UDC _____ 编号 _____

雲南大學

硕士研究生学位论文

题 目 车联网环境下能耗优先的任务调度算法研究

Title Research on Energy Priority Task Scheduling Algorithm in the Vehicle Internet Environment

学院（所、中心） 信息学院

专 业 名 称 计 算 机 软 件 与 理 论

研 究 方 向 智 能 算 法

研究生姓名 XXX 学号 12021115016

导 师 姓 名 XX 职称 教授

2023 年 5 月

<table>
<tr><td rowspan="12">云南大学硕士研究生学位申请简况表</td><td rowspan="6">论文预审</td><td colspan="4">论文预审结果：</td></tr>
<tr><td>专家姓名</td><td>职称</td><td colspan="2">所在单位（校内：学院/校外：所在单位）</td></tr>
<tr><td></td><td></td><td colspan="2"></td></tr>
<tr><td></td><td></td><td colspan="2"></td></tr>
<tr><td></td><td></td><td colspan="2"></td></tr>
<tr><td></td><td></td><td colspan="2"></td></tr>
<tr><td rowspan="5">论文送审</td><td>专家姓名</td><td>职称</td><td>所在单位（校内：学院/校外：所在单位）</td><td>结果</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="6">论文答辩</td><td colspan="4">答辩结果：</td></tr>
<tr><td>答辩专家</td><td>职称</td><td colspan="2">所在单位（校内：学院/校外：所在单位）</td></tr>
<tr><td></td><td></td><td colspan="2"></td></tr>
<tr><td></td><td></td><td colspan="2"></td></tr>
<tr><td></td><td></td><td colspan="2"></td></tr>
<tr><td></td><td></td><td colspan="2"></td></tr>
</table>

论文独创性声明及使用授权

本论文是作者在导师指导下取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，不存在剽窃或抄袭行为。与作者一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

现就论文的使用对云南大学授权如下：学校有权保留本论文（含电子版），也可以采用影印、缩印或其他复制手段保存论文；学校有权公布论文的全部或部分内容，可以将论文用于查阅或借阅服务；学校有权向有关机构送交学位论文用于学术规范审查、社会监督或评奖；学校有权将学位论文的全部或部分内容录入有关数据库用于检索服务。

（内部或保密的论文在解密后应遵循此规定）

研究生签名：_____ 导师签名：_____ 日　期：_____

目录

摘要

鉴于当今我国人口规模庞大的基本国情，医疗系统一直承受着巨大压力。特别是新冠病毒的袭来，医院人满为患，医护人员疲于奔命，不仅自身健康得不到保证，工作效率还会随着疲劳增加而降低。虽然抗原试剂的出现可以帮助人们在家自我检测，但抗原试剂有检测结果滞后性这个弊端。对于那些本身就患有基础性疾病的老年患者来说，越早发现，生命也就多一分保障。因此结合计算机辅助诊断势在必行也迫在眉睫。本文基于卷积神经网络技术搭建一套肺部 CT 图像识别模型。以下几个方面为本文主要进行的工作：

首先是在肺部 CT 图像分类任务中，过往的模型由于参数众多，计算量大而导致等待时间过长，不适合应用在实际中。本文采用的 Inception-ResNet 模型结合 GoogLeNet 和 ResNet 两大网络的优点，通过多个不同尺寸的卷积核进行信息提取的同时，借鉴残差模块的思想，防止模型出现过拟合。

然后在图像分割任务方面，本文主要选用 U-Net 网络作为基础网络框架，因为在医学图像的分割效果上，U-Net 取得了不错的效果。本文吸收了迁移学习的思想，选择 effentnet 族 CNN 模型作为编码器模块。利用预训练得到的信息解决新问题，同时需要的数据量更少。由于传统 U-Net 网络提取的浅层特征图有很多的冗余信息，会浪费大量的计算时间，引入注意力机制后，网络模型会在大量信息传输给解码器之前，找到当前任务中最重要的信息，通过抑制非相关信息的特征去提高网络的性能。本文改进后的 U-Net 模型与 CNN、FCN 等模型进行比较，可以发现模型精度提高的同时切割出的病灶部位特征也更明显。正好能弥补 Inception-ResNet 模型对于某些病灶部位模糊而导致误判普通肺炎和新冠肺炎的缺点。

关键词：肺炎；卷积神经网络；图像语义分割；肺炎 CT 图像识别

Abstract

Given the basic conditions of our country today with a large population, the healthcare system has been under tremendous pressure. Especially with the onset of the new coronavirus, hospitals are overcrowded and health care workers are overwhelmed, not only their own health is not guaranteed, but their work efficiency also decreases as fatigue increases. Although the advent of antigen reagents can help people to self-test at home, antigen reagents have the disadvantage of lagging test results. For elderly patients with underlying diseases, the earlier they are detected, the more life is guaranteed. Therefore, it is imperative and urgent to incorporate computer-aided diagnosis. In this paper, we build a lung CT image recognition model based on convolutional neural network technology. The following aspects are the main work carried out in this paper:

The first is that in the lung CT image classification task, the past models are not suitable for application in practice because of the long waiting time due to numerous parameters and large computation. The Inception-ResNet model used in this paper combines the advantages of two major networks, GoogLeNet and ResNet, to extract information through multiple convolutional kernels of different sizes while drawing on the idea of residual modules to prevent the model from overfitting.

Then, for the image segmentation task, this paper mainly selects the U-Net network as the basic network framework, because U-Net has achieved good results in the segmentation effect of medical images. In this paper, we absorb the idea of migration learning and choose the effentnet family CNN model as the encoder module. The information obtained from pre-training is used to solve the new problem while requiring less data. Since the shallow feature map extracted by the traditional U-Net network has a lot of redundant information, which will waste a lot of computation time, after introducing the attention mechanism, the network model will find the most important information in the current task before a large amount of information is transmitted to the decoder, and go to improve the performance of the network by suppressing the features of non-relevant information. In this paper, the improved

U-Net model is compared with CNN, FCN and other models, and it can be found that the model accuracy is improved while the features of the cut-out lesion sites are more obvious. The Inception-ResNet model is able to compensate for the disadvantage of misclassification of common pneumonia and neocoronary pneumonia due to the ambiguity of some lesion sites.

