

# 基于改进的混合遗传算法的车联网任务卸载策略研究

丛玉良<sup>1</sup>, 孙闻晔<sup>1</sup>, 薛科<sup>2</sup>, 钱志鸿<sup>1</sup>, 陈绵书<sup>1</sup>

(1. 吉林大学通信工程学院, 吉林 长春 130012; 2. 中国科学院长春光学精密机械与物理研究所, 吉林 长春 130012)

**摘要:** 针对多车多服务器车联网场景下的卸载决策造成资源分配不合理的问题, 提出了一种两阶段启发式的车联网任务卸载策略。该策略利用改进的混合遗传算法 (IGHA) 与改进的人工鱼群算法 (AFSA), 结合系统对其内部的平均开销、时延与能耗的要求, 对 2 种改进算法进行多次迭代求解, 在任务卸载过程中实现最优资源分配。仿真实验结果表明, 相比于基准方案, 所提方案可以有效地降低系统内开销并提高任务卸载效率。

**关键词:** 车联网; 卸载; 遗传算法; 人工鱼群算法

**中图分类号:** TN929.5

**文献标志码:** A

**DOI:** 10.11959/j.issn.1000-436x.2022188

## Research on task offloading strategy of Internet of vehicles based on improved hybrid genetic algorithm

CONG Yuliang<sup>1</sup>, SUN Wenxi<sup>1</sup>, XUE Ke<sup>2</sup>, QIAN Zhihong<sup>1</sup>, CHEN Mianshu<sup>1</sup>

1. College of Communication Engineering, Jilin University, Changchun 130012, China

2. Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130012, China

**Abstract:** Aiming at the problem of unreasonable resource allocation caused by the unloading decision in the multi-vehicle and multi-server IoV scenario, a two-stage heuristic IoV task offloading strategy was proposed. This strategy used the improved hybrid genetic algorithm (IHGA) and the improved artificial fish swarm algorithm (AFSA), combined with the system's internal average overhead, delay and energy consumption requirements, the two improved algorithm was used for multiple iterations to achieve optimal resource allocation in the process of task unloading. The simulation results show that the proposed scheme can effectively reduce the system internal overhead and improve the task offloading efficiency compared with the benchmark scheme.

**Keywords:** Internet of vehicles, offloading, genetic algorithm, artificial fish swarm algorithm

## 0 引言

随着智能汽车和无线通信的快速发展, 车联网中出现了许多先进应用, 如自动驾驶和视频辅助实时导航等。这些应用程序的实现需要强大的计算能力来管理各种计算密集且对时延敏感的任务<sup>[1-2]</sup>。为了解决车辆终端计算能力等问题, 移动边缘计算 (MEC, mobile edge computing) 被认为是一种很有前途的范例<sup>[3]</sup>。有了 MEC, 车辆的计算任务通过无

线链路上传到 MEC 服务器上计算<sup>[4]</sup>, 可以减少本地计算时延。然而, 随着越来越多的任务被转发到 MEC 服务器, 由于无线和计算资源有限, MEC 服务器将承受巨大压力, 这将导致车辆的高时延和低可靠性。面对多车多服务器的场景, 合理的任务卸载决策和计算资源分配值得研究。

车联网任务卸载决策主要解决的是终端是否将任务卸载至服务器, 以及卸载多少的问题, 是一个典型的 0-1 整数规划问题。近年来, 遗传算法因

收稿日期: 2022-05-07; 修回日期: 2022-09-23

通信作者: 陈绵书, chenms@jlu.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61771219)

**Foundation Item:** The National Natural Science Foundation of China (No.61771219)

其具有全局优化性、简单性及稳健性等特点,得到了许多学者的青睐。余翔等<sup>[5]</sup>面对移动边缘计算的车联网场景,考虑并发多个优先级不同的计算任务,提出了一种基于遗传算法的任务卸载策略。邓添等<sup>[6]</sup>提出了一种在多信道环绕下的任务卸载模型,并采用遗传算法来求解。高基旭等<sup>[7]</sup>考虑了本地、边缘服务器和云端协同计算的任务卸载模型,提出了一种基于遗传算法的求解方案,从而得到同时考虑时延和能耗的最小系统代价。但是上述文献均没有对基本的遗传算法进行改进,传统的遗传算法后期易出现早熟现象且寻优能力无法达到理想效果。并且上述文献只考虑了任务卸载决策这一个问题,未对 MEC 的计算资源进行优化。

Zhao 等<sup>[8]</sup>在云辅助的情况下,通过联合优化任务卸载决策和计算资源分配,建立了以系统效用最大化的优化问题,提出了一种分布式计算卸载和资源分配算法。Yang 等<sup>[9]</sup>研究了车载边缘计算网络中高效的任務卸载策略。车辆以最佳方式执行卸载时间选择、通信和计算资源分配,并考虑车辆的移动性和任务的最大时延。梁颖杰等<sup>[10]</sup>提出了一种基于 MEC 和任务优先级的智能卸载策略以降低由时延所组成的系统总成本,并根据任务优先级对任务卸载位置进行选择。

文献[8-10]虽同时考虑了任务卸载决策和资源分配 2 个因素,但均只考虑了时延这一个指标。近几年来,能源短缺正成为限制车联网发展的一个关键障碍。根据统计数据<sup>[11]</sup>,在不断增长的能源需求中,电力占 16.5%,传统可再生能源占 11.9%,煤炭占 6.4%,而化石燃料占 49.4%,主要包括现有车辆消耗。因此,考虑车辆应用的能耗意义重大。特别是,电动汽车或混合动力电动汽车将在不久的将来主导市场,这促使人们考虑车联网中的能源消耗。

在现实生活中,每辆车由于计算任务大小的不同通常会需要不同计算资源。在计算任务卸载过程中,有限的计算资源在车辆之间共享,每辆车需要确定是否进行卸载,每台 MEC 服务器也应在多车之间进行合理的资源分配。考虑到上述问题,本文的主要研究工作如下。

