



西安理工大学学报
Journal of Xi'an University of Technology
ISSN 1006-4710, CN 61-1294/N

《西安理工大学学报》网络首发论文

题目：基于 NSGA2 的车联网边缘计算任务卸载方案
作者：刘建华，罗荣鑫，刘佳嘉，李炜，史小妮
收稿日期：2022-09-01
网络首发日期：2023-01-31
引用格式：刘建华，罗荣鑫，刘佳嘉，李炜，史小妮. 基于 NSGA2 的车联网边缘计算任务卸载方案[J/OL]. 西安理工大学学报.
<https://kns.cnki.net/kcms/detail//61.1294.n.20230130.1410.002.html>



网络首发：在编辑部工作流程中，稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定，且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式（包括网络呈现版式）排版后的稿件，可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定；学术研究成果具有创新性、科学性和先进性，符合编辑部对刊文的录用要求，不存在学术不端行为及其他侵权行为；稿件内容应基本符合国家有关书刊编辑、出版的技术标准，正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性，录用定稿一经发布，不得修改论文题目、作者、机构名称和学术内容，只可基于编辑规范进行少量文字的修改。

出版确认：纸质期刊编辑部通过与《中国学术期刊（光盘版）》电子杂志社有限公司签约，在《中国学术期刊（网络版）》出版传播平台上创办与纸质期刊内容一致的网络版，以单篇或整期出版形式，在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊（网络版）》是国家新闻出版广电总局批准的网络连续型出版物（ISSN 2096-4188，CN 11-6037/Z），所以签约期刊的网络版上网络首发论文视为正式出版。

基于 NSGA2 的车联网边缘计算任务卸载方案

刘建华, 罗荣鑫, 刘佳嘉, 李 炜, 史小妮

(中国民用航空飞行学院 航空电子电气学院, 四川 德阳 618000)

摘要: 针对如何降低基于车联网的边缘计算的时延与能耗, 提升移动用户体验的问题, 提出了一种在车联网边缘计算架构中基于非支配排序遗传算法 II(Non-dominated Sorting Genetic Algorithm II, NSGA2)算法的任务卸载方案。首先, 设计一个面向车联网的边缘计算体系, 路侧单元接受来自传输范围内车辆提供的信息, 再传输至移动边缘计算服务器进行处理。然后, 基于本地计算与计算卸载分别构建边缘计算的时延与能耗的数学模型。最后, 将体系中的能耗和时延作为决策目标, 以用户的总成本最小化作为目标函数, 设计基于 NSGA2 的算法进行多目标函数的优化。实验结果显示, 与其他方案相比, 所提出方案的收敛速度提高了 13% 左右, 总成本降低了 30% 左右。因此, 所提出的方案更适合用于车联网的场景中。

关键词: 车联网; 边缘计算; 非支配排序遗传算法 II; 计算卸载
中图分类号: TN914.5 **文献标志码:** A

Task offloading scheme of Internet of Vehicles edge computing based on NSGA2

LIU Jianhua, LUO Rongxin, LIU Jiajia, LI Wei, SHI Xiaoni

(School of Avionics and Electrical, Civil Aviation Flight University of China, De'yang 618000, China)

Abstract: Aiming at the problem of how to reduce the delay and energy consumption of edge computing based on Internet of Vehicles and improve the experience of mobile users, a task offloading scheme based on Non-dominated Sorting Genetic Algorithm II (NSGA2) algorithm in the edge computing architecture of Internet of Vehicles is proposed. First, an edge computing system for the Internet of Vehicles is designed, in which the roadside unit receives the data information provided by the vehicles within the transmission range, and then transmits it to the mobile edge computing server for processing. Then, mathematical models of delay and energy consumption of edge computing are constructed, respectively, based on local computing and computing offloading. Finally, taking the energy consumption and delay in the system as the decision-making objectives, and minimizing the total cost of users as the objective function, the algorithm based on NSGA2 is designed to optimize the multi-objective function. The experimental results show that compared with other schemes, the convergence speed of the proposed scheme is increased by about 13%, and the total cost is reduced by about 30%. Therefore, the proposed scheme is more suitable for the scenario of Internet of Vehicles.

Key words: Internet of Vehicles; edge computing; NSGA2; computing offloading

近年来, 随着汽车工业技术的不断发展和人们生活水平的不断提高, 汽车的拥有数量及其使用率逐年增高, 车联网(Internet of Vehicles, IoV)^[1]已成为改善道路拥堵、降低交通事故率以及为在途人员和设备提供信息计算和处理服务的一个关

键基础设施。移动边缘计算(Mobile Edge Computing, MEC)能为各种移动应用提供边缘计算服务, 在分担远程云端的计算压力的同时, 提高移动应用的反应速度, 并降低传输时延和能耗。随着物联网与人工智能的迅速发展, 车联网

收稿日期: 2022-09-01

基金项目: 四川省科技厅项目(2022JDKP0093); 四川省科技创新创业苗子工程重点项目(2022JDRC0076); 中央高校基本科研业务费基金项目(ZHMH2022-004, J2022-025)

第一作者: 刘建华, 男, 博士, 副教授, 硕导, 研究方向为边缘计算、信息安全。E-mail: jianhuacafuc13@cafuc.edu.cn

通信作者: 罗荣鑫, 男, 硕士生, 研究方向为边缘计算。E-mail: 492364397@qq.com

所需的计算数据量也更加庞大, 例如自动驾驶、路况识别、提升安全性等应用服务需要大量的计算和存储资源, 对时延和能耗的需求非常大, 目前车联网计算和通信能力等方面的不足, 这使得 IoV 在应用起来面临巨大的瓶颈。将移动边缘计算 (MEC) 技术应用到车联网当中^[2], 使得车辆与车辆、车辆与路侧单元 (RSU) 形成的接入网络中拥有计算、存储的能力, 可以大大节省数据远距离传输所带来的时延和能耗。同时, 车载的 MEC 服务器可以提供大量的计算与存储资源, 在提升本地计算速度的同时也丰富了车载资源。然而, IoV 复杂的场景模型也给移动边缘计算技术的应用带来了更多的问题。首先, 道路上高速移动的车辆带来的多普勒效应以及设置拓扑结构的多样性使得计算任务卸载的工作变得愈加复杂。其次, 在任务卸载中, 如何合理地卸载计算任务使得移动应用的计算任务更合理、高效地执行, 同时降低计算、传输所产生的时延和能耗, 使得成本较小是一个值得深入讨论的问题。

