

基于车载以太网的任务卸载方案

涂岩恺

(厦门雅迅网络股份有限公司,福建 厦门 361008)

摘要:现代汽车总线的电子电气(E/E)架构开始从基于CAN总线的分布式架构向通信容量更大的基于车载以太网的集中式架构发展,但车内计算资源分布不平均,集中式网络的中央设备计算能力显著高于其他边缘端设备。由于智能汽车边缘端设备的计算任务不断在增长,为应对增长的边缘端计算任务需求,降低计算延时,研究在集中式E/E架构中实现有效的计算任务卸载方案,对车载以太网的通信特点进行分析,以车载中央计算机为核心对车内域控制器的计算任务进行卸载。针对车载以太网AVB通信协议的特点,结合数据分级和整流算法获得计算卸载的延时模型,进而建立边缘计算任务延时最小化的目标优化方程,并采用改进遗传算法进行迭代求解,最终收敛求出任务卸载的最优参数。实现了在满足复杂车载条件约束条件下降低车内计算任务时延的目标,仿真实验结果表明所提方法可以有效降低车内计算任务延时,在集中式E/E架构的车内网络上实现了计算资源的充分利用。

关键词:汽车边缘计算;车载以太网;AVB协议;任务卸载;任务延时;域控制器

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2023)05-0116-06

doi:10.3969/j.issn.1673-629X.2023.05.018

Task Offloading Scheme Based on Automotive Ethernet

TU Yan-kai

(Xiamen Yaxon Network Co., Ltd., Xiamen 361008, China)

Abstract: The electrical and electronic (E/E) architecture of modern automotive bus has started to evolve from a distributed architecture based on CAN bus to a centralized architecture based on Automotive Ethernet with larger communication capacity, but the computational resources are not evenly distributed in the vehicle, and the computational capacity of the centralized network is significantly higher than that of other edge devices. Since the computational tasks of edge devices in smart cars are growing, in order to cope with the growing demand of edge computational tasks and reduce the computational latency, an effective computational task offloading scheme is studied in the centralized E/E architecture, and the communication characteristics of Automotive Ethernet are analyzed to offload the computational tasks of in-vehicle domain controllers to the in-vehicle central computer. For the characteristics of AVB communication protocol of Automotive Ethernet, the delay model of computation offloading is obtained by combining data classification and traffic shaping algorithms, and then the objective optimization equation for minimizing the delay of edge computation tasks is established and iteratively solved by using improved genetic algorithm, and finally the optimal parameters of task offloading are converged. The objective of reducing the in-vehicle computation task latency under the constraints of complex automotive conditions is achieved, and the simulation results show that the proposed method can effectively reduce the in-vehicle computation task latency and realize the full utilization of computational resources on the in-vehicle network of centralized E/E architecture.

Key words: automotive edge computing; automotive Ethernet; AVB protocol; task offloading; task latency; domain controller

0 引言

随着智能汽车时代的到来,汽车需要处理的数据量呈指数级增长,智能传感器、高精度地图、V2X协同、AR/VR等应用使得汽车对ECU数量、运算能力和带宽的需求激增,从而使传统汽车电气电子(E/E)架构不堪重负,无法适应现代智能汽车的需求。如果使

用车-云^[1]架构处理计算任务,将难以解决网络带宽消耗和计算实时性问题。

边缘计算具有靠近数据源、节省通信带宽、实时性强的优势,因此在实时性要求高的智能汽车领域,具有较大的应用潜力与产业价值,在国际上已经成立汽车边缘计算联盟(AECC),旨在利用边缘计算技术驱动

收稿日期:2022-08-18

修回日期:2022-12-22

基金项目:浙江省重点研发计划项目(2021C01038);厦门市科技计划项目(3502Z20221027)