1) 构建了基于 MEC 的车联网任务卸载模型,在任务卸载决策和资源分配的约束下,通过联合优化任务卸载决策和边缘服务器的资源分配来最小化系统内的平均成本。成本是基于任务时延、能耗和权重因子这 3 个指标定义的。

2) 将任务卸载决策和资源分配建模为一个混合整数非线性规划问题,针对这个问题,提出了一种基于改进的混合遗传算法的两阶段启发式方案。通过将优化问题分成 2 个子问题,利用改进的混合遗传算法 (IGHA, improved hybrid genetic algorithm) 求解任务卸载决策问题,利用改进的人工鱼群算法 (AFSA, artificial fish-swarm algorithm) 求解资源分配问题,再将二者不断迭代求解,直至达到终止条件。

3) 对基本的遗传算法进行了改进。充分考虑了 MEC 服务器出现过载或者对计算资源利用不足的 2 种情况,具体为通过对经过交叉、变异后得到的不满足约束条件的染色体和满足约束条件但对 MEC 的计算资源并未充分利用的染色体使用了贪婪修正法进行修正,使其转换为对 MEC 的计算资源充分利用的染色体,从而提高算法的收敛速度和寻优能力。

4) 仿真结果表明,相比于基准方案, IHGA-AFSA 方案降低了系统内的平均开销、时延及能耗。

## 1 系统模型

### 1.1 网络模型

如图 1 所示,本文所研究的基于 MEC 的车联网包括  $N$  辆单向行驶在道路上的汽车,道路沿线有  $M$  个路边单元 (RSU, road side unit), 每个 RSU 都配备一个 MEC 服务器,二者通过光纤有线链路连接。RSU 的集合定义为  $\mathcal{M} = \{1, 2, \dots, M\}$ 。车辆的集合定义为  $\mathcal{N} = \{1, 2, \dots, N\}$ , 每辆车有且仅有一个计算任务  $Q_n = \{D_n, C_n, T_n^{\max}\}$ ,  $n \in \mathcal{N}$ 。其中,  $D_n$  是输入数据的大小,  $C_n$  是完成任务  $Q_n$  所需的 CPU 周期数,  $T_n^{\max}$  是任务完成可容忍的最大时延。任务可以在本地处理,或者卸载到 MEC 服务器处理,定义  $a_n$  表示任务卸载决策变量,  $a_n = 1$  表示车辆将任务卸载至边缘服务器计算,  $a_n = 0$  表示任务在车辆本地计算。

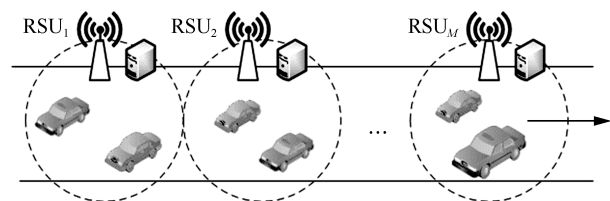


图 1 车联网场景

车辆与 RSU 之间的通信是通过直连的无线链路进行的,车辆到 RSU 的上传链路设定为频率平

坦型快衰落瑞利信道。根据香农公式，可以计算出上传链路的传输速率为

$$R_n = B_n \text{lb} \left( 1 + \frac{G_n P_n}{\sigma^2} \right) \quad (1)$$

其中， $B_n$  表示 MEC 服务器与车辆之间的带宽大小， $P_n$  表示车载设备的发射功率， $G_n$  表示车辆与 RSU 之间的信道增益， $\sigma^2$  表示无线信道的传输噪声。

## 1.2 计算模型

### 1) 本地计算

当车辆在本地处理其计算任务时，计算时延取决于其自身的计算能力，设  $f_n^{\text{loc}}$  是车辆  $n$  的计算能力，由放置在车辆上的车载单元（OBU, on-board unit）确定<sup>[12]</sup>。故本地计算时延为

$$T_n^{\text{loc}} = \frac{C_n}{f_n^{\text{loc}}} \quad (2)$$

同时可以获得车辆用户进行任务卸载的能耗为

$$E_n^{\text{loc}} = \kappa (f_n^{\text{loc}})^2 C_n \quad (3)$$

其中， $\kappa (f_n^{\text{loc}})^2$  表示执行一个 CPU 周期的能耗， $\kappa$  是取决于芯片结构的有效开电容系数。

### 2) 边缘服务器计算

与其他研究<sup>[12-13]</sup>一样，在语音识别等许多应用中，其计算结果的大小远小于输入数据的大小，接收计算结果的时延可以忽略不计。在这种情况下，总时延具体包括上行链路传输时延及计算任务执行时延。车辆  $n$  的总时延为

$$T_n^{\text{off}} = \frac{D_n}{R_n} + \frac{C_n}{f_n} \quad (4)$$

其中， $f_n$  表示 MEC 服务器分配给车辆  $n$  的计算资源，定义  $F_m^{\text{max}}$  为服务器  $m$  的总计算资源，有  $\sum_{n=1}^N a_n f_n \leq F_m^{\text{max}}$ ； $R_n$  表示计算任务所属车辆与服务

器之间的无线传输速率。

车辆任务卸载时的能耗主要来自卸载数据到边缘服务器的过程中的能耗，即

$$E_n^{\text{off}} = p_n \frac{D_n}{R_n} \quad (5)$$

## 1.3 问题表述

由于每个任务可以在本地或者 MEC 服务器上执行，因此车辆  $n$  的总时延为

$$T_n = (1 - a_n) \frac{C_n}{f_n^{\text{loc}}} + a_n \left( \frac{D_n}{R_n} + \frac{C_n}{f_n} \right) \quad (6)$$

车辆  $n$  的能耗包括本地能耗和上传能耗，表示为

$$E_n = (1 - a_n) \kappa (f_n^{\text{loc}})^2 C_n + a_n P_n \frac{D_n}{R_n} \quad (7)$$