文献[4][5]将人工智能与 IoV 边缘计算相结合以期更好地解决边缘计算任务卸载中的资源分配和调度问题, 但他们的方案却存在时间复杂度过大、效率不高的问题。文献[6]引入了隐私保护作为目标并运用 BES 算法对其进行研究, BES 算法运算速度快但是其准确度却随着迭代次数的增加而降低。文献[13]则是在车联网边缘计算架构中采用深度学习中的双层深度网络来解决网络状态实时变化的问题, 但是所需要训练样本过大, 难以在较少训练样本的情况下实现近乎最优的效果。上述工作大都在较为理想的应用场景下进行研究并且都为单个目标却忽略了各优化目标之间的相互影响。基于以上考虑, 本文提出了一种车联网场景下边缘计算架构中基于 NSGA2 算法的任务卸载方案, 通过优化应用场景, 不断地改进卸载比例与遗传算子并与其他算法进行对比, 从而验证该方案的有效性。主要贡献如下:

1. 设计了车联网的边缘计算架构, 并将该架构分成多个部分。每个部分设置一个 RSU, 负责接收该区域的车辆卸载任务, 在将其传输至 MEC 服务器进行数据的处理、分析, 以期更加合理地利用本地和 MEC 服务器的计算资源。

2. 针对复杂的 IoV 应用场景, 在边缘计算架构中提出了计算模型与通信模型, 并以总时延和总能耗为目标函数, 构建多目标函数的最小化模

型。然后使用 NSGA2 算法对优化函数进行优化, 达到高效地进行任务卸载的目的。

3. 仿真结果显示使用该卸载方案和算法能达到使用户总成本函数较小化的实验效果。

本文的组织结构如下: 第 1 章介绍了相关工作。第 2 章描述了构建的模型并阐述了问题。第 3 章重点介绍了所用的算法的流程及设计细节。第 4 章是仿真结果的展示。第 5 章对本文进行了总结以及展望。

1 相关工作

车联网可以通过车对车 (V2V) 通信通过数据中继和协同计算相互协作, 建立了一个包含通信、计算、缓存和协同计算的统一框架, 并开发协同数据调度方案, 以在保证应用程序延迟约束的情况下最小化系统范围内的数据处理成本[3]。文献[4][5]将人工智能启发的计算、缓存和通信资源推到智能车辆附近, 从而在 IoV 框架中共同实现 RSU 对等卸载、车辆到 RSU 卸载和内容缓存。为了解决车辆网络中由于车辆有限的计算能力而导致的任务处理延迟和能量消耗过大的问题, 考虑了 MEC 的任务和特性, 并根据数据的大小 (即是否影响分割后的功能) 将任务划分为不可分割的任务和可分割的任务。然后文献[6]提出 BES 算法, 使物联网能更快的获取信息。为实现边缘节点的负载平衡, 文献[7]在 5G 网络体系结构下构建 IoV 计算卸载的边缘计算框架。提出了一个多目标优化问题来选择合适的目的地边缘节点。为了保证延迟敏感 IoV 服务的高可靠性完成, 文献[8]在边缘计算辅助 (EC-SDIoV) 中引入了部分计算卸载和带再处理机制的可靠任务分配, 并设计了一种启发式算法——容错粒子群优化算法。根据实际的通信环境, 文献[9]提出了一种基于模拟退火算法 (SAA) 的 Cellular Vehicle to Everything (C-V2X) 车联网 (IoV) 任务卸载和资源分配方案。为了满足车联网中车辆到一切 (V2X) 通信中信息交互的需求, 文献[10]采用了区块链技术, 来实现安全、分布式、计算能力增强的车联互联网。文献[11]在考虑以上问题的同时, 还引入了卸载效率、丢包率、负载分担率等, 该工作提出了软件定义虚拟网络 (Software Defined Virtual Network, SDVN) 下的边缘计算任务卸载和任务迁移的 IoV 模型, 采用 JDE-VCO 算法在一定程度上降低了系统总能耗。针对边缘计算高效计算的需

求, 文献[12]提出了深度强化学习与边缘计算中的任务卸载以及卸载算法相结合的方法, 解决网络多样性的问题。文献[13]针对 IoV 在能量转换设备会在过多的服务请求下过载, 从而降低服务质量 (QoS) 的情况下提出了一种基于深度强化学习的任务卸载方法 SOL, 并利用深度 Q 网络将深度学习和强化学习的价值函数相近似结合证明其有效性和适应性强的特性。

不幸的是, 上述近年来的研究较少有对任务卸载中产生的时延、能耗、隐私熵等目标进行多目标优化的。针对该问题, 本文在 IoV 的场景中建立边缘计算的模型, 并将车辆任务卸载产生总时延和总能耗作为目标函数, 运用 NSGA2 算法^[14]进行多目标的优化^[15], 使其达到用户总成本较小化的目的。

2 模型建立和问题描述

本节介绍边缘计算中的系统模型和其中任务卸载的计算模型, 所用到的主要符号及含义见表 1。

表 1 主要符号列表

Tab1 Main symbol list

符号	含义
U	需要进行任务卸载计算的车辆集合
V	移动边缘计算服务器的部署集合
W_k	车辆产生的任务数据量大小(bit)
S_k	完成计算任务所需要的 CPU 周期数(cycles)
f_k	对终端设备的计算能力(cycles/s)
ρ	比例系数
P_k	车辆到 MEC 服务器上行链路的传输功率(W)
h_k	车辆到 MEC 服务器上行链路的信道增益(db)
N_0	车辆到 MEC 服务器信道内部的高斯噪声功率(W)
T	任务卸载过程产生的总时延(ms)
E	任务卸载过程消耗的总能耗(J)

2.1 系统模型

首先建立一个车联网的系统场景, 道路上每一辆机动车装备有一台具有一定信息处理能力的设备以及无线传输模块, 道路的两侧部署了一定数量的路侧单元设备 (RSU), 用于接收并采集车辆所发出的任务处理请求和数据信息。每个路侧单元覆盖一块区域范围, 我们假定其覆盖范围的半径为 r , 目标车辆中心到路侧单元的垂直距离用 d 表示。在范围内的车辆就近将计算任务发送给 RSU, 每个 RSU 都配备有移动边缘计算服务器

(MEC), 然后再将其传输到移动边缘计算服务器 (MEC) 中进行复杂的数据计算处理与分析。最后, 将处理完成的数据信息直接传送给正在移动的车辆, 完成任务的反馈。

控制中心对边缘计算集中控制, 用来接收多个基站覆盖区域内的路侧单元和移动边缘计算服务器 (MEC) 发送的数据信息, 道路上的车辆也可以通过访问控制中心来获取前方道路的路况或单元信息。车联网场景建模: 设定需要卸载计算任务的车辆为选取的指定车辆, 用符号 k 表示, 那么所有需要进行任务卸载计算的车辆集合为