作者简介:涂岩恺(1983-),男,博士,高级工程师,从事人工智能与车联网应用研究。

智能汽车生态发展。Sun 等人^[2]基于移动计算资源的不稳定性、车辆计算能力的异构性以及计算任务的相互依赖性,采用蝙蝠算法优化汽车边缘计算卸载的协同任务调度,最小化任务执行延迟;李波^[3]提出基于最小执行时间算法的计算切换策略,采用 Docker 的容器迁移技术来降低任务卸载迁移时间;范艳芳等人^[4]发明了基于深度强化学习的协同计算卸载方案,解决高速公路场景下车辆移动性与边缘服务器静态部署的矛盾,降低计算卸载失败率;Huang 等人^[5]提出了一种利用停泊车辆边缘计算的方案,利用附件停车场停泊车辆的闲置计算资源与边缘服务器协同执行汽车计算任务的卸载。

当前研究热点主要是汽车与外部边缘设备或其他车辆之间的任务卸载、任务缓存、调度与优化,算法多基于车外网络架构与通信协议特点。实际上随着车载高性能芯片技术的发展,已经开始出现高性能的本地车载计算设备,并带动了新的汽车 E/E 架构的演进。在整车 E/E 架构上,特斯拉大范围合并车内网络域,将电子构架升级为三大中央计算机为基础的域控架构^[6],提升汽车自身计算能力的同时极大地缩减了 ECU 数量;在车内网络通信上,美国博通公司发明了 100BASE-T1 车载以太网^[7]替代传统的 CAN 总线网络,实现 100 Mbps 速率可靠传输,可为车内大量数据的任务卸载提供高效实时的通信方式。因此在以中央

计算机为核心,高速车载以太网为通信总线的新汽车 E/E 架构下,车载中央计算机可以作为车内边缘计算服务器的角色,将计算任务通过车载以太网在中央计算机与普通控制器之间进行卸载,相对于车外网络的计算卸载能更获得更高的实时性。

该文根据车载以太网通信协议的特点,建立车内网任务卸载延时方程,并采用改进遗传算法进行调度方程延时最小的最优化求解,在保持车辆总线任务优先级和功耗约束条件的基础上降低计算延时。

1 E/E 架构与任务卸载需求

传统的汽车 E/E 架构为分布式架构,车辆各功能由单一的电子控制单元控制。随着汽车智能化的发展使整车中 ECU 数量激增,分布式 E/E 架构很难高效地分配和承载数据计算和过多的复杂功能,难以满足智能汽车更高算力与更强通讯能力的需求。于是汽车 E/E 架构开始向集中式演化,将各功能合并和划分为新的功能域,对应的 ECU 合并成一个域控制器进行车辆控制,在车内网络设置一个中央计算机平台执行智能化程序,与域控制器采用以太网连接,域控制器受中央计算平台统一管理。如图 1 所示,实线框表示中央计算机及域控制器,虚线框表示在这些设备上执行的功能或服务。