Key Words: Pneumonia； Convolution neural network； Image semantic segmentation；Pneumonia CT image recognition

第 1 章绪论

本章主要讲述了车联网中任务优先调度优先算法的意义，以及其研究现状和成果，提出了本文所要做的任务，阐述了本文的主要研究内容和介绍了本文的组织结构。

1.1 研究背景及意义

未来将是“万物互联”的时代，随着射频识别技术、网络协议、数据存储以及传感器技术日新月异的发展，越来越多的物理机器与互联网连接起来，实现了物体之间的信息交互，以及赋予了机器以“智能”。这将深刻地改变人们的生产生活方式，为人们提供更加便利、智能和高效的生活和工作体验。

车联网是“万物互联”时代的典型应用，汽车的功能越来越超出其原本的范围，不再只是交通出行的工具，而变成了一个智能互联的计算系统，车辆将承担起对车辆自身的调度与管理的任务，以及满足影音娱乐等用户额外的需求。在对车辆自身进行调度与管理的任务中，有些任务对计算资源的要求较高，且对时延敏感，因此需要大量的稳定的计算资源来保证其服务质量。显而易见，车载处理器的算力受限于汽车电池，计算资源有限，无法满足对互联的计算系统的要求。

为了解决算力受限的问题，基于云的解决方案通过将任务上传到云端（远端、远程云）来进行处理（如图 1），云端本身就是一个计算中心，通常具有超大的规模，能赋予用户超强的计算能力，对大规模数据聚合、数据挖掘，分析、优化和存储复杂数据，云端可以在短的时间内计算得到结果。

车辆应用产生的计算任务可以通过网络传输到远程云，在云端计算并将结果返回给车辆。由于云端超强的计算能力，算力能满足车辆任务的需求，因此能够计算和分析复杂的计算任务，并能够提供较高的计算性能，但是因为云端位于网络的远端，将车辆计算任务传输到远程云并得到相应的返回结果需要很大的传输时间开销。因此对于数据密集型和延迟敏感型的计算任务，即使计算延迟较小（数据量较小），较大的传输延迟仍会使任务总卸载延迟较大。同时，为了将车辆产生的大量数据全部上传到云端，这将会导致网络中的数据量大幅增加，占用大量的网络带宽，使得网络设施负担加重，从而造成网络性能和稳定性的下降。

图：将任务分配给云服务器

为了解决云计算的时延以及流量问题，移动边缘计算（MEC）选择了在网络边缘中（边缘端）处理数据，与传统的云计算相比，边缘计算将服务部署在距离用户更近的地方，将计算服务放置到网络的边缘，可以实现更低的时延和更高的带宽，减少了核心网络的负载，并且有利于数据安全以及隐私保护。

边缘计算也可以通过与物联网技术结合，实现一些与智能车联网的应用，如自动驾驶、智慧城市等【Joint task management in connected vehicle networks by software-defined networking, computing and caching】。美国自然基金委和英特尔在 2016 年开始合作讨论建设无线边缘网络。同一年，中国也兴起了移动边缘计算的研究，在科研院所、高等院校、企业、行业协会等各方单位共同努力下，边缘计算产业联盟成立。

边缘端位于云端和车辆之间，它由拥有计算能力的边缘设备组成，它是移动边缘计算中的重要组成部分，是解决远程云传输延迟高的一种有效替代方案。边缘云可由边缘服务器、RSU、BS、无线接入点（Access Point，AP）以及移动设备等其他具备计算、存储和通信能力的边缘设备组成。因为边缘云处在网络的边缘，更靠近车辆，车辆到 EC 的传输延迟可以大大降低。因此边缘云可以为车辆提供低延迟计算、高速缓存、位置感知、紧急管理等服务，并能用于实时交互，对于数据密集型和延迟敏感型的车辆计算任务，例如：增强现实，实时视频分析，

人类行为识别等任务，它们需要极低的延迟以便快速响应和决策，此时边缘云比远程云具有更大的优势。

边缘计算并不是云计算的反面，边缘计算是云计算的延伸，未来将与云计算协同配合，为用户提供更高质量的服务。

为了使车联网支持云计算和边缘计算，科研组织和联盟实现了车到万物通信(V2X)技术，V2X 主要包括车辆对车辆通信(V2V)、车辆对基础设施通信(V2I)。

任务卸载是指将移动设备（如智能手机、车载终端等）上的计算任务分解成若干个子任务，其中一部分在本地完成，另一部分则通过网络卸载到云端或者其他设备上进行计算，最终将结果返回到本地设备，从而提高移动设备的计算性能和能耗效率的一种技术。

根据卸载的目标，任务卸载可以分为以下两类：

（1）计算卸载：主要目的是将计算密集型任务卸载到云端或其他远程设备上，减少本地设备的计算负担，提高计算性能和能耗效率；

（2）通信卸载：主要目的是将数据密集型任务卸载到云端或其他远程设备上进行处理，减少本地设备的数据传输负担，提高通信性能和能耗效率。

在本文中，主要研究的是车联网中的通信卸载。在车联网中，任务卸载主要应用于以下几个方面：

（1）智能驾驶：智能驾驶需要大量的计算因此需要强大的数据处理能力，通过任务卸载技术，可以将一部分计算和数据处理任务卸载到云端或其他设备上，提高计算性能和能耗效率。

（2）车辆监控：车辆监控需要实时收集和处理大量的数据，并进行数据分析和决策，通过任务卸载技术，可以将一部分数据处理任务卸载到云端或其他设备上，提高数据处理效率。

（3）车辆诊断：车辆诊断需要进行大量的数据分析和处理，通过任务卸载技术，可以将一部分数据处理任务卸载到云端或其他设备上，提高诊断效率和精度。

智能车辆的任务并不是都可以卸载的，车辆任务按照其关键程度分为三类：关键任务（Crucial Tasks，CTs）、高优先级任务（High-Priority Tasks，HPTs）和低优先级任务（Low-Priority Tasks，LPTs）。

CTs 是和车辆安全相关的应用，是保证车辆和乘客安全的关键应用程序，如车辆控制、系统监控和事故预防等。由于 CTs 和安全紧密相关，因此享有最高的优先级，必须保留充足的计算资源给它，不能因为 HPTs 和 LPTs 的存在而影响 CTs 的正常运行，因此这类任务也不允许卸载，只允许在本地执行，不属于计算任务卸载的范畴。该类任务的实例是：车辆控制、碰撞预警、红绿灯警告、网上车辆诊断、道路湿滑程度检测等。

HPTs 包括与驾驶相关的应用和可选的安全增强应用，这类应用程序对车辆而言是重要但不是必须的，拥有较高的执行优先级，例如实时路径规划和路况提醒等。这类应用允许出现延迟或卸载失败的情况，但不能影响 CTs。该类任务的实例是：地图导航、平视显示器、视野增强、车辆传感等。