$U = \{U_k | k \in N^*\}$ 。在 V2V 卸载过程中, 车辆 k 的

卸载任务可能会经过很多个跃点, 最终在某个车辆上完成计算任务。设完成计算任务的车辆为目标车辆, 用符号 g 表示。但是本文只讨论 V2I 的卸载情况。在路旁部署的移动边缘计算服务器用符号 m 表示, 即部署集合可表示为 $V = \{V_m | m \in N^*\}$ 。具体的 IoV 系统的任务卸载框架如图 1 所示。

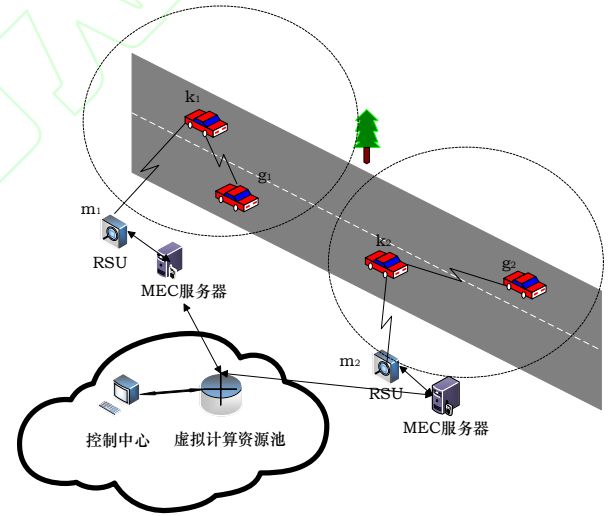


图 1 IoV 系统的任务卸载框架

Fig1 Task unloading framework of IOV system

假设道路上所有车辆均服从泊松分布, 假设车辆路过路侧单元设备使产生一个可卸载的计算任务 R , 可提取 R 的两个属性将其表示为 $R = \{w_k, s_k\}$ 。其中 W_k 表示为车辆产生的任务数据量大小, S_k 表示为完成计算任务 R 所需要的 CPU

周期数。

通过 RSU 的覆盖范围半径 r 以及设定的从目标单元到 RSU 的垂直距离 d 可计算出目标车辆从进入区域到离开区域所经过的总路程 s :

$$s = 2\sqrt{r^2 - d^2} \quad (1)$$

所以目标车辆通过单个 RSU 的覆盖区域所用的时间 t_b 为:

$$t_b = s / v_s \quad (2)$$

其中, v_s 代表目标车辆通过区域时的车速, 为简化模型, 假定车辆以恒定的速度前进。

最后定义任务卸载的比例因子, 将任务部分卸载到本地计算, 部分卸载到移动边缘计算服务器进行计算, 形成两者协同卸载计算的模型。定义任务卸载到本地的时延比例因子为 α , 卸载到本地的能耗比例因子为 β 。那么任务卸载到移动边缘计算服务器的比例因子分别为 $(1 - \alpha)$ 和 $(1 - \beta)$, 且满足 $0 \leq \alpha, \beta \leq 1$ 。

2.2 计算模型

本文主要讨论车辆计算任务本地卸载和 V2I 卸载过程两个部分, 具体计算模型如下:

1) 本地计算模型。如果车辆用户 k 选择在本地执行计算任务, 定义部分任务在本地计算的时延为 t_k^l , 因此时延 t_k^l 为:

$$t_k^l = \frac{w_k}{f_k} \quad (3)$$

其中, f_k 表示对终端设备的计算能力。

定义部分任务在本地计算消耗的能量为 e_k^l , 则 e_k^l 表示为:

$$e_k^l = \rho w_k \quad (4)$$

其中, ρ 表示比例系数, 由车辆自身结构决定。

2) 任务卸载模型。V2I 卸载过程主要经过三个步骤: 1. 车辆通过 V2I 通信方式将计算任务上传到 RSU 上; 2. RSU 相对应的 MEC 服务器对该计算任务进行计算; 3. MEC 服务器完成计算处理时, 将计算结果从 RSU 传回车载设备。整个卸载过程会产生任务数据传输、任务处理计算以及任

务回传三个部分的时延。但是上传的任务经过 MEC 服务器一系列的边缘卸载计算处理之后, 所回传给车辆的任务数据量将远小于原始的数据量, 下行链路传输速率较快, 占用带宽较小, 所需时延较短。故在建立卸载计算模型时可以将下行链路的回传时延忽略不计, 不带入公式进行计算^[16]。以下将详细介绍任务由车辆卸载到 MEC 服务器进行计算的时延和能耗模型。

因为任务数据的上传需要经过上行链路中的信道进行传输, 所以我们首先建立信道模型。对于无干扰的理想信道模型来说, 我们多采用时分多址接入 (TDMA) 或正交频分多址接入 (OFDMA) 技术, 即由车辆节点将任务卸载到

MEC 服务器过程的传输速率 r_k 可以根据香农公式^[17]定义为:

$$r_k = B \log_2 \left(1 + \frac{P_k h_k^2}{N_0} \right) \quad (5)$$

其中, B 表示车辆到 MEC 服务器的信道带宽;

P_k 表示车辆到 MEC 服务器上行链路的传输功率;

h_k 表示车辆到 MEC 服务器上行链路的信道增益;

N_0 表示车辆到 MEC 服务器信道内部的高斯噪声功率。

任务数据上传的时延 $t_k^{T_1}$ 和能耗 $e_k^{T_1}$ 分别定义为:

$$t_k^{T_1} = \frac{S_k}{r_k} \quad (6)$$

$$e_k^{T_1} = P_k \cdot t_k^{T_1} = P_k \cdot \frac{S_k}{r_k} \quad (7)$$

任务卸载到 MEC 服务器进行计算产生的时延 $t_k^{T_2}$ 和能耗 $e_k^{T_2}$ 分别定义为:

$$t_k^{T_2} = \frac{S_k}{f_k} \quad (8)$$

$$e_k^{T_2} = P_k \cdot t_k^{T_2} = P_k \cdot \frac{S_k}{f_k} \quad (9)$$

综合以上公式可得总时延 T 和总能耗 E 的定义分别为:

$$T = \frac{w_k}{f_k} + \frac{S_k}{f_k} + \frac{S_k}{B \log_2 \left(1 + \frac{P_k h_k^2}{N_0} \right)} + \frac{s}{v_s} \quad (10)$$

$$E = \rho W_k + P_k \frac{S_k}{f_k} + P_k \frac{S_k}{\text{Blog}_2 \left(1 + \frac{P_k h_k^2}{N_0} \right)} \quad (11)$$