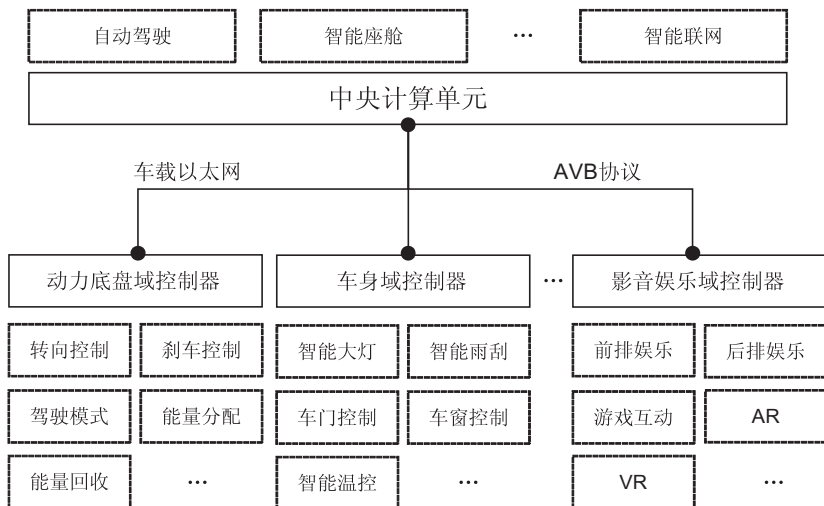


图1 集中式汽车 E/E 架构

域控制器采集和处理大量的传感器数据,对数据分析计算产生功能决策,在自身计算资源相对有限的情况下,需要通过高速车载以太网,将计算数据和计算任务卸载到中央计算单元进行计算。同时以太网总线还需要基于以太网音视频桥接 (AVB)^[8] 协议为影音娱乐域提供稳定的带宽。因此,需要在保证原有音视频数据流任务优先级的前提下,满足车内网计算任务卸载的优化调度需求。

2 车载以太网信号延迟特征

2.1 流量整形

传统以太网使用带有冲突检测的载波侦听多路访问 (CSMA/CD) 技术,所有数据流都是串行通讯,基于 Best Effort 尽力服务机制转发,这会导致不同节点的数据流发生带宽重叠的冲突,如图 2 所示,部分冲突的数据会被丢弃重传,从而影响消息实时性,这在总线型

网络架构中影响尤为明显。

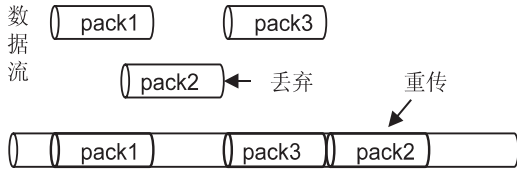


图 2 数据流重叠冲突影响通信实时性

为了避免带宽重叠问题,车载以太网采用流量整形算法^[9]对不同等级的数据流进行排序发送,以达到提高可靠实时传输的目的。

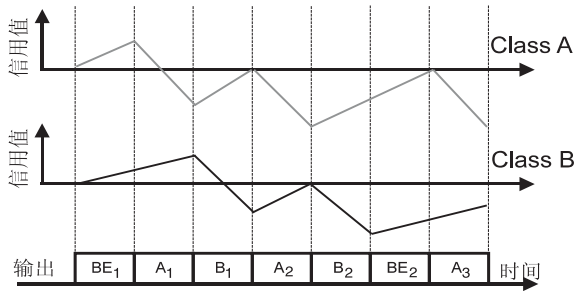


图 3 流量整形算法

首先,将车载以太网网络数据划分成 3 个等级,其中 A 等级为高优先级对应高实时数据流,一般是音视频数据流、车辆控制信号帧等;B 等级为次高优先级对应计算任务卸载类的数据流,A、B 等级数据均要进行整形,以保证实时性;BE 等级传输其他数据,其优先级最低不参与整形,采用传统以太网 Best Effort 机制发送,在整形数据中见缝插针式进行传输。如图 3 所示,整形算法建立在信用的累积和消耗上,在该类别信用值大于等于 0 时才可以传输,完成传输后信用值骤减为负值,之后按照该等级变化速率增加信用值。这样对 A、B 等级高实时性要求的数据流在单独时段占有带宽,BE 等级数据流用见缝插针式的方式进行整形,从而提高带宽的占用率,整体上提高总线数据传输实时性。流量整形效果如图 4 所示。

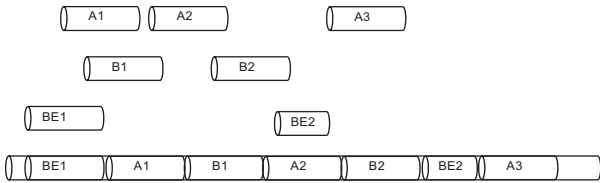


图 4 流量整形效果

2.2 延时特性

车载以太网延迟包含输入等待时延、存储和转发时延、干扰时延、传输时延等几个部分,其中,由流量冲突引起的干扰时延是主要时延^[10]。干扰时延包括高优先级流量阻塞时延 t_{HPB} 、低优先级流量阻塞时延 t_{LPB} 、流量整形时延 t_{TSB} 和数据帧传输时延 t_{trans} 。