将时延能耗开销（LEC, latency-energy cost）即车辆开销，定义为任务执行的时延和能耗的加权和。因此，车辆  $n$  的 LEC 表达式为

$$\varphi_n = \Lambda_n^t T_n + \Lambda_n^e E_n \quad (8)$$

其中， $\Lambda_n^t$  和  $\Lambda_n^e$  分别表示时延和能耗的权重因子，表示用户对时延和能耗的重视程度，并满足约束条件  $\Lambda_n^t + \Lambda_n^e = 1, 0 \leq \Lambda_n^t, \Lambda_n^e \leq 1$ 。为满足特定用户的要求，允许车辆选择不同的权重进行决策。例如，具有能源优先权的车辆会选择更大的  $\Lambda_n^e$  以节省更多能源。同时，当车辆正在执行一些对时延敏感的应用程序时，例如目标检测，则最好使用更大的  $\Lambda_n^t$  来最小化时延。

本节将联合任务卸载决策和资源分配作为优化问题，目标是 minimized 系统内平均开销。定义  $a = \{a_1, a_2, \dots, a_n\}$  为 MEC 服务器任务卸载决策变量， $f = \{f_1, f_2, \dots, f_n\}$  为计算资源变量，将优化问题表示为

$$\begin{aligned} \text{P1: } & \min_{a, f} \frac{1}{N} \sum_{n=1}^N \varphi_n(a, f) \\ \text{C1: } & \sum_{n=1}^M a_n = 1, \forall n \in \mathcal{N} \\ \text{C2: } & a_n \in \{0, 1\}, \forall n \in \mathcal{N} \\ \text{C3: } & 0 \leq f_n, \forall i \in \mathcal{N} \\ \text{C4: } & \sum_{n=1}^N a_n f_n \leq F_m^{\text{max}}, \forall n \in \mathcal{N}, m \in \mathcal{M} \end{aligned} \quad (9)$$

其中，C1 和 C2 表示一个计算任务只能卸载至某一个 MEC 服务器，不能同时卸载至 2 个服务器；C3 表示 MEC 服务器分配给车辆  $n$  的计算资源不能为负数；C4 表示 MEC 服务器总计算资源的约束。

## 2 算法设计

解决 P1 的关键挑战是求解整数变量  $a_n \in \{0, 1\}$ ， $a_n$  使 P1 是一个混合整数非线性规划问题，这类问

题通常都是非凸的 NP-hard<sup>[14]</sup>, 求解起来具有一定的难度, 对此本文提出了一种基于改进的混合遗传算法的求解方案, 具体将 P1 划分为 2 个子问题。

1) 任务卸载决策问题。假设资源分配已给定, 即  $f = f^{(0)}$ , 那么原优化问题就变成了只关于变量  $a$  的 0-1 整数规划问题, 本文采用 IHGA 求解。

2) 资源分配问题。在该子问题的基础上, 确定  $a = a^*$  后, 问题 P1 转化为关于  $f$  的连续问题, 本文采用改进的 AFSA 求解。

### 2.1 任务卸载决策

当  $f = f^{(0)}$  给定时, P1 转化为 P2

$$\begin{aligned} \text{P2: } & \min_a \frac{1}{N} \sum_{n=1}^N \varphi_n(a, f^{(0)}) \\ \text{C1: } & \sum_{n=1}^M a_n = 1, \forall n \in \mathcal{N} \\ \text{C2: } & a_n \in \{0, 1\}, \forall n \in \mathcal{N} \\ \text{C5: } & \sum_{n=1}^N a_n f_n^{(0)} \leq F_m^{\max}, \forall n \in \mathcal{N}, m \in \mathcal{M} \end{aligned} \quad (10)$$

此时, P1 转化为关于  $a$  的优化问题 P2,  $a$  为 0-1 整数变量, 采用遗传算法进行求解。遗传算法不要求求解的问题是连续的或可微的, 也不需要控制算法的执行方向, 具体步骤如下。

**步骤 1** 染色体。遗传算法借鉴了自然选择的思想<sup>[15]</sup>, 引入了染色体的概念, 即可能的解。本文中的任务卸载决策构成了个体的染色体信息, 对于第  $n$  条染色体, 其染色体信息可表示为

$$I_i = [a_1^i, a_2^i, \dots, a_N^i] \quad (11)$$

编码及种群初始化。根据任务的卸载模型, 共有  $N$  个任务要经过卸载决策之后决定是否卸载。因此在遗传算法中, 每个染色体都应该由  $N$  个基因组成, 每个任务都可以选择在本地计算还是卸载至 MEC 服务器计算, 故每个基因都有 2 个可能的值 (本地计算则基因为 0, MEC 服务器计算则基因为 1), 并得到初始的染色体种群  $I(0)$ 。

**步骤 2** 染色体修正。对初始化染色体进行修正, 修正主要分为以下两步。

1) 对不符合约束条件的染色体进行修正。对于该部分染色体, 基于文献[16]中混合遗传算法的思想使其转换为符合约束条件的染色体, 如果转换后的染色体充分利用了资源, 则修正结束; 否则转至

2)。具体为, 当  $\sum_{n=1}^N a_n f_n > F_m^{\max}$  时, 将  $a_n = 1$  的车辆

按照  $\frac{\varphi_n}{f_n}$  从大到小的顺序进行排序, 然后根据排序

将选择卸载的车辆进行任务卸载, 直至接近但不超过服务器的计算总资源, 将剩余选择卸载的车辆由状态  $a_n = 1$  改变为  $a_n = 0$ 。

2) 对满足约束条件但对 MEC 的计算资源利用不足的染色体, 运用贪婪修正法进行修正。设  $X_i = [a_1^i, a_2^i, \dots, a_N^i]$  是一个对 MEC 的计算资源利用不足的染色体, 该染色体有  $r$  个车辆  $a_n = 0$ , MEC 的剩余计算资源为  $F_{m_0}$ , 通过贪婪修正法将其修正为充分利用计算资源的染色体  $Y_i = [a_1^i, a_2^i, \dots, a_N^i]$ 。

将染色体  $X_i$  中  $a_n = 0$  的车辆按照  $\frac{\varphi_n}{f_n}$  从大到小的顺序进行排序, 形成长度为  $r$  的序列  $\{b_1, b_2, \dots, b_r\}$  (其中,  $b_1$  为最大的序号,  $b_r$  为最小的序号), 然后按