然后在原有计算模型的基础上, 对信道模型以及计算资源分配问题进行进一步的改进, 使得模型更接近真实的车联网边缘计算架构, 具体改进细节如下:

改进一: 考虑存在干扰的信道模型, 采用码分多址接入 (CDMA) 技术, 根据香农公式, 此时的信道传输速率 r'_k 为:

$$r'_k = \text{Blog}_2 \left(1 + \frac{P_k h_k^2}{N_0 + \sum_{i \neq k} P_i h_i^2} \right) \quad (12)$$

此公式反映了不同设备间的互相干扰^[18]。

改进二: 考虑计算资源分配, 每个基站侧的 MEC 服务器可以同时向多个用户提供计算卸载服务。将分配策略定义为 $f_{m,k}$, 其中 $f_{m,k}$ 代表用户 k 卸载到 MEC 服务器 m 后分配的计算能力。

此时分配策略必须满足计算资源约束:

$$\sum_{k=1}^n f_{m,k} \leq f_m, \quad \forall m \in N^*$$

即改进后的总时延 T 和总能耗 E 分别为:

$$T = \frac{W_k}{f_k} + \frac{S_k}{f_{m,k}} + \frac{S_k}{\text{Blog}_2 \left(1 + \frac{P_k h_k^2}{N_0 + \sum_{i \neq k} P_i h_i^2} \right)} + \frac{s}{v_s} \quad (13)$$

$$E = \rho W_k + P_k \frac{S_k}{f_{m,k}} + P_k \frac{S_k}{\text{Blog}_2 \left(1 + \frac{P_k h_k^2}{N_0 + \sum_{i \neq k} P_i h_i^2} \right)} \quad (14)$$

再分别代入时延和能耗的比例因子, 组成多目标优化模型, 最终的目标成本函数 C 为:

$$\text{Min}(C) = \text{Min}\{\alpha t^L + (1 - \alpha)t^T + \beta e^L + (1 - \beta)e^T\} \quad (15)$$

$$\begin{aligned} \alpha, \beta &\in [0, 1], \forall U_k \in U, V_m \in V \\ 0 < P_k < P_{MAX}, \forall U_k \in U, V_m \in V \\ f_{m,k} &> 0, \forall U_k \in U, V_m \in V \end{aligned}$$

$$\sum_{k=1}^n f_{m,k} \leq f_{MEC}, \forall U_k \in U, V_m \in V$$

3 NSGA2 算法设计与分析

本文以任务卸载的时延和能耗作为两个决策变量建立计算模型, 此为多目标的优化模型, 对

该模型采用 NSGA2 算法进行优化。NSGA2 算法是一种多目标遗传算法, 用来分析并解决多目标优化的问题, 其核心思想是根据目标函数之间的关系, 找出使得各目标函数值都尽可能大或尽可能小 (视具体要求而定) 的函数值最优解集。具体原理及操作步骤见参考文献^[19]。

3.1 NSGA2 算法步骤

3.1.1 快速非支配排序

首先需要对种群个体中非劣解的水平进行判断, 根据其找出对应个体集合进行排序并标记非支配序值, 这是一个迭代的非支配排序的过程^[20]。比如, 在第一次迭代中, 对找出的非支配集合记为 F_1 , 标记集合中的非支配序值 $\text{rank} = 1$, 然后将

其去除并对余下的种群进行相同的操作, 找出 F_2 并将其个体标记 $\text{rank} = 2$, 循环往复, 直至整个种群被分层。

3.1.2 拥挤度

对种群分好层级后, 我们还需要对同一层目标空间的分布密度进行调节, 这一指标为个体的拥挤距离。而计算拥挤度距离的具体步骤为: 首先对同非支配层的个体距离初始化, 对处在排序边缘的个体给予优先选择。而对于位于中间的个体, 计算其拥挤度则按照公式 $i_d = \sum_{j=1}^m (|f_j^{i+1} - f_j^{i-1}|)$, 其中 i_d 表示个体 i 的拥挤度, f_j^{i+1} 表示 $i+1$ 点第 j 个目标函数的函数值^[21], 如图 2 所示。接着更换目标函数并重复以上步骤, 选择距离较大的个体可以防止个体局部的聚集, 来更好的保持种群多样性。

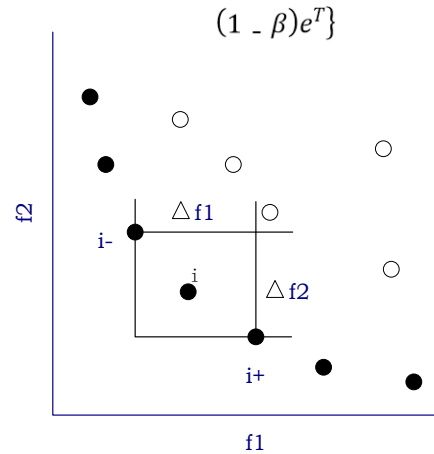


图 2 拥挤度的确定

Fig2 Determination of crowding degree

3.1.3 精英策略

将父代种群直接与其子代合并, 这样做的目的是避免 pareto 最优解丢失。再将合并后的种群依据非支配序值rank从低到高依次放入新的父代中, 直到放入某一非支配集 F_i 后, 此时种群数超过原始种群数。最后根据拥挤距离大小来筛选个体使其种群数不变^[22], 具体步骤如图3所示。

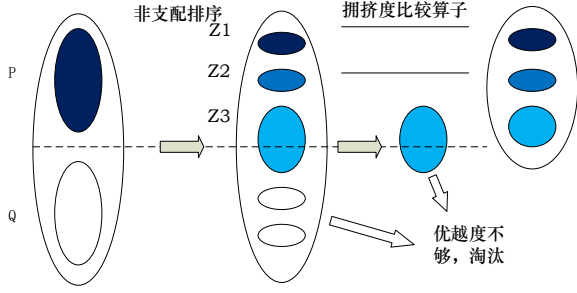


图3 精英策略执行步骤

Fig3 Elite strategy implementation steps

3.2 具体应用与分析

基于 NSGA2 的车联网边缘计算任务卸载。将车辆任务卸载比例当作种群中的个体, 不同的个体对应产生不同的时延 T 和能耗 E (根据公式(13)、(14)与卸载比例计算可得)。首先, 对初始种群进行快速非支配排序, 在第一轮排序中, 对于任意个体, 记录种群中时延 T 和能耗 E 都小于该个体的数量以及两者都比该个体大的个体集合。在第二轮及后面的排序中, 只需要对数量与集合进行处理即可判断个体是否为非支配解, 完成排序后记录每个个体的非支配等级。接着计算每个个体的拥挤度距离, 首先对每个子目标进行降序排序, 接着对排好序的种群个体计算其拥挤度距离, 其中时延作为横坐标, 能耗作为纵坐标将其数值带