对于 A 类数据,没有高优先级阻塞时延。其流量整形时延即等待信用值恢复至 0 的时间:

$$t_{TSB}^A = \left\lceil \frac{\text{loCredit}}{S_{idle}^A} \right\rceil \quad (1)$$

其中, loCredit 为信用值允许的最小值, S_{idle} 为信用值的增加速率,一般等于数据发送方预留带宽。在整形算法允许 A 类数据发送时,低等级数据帧可能正在传输,则需要等待低等级数据包发送完成,时延计算如下:

$$t_{LPB}^A = \frac{M_{B, BE}}{C} \quad (2)$$

其中, C 为以太网带宽, $M_{B, BE}$ 为当前低等级数据包剩余未完数据长度。A 类数据自身传输时延为:

$$t_{trans}^A = \frac{M_A}{C} \quad (3)$$

其中, M_A 为当前发送的 A 类数据帧长度。因此, A 类数据时延为:

$$W_A = t_{TSB}^A + t_{LPB}^A + t_{trans}^A \quad (4)$$

对于 B 等级的任务卸载数据帧来说,流量整形时延为:

$$t_{TSB}^B = \left\lceil \frac{\text{loCredit}_B}{S_{idle}^B} \right\rceil \quad (5)$$

低优先级流量阻塞时延计算为:

$$t_{LPB}^B = \frac{M_{BE}}{C} \quad (6)$$

其传输时延为:

$$t_{trans}^B = \frac{M_B}{C} \quad (7)$$

在 B 类数据帧经整形完毕可以发送时,考虑 A 类数据帧可能存在还未传输部分的长度,则产生高优先级流量阻塞:

$$t_{HPB}^B = W_A - t_{LPB}^B \quad (8)$$

因此,计算任务卸载类数据帧的时延为:

$$W_B = t_{TSB}^B + t_{HPB}^B + t_{LPB}^B + t_{trans}^B \quad (9)$$

3 基于车载以太网的任务卸载方案

3.1 卸载代价模型

设任务卸载率为 $0 \leq x_{i,j} \leq 1$, 表示域控制器 i 的第 j 个任务卸载到中央计算器的比例,剩余的 $1 - x_{i,j}$ 的计算任务留在域控制器本地进行计算。则本地时延为:

$$T_{i,j}^{\text{LOCAL}} = \frac{(1 - x_{i,j}) D_{i,j} c_k}{f_i} \quad (10)$$

其中, $D_{i,j}$ 表示该任务需要计算的数据长度, c_k 单位是 cycles/bit, 表示处理 1 bit 数据需要的指令周期,其由计算任务本身采用的算法决定; f_i 单位为 cycles/second, 表示域控制器 i 的计算能力。

本地计算的能耗 $E_{i,j}$ 计算公式为:

$$E_{i,j}^{\text{LOCAL}} = (1 - x_{i,j}) D_{i,j} c_k Z_i \quad (11)$$

其中, Z_i 单位为 J/cycle, 表示域控制器 i 单个 CPU 指令周期消耗的能量。

任务卸载的计算时延为:

$$T_{ij}^{\text{COM}} = \frac{x_{ij} D_{ij} c_k}{f_j^M} \quad (12)$$

其中, f_j^M 代表中央计算机分配给该任务 j 的计算能力。由式(9)可得任务卸载的传输时延为:

$$T_{ij}^{\text{TRAN}} = 2t_{\text{TSB}}^B + 2t_{\text{HPB}}^B + 2t_{\text{LPB}}^B + (1 + \varepsilon) t_{\text{trans}}^B \quad (13)$$

其中, ε 是计算结果与计算任务数据量的比例, 由于计算结果的数据长度一般远小于原始计算数据长度, 为易于分析验证, 这里取任务的平均比例均值 0.1。因此, 任务卸载总延时为:

$$T_{ij}^{\text{EDGE}} = T_{ij}^{\text{COM}} + T_{ij}^{\text{TRAN}} \quad (14)$$

任务卸载计算的总能耗为:

$$E_{ij}^{\text{EDGE}} = x_{ij} D_{ij} [c_k Z_M + (1 + \varepsilon) W_{ij}] \quad (15)$$