LPTs 是一类为用户提供影音娱乐服务的应用程序，它的优先级较低，例如语音识别，它允许驾驶员发出各种声音命令，通过语音识别命令计算机做一些响应，而不会使驾驶员分心。该类任务的实例是：虚拟现实、语音识别、视频处理、在线游戏等。

HPTs 和 IPTs 已经被部署到越来越多的车辆上，由于 HPTs 和 LPTs 不会涉及到安全，因此可以将其进行卸载，来提高资源的利用率。

1.2 国内外研究现状

1.3 研究的主要内容

本文主要研究了车联网任务卸载算法，在车辆任务不能完全卸载情况下，即有部分任务必须车辆在本地执行的情况下,

考虑了车辆移动性对无线信道以及任务卸载决策和资源分配的影响，设计了一种权衡时延和能耗的效用评价函数，提出了一个联合任务卸载和计算资源分配的优化问题，提出了基于数值方法的计算资源分配算法和一种启发式任务卸载算法。然后，在车辆任务部分卸载情况下，针对多用户多服务器场景，根据系统总时延以及服务器均衡目的设计了一种效用函数，并提出一个联合任务卸载、计算资源分配和卸载率的系统效用最大化问题，接着将联合问题分解为任务卸载问题、计算资源分配问题以及卸载率优化问题，最后提出了一种基于 PSO 的任务卸载算法，并根据拉格朗日乘子法和 KKT 条件求解最优计算资源和卸载率

。

1.4 论文组织结构

论文的具体内容安排如下：

第一章，绪论。本章对全文内容进行概述和章节安排，首先介绍了车联网和移动边缘计算研究背景和意义，接着简述了边缘计算中任务卸载的国内外研究现状，然后对全文内容进行概述，最后给出论文组织章节安排。

第二章，相关背景知识介绍。本章首先对车联网架构和通信范式、车联网关

键技术以及车联网相关应用和服务进行概述，接着阐述了移动边缘计算的概念和架构，最后介绍了常用的移动边缘计算任务卸载和资源分配方法。

第三章，车联网中基于 MEC 的移动感知多用户任务卸载。本章在车辆任务完全卸载情况下，考虑了车辆移动性对无线信道以及任务卸载决策和资源分配的影响，设计了一种权衡时延和能耗的效用评价函数，提出了一个联合任务卸载和计算资源分配的优化问题，提出了基于数值方法的计算资源分配算法和一种启发式任务卸载算法。最后，通过仿真验证了所提算法的有效性。

第四章：基于改进的 U-Net 网络的肺炎 CT 图像语义分割研究。首先对图像进行增强，之后将增强后的数据带入改进的网络里。最后将得到的结果与 CNN、FCN 网络的结果进行对比，验证改进的网络的性能。

第五章：全文总结与展望。

第 2 章车联网中能耗优先的任务调度算法研究

2 车联网的物理设备

与车联网有关的物理设备包含了多个组成部分，包括车辆、路边设施（RSU）、数据中心等，可以实现车辆之间、车辆和路边设施之间、车辆和数据中心之间的数据共享和协同处理。

车联网中的车辆可以通过各种传感器收集大量的数据，包括车辆状态、位置信息、驾驶行为、环境信息等等。这些数据可以通过车载计算机进行处理和分析，然后传输到数据中心或其他车辆进行分析以进行共享和协同处理。

路边设施也可以通过传感器收集相关数据，例如交通信号灯、路况监测设备等。这些数据可以和车辆收集的数据一起传输到数据中心进行处理和分析。

数据中心作为车联网的核心，它可以接收、处理、存储和共享车辆和路边设施收集的数据。在数据中心中，可以使用各种计算模型进行数据分析、模型训练和预测等。

车联网的数据中心由远程云、边缘云和车辆云三层云结构组成。其中远程云是传统云计算中的云资源，边缘云位于网络的边缘，可以提供更低延迟和更高带宽的服务，车辆云与边缘云通过无线通信连接。

远程云（Remote Cloud）通常具有超大的规模，能赋予用户超强的计算能力，对大规模数据聚合、数据挖掘，分析、优化和存储复杂数据，远程云可以在短的时间内计算得到结果。车辆应用产生的计算任务可以通过网络传输到远程云，在远程云计算并将结果返回给车辆。由于远程云超强的计算能力，因此能够计算和分析复杂的计算任务，并能够提供较高的计算性能，但是因为远程云位于网络的远端，将车辆计算任务传输到远程云并得到相应的返回结果需要很大的传输时间开销。因此对于数据密集型和延迟敏感型的计算任务，即使计算延迟较小，较大的传输延迟仍会使任务总卸载延迟较大。同时，车辆产生的大量数据全部上传到远程云会占用大量的网络带宽，从而造成网络性能和稳定性的下降。

车辆云（Vehicular Cloud，VC）是汽车组成的资源系统，随着汽车工业

中技术的演变和智能网联车辆的发展，车辆已被赋予更多的计算、存储、通信和传感资源。城市地区有大量的车辆，将空闲的车辆资源加以利用，可以为移动设备提供巨大的资源和价值，通过整合车辆空闲的计算资源形成车辆云，为其他车辆或者移动用户提供服务，可以大大提高服务和应用的质量。智能网联车辆包含使用无线通信议的设备，车辆能以较低的通信延迟接发计算任务，充分利用车辆上空闲的计算、存储资源，在保证车辆安全行驶的前提下，降低任务处理延迟，提高用户的驾驶体验。车辆云中车辆的角色不是固定的，当车辆资源空闲时可以作为服务提供者，接受来自其他车辆或是移动设备的任务请求；当车辆需要卸载任务时，可以作为服务请求者，将任务卸载至其他拥有空闲计算资源的车辆，来实现资源的整合和资源利用率的提高。

2.2 通信技术

2.2.1 车辆间通信

车联网中的设备需要通过无线通讯网络来进行通讯。

计算任务卸载过程需要通过无线通讯网络将计算任务及其相关参数从服务请求者传输至服务提供者。

随着车联网的发展，车载网络技术已经相对成熟，车辆之间（Vehicle--To-Vehicle，V2V）的通信可以基于专用短程通信（Dedicated Short-Range Communications，DSRC）来实现，DSRC 是基于 IEEE802.11p 设计的，在 300m 覆盖内 DSRC 的数据速率可以达到 27Mb/s 左右。

车辆与固定基础设施之间（Vehicle-To-Infrastructure，V2I）的通信则可以使用 Long-Term Evolution（LTE）和 DSRC，与 DSRC 相比，LTE 的覆盖范围更广，服务质量更具有保障，LTE 还支持 350km/h 的高移动性的用户设备，可以很好地适应高速移动的车辆。