排序将  $a_n = 0$  的车辆转变为  $a_n = 1$ , 直至接近服务器的计算总资源, 但不能超过其计算资源。

**步骤 3** 遗传算法中个体 (解决方案) 的质量由适应度值来评估。由于本文研究了最小化问题, 因此将适应度函数设置为

$$\text{fitness} = \frac{1}{\frac{1}{N} \sum_{n=1}^N \varphi_n} \quad (12)$$

适应度值越高, 目标函数的成本就会越小, 表明该卸载策略越优。

**步骤 4** 选择。采用轮盘赌算法筛选个体, 即随机旋转轮盘进行选择, 每个个体都可能被反复挑选。其基本思想是每个个体被选中的概率与其适应度成正比。设  $P(d_i)$  为下一代选择染色体  $d_i$  被选中的概率, 即

$$P(d_i) = \frac{f_{\text{fitness}}(d_i)}{\sum_{j=1}^N f_{\text{fitness}}(d_j)} \quad (13)$$

**步骤 5** 交叉。如图 2 所示, 采用单点交叉的方式, 在染色体串中随机设置一个交叉点, 然后进行基因交换, 生成 2 个新个体。其意义在于通过为下一代群体保留具有更好亲本的基因来改善新群体的适应性。

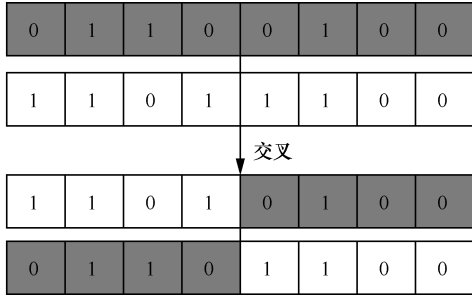


图 2 交叉

**步骤 6 变异。**防止种群中的所有解落入局部最优解。对于二进制编码，随机选择几个位置从 1 变成 0 或从 0 变成 1，如图 3 所示。

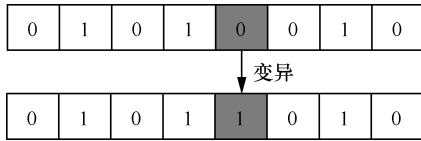


图 3 变异

## 2.2 资源分配

当求出较优的任务卸载决策  $a^*$  后，代入 P1，得到问题 P3

$$\begin{aligned}
 \text{P3: } & \min_f \frac{1}{N} \sum_{n=1}^N \varphi_n(a^*, f) \\
 \text{C3: } & 0 \leq f_n, \forall n \in \mathcal{N} \\
 \text{C6: } & \sum_{n=1}^N a_n^* f_n \leq F_m^{\max}, \forall n \in \mathcal{N}, m \in \mathcal{M} \quad (14)
 \end{aligned}$$

特别是，AFSA 有一种机制，可以使搜索行为跳出局部极值点并获得全局优化解。因此，采用 AFSA 求解资源分配问题<sup>[17]</sup>。通常，AFSA 有 4 个关键行为，具体如下。

1) **觅食行为：**人工鱼趋向于食物的一种行为。人工鱼通过视觉或味觉来感知水中食物量或食物浓度来选择行动的方向。觅食行为是算法收敛的基础。假设第  $i$  条人工鱼的当前位置和适应度值分别为  $X_i$  和  $Y_i$ ，人工鱼视野内的另一个位置为  $X_j$  和适应度值为  $Y_j$ ，若  $Y_j > Y_i$ ，则按式(15)向新选择的位置靠近一步；否则，重新获取新位置，判断是否满足条件，若还不满足条件，则按式(16)随机移动一步，人工鱼  $X_i$  在其视野内随机选择一个位置  $X_j$ ，其表达式为

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{\|X_j - X_i^t\|} \cdot \text{Step} \cdot \text{Rand()} \quad (15)$$

$$X_j = X_i + \text{Visual} \cdot \text{Rand()} \quad (16)$$

2) **聚群行为：**促使人工鱼自发地聚集在一起以

避免伤害。如果鱼群中心有许多“营养食物”，并且它们不是太拥挤，即人工鱼  $X_i$  搜索当前视野内 ( $d_{ij} < \text{Visual}$ ) 的伙伴数目  $n_f$  和中心位置  $X_c$ ，若  $\frac{Y_c}{n_f} > \delta Y_i$ ，则  $X_i$  朝伙伴的中心位置移动一步

$$X_i^{t+1} = X_i^t + \frac{X_c - X_i^t}{\|X_c - X_i^t\|} \cdot \text{Step} \cdot \text{Rand()} \quad (17)$$

3) **追尾行为：**当一个人工鱼找到一个食物充足且不太拥挤的最佳区域时，附近的人工鱼将跟随它并快速到达其附近的位置。人工鱼  $X_i$  搜索当前视野内 ( $d_{ij} < \text{Visual}$ ) 的伙伴中函数  $Y_j$  最优伙伴  $X_j$ ，若  $\frac{Y_j}{n_f} > \delta Y_i$ ，则  $X_i$  朝此伙伴移动一步

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{\|X_j - X_i^t\|} \cdot \text{Step} \cdot \text{Rand()} \quad (18)$$

4) **随机行为：**为了使人工鱼在更大范围内寻找食物或同伴，会按式(19)在水中随机地游来游去。当发现食物时，会向食物逐渐增多的方向快速移动。

$$X_i^{t+1} = X_i^t + \text{Visual} \cdot \text{Rand()} \quad (19)$$

基本的 AFSA 步长和视野一般是固定值，这样前期的搜索易陷入盲目搜索，故本文引入自适应因子使视野和步长能根据外界环境信息的变化而动态地调整，这样算法在前期具有更好的全局搜索能力，还可以避免算法在后期容易发生振荡现象。二者的变化满足

$$\begin{cases} r = \frac{k}{100} \\ \text{Visual} = \frac{1.25 \text{Visual}}{(r+1)^2} \\ \text{Step} = a \text{Step} \end{cases} \quad (20)$$