其中, Z_M 单位为 J/cycle, 表示中央计算机单个 CPU 指令周期消耗的能量, W_{ij} 单位为 J/bit, 表示以太网收发 1 bit 数据需要消耗的能量。

3.2 求解问题描述

由中央计算机对域控制器计算任务卸载的时延和能耗代价进行评估, 并通过比较本地和卸载的代价花费, 做出相应的卸载率 x_{ij} 、预留带宽 S_{idle}^B 和计算资源 f_{ij}^M 分配的决定, 实现车内边缘计算的延时代价最小化的目的。同时, 假设域控制 i 的设计最大能耗为 E_i^{MAX} , 计算卸载后的总能耗, 不能大于域控制器自身设计最大能耗的上限。优化目标函数如式(16)所示:

$$\begin{cases} \min \{ \max(T_{ij}^{\text{LOCAL}}, T_{ij}^{\text{EDGE}}) \} \\ \text{s. t. } C_1: x_{ij} \in [0, 1] \\ C_2: 0 \leq \sum_j f_{ij}^M \leq f^M \\ C_3: S_{\text{idle}}^A + S_{\text{idle}}^B < C \\ C_4: E_{ij}^{\text{LOCAL}} + E_{ij}^{\text{EDGE}} \leq E_i^{\text{MAX}} \end{cases} \quad (16)$$

这是一个涉及多变量的单目标优化问题, 梯度下降法^[11]是解决该类问题的传统算法, 但其在接近最优解的区域收敛速度明显变慢, 且容易陷入局部最优解。其他针对边缘计算的最优化算法包括深度确定性策略梯度算法^[12]、基于增强学习的 A3C 算法^[13]、内点法^[14]等。考虑到车内计算卸载的应用场景下中央计算机本身也是车载设备, 存在功耗、带宽等多个车载环境的复杂约束条件, 而遗传算法适于求解复杂系统问题^[15-16], 对问题的种类有较好鲁棒性, 这里采用遗传算法进行优化求解。

3.3 改进遗传算法求解

传统遗传算法收敛速度慢且容易局部收敛, 为解决这一问题将轮盘赌选择法与最优保存策略相结合, 保证算法的收敛性。种群由式(17)表示, 其中下标 a

为种群个体编码, b 为个体基因, 对应卸载率、预留带宽等解空间中的某一维解, 自变量为种群进化代数, 先建立初始化种群, 并对种群的解空间在约束条件下限 y_b^B 和上限 y_b^U 间进行随机赋值:

$$\{Y_a(0) \mid y_{a,b}(0) = \text{rand}(y_b^B, y_b^U)\} \quad (17)$$

计算种群个体适应度值, 用于衡量个体表征的解空间与最优化目标的接近程度, 由于目标函数是一个最小化函数, 因此适应度计算方程如下式所示:

$$\text{Fit}_a = f_{\text{max}} - f(y_{a,b}) \quad (18)$$

其中, f_{max} 是本代种群目标函数的最大值。种群的平均适应度为 f_{ave} , 若 $f_a > f_{\text{ave}}$, 则个体 a 予以保留, 其余个体采用轮盘赌选择法, 计算种群个体在下一代进化中被选入交配池的概率为:

$$P_a = \frac{\text{Fit}_a}{\sum_a \text{Fit}_a} \quad (19)$$

计算出每个个体的累积概率:

$$q_i = \sum_{a=1}^i P_a \quad (20)$$

在 $[0, 1]$ 区间产生一个随机数 r , 若 $r < q_i$, 则选择个体 $a = i$, 否则选择个体 a , 使得 $q_{a-1} < r < q_a$ 。不断重复直到交配池种群达到预定数量。由于进化初期的种群中的优秀个体适应度值大, 故交叉率和变异率比较低, 搜索容易陷入局部收敛^[17]。针对这个问题, 在保留高适应度个体的前提下, 增加在平均适应度附近的个体数量, 形成一个稳定的种群, 引入 S 型生长曲线函数作为交叉率和变异率的自适应调整曲线, S 型生长曲线上、下段变化平缓, 中间段变化接近线性变化且变化迅速:

$$P_c = \begin{cases} P_{c_{\text{max}}}, & f \leq f_{\text{ave}} \\ \frac{(e^{f_{\text{ave}}} - e^{f_{\text{min}}}) P_{c_{\text{max}}} P_{c_{\text{min}}}}{(e^{f_{\text{ave}}} - e^f) P_{c_{\text{max}}} + (e^f - e^{f_{\text{min}}}) P_{c_{\text{min}}}}, & \text{otherwise} \end{cases} \quad (21)$$

$$P_m = \begin{cases} P_{m_{\text{max}}}, & f \leq f_{\text{ave}} \\ \frac{(e^{f_{\text{ave}}} - e^{f_{\text{min}}}) P_{m_{\text{max}}} P_{m_{\text{min}}}}{(e^{f_{\text{ave}}} - e^f) P_{m_{\text{max}}} + (e^f - e^{f_{\text{min}}}) P_{m_{\text{min}}}}, & \text{otherwise} \end{cases} \quad (22)$$

式中, P_c 为交叉概率, P_m 为变异概率, e 为指数常数, f 为待变异的个体的适应度值; $P_{c_{\text{min}}}$ 、 $P_{c_{\text{max}}}$ 分别为交叉概率的最小值和最大值; $P_{m_{\text{min}}}$ 、 $P_{m_{\text{max}}}$ 分别为变异概率的最小值和最大值。设第 g 代交配的 3 个种群个体为 $Y_{r1}(g)$ 、 $Y_{r2}(g)$ 、 $Y_{r3}(g)$, 则个体变异结果为:

$$V_a(g+1) = Y_{r1}(g) + P_m(Y_{r2}(g) - Y_{r3}(g)) \quad (23)$$

根据交叉生成下一代种群:

$$Y_a(g+1) = \begin{cases} V_a(g+1), & \text{if } \text{rand}(P_{e_{\min}}, P_{e_{\max}}) \leq P_e \\ Y_a(g), & \text{otherwise} \end{cases} \quad (24)$$

当最优值已经连续若干代没有变化,或种群迭代数达到预设值时,结束算法迭代,得到求解结果。

4 仿真实验分析

使用 ANDi 软件结合 100Base-T1 的 Ethernet HIL 进行仿真测试,进行 2 个场景下的仿真实验:(1)保持等级 A 的音视频数据始终处于发送状态;(2)音视频数据停止发送。根据提出的改进遗传算法,求得仿真域控制器在两种情况下通过车载以太网卸载计算任务至中央计算机的任务执行时间,与不进行计算任务卸载的本地计算进行对比。实验参数设置如表 1 所示。

表 1 仿真参数设置

参数	数值
车载以太网总带宽/Mbps	100
域控制器个数	1~10
域控制器 CPU 计算能力/(cycles/s)	5×10^8
中央计算机 CPU 计算能力/(cycles/s)	6×10^9
等级 A 数据预留带宽/Mbps	60
各域控制器计算任务大小/MB	1
域控制器单指令周期能耗/(J/cycle)	7×10^{-9}
中央计算机单指令周期能耗/(J/cycle)	1.2×10^{-8}
以太网收发数据单位能耗/(J/bit)	2.1×10^{-11}
域控制器设计最大功率/W	4.8
等级 A 数据时间间隔/ μs	125
等级 B 数据时间间隔/ μs	250
等级 A 数据帧大小/KB	1 486
等级 BE 数据帧大小/KB	2