此外，迅速发展的 5G 通信网络提供了一种全新的网络架构，提供 10Gbps 以上的峰值速率、更佳的移动性能、毫秒级时延和超高密度连接，国际电信联盟无线电通信局定义了 5G 的三大典型应用场景为增强型移动宽带（eMBB）、超

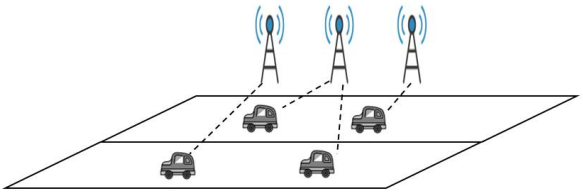
低时延高可靠通信（uRLLC）和海量大规模连接物联网（mMTC）。其中 uRLLC 由于其即时、可靠、高效数据传输的特点，如车联网、自动驾驶、自动化无人机等都是其主要实际应用，基于 5G 的车联网络将是未来车辆计算卸载的通信技术之一。

DSRC、LTE、5G 等车联网络技术为 VEC 计算任务卸载提供了相对可靠的网络传输通道，为高速移动下车辆的计算任务卸载提供可靠的通信保障。。

。

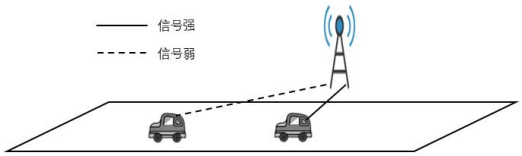
2.3 车联网中的任务卸载

在车联网中，



图：将任务交给边缘服务器

如果将任务交给边缘服务器进行处理（图 3），因为边缘设施固定在道路两旁，在车辆高速行驶时，可能会导致传输速率低和传输失败等问题。



图：高速行驶导致车辆信号弱

在最近流行的边缘计算技术中，我们不仅可以将任务分配给附近的边缘服务器，还可以分配给附近资源丰富的用户。但是其中很多问题需要解决，例如，如何衡量计算资源，如何实时的分配任务，如何分配任务更加公平。

图：将任务交给边缘服务器

图：将任务交给边缘服务器

第 3 章基于基因算法的车辆任务卸载算法

3.1 引言

作为下一代智慧交通的电动智能汽车有许多的问题需要解决，其中电池问题尤为重要，这关系着汽车的行驶里程。而任务卸载是其中较有希望的一个减少能源消耗，提高能源利用率的一种方式。

在最近流行的边缘计算技术中，不仅可以将任务分配给附近的边缘服务器，还可以分配给附近资源丰富的用户。但是其中很多问题需要解决，例如，如何衡量计算资源，如何实时的分配任务，如何分配任务更加公平。

在该问题中，将任务分配给附近的车辆，来提高资源的利用率以及最小化系统的能耗。我们将移动边缘计算场景车联网中任务卸载的问题抽象为一个数学规划的形式，其中，目标是最小化所有能源消耗的平方，这可以兼顾能耗最小化与公平。

约束包括：所有的任务必须有车辆来做，所有的车辆能够在规定时间内完成任务，以及为了保证通讯的质量，参与资源共享的智能汽车的距离要在一定的距离内。

该问题是一个非线性整数规划问题，没有多项式时间范围内的解决办法。

在本文中，我们提出了基于用于电动智能汽车的遗传算法。我们评估通过仿真实验验证了算法的性能实验。实验结果表明，我们的算法可以达到节能的目的。在未来的研究中，我们计划建立在数据大小和指令数量之间函数关系并考虑最后期限限制，同时，也将考虑卸载过程的真实性。

3.2 系统模型

本节主要介绍了系统模型，分为两个部分，系统场景，车辆计算能力模型，以及车辆能耗模型。

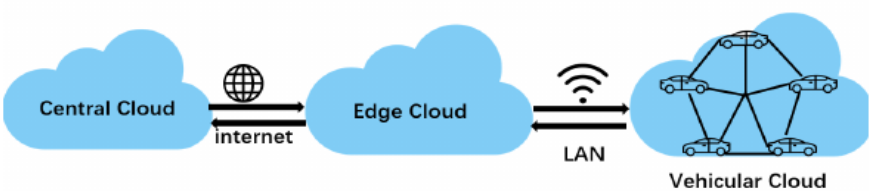
3.2.1 系统场景

在本文中，如图 4 所示，层次结构由车辆云、边缘云和中心云组成。中心云负责全局调度，它执行的任务包括执行复杂的计算任务和进行全局的决策。边缘云是车辆云的控制器，负责创建、维护和删除车辆云。车辆云由一定范围内的、参与共享其计算资源的智能车辆组成。基站、边缘服务器和远程的中心云服器通过有线来连接，基站和边缘服务器物理上距离十分接近，因此可以看作一体，后文中以边缘云来称呼这两者的总和。

每辆车都可以访问边缘云并与其进行通讯。车辆可以通过将任务卸载到其他车辆的方式来节省能源。这样可以提高整体资源利用率。

任务卸载过程如下：首先，实时流量信息被传输到云服务器进行分析。实时信息包括目的地、当前位置、时间、车速等。云服务器在汇总数据后，进行大数据分析以获取道路拥堵情况，然后推断未来某个时间段内车辆的位置信息，并将结果返回到车辆附近的边缘云。

这些车辆在未来一段时间内将会参与资源共享。然后，边缘云根据这些信息分配任务，并将车辆分为提供者和请求者，他们在 T 时间段共享资源。



图：系统场景层次结构

3.2.2 车辆计算能力模型

如何定义车辆的计算资源是建立起问题描述非常重要的一步，在此，我使用计算每秒执行的指令条数来刻画计算资源，我们将资源共享时间定义为 $\mathcal{T}=\{1,\ldots,T\}$.。车辆数量定义为 N。

我们使用元组 $J_{it}=\{r_{it},r'_{it},d_{it}\}$ 来表示时间为 t 时，车辆 i 的任务大小，d_{it} 是任务的数据大小。

任务分为必须在本地执行的任务，以及可以分配出去的任务。这一点并不抽象， 例如有些涉及隐私的任务，或者是要求实时性的任务，就必须在本地执行。

在时间 t 时，r_{it} 是车辆 i 必须要在本地执行的任务，r'_{it} 是车辆 i 能够分配给别人的任务，并且，它们的值是由需要的指令条数来表示的。

为了更好的描述这个模型，我们定义 $\mathbf{X}_t=\left\{x_{ij}\right\}\in\{0,1\}^{N\times N}$ 为分配矩阵，如果 $x_{ij}=1$，代表车辆 i 执行车辆 j 所卸载的任务。注意，如果 $x_{ii}=1$，车辆 i 所有的任务都在本地执行。