其中， $k$  为迭代次数， $a \in (0,1)$  为自适应因子。

基于改进的 AFSA 求解资源分配的具体步骤如下。

**步骤 1** 初始化设置，包括种群规模  $N$ ，每条人工鱼的初始位置、视野  $\text{Visual}$ 、步长  $\text{Step}$ 、拥挤度因子  $\delta$ 、重复次数  $k$ ；

**步骤 2** 根据 P3 计算初始个体的适应值；

**步骤 3** 对每个个体进行评价，从觅食、聚群、追尾和随机中选择一个行为；

步骤 4 执行行为, 更新自己, 生成新鱼群;

步骤 5 评价所有个体;

步骤 6 达到迭代次数上限时算法结束, 否则转到步骤 3。

### 2.3 全局算法

综上, 本文将任务卸载决策和资源分配建模为混合整数非线性规划问题, 对此提出了一个基于改进的混合遗传算法的两阶段启发式算法的方案。将问题分成 2 个子问题, 通过将初始化资源分配代入 P1, 得到只关于任务卸载决策的 P2, 通过 IHGA 进行求解。将得到的任务卸载决策解代入 P1, 得到只关于资源分配问题的 P3。利用改进的 AFSA 进行求解, 基于二者的耦合关系, 不断迭代求解, 具体步骤如下。

步骤 1 初始化资源分配为  $f^{(0)}$ , 代入 P1, 设置当前迭代次数为  $r=0$ ;

步骤 2 将资源分配  $f^{(r)}$  代入 P2, 通过 IHGA 得到较优的任务卸载决策  $a^{(r+1)}$ ;

步骤 3 通过 IHGA 得到的任务卸载决策  $a^{(r+1)}$  代入 P1, 得到 P3, 通过改进的 AFSA 求解得到资源分配方案  $f^{(r+1)}$ ;

步骤 4 判别相邻两次目标函数的增长值是否小于阈值  $\tau$ , 若大于或等于则  $r=r+1$ , 并同时继续步骤 2; 否则输出当前最优的任务卸载决策  $a_n^*$  和资源分配方案  $f^*$ 。

全局算法流程如图 4 所示。

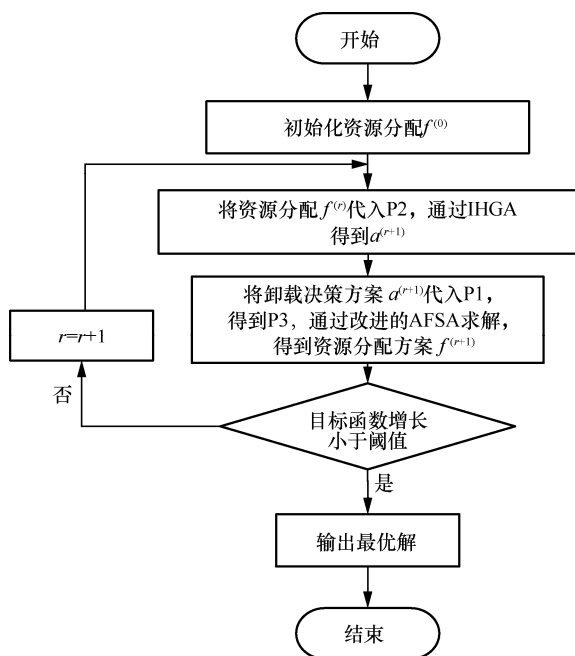


图 4 全局算法流程

### 3 仿真实验及数据分析

为了验证本文针对任务卸载决策和资源分配所提的算法, 在 MATLAB 平台进行仿真实验。参考文献[7-9,12], 以及本文的实际仿真环境, 有如下参数设置。考虑一条长为 2 000 m 的单向道路, 放置 5 个 RSU, 每个 RSU 的通信覆盖范围的半径为 500 m, RSU 分配给每辆车的子信道带宽为 2 MHz, 每辆车的发射功率为 46 dBm, 高斯白噪声为  $\sigma^2 = -147\text{dBm}$ , 车辆用户的信道衰减模型为  $u = 127 + 30 \log d$ , 其中  $d$  是每辆车与 RSU 中心的距离, RSU 与车辆  $i$  之间的信道增益为  $G_i = 10^{\frac{-u}{10}}$ 。IHGA 中的交叉概率为 0.4, 变异概率为 0.02, 种群数量为 50, 迭代次数为 50。改进的 AFSA 中<sup>[18]</sup>的人工鱼数量为 50 只, 感知距离为 1, 步长为 1, 拥挤度因子为 0.618, 最多迭代次数为 50, 最多试探次数为 100, 自适应因子为 0.6。设置时延因子  $A_n^t = 0.8$ , 能耗因子  $A_n^e = 0.2$ 。IHGA-AFSA 方案的阈值  $\tau = 0.1$ 。

为了验证本文提出的 IHGA-AFSA 方案的性能, 对比了以下 3 种基准方案。

- 1) 本地计算方案: 计算任务不卸载到 MEC 服务器上计算, 而是全部留在车辆本地计算。
- 2) 随机卸载策略: 车辆随机选择是否进行卸载, 若卸载则利用改进的 AFSA 求解资源分配问题。
- 3) IHGA-平均方案: IHGA 求解任务卸载决策, 将 MEC 的计算资源平均分配给车辆。

#### 3.1 任务所需 CPU 周期数性能分析

本节实验中一共有 100 辆车, 每辆车均有一个计算任务, 故共有 100 个计算任务。每个 RSU 配备一台 MEC 服务器, 每台 MEC 服务器的最大计算资源为 400 GHz, 每辆车的计算频率在 1.8~2.0 GHz 随机分布。计算任务的输入数据量大小为 1.5 MB, 图 5~图 7 分别展示了完成任务所需计算量(用 CPU 周期数表示)为 1 300~2 000 Megacycles 时系统内开销、时延及能耗的变化。为了消除随机性, 一共进行 100 次实验, 最后结果取平均。