入公式 $i_d = \sum_{j=1}^m (|f_j^{i+1} - f_j^{i-1}|)$ 进行计算, 得到

拥挤度算子。通过非支配等级与拥挤度算子共同筛选得到下一代种群, 将子代与父代种群相结合并通过精英策略与交叉、突变的遗传方式构造新的种群并保持种群数不变。

经过反复地迭代直至收敛并不再出现新的最优解集合来替代, 此时的最优解集合由第一非支配层的个体组成, 即通过 NSGA2 算法优化后得到的最优时延 T 和能耗 E 集合。再根据目前的最优解集合与其对应的卸载比例因子加权求和得到最优总成本 C (公式(15))。

3.3 复杂度分析

综合以上步骤, 对于快速非支配排序的复杂度为 $O(MN^2)$, 其中 M 为目标数, N 为个体数, 拥挤度计算的复杂度为 $O(MN \log N)$, 精英策略的复杂度为 $O(N \log N)$ 。

该算法的复杂度由以上三个部分组成, 所以算法的总复杂度为 $O(MN^2)$ 。

4 实验方案及分析结果

在本次实验中, NSGA2 算法的性能将在配置了 Intel(R) Core(TM) i5-7200U 处理器、4GB RAM 的 PC 机上进行模拟, 实现仿真的平台采用 MATLAB 2016a。实验场景我们考虑多个区域和多个 MEC 服务器的情况, 并且每个区域部署一个 MEC 服务器, 保证单元范围内的 IoV 用户分布均匀。另外还部署了中央控制器和虚拟计算资源池在整个网络中, 以调度范围内所有的基站以及 MEC 服务器。在所用符号和基本参数的大小如表2所示。这些模拟参数都是从许多实验以及丰富的经验中得来的, 并且参考了大量类似的案例。

表2 仿真参数

Tab2 Simulation parameters

参数	值	参数	值
P_i	$4 \times 10^8 W$	h_k	-20db
h_i	-15db	N_0	2000W
w_k	$20 \times 10^6 \text{bit}$	B	$9 \times 10^6 \text{Hz}$
S_k	$22.5 \times 10^6 \text{cycles}$	$f_{m,k}$	$2 \times 10^6 \text{cycles/s}$
f_k	$10 \times 10^6 \text{cycles/s}$	s	1000m
ρ	0.59	v_s	60km/h
P_k	$512 \times 10^6 W$		

4.1 车速对目标函数的影响分析

车速不是我们主要研究的变量, 但是车速度的变化会导致信道传输条件的变化, 从而间接的影响任务卸载及本地计算产生的时延和能耗。所以我们为了更好的研究车速对目标函数的影响, 模拟复杂多变的真实场景, 假定了三种不同的道路环境, 依次为乡间道路 (30km/h)、城市道路 (60km/h) 以及高速公路 (100km/h), 其迭代次数都是 200 次。如图4所示, 我们可以看出随着车速的逐渐增大, 边缘卸载所产生的能耗几乎没有变化, 所以车速对能耗的影响非常小。而车速与时延的关系如图5所示, 我们能看出时延随

着车速的增加而减少, 并且车速小时时延减少快, 速度大时时延减少慢。

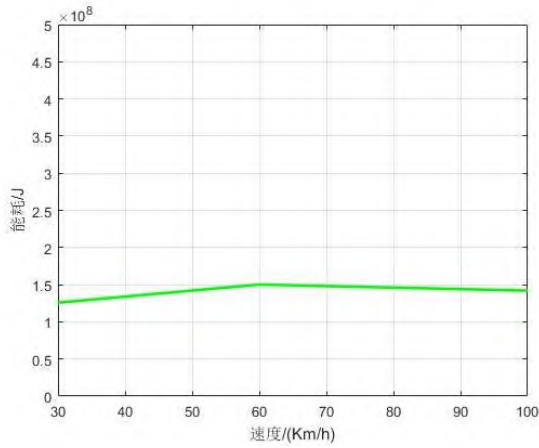


图4 车速与能耗的关系

Fig4 Relationship between vehicle speed and energy consumption

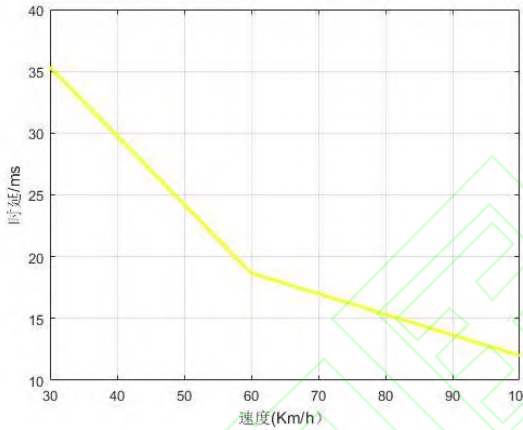


图5 车速与时延的关系

Fig5 Relationship between vehicle speed and time delay

4.2 收敛性分析

为了探究该卸载策略对多个目标函数值(能耗、时延)是否具有收敛效果, 我们进行了对比分析。如图 6-8 所示分别为迭代次数为 1 次、100 次、200 次的非支配解集, 即为能耗和时延都最优的解集。从图中我们可以看出随着迭代次数的增加, 能耗由最开始的 3.6×10^9 减少到迭代 100 次时的 1.807×10^8 再收敛到最后的 1.704×10^8 , 同理我们可以看到时延也由开始的 28.7 收敛到最后的 18.667。所以, 该卸载策略对该目标函数具有收敛性。

下面分析收敛速度以及迭代次数与收敛效果的关系。从图 9、图 10 可以看出, 随着迭代次数的增加能耗与时延都在减少, 并且收敛的速度逐渐减慢。开始迭代时收敛速度很快, 收敛期间能