我们将车辆 i 的容量定义为 C_{it} 在时间为 t 时。在时间 t，当车辆 i 被选为提供者（P），所有需求者需要的资源需要少于这辆车的容量。公式化表达如下：

$$\sum_{j=1}^{N}x_{ij}^{t}\cdot r'_{jt}\le C_{it},i=1,\ldots,N \tag{3.1}$$

一个关键的步骤是怎样衡量车辆的计算资源，在本篇文章中，计算资源定义在每秒能够执行的指令条数上。

公式化表达如下：

$$C_{it}=M_{it}\cdot\Delta T-r_{it} \tag{3.2}$$

其中，ΔT 是资源共享的持续时间，并且它是一个比建立车辆边缘网络的更小的时间粒度，在 ΔT 秒后，分配矩阵会发生变化。

对于一个单核的 CPU，每秒能够执行的指令条数（m_{it}） 和 CPU 的频率（f_{it}）有如下的关系:

$$M_{it} = v_i \cdot f_{it} + \theta_i \tag{3.3}$$

其中，v_i和θ_i是待估计的参数。

最后，公式（2）中的CPU的容量C_{it}在能被下面的公式计算出来：

$$C_{it} = (v_i \cdot f_{it} + \theta_i) \times \Delta T - r_{it} \tag{3.4}$$

3.2.3 车辆能耗模型

当我们考虑车辆的能源消耗时，我们将它分为两个部分（1）计算所需要的能量以及（2）传输所需要的能量。

根据[9][10][11][12]，计算所需要的能耗能够被下列公式计算：

$$E = \lambda_i \cdot f_{it}^3 \cdot \Delta T \tag{3.}$$

其中f_{it}是CPU在t时刻的频率。如果一辆车被选为了需求者（R），这辆车的频率会下降$f_{it}^{'}$。因为任务被分配出去了，所以他消耗的能量会减少。消耗的能量能够被以下公式计算：

$$E_i^{save} = (\lambda_i \cdot f_{it}^3 - \lambda_i \cdot f_i^{'3}) \times \Delta T, \forall i \in R \tag{3.}$$

传输能耗和传输的时间有线性关系，传输的时间取决于数据大小和传输速率的比值(b_{ij})：

$$E = P_0 \cdot \frac{d_{it}}{b_{ij}} \tag{3.}$$

其中，P_0是传输率。

最大的传输速率（b）可以通过香农公式计算：

$$b_{ij} = W \log(1 + \mathrm{SNR}) \tag{3.}$$

其中，SNR是信噪比，W是频道的带宽。因为它和每一个场景相关，我们将它考

虑为一个常数。

对于每一辆车，接收信息所消耗的能量是：

$$E_i^{rec} = \sum_{j=1, j\neq i}^{N} x_{ij}^t \cdot P_0 \cdot \frac{d_{jt}}{W \log(1+\mathrm{SNR})}, \quad i = 1, \ldots, N \tag{3.9}$$

对于每一辆车，发送信息所需要的能量是：

$$E_i^{send} = \sum_{i=1, i\neq j}^{N} x_{ij}^t \cdot P_0 \cdot \frac{d_{it}}{W \log(1+SNR)}, \quad j = 1, \ldots, N \tag{3.}$$

定义 E_t^{blnc} 是车辆 i 在时刻 t 所消耗的能量总和：可以被这么计算

$$E_{it}^{blnc} = \sum_{j=1, j\neq i}^{N} x_{ij}^t \cdot P_0 \cdot \frac{d_{jt}}{W \log(1+\mathrm{SNR})} + \sum_{j=1,, j\neq i}^{N} x_{ji}^t \cdot P_0 \cdot \frac{d_{it}}{W \log(1+\mathrm{SNR})} + \lambda_i \cdot f_{it}^3 \cdot \Delta T, \quad i = 1, \ldots, N$$

(1)

我们算法的设计目标是:最小化所有车辆能量消耗的平方。平方和和普通的和相比，平方和更容易受到极端值的影响，因为平方项会使得较大或较小的值对平方和的贡献更加显著。

这个目标既能够考虑最小化能量消耗，又能考虑公平，也就是说平衡了能量消耗和公平。公式表示如下文：

$$\min \sum_{t=1}^{T} \sum_{i=1}^{N} (E_{it}^{blnc})^2$$

3.3 问题建模

在本节中，我们将问题分为两种情况来建模，一种是任务可分情况，另一种是任务不可分情况。当任务可分的情况下，问题能够直接通过规划的方式解决；

当任务不可分时，不能通过规划的方式来解决。

3.3.1 任务可分情况

当车辆在进行某些同质任务时，可以将大任务任意的拆成一些小任务，因为任务的粒度太小，因此可以近似看成任务可以随意分配，也就是任务可分的情况。

在任务可分时，根据本地留存的高优先级任务的大小又分为：高于平均任务大小，和低于平均任务大小。

当本地任务低于平均任务大小（小任务）时，该问题较为简单，直接将所有任务平均分配到车辆上就能解决。

该问题等价于：首先将固定值 C 切制成 n 个数，是的他们的三次方之和最小。设切分的数分别为 $a_1, a_2, \cdots, a_n$，根据均值不等式：

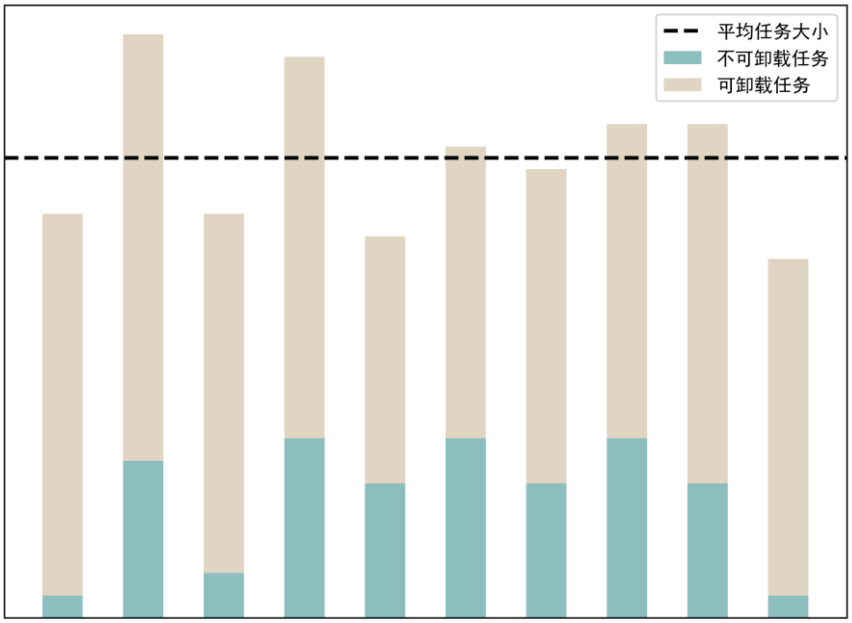
$$\sqrt[3]{\frac{\sum_{i=1}^{n} a_i^3}{n}} \geq \frac{\sum_{i=1}^{n} a_i}{n}$$