图 5 分析了任务所需计算量与系统内开销的关系。从图 5 可以看出, 当任务所需计算量增加时, 系统内开销不断增加。其中, 本地计算方案的开销最大, IHGA-AFSA 方案与随机卸载策略相比, 资源分配均采用改进的 AFSA, 而 IHGA-AFSA 方案所产生的开销更小, 说明本文提出的 IHGA 对任务卸载决策

起到了优化作用。IHGA-AFSA 方案与 IHGA-平均方案相比，卸载决策均采用 IHGA，而 IHGA-AFSA 方案所产生的开销也是更小，说明本文提出的改进的 AFSA 对资源分配起到了优化效果。因此本文提出的 IHGA-AFSA 方案在优化系统开销方面性能优于本地计算、随机卸载和 IGHA-平均方案。

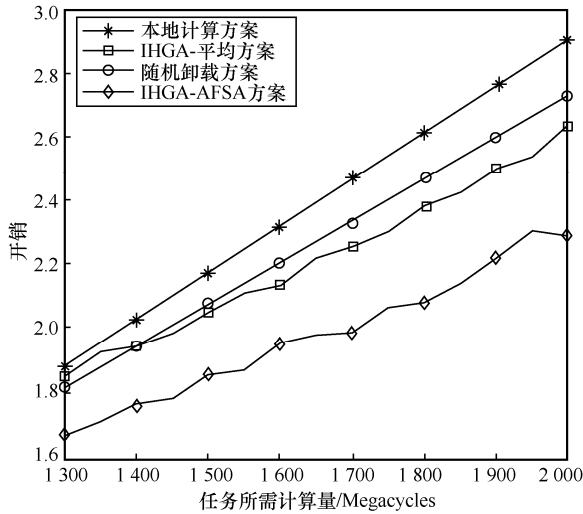


图 5 任务所需计算量与系统内开销的关系

图 6 分析了任务所需计算量与系统内时延的关系。从图 6 可以看出，当任务所需计算量增加时，系统内时延不断增加。其中，本地计算方案的时延最大，其余 3 种方案均小于本地时延，说明车联网与 MEC 技术的结合更加符合时延敏感型任务对时延的苛刻要求。当任务所需计算量为 1 700 Megacycles 时，IHGA-AFSA 方案与本地计算方案相比时延减少了约 17.4%。在 4 种方案中，本文所提方案时延最低，因此本文提出的 IHGA-AFSA 方案在优化系统时延方面性能强于本地计算、随机卸载和 IHGA-平均方案。

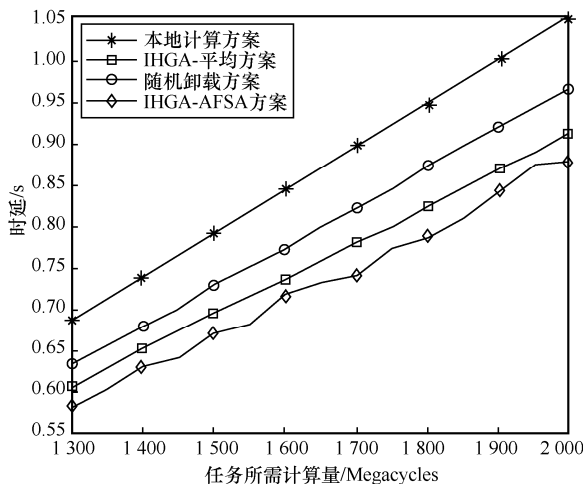


图 6 任务所需计算量与系统内时延的关系

图 7 分析了任务所需计算量与系统内能耗的关系。当任务所需计算量增加时，系统内能耗不断增加。其中，本地计算方案的能耗最大，其次是随机卸载策略，再次是 IHGA-平均方案，本文提出的方案在优化能耗方面性能是最优的，产生的能耗是最低的。IHGA-AFSA 方案与随机卸载策略相比，资源分配均采用改进的 AFSA，而 IHGA-AFSA 方案所产生的能耗更小，说明本文提出的 IHGA 不仅对任务卸载决策起到了优化作用，对能耗也有效果，在任务所需计算量为 1 700 Megacycles 时，能耗减少约 16.6%。

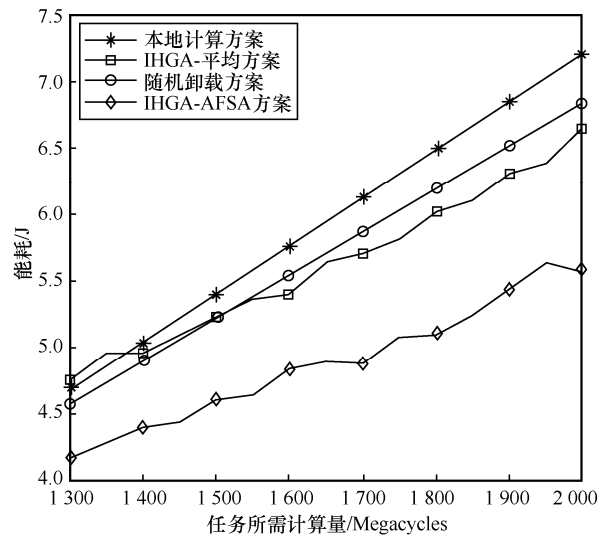


图 7 任务所需计算量与系统内能耗的关系

### 3.2 任务数据量大小性能分析

本节实验中一共有 70 辆车，每辆车均有一个计算任务，故共有 70 个计算任务。每个 RSU 配备一台 MEC 服务器，每台 MEC 服务器的最大计算资源为 400 GHz，每辆车的计算频率在 1.8~2.0 GHz 随机分布。计算任务所需的 CPU 周期数为 2 000 Megacycles，图 8~图 10 分别展示了任务数据量大小为 0.5~1.5 MB 时系统内开销、时延及能耗的变化。为了消除随机性，一共进行 100 次实验，最后结果取平均。