耗和时延都有小幅度的波动, 然后收敛的速度逐渐减慢趋于平缓。能耗大约是在迭代 100 次左右收敛到最佳, 而时延是在 80 次左右收敛到最优的效果。

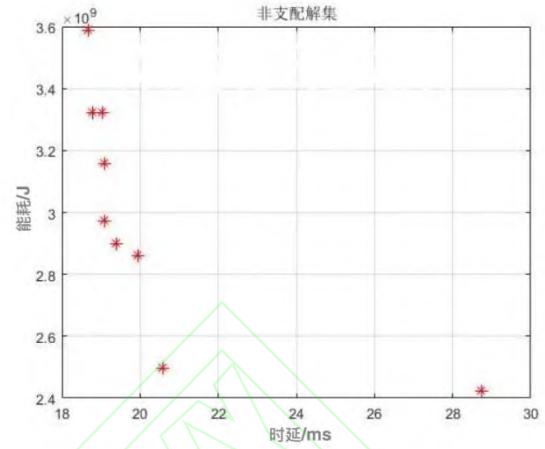


图6 迭代 1 次

Fig6 Iteration 1 time

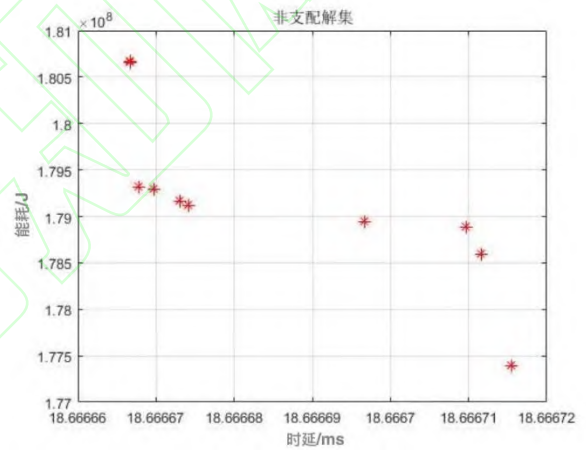


图7 迭代 100 次

Fig7 Iteration 100 times

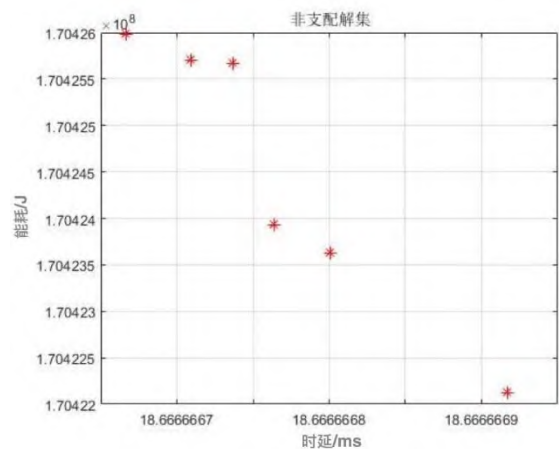


图8 迭代 200 次

Fig8 Iteration 200 times

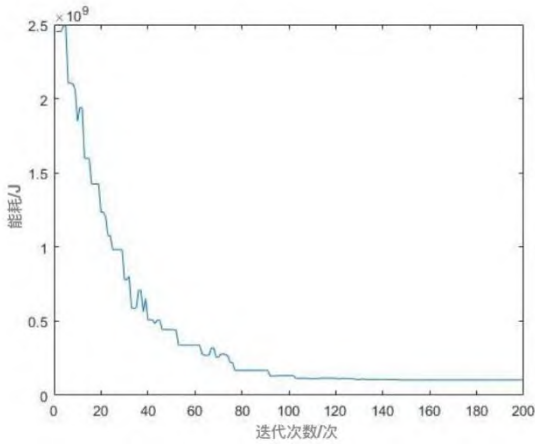


图9 迭代次数与能耗的关系

Fig9 Relationship between iteration times and energy consumption

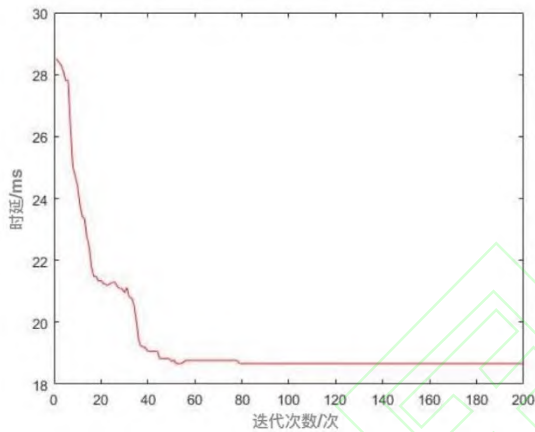


图10 迭代次数与时延的关系

Fig10 Relationship between iteration times and delay

4.3 与其他算法的对比分析

为了分析所提出卸载策略及算法的性能, 下面将与 SPEA2 算法^[23]、PESA2 算法^[24]以及 SA、GA、PSO 等算法^[25]进行对比。图 11 为迭代次数的收敛曲线, 图像曲线的收敛性都具有一定的波动, 可以看出虽然 SPEA2 算法和 PESA2 算法的收敛速度更快, 但是在达到最佳所用的收敛次数以及收敛的效果上, NSGA2 算法都是更好的。NSGA2 算法收敛到最佳的值大致为 1.704×10^8 , 比 SPEA2 算法的要好 30% 左右, 比 PESA2 算法的效果要好 36% 左右。具体各数值如表 3 所示。

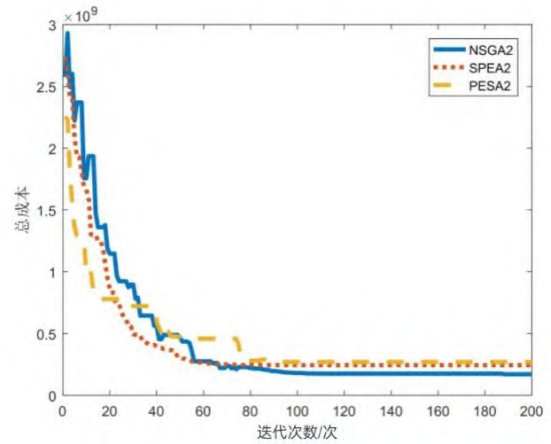


图11 NSGA2 算法与 SPEA2 算法对比

Fig11 Comparison between NSGA2 algorithm and SPEA2 algorithm

表3 各算法的总成本随迭代次数的变化

Tab3 The total cost of each algorithm varies with the number of iterations

(10 ⁸)	1 次	10 次	50 次	100 次	150 次	200 次
NSGA2	25.763	17.555	3.345	1.807	1.704	1.704
SPEA2	27.345	16.554	3.044	2.439	2.436	2.436
PESA2	22.546	11.138	4.728	2.706	2.694	2.694