即

$$\sum_{i=1}^{n} a_i^3 \geq \frac{n(\sum_{i=1}^{n} a_i)^3}{n^3} = \frac{C^3}{n^3}$$

因此，要使 $\sum_{i=1}^{n} a_i^3$ 最小，需要让 $\sum_{i=1}^{n} a_i$ 的值尽可能平均分配给 n 个数，即令 $a_1 = a_2 = \cdots = a_n = \frac{C}{n}$。

当本地任务是小任务时，能搞保证参与资源共享的车辆，任务的大小可以卸载为平均值。



图：情况 1

但是当任务高于平均任务大小（大任务）时，该问题需要进行分类的讨论，是否还需要将再给有大任务的汽车分配任务。

该问题等价于：有一个固定值 C，将其切割为 n 个数，要求这 n 个数的三次方最小，知道有 m 个数必须大于 $\frac{C}{n}$，如何切割。

在有限制条件的情况下，该问题可以使用拉格朗日乘数法来求解。设切分后的数为 $a_1, a_2, \cdots, a_n$，且有 m 个数大于 $\frac{C}{n}$。

有以下约束条件：

$$\begin{cases} \sum_{i=1}^{n} a_i = C, \\ \sum_{i=1}^{n} \text{count}(a_i, \frac{C}{n}) = m, \end{cases}$$

其中，count 表示指示函数，当 $a_i > \frac{C}{n}$ 时，取值为 1，否则为 0。

目标为：

$$\min \sum_{i=1}^{n} a_i^3 \tag{3.}$$

根据拉格朗日乘数法，构造带拉格朗日乘数 $\lambda_1, \lambda_2, \cdots, \lambda_n$ 的拉格朗日函数：

$$L\left(\{a_i\}_{i=1}^{n}, \{\lambda_i\}_{i=1}^{n}\right) = \sum_{i=1}^{n} a_i^3 + \sum_{i=1}^{n} \lambda_i \left(a_i - \frac{C}{n}\right) + \mu \sum_{i=1}^{n} \text{count}\left(a_i, \frac{C}{n}\right) \tag{3.}$$

其中，μ 是 Lagrange 乘子。

对 L 求导并令其等于 0，可得：

$$3a_i^2 + \lambda_i - \mu \text{count}\left(a_i, \frac{C}{n}\right) = 0$$

将上式分为两种情况：

（1） 当 $a_i \le \frac{C}{n}$ 时，μ=0，则有 $a_i = -\frac{1}{3}\lambda_i$。

（2） 当 $a_i > \frac{C}{n}$ 时，则有 $\lambda_i = -3a_i^2$， μ 取任意值。

将 a_i 代入约束条件中可得：

$$m = \sum_{i=1}^{n} \text{count}(a_i, \frac{C}{n}) = \sum_{i=1}^{n} \text{count}(\lambda_i, 3(\frac{C}{n})^2) \tag{3.}$$

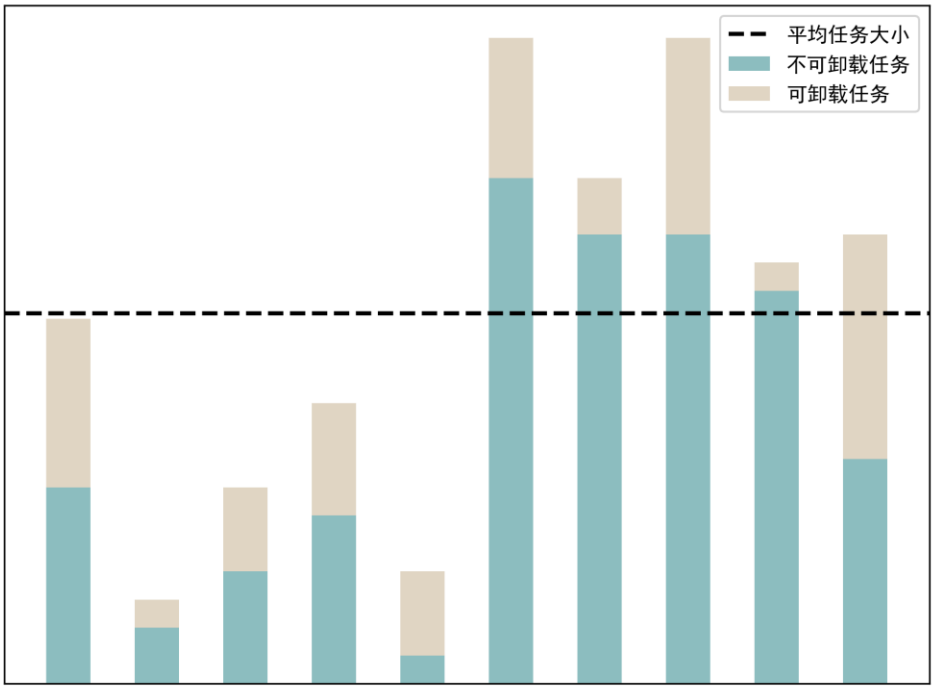
即有 m 个 λ_i 满足 $\lambda_i > 3(\frac{C}{n})^2$。假设它们的下标 $i_1, i_2, \cdots, i_m$，则有：

$$\sum_{j=1, j\neq i_k}^{n} a_j = C - \frac{\sum_{k=1}^{m} \lambda_{i_k}}{n}, \quad k = 1, 2, \cdots, m \tag{3.}$$

进一步代入目标函数 $\sum_{i=1}^{n} a_i^3$ 中，可以得到：

$$\sum_{i=1}^{n} a_i^3 = (\frac{C - \sum_{k=1}^{m} \lambda_{i_k}}{n})^3 \cdot (n-m) + \sum_{k=1}^{m} \lambda_{i_k}^3 \tag{3.}$$

要求 $\sum_{i=1}^{n} a_i^3$ 最小，就要使 $\sum_{k=1}^{m} \lambda_{i_k}^3$ 最小。该问题是一个无约束优化问题，可以使用导数来求解。可得，最优的分配方案是，将那几个本地任务是大任务的车辆将他们所有可分得任务分出去，然后再均分剩下的任务。