图 8 分析了任务数据量大小与系统内开销的关系。当任务数据量大小增加时，本地计算方案所产生的开销波动较小，这是因为开销是时延和能耗的加权，时延和能耗波动均比较小，故开销波动也小。但本地计算方案产生的开销是最大的，其次是随机卸载策略，本文提出的 IHGA-AFSA 方案与 IHGA-平均方案相比产生的开销更小。这说明 IHGA-AFSA 方案在优化开销方面强于本地计算、随机卸载、IHGA-平均

方案。当任务数据量大小为 1 MB 时，本文方案相比于本地计算、随机卸载、IHGA-平均方案所产生的开销降低了约 29.1%、23.3%、14.4%。

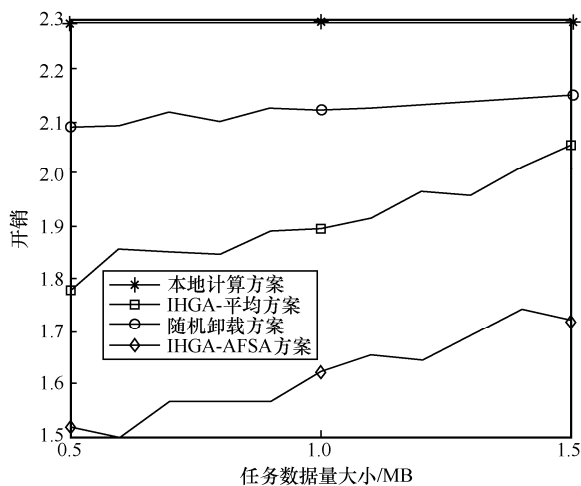


图 8 任务数据量大小与系统内开销的关系

图 9 分析了任务数据量大小与系统内的时延的关系。随着任务数据量大小的增加，本地计算方案的时延波动最小，这是因为任务数据量大小与本地时延没有直接的关系。其余 3 种方案的时延随着任务数据量大小的增加并没有明显的递增性，这是因为任务数据的大小只会对任务上传的时延造成影响，而上传链路的传输速率较快，故总体的时延不会有很明显的递增表现。但是在 3 种方案中，本文提出的 IHGA-AFSA 方案产生的时延最小，证实了其在优化时延方面的优越性。当任务数据量大小为 1 MB 时，本文方案相比于本地计算、随机卸载、IHGA-平均方案所产生的时延降低了约 23.4%、16.5%、15.5%。

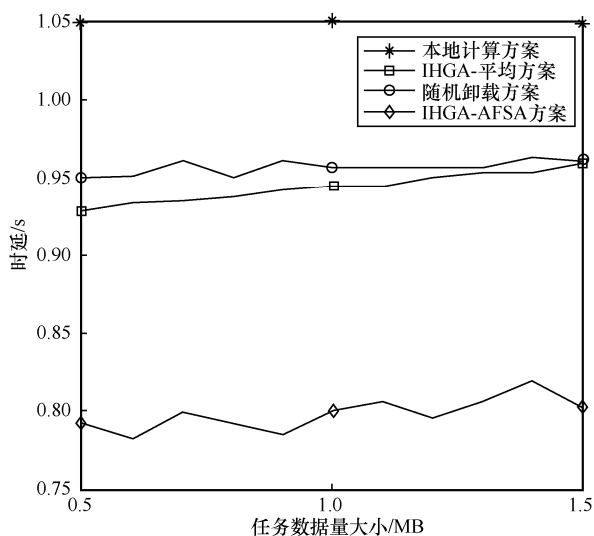


图 9 任务数据量大小与系统内时延的关系

图 10 分析了任务数据量大小与系统内能耗的关系。随着任务数据量大小的增加，本地计算方案的能耗波动最小，这是因为任务数据量大小与本地能耗没有直接的关系。其余 3 种方案的能耗随着任务数据量大小的增加而增加，这是因为任务数据量大小会对任务上传的能耗造成影响，在这种情况下本文提出的 IHGA-AFSA 方案在能耗优化方面效果是最好的。随机卸载策略中，卸载决策是随机选择的，并且本文时延因子  $\lambda'_n$  设置的较大，更倾向于对时延的优化，在图 10 中也证实了该方案对能耗优化的有效性。当任务数据量大小为 1 MB 时，IHGA-AFSA 方案相比于本地计算、随机卸载、IHGA-平均方案所产生的能耗降低了约 32.1%、27.2%、13.6%。

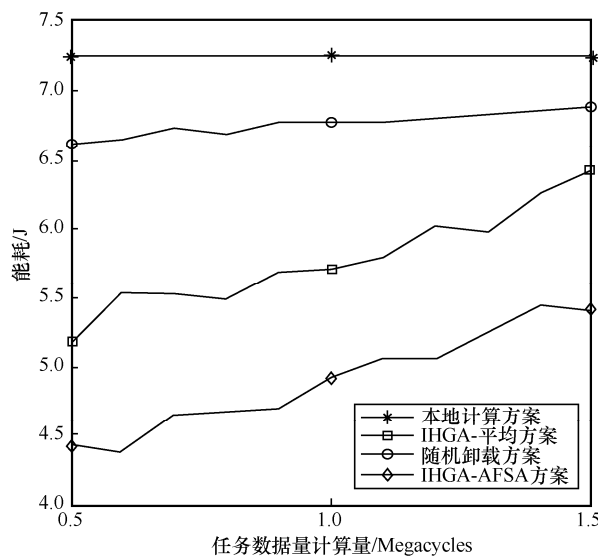


图 10 任务数据量大小与系统内能耗的关系

#### 4 结束语

本文对基于 MEC 车联网中的任务卸载决策和资源分配进行了研究。考虑决策和资源的约束，建立了以最小化系统内平均开销为目标的计算模型。针对优化问题的 NP-hard，将优化问题划分成 2 个子问题，提出了一个基于改进的混合遗传算法的两阶段启发式方案。利用 IHGA 求解任务卸载决策，AFSA 求解资源分配，再将二者不断迭代优化求解，直至达到阈值条件。仿真结果表明，所提方案与基准方案相比降低了系统内的平均开销、时延及能耗。