影响总成本大小的还有边缘节点、任务量以及用户数量等因素。我们将三种算法分开比较, 进一步探究三种条件的影响以及不同算法的效果, 如图 12-14 所示。从图 12 可以看出, 随着车联网中边缘节点的增加, 数据计算能力也同样增加, 总成本逐渐下降, NSGA2 算法的优化效果始终优于 SPEA2 和 PESA2 算法。从图 13 可得, 随着任务量的增加, 总成本逐渐增加, 这是因为更多的任务量必然消耗更多的能量和需要更多的时延。当任务量较少时, PESA2 所需总成本最多, 随着任务量的逐渐增加, 三种方案所需的总成本都逐渐增加, 但 NSGA2 始终是最优的, 当任务量为 10×10^7 时, NSGA2 比其他两种方案所消耗的总成本至少低 18% 左右。从图 14 可得, 当用户数量为 10 时, 三种方案所消耗的总成本非常接近, 随着用户增加, 但 NSGA2 所消耗的总成本增加的最慢, 当用户数量增加到 30 时, NSGA2 比其它两种方案所需的总成本低 35% 左右, 当用户数量增加到 50 时, NSGA2 所需总成本至少比其他两种方案低 60%。综上, NSGA2 算法优化效果是最佳的。具体各算法下的影响因素与总成本的关系如表 4 所示。

在与 SA、GA、PSO 等算法的对比上, 由图

15 迭代次数的收敛曲线可以看出, 随着迭代次数的增加, 各种算法均在趋于收敛, 消耗的总成本均在降低, 显然, 对于不同的迭代次数, NSGA2 所需总成本始终最低。NSGA2 与 GA 均在 80 次左右收敛, SA 与 PSO 均在 90 次左右收敛, NSGA2 的收敛速度提高了 13% 左右, 并且所需总成本最

低。因此, 在收敛速度以及消耗总成本方面, NSGA2 算法优于其他算法。其主要原因是所提出的算法在延用了遗传算法的基础上还引用了拥挤度和精英策略的概念, 使得每次迭代选择的更加随机, 避免过早的陷入了局部最优的局面。

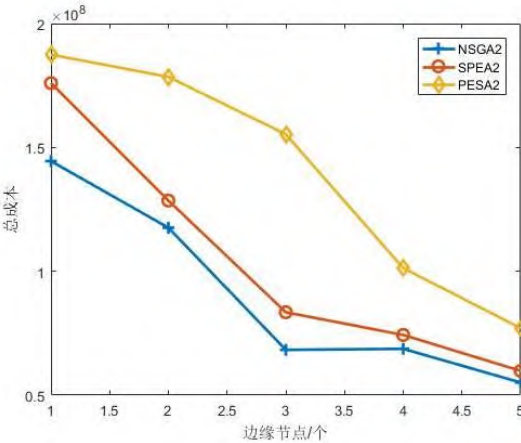


图 12 边缘节点与总成本的关系对比
Fig12 Comparison of the relationship between edge nodes and total cost

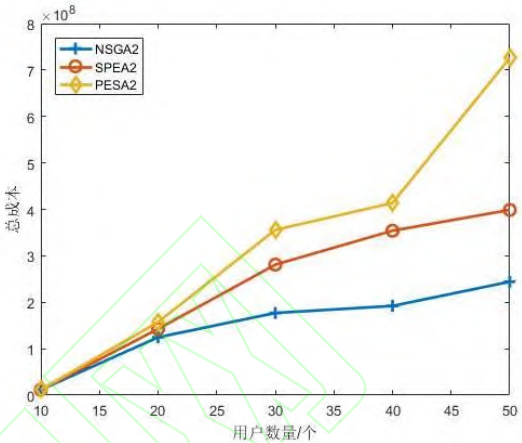


图 14 用户数量与总成本的关系对比
Fig14 Comparison of the relationship between the number of users and the total cost

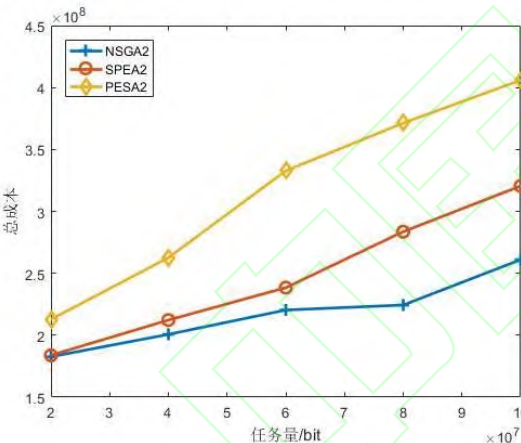


图 13 任务量与总成本的关系对比
Fig13 Comparison of the relationship between task volume and total cost

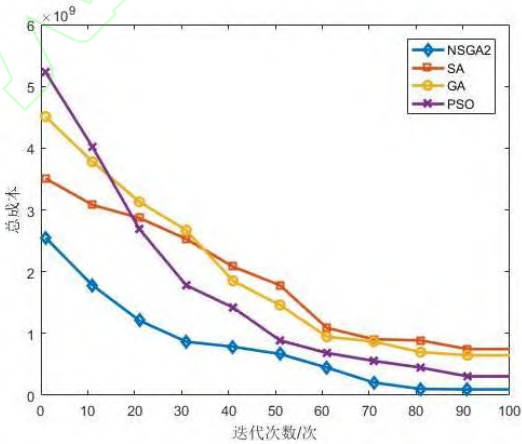


图 15 NSGA2 算法收敛性能与各算法比较
Fig15 Convergence performance of NSGA2 algorithm and comparison with other algorithms

表 4 各算法影响因素与总成本的关系

Tab4 Relationship between influencing factors of each algorithm and total cost

NSGA2	用户数量					边缘节点数					任务量 (10 ⁷)				
	10	20	30	40	50	1	2	3	4	5	2	4	6	8	10
总成本 (10 ⁸)	0.118	1.246	1.770	1.921	2.442	1.445	1.176	0.683	0.687	0.551	1.827	2.007	2.204	2.244	2.610
SPEA2	用户数量					边缘节点数					任务量 (10 ⁷)				
	10	20	30	40	50	1	2	3	4	5	2	4	6	8	10

总成本 (10^8)	0.118	1.425	2.810	3.541	3.987	1.760	1.285	0.835	0.743	0.598	1.838	2.122	2.384	2.836	3.205
PESA2	用户数量					边缘节点数					任务量 (10^7)				
	10	20	30	40	50	1	2	3	4	5	2	4	6	8	10
总成本 (10^8)	0.118	1.573	3.558	4.138	7.260	1.875	1.785	1.553	1.013	0.771	2.128	2.626	3.328	3.713	4.059