图：情况 2

3.3.2 任务不可分情况

在任务不可分时，该问题没有多项式时间内的解决方法，因此，在下一节中，本文打算通过智能算法来获得一个近似解，来满足靠近最优解和卸载方案计算时间的平衡。

该问题的优化目标是通过调度各个车辆的任务，最小化所有能耗的平方，该问题最终被建模为如下形式：

$$\min \quad \sum_{t=1}^{T}\sum_{i=1}^{N}(E_{it}^{blnc})^2 \tag{2}$$

$$\text{s.t.} \quad \sum_{j=1,j\neq i}^{N} x_{ij}^{t}\cdot P_0\cdot\frac{d_{jt}}{W\log(1+\mathrm{SNR})} + \sum_{j=1,j\neq i}^{N} x_{ji}^{t}\cdot P_0\cdot\frac{d_{it}}{W\log(1+\mathrm{SNR})} + \lambda_i\cdot f_{it}^{3}\cdot\Delta T,\quad i=1,\ldots,N \tag{3}$$

$$\sum_{j=1}^{N} x_{ij}^{t}\cdot r'_{jt}\le C_{it} \tag{4}$$

$$\sum_{i=1}^{N} x_{ij}^{t}\ge 1 \tag{5}$$

$$f_{it}\ge 0 \tag{6}$$

$$x_{it}^{t}\in\{0,1\} \tag{7}$$

正如前文所提到的，公式（14）说明了对于每一辆车辆 i，需求的总和不能超过他的容量。公式（15）说明了车辆 i 的任务，必须要被执行，无论是本地执行还是分配给其他车辆。

3.4 基于基因算法的问题求解

因为[公式]这是一个 NP 难（NP-Hard） 问题，多项式时间内求解不出该问题的精确解，直接计算法求解是不现实的，因此采用了基因算法求解该问题

基因算法是一种典型的启发式智能优化算法，其鲁棒性来源主要有以下三点：首先来源于个体多样性，通过引进交叉、变异等操作，使得不同个体之间进行信

息的交流，有助于保持种群的多样性，从而不易陷入问题的局部最优解。其次来源于选择策略。基因算法中的选择策略来自于达尔文自然选择学说中的“适者生存”原则，通过对适应度高的个体进行选择和繁殖，逐步提高整个种群的适应性，这提高了算法的鲁棒性。最后来源于参数设置。基因算法中的参数设置，例如交叉的概率、变异的概率，这对算法的表现具有较大影响。通过对参数的合理设置，可以提高算法的鲁棒性，并且使其更容易达到全局最优解。因此本文选择了能适用于离散问题的基因算法来求解该问题。

以下是基因算法的实现步骤：

(1) 使用初始化参数来初始化种群：设置种群的大小，每个个体的基因编码方式和以及初值范围，设置循环最大次数为 $it_{\max}$，然后根据这些参数生成初始种群。同时设置目标函数 $f(x)$ 以及约束函数族 $G(x)$。

(2) 评估适应度：将每个个体的基因编码字符串解码成相应的解，然后计算其适应度值。适应度值是个体在解空间中执行任务的优劣程度的量化指标。

基因算法二进制解码公式如下：

$$\delta = \frac{U_x - L_x}{2^l - 1} \tag{3.}$$

$$x = L_x + \delta \cdot \sum_{i=1}^{l} A_i 2^{i-1}$$

其中，δ 是偏移量，l 是二进制码的长度，A_i 是第 i 位二进制位的值，x 是解码后的值。例如，给定 $\mathrm{U_x} = 5$，$\mathrm{L_x} = -5$ 以及 $l = 10$，那么 $\delta = \frac{5-(-5)}{2^{10}-1} \approx 0.009775$。

基因算法二进制编码二进制长度公式如下：

$$l = \left\lceil \log_2 \left(\frac{U_x - L_x}{\mathrm{SearchAccuracy}} \right) \right\rceil \tag{3.}$$

其中，$\mathrm{U_x}$ 和 $\mathrm{L_x}$ 分别是解的最大值和最小值，SearchrAccuracy 是求解的精度。例如，给定 $\mathrm{U_x} = 5$ 和 $\mathrm{L_x} = -5$，精度是 0.01，那么 $\mathrm{l} = \left\lceil \log_2 \left(\frac{5-(-5)}{0.01} \right) \right\rceil = 10$

适应度计算公式如下：

$$fit_i = e^{-\frac{y_i}{mean(y)}} \tag{3.}$$

其中，$y_i = f(x_i)$，$mean(y)$是种群中所有的个体的平均值。

(3) 选择操作：按照轮盘赌选择的规则，从种群中选择一部分较好的个体进行进化操作，以期望得到更加优秀的后代个体。在进行轮盘赌选择时，标准化处理的公式为：

$$p_i = \frac{fit_i}{\sum_{i=1}^{n} fit_i} \tag{3.}$$

其中，n是种群中的个体数量。

(4) 进行遗传操作：包括交叉操作和变异操作。交叉操作是将两个个体的染色体进行配对，然后按照一定的概率产生新的后代染色体；变异操作是随机改变个体染色体的一个或多个基因，以增加种群的多样性。

(5) 更新种群：这是“适者生存”法则在基因算法中的应用。将新生成的后代个体插入到种群中，也就是按照一定的比例，某些适应度高的个体替换掉适应度低的个体。

(6) 判断遗传算法是否满足停止准则：停止的准则可以设置最大迭代次数或者在种群中出现一定数量的相似个体等条件，决定是否终止遗传算法。如果不满足条件则跳转到第（2）步。

3.5 仿真实验

3.5.1 实验环境

本章实验环境为：处理器为：Intel(R) Core(TM) i7-7700HQ；显卡的型号为：NVIDIA GeForce RTX 1050；机带 RAM 为 8GB；操作系统为 Windows10 专业版；Python 版本为 3.8，Numpy 版本为 1.19.2，matplotlib 版本为 3.3.2。

3.5.2 仿真参数设计

表 1：仿真参数

参数名称	值/分布	参数名称	值/分布
f_i	[700, 1900]	ΔT	1
v_i	7.683	r_{it}	U[820, 10000]
θ	-4558.52	r'_{it}	U[200, r_{it}]
λ	0.00125	P_0	0.2
W	10 MHz	SNR	677

本文将变异概率设置为 0.01，参数用于控制每个个体发生变异的概率。该概率越高，个体发生变异的可能性就越大。一般情况下，变异概率会设置为较小的值，以保持种群的多样性，并防止算法陷入局部最优解。

