 8: 173779-173789.
- [16] WANG D, YUE J, CHAI P, et al. Calibration of camera internal parameters based on grey wolf optimization improved by Levy flight and mutation[J]. Scientific Reports, 2022, 12(1): 7828.
- [17] MILOSEVIC M, CEP R, CEPOVA L, et al. A hybrid grey wolf optimizer for process planning optimization with precedence constraints[J]. Materials, 2021, 14(23): 7360.
- [18] MEHTA S, SINGH M. Revolutionizing traffic law enforcement: IoT-enabled vehicle control systems for smart cities[C]//Proceedings of the 2024 IEEE International Conference on Intelligent Signal Processing and Effective Communication Technologies, 2024: 1-5.
- [19] HE X L, ZHANG G, CHEN Y H, et al. WOA-SVM multi-classification algorithm incorporating Lévy flight and elite opposition-based learning[J]. Application Research of Computers, 2021, 38(12): 3640-3645.  
何小龙, 张刚, 陈跃华, 等. 融合 Lévy 飞行和精英反向学习的 WOA-SVM 多分类算法[J]. 计算机应用研究, 2021, 38(12): 3640-3645.
- [20] CHAABOUNI N, MOSBAH M, ZEMMARI A, et al. Network intrusion detection for IoT security based on learning techniques[J]. IEEE Communications Surveys & Tutorials, 2019, 21(3): 2671-2701.
- [21] WANG Z, PAN T. A gray wolf optimization algorithm using position mutation strategy[C]// Proceedings of the 41st Chinese Control Conference, 2022: 1957-1960.
- [22] ZHANG M. A binary hybrid grey wolf optimizer for MEC offloading[C]// Proceedings of the 2022 12th International Conference on Information Technology in Medicine and Education, 2022: 525-529.
- [23] LIN F, LI H, LUO C W, et al. Joint computation offloading and resource allocation algorithm for dependent tasks in Internet of Vehicles[J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2023, 35(5): 826-837.  
林峰, 李华, 罗斌文, 等. 车联网中面向依赖任务的联合计算卸载及资源分配算法[J]. 重庆邮电大学学报(自然科学版), 2023, 35(5): 826-837.

(责任编辑:尹晨茹)