5 结 论

车联网的飞速发展, 其产生的网络数据也呈现爆炸式的增长。为了缓解信息处理的网络压力, 并且满足网络延迟在车联网应用中的需求, 我们提出了一种基于 NSGA2 算法的车联网边缘计算架构任务卸载策略。首先, 设计一个面向车联网的边缘计算体系, 路侧单元接受来自传输范围内车辆提供的数据信息, 再传输至移动边缘计算服务器进行处理。然后, 基于本地计算与任务卸载分别构建边缘计算的时延与能耗的数学模型。最后, 将体系中的能耗和时延作为决策目标, 考虑时延和能耗两大因素的基础上, 用 NSGA2 算法进行多目标的优化, 达到使用户总成本最小化的目的。实验仿真结果表明, 所提出的卸载策略能够快速收敛, 并且随着车辆的车速以及迭代次数的增加, 用户的总成本是在不断降低的。由此可以看出, 所提出方案具有较好的性能, 能很好地适用于车联网场景。另外, 用什么样的通信方式来提升车联网中的通信可靠性也是今后值得探究的问题。

参考文献:

- [1] 谷晓会, 章国安. 移动边缘计算在车联网中的应用综述[J]. 计算机应用研究, 2020, 37(6): 1615-1621.
GU Xiaohui, ZHANG Guoan. Overview of application of mobile edge computing in vehicle network [J] Computer Application Research, 2020, 37 (6): 1615-1621.
- [2] ZHANG J, LETAIEF K B. Mobile edge intelligence and computing for the internet of vehicles[J]. Proceedings of the IEEE, 2019, 108(2): 246-261.
- [3] Luo Q, Li C, Luan T H, et al. Collaborative data scheduling for vehicular edge computing via deep reinforcement learning[J]. IEEE Internet of Things Journal, 2020, 7(10): 9637-9650.
- [4] NING Z, ZHANG K, WANG X, et al. Intelligent edge computing in internet of vehicles: a joint computation offloading and caching solution[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(4): 2212-2225.
- [5] ZHANG Y, ZHANG M, FAN C, et al. Computing resource allocation scheme of IOV using deep reinforcement learning in edge computing environment[J]. EURASIP Journal on Advances in Signal Processing, 2021, 2021(1): 1-19.
- [6] ALMASHHADANI H A, DENG X, LATIF S N A, et al. An Edge-Computing Based Task-Unloading Technique with Privacy Protection for Internet of Connected Vehicles[J]. Wireless Personal Communications, 2021(1):1-22.
- [7] ZHOU P, CHEN X, LIU Z, et al. DRLE: Decentralized reinforcement learning at the edge for traffic light control in the IoV[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(4): 2262-2273.
- [8] HOU X, REN Z, WANG J, et al. Reliable computation offloading for edge-computing-enabled software-defined IoV[J]. IEEE Internet of Things Journal, 2020, 7(8): 7097-7111.
- [9] 李智, 薛建彬. C-V2X 车联网中基于模拟退火算法的任务卸载与资源分配[J]. 计算机应用, 2022, 42(10): 3140-3147.
LI Zhi, XUE Jianbin. Task offloading and resource allocation based on simulated annealing algorithm in C-V2X internet of vehicles[J]. Journal of Computer Applications, 2022, 42(10): 3140-3147.
- [10] LI W, JIN S. Performance evaluation and optimization of a task offloading strategy on the mobile edge computing with edge heterogeneity[J]. The Journal of Supercomputing, 2021, 77(11): 12486-12507.

- [11] LV Z, CHEN D, WANG Q. Diversified technologies in internet of vehicles under intelligent edge computing[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(4): 2048-2059.
- [12] WANG J, WANG L. Mobile edge computing task distribution and offloading algorithm based on deep reinforcement learning in internet of vehicles[J]. Journal of Ambient Intelligence and Humanized Computing, 2021, 8: 1-11.
- [13] 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.
- [14] 佚名. NSGA-II 的改进算法研究[D]. 燕山大学, 2013. <https://www.docin.com/p-1967125919.html>.
Anonymous. Research on improved algorithm of NSGA-II [D]. Yanshan University, 2013.
- [15] 周研. 光网络组播路由多目标进化算法研究[D]. 湖南大学, 2018.
ZHOU Yan. Research on Multi-objective Evolutionary Algorithm for Multicast Routing in Optical Networks [D]. Hunan University, 2018.
- [16] JIANG H, LIU B, WANG Y, et al. Multiobjective TOU pricing optimization based on NSGA2[J]. Journal of Applied Mathematics, 2014: 104518-104526.
- [17] WANG R. An improved nondominated sorting genetic algorithm for multiobjective problem[J]. Mathematical Problems in Engineering, 2016: 1519542-1519549.
- [18] LIN H, ZEADALLY S, CHEN Z, et al. A survey on computation offloading modeling for edge computing[J]. Journal of Network and Computer Applications, 2020, 169(1): 102781-102789.
- [19] 李莉. 基于遗传算法的多目标寻优策略的应用研究[D]. 无锡: 江南大学, 2008.
LI Li. Application research on the multi-objectives optimization based on genetic algorithm[D]. Wuxi: Jiangnan University, 2008.
- [20] 张宁. 柴油/天然气发动机燃烧过程优化研究[D]. 北京交通大学, 2018.
ZHANG Ning. Research on combustion process optimization of diesel/natural gas engine[D]. Beijing Jiaotong University, 2018.
- [21] 沈俊宇. 区域冷链物流配送中心选址及配送路径优化研究[D]. 西南交通大学, 2018.
SHEN Junyu. Research on site selection and distribution route optimization of regional cold chain logistics distribution center[D]. Southwest Jiaotong University, 2018.
- [22] 孙志伟. 电火花加工参数优化的研究[D]. 华南理工大学, 2011.
SUN Zhiwei. Research on optimization of EDM parameters[D]. South China University of Technology, 2011.
- [23] BI J, YUAN H, DUANMU S, et al. Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization[J]. IEEE Internet of Things Journal, 2020, 8(5): 3774-3785.
- [24] JAIN A, LALWANI S, LALWANI M. A comparative analysis of MOPSO, NSGA-II, SPEA2 and PESA2 for multi-objective optimal power flow[C]//2018 2nd International Conference on Power, Energy and Environment: Towards Smart Technology (ICEPE). IEEE, 2018: 1-6.
- [25] JEET S, BARUA A, BAGAL D K, et al. Comparative investigation of CNC turning of nickel-chromoly steel under different cutting environment with a fabricated portable mist lubricator: a super hybrid taguchi-WASPAS-GA-SA-PSO approach[M] //Advanced Manufacturing Systems and Innovative Product Design. Springer, Singapore, 2021: 515-531.