图 5 给出了在两组场景下改进遗传算法对时延最小化的目标求解的结果,域控制器数量为 10 个,其中 MGA-1 曲线对应等级 A 的音视频数据始终处于发送状态情况,MGA-2 曲线对音视频信号停发情况,LC (Local Computing)为本地计算策略,指计算任务全部在域控制器本地进行计算的延时,为固定值 0.22 秒。可以看出,改进遗传算法能够有效地求解该最小化问题。通过对比可看出,在音视频信号停发时,由于以太网总线上没有了高优先级流量阻塞时延的干扰因素,因此整个求解空间复杂度相对小,遗传算法在约 40 代时就趋近收敛,计算任务卸载后最小延时约 0.15 秒,相对于本地计算延时降低 31.8%;在引入等级 A 的音视频数据流量时,由于高优先级流量阻塞时延的影响,导致求解复杂度增大,遗传算法的收敛曲线波动变大且收敛速度变慢,到约 50 代时收敛,计算任务卸载后

最小延时约 0.18 秒,相对于本地计算延时降低了 18.2%。通过对比也可以看出在音视频数据停发时,由于有足够带宽卸载更多的计算任务到中央计算机,延时相对于有音视频数据时降低的更显著。

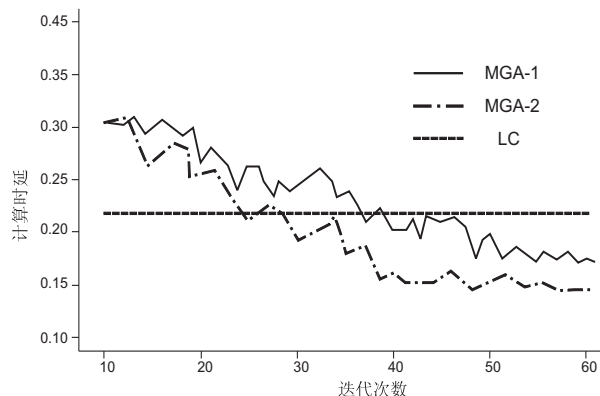


图 5 不同场景下的卸载仿真结果对比

将提出的 MGA 方法与梯度下降法(GD)、内点法(IP)和 A3C 算法进行比较,在存在等级 A 数据发送的场景下多次实验的平均结果如图 6 所示。

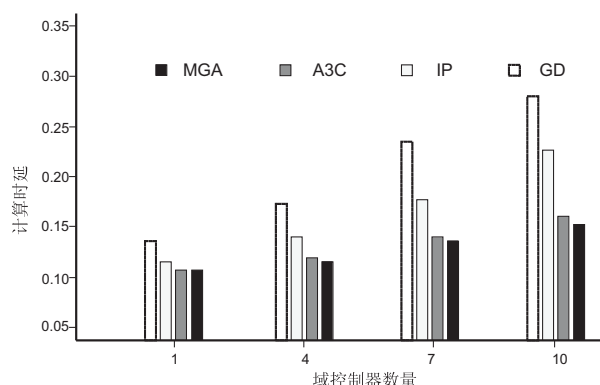


图 6 不同优化算法对比

由于车载以太网通信涉及流量整形的非线性因素,是一个非凸优化问题,因此梯度下降和内点法对卸载延时的优化效果并不理想,容易陷入局部收敛,特别是当车内网络的域控制器数量增加,系统非线性复杂度增大时其效果劣化程度越大,卸载延时呈现非线性增长。相比之下改进遗传算法与 A3C 方法能够保证该非线性优化问题的收敛,可看出延时增加基本与域控制器数量增加呈线性关系,总体来看 MGA 算法也要略优于 A3C 算法,多次实验能取得更低的卸载计算延时。

5 结束语

基于现代汽车新 E/E 架构的演化趋势,该文提出了车内边缘计算的方法,通过 100BASE-T1 高速车载以太网进行车内域控制器的计算任务卸载,充份发挥高性能车载中央计算机的计算性能。该方法通过对数据等级的分类和 AVB 协议延时特征的研究,能够在现