#### 参考文献:

[1] CHEN S Z, HU J L, SHI Y, et al. A vision of C-V2X: technologies, field testing, and challenges with Chinese development[J]. IEEE In-

- ternet of Things Journal, 2020, 7(5): 3872-3881.
- [2] 钱志鸿, 田春生, 郭银景, 等. 智能网联交通系统的关键技术与发展[J]. 电子与信息学报, 2020, 42(1): 2-19.  
QIAN Z H, TIAN C S, GUO Y J, et al. The key technology and development of intelligent and connected transportation system[J]. Journal of Electronics & Information Technology, 2020, 42(1): 2-19.
- [3] 吕品, 许嘉, 李陶深, 等. 面向自动驾驶的边缘计算技术研究综述[J]. 通信学报, 2021, 42(3): 190-208.  
LYU P, XU J, LI T S, et al. Survey on edge computing technology for autonomous driving[J]. Journal on Communications, 2021, 42(3): 190-208.
- [4] YAN J, BI S Z, ZHANG Y J A. Offloading and resource allocation with general task graph in mobile edge computing: a deep reinforcement learning approach[J]. IEEE Transactions on Wireless Communications, 2020, 19(8): 5404-5419.
- [5] 余翔, 刘一勋, 石雪琴, 等. 车联网场景下的移动边缘计算卸载策略[J]. 计算机工程, 2020, 46(11): 29-34, 41.  
YU X, LIU Y X, SHI X Q, et al. Mobile edge computing offloading strategy under Internet of vehicles scenario[J]. Computer Engineering, 2020, 46(11): 29-34, 41.
- [6] 邓添, 沈艳, 史奎锐. 基于遗传算法的移动边缘计算混合关键任务卸载[J]. 信息与电脑(理论版), 2021, 33(11): 26-29.  
DENG T, SHEN Y, SHI K R. Mixed-critical task offloading algorithm for mobile edge computing based on genetic algorithm[J]. China Computer & Communication, 2021, 33(11): 26-29.
- [7] 高基旭, 王珺. 一种基于遗传算法的多边缘协同计算卸载策略[J]. 计算机科学, 2021, 48(1): 72-80.  
GAO J X, WANG J. Multi-edge collaborative computing unloading scheme based on genetic algorithm[J]. Computer Science, 2021, 48(1): 72-80.
- [8] ZHAO J H, LI Q P, GONG Y, et al. Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(8): 7944-7956.
- [9] YANG C, LIU Y, CHEN X, et al. Efficient mobility-aware task offloading for vehicular edge computing networks[J]. IEEE Access, 2019, 7: 26652-26664.
- [10] 梁颖杰, 马星, 赵川斌. 车联网中基于 MEC 和任务优先级的智能卸载策略研究[J]. 通信与信息技术, 2021(2): 41-45.  
LIANG Y J, MA X, ZHAO C B. Research on intelligent offloading strategy based on MEC and task priority in Internet of vehicles[J]. Communication & Information Technology, 2021(2): 41-45.
- [11] EMPRECHA S, PATTARAPRAKORN W, CHUTIPRAPAT V, et al. The study on the effect of electric bus (non-fixed route) to energy consumption in Thailand[C]//Proceedings of 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). Piscataway: IEEE Press, 2016: 1-5.
- [12] ZHANG K, MAO Y M, LENG S P, et al. Optimal delay constrained offloading for vehicular edge computing networks[C]//Proceedings of 2017 IEEE International Conference on Communications. Piscataway: IEEE Press, 2017: 1-6.
- [13] ZHANG K, MAO Y M, LENG S P, et al. Delay constrained offloading for Mobile Edge Computing in cloud-enabled vehicular networks[C]// Proceedings of 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM). Piscataway: IEEE Press, 2016: 288-294.
- [14] GAREY M R, JOHNSON D S. Computers and intractability: a guide to the theory of NP-completeness[M]. New York: W. H. Freeman, 1990.
- [15] 金玲, 刘晓丽, 李鹏飞, 等. 遗传算法综述[J]. 科学中国人, 2015(27): 230.  
JIN L, LIU X L, LI P F, et al. A survey of genetic algorithms[J]. Scientific Chinese, 2015(27): 230.
- [16] 徐宗本. 计算智能—模拟进化计算[M]. 北京: 高等教育出版社, 2005.  
XU Z B. Computational intelligence - simulating evolutionary computation[M]. Beijing: Higher Education Press, 2005.
- [17] JIA S Y, ZHOU K, ZHOU S S, et al. An artificial fish swarm algorithm for a multi-objective grain transportation problem[J]. Evolutionary Intelligence, 2020, 13(1): 9-19.
- [18] 赵有生, 邸晟钧, 王占全, 等. 改进的人工鱼群算法采煤机调高控制策略[J]. 煤炭工程, 2020, 52(2): 136-141.  
ZHAO Y S, DI S J, WANG Z Q, et al. Improved artificial fish swarm algorithm for height control strategy of shearer[J]. Coal Engineering, 2020, 52(2): 136-141.

## [作者简介]



丛玉良(1966-), 女, 山东威海人, 博士, 吉林大学教授, 主要研究方向为智能信息处理、车联网、阵列信号处理等。



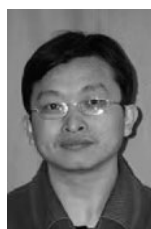
孙闻晞(1995-), 女, 吉林通化人, 吉林大学硕士生, 主要研究方向为车联网和移动边缘计算, 具体为基于 MEC 的车联网中的卸载问题。



薛科(1995-), 男, 吉林长春人, 吉林大学硕士生, 主要研究方向为车联网、移动边缘计算等。



钱志鸿(1957-), 男, 吉林长春人, 博士, 吉林大学教授、博士生导师, 主要研究方向为无线网络与通信理论, 包括 D2D、车联网技术的未来移动通信网络的移动性和大规模接入问题。



陈绵书(1973-), 男, 辽宁海城人, 博士, 吉林大学副教授、硕士生导师, 主要研究方向为图像处理、计算机视觉。