有以太网音视频数据业务正常使用的基础上,通过计算卸载方案显著提高车内计算任务的效率,降低计算延时,同时保证计算能耗不超出设计范围。改进遗传算法对于复杂问题空间的非线性问题求解的鲁棒性保证了对车内卸载任务调度问题的最优化求解,仿真结果证明了所提出的计算卸载策略的可行性和有效性,最优化求解的收敛结果在 AVB 协议数据流空闲时可降低 31.8% 的计算延时,在 AVB 协议数据流通信过程中也可降低 18.2% 的计算延时。在未来的工作中,将进一步考虑将车辆内外网络一体化进行计算任务集成协同的方法,实现更为有效的边缘计算卸载方案。

参考文献:

- [1] 刘健全. 基于高速移动网络的智能车视频驾驶系统设计与实现[D]. 杭州:浙江大学,2015.
- [2] SUN J, GU Q, ZHENG T, et al. Joint communication and computing resource allocation in vehicular edge computing [J]. International Journal of Distributed Sensor Networks, 2019, 15(3): 1-13.
- [3] 李波, 黄鑫, 牛力, 等. 车载边缘计算环境中的任务卸载决策和优化[J]. 微电子学与计算机, 2019, 36(2): 78-82.
- [4] 范艳芳, 袁爽, 蔡英, 等. 车载边缘计算中基于深度强化学习的协同计算卸载方案[J]. 计算机科学, 2021, 48(5): 270-276.
- [5] HUANG X, YU R, LIU J, et al. Parked vehicle edge computing: exploiting opportunistic resources for distributed mobile applications[J]. IEEE Access, 2018, 6: 66649-66663.
- [6] 李星宇, 余凯. 向超级中央计算机迈进——智能汽车电子电气构架变革迎接数字化重塑[J]. 微纳电子与智能制造, 2019, 1(2): 62-71.
- [7] 张宇, 高雅洁, 薛燕芳, 等. 基于环视系统的车载以太网时延评估与分析[J]. 江苏大学学报: 自然科学版, 2020, 41(4): 411-418.
- [8] 朱永健, 韩光省, 崔根群, 等. 基于 AVB 的车载千兆以太网交换机设计[J]. 现代电子技术, 2019, 42(23): 100-103.
- [9] 李二帅, 何锋, 熊华钢. AVB 网络流量整形模型端到端延迟计算[J]. 北京航空航天大学学报, 2017, 43(7): 1442-1449.
- [10] 洪俊. 车载以太网 AVB 数据传输时延特性数学模型研究[D]. 镇江: 江苏大学, 2020.
- [11] 孙娅楠, 林文斌. 梯度下降法在机器学习中的应用[J]. 苏州科技大学学报: 自然科学版, 2018, 35(2): 26-31.
- [12] NING Z L, ZHANG K Y, WANG X J, et al. Joint computing and caching in 5G-envisioned Internet of vehicles: a deep reinforcement learning-based traffic control system[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(8): 5201-5212.
- [13] QI Q, WANG J Y, MA Z Y, et al. Knowledge-driven service offloading decision for vehicular edge computing: a deep reinforcement learning approach[J]. IEEE Transactions on Vehicular Technology, 2019, 68(5): 4192-4203.
- [14] LIU P, LI J, LI H, et al. Convex optimisation-based joint channel and power allocation scheme for orthogonal frequency division multiple access networks[J]. IET Communications, 2015, 9(1): 28-32.
- [15] GHODOUSIAN A, BABALHAVA E J. An efficient genetic algorithm for solving nonlinear optimization problems defined with fuzzy relational equations and max-Lukasiewicz composition[J]. Applied Soft Computing, 2018, 69: 475-492.
- [16] GARAI G, CHAUDHURI B B. A novel hybrid genetic algorithm with Tabu search for optimizing multi-dimensional functions and point pattern recognition[J]. Information Sciences, 2013, 221: 28-48.
- [17] SRINIVAS M, PATNAIK L M. Adaptive probabilities of crossover and mutation in genetic algorithms[J]. IEEE Transactions on Systems Man & Cybernetics, 2002, 24(4): 656-667.