本文选择了 7 种种类的参与资源共享的车辆数目，分别是{10，15，20，25，30，35，40}。在算法中，种群数量被设置为 40。而当车辆数量为 35 和 40 时，种群数量被设置为 150。因为当车辆数量为 35 和 40 时，种群的规模太小，算法不会收敛或者不能很好的收敛到最优值附近。

3.5.3 评价标准

我们设置了个指标来衡量我们的实验，分别是，性能指标，公平指标，种群离散程度。

在实验中，我们的性能指标是指节能的百分比。节能的百分比，其定义如下：

$$P = \min\left(100 \cdot \left(1 - \frac{\sum_{i=1}^{N} E_{it}^{blnc}}{E_{loc}}\right)\right), t = 1, \ldots, T \tag{3.}$$

其中，E_{loc} 是本地执行任务所需要消耗的能量，

为了反映我们任务分配的公平性，我们在标准差的基础上定义了公平系数（FC）。与标准差一样，数值越小，公平性越高。标准差定义如下：

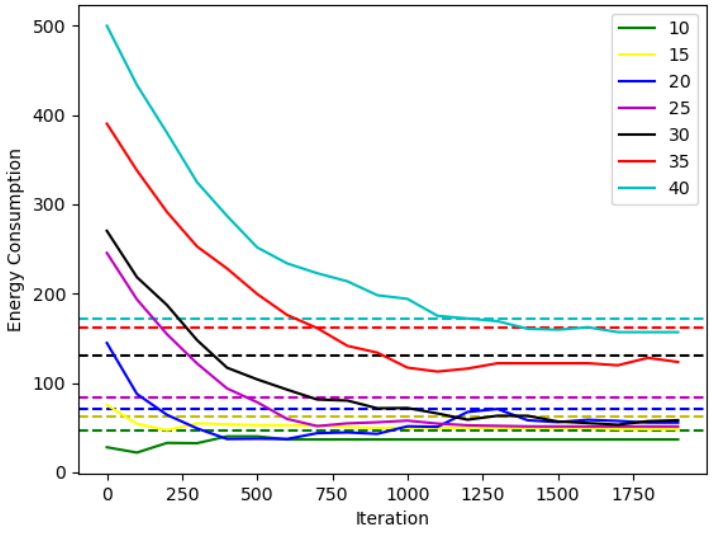
$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}}$$

FC 的定义如下。

$$FC = \max \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(E_{it}^{\text{blnc}} - \bar{E}\right)^2}}{\bar{E}}, t = 1, \ldots, T$$

其中，t 是迭代的次数，$\bar{E}$ 是所有车辆能耗的平均值。

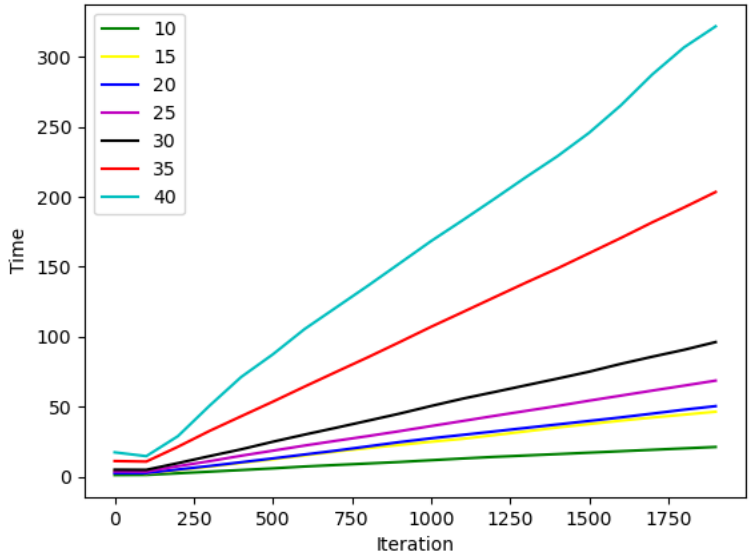
3.5.4 实验结果



图：

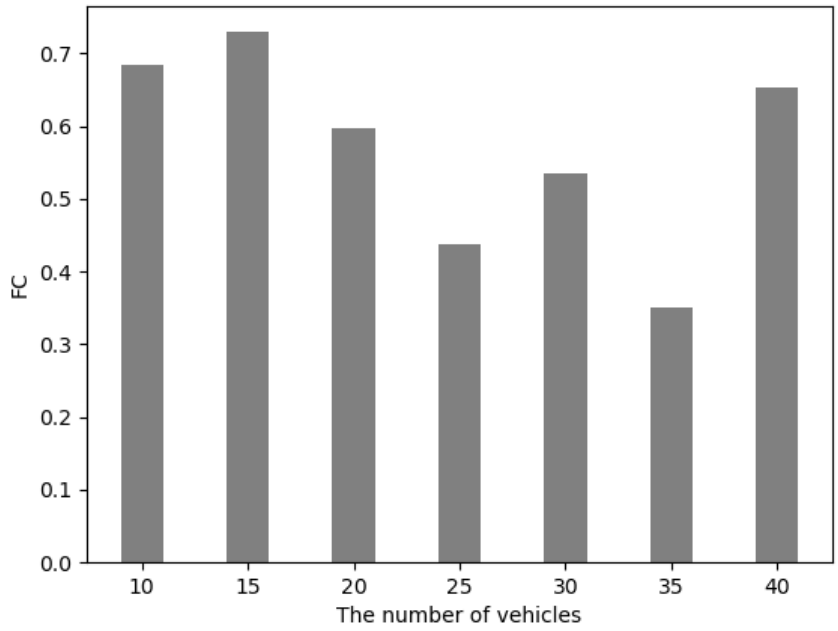
在图 2 中，虚线是任务没有负载出去而在本地执行所需要消耗的能量。我们可以看到，当车辆数为 10 时，该算法迭代几十轮就可以得到很好的结果。当车辆数为 20 时，需要 350 次迭代才能得到一个较好的结果。而当车辆数量为 30 时，能耗在 750 次迭代之前变化较大，而在 750 次迭代之后变化就会明显变小。也就是说，能源消耗的减少率在 750 次迭代前较快，而在 750 次迭代后，能源消耗的减少率就会降低。当车辆数为 40 时，几乎没有节能效果。

我们可以得出结论，随着车辆数量的增加，迭代的次数也在增加。因此，当我们在边缘服务器上运行该算法时，我们需要合理地调整迭代次数，因为车辆数量的变化会显著的影响算法的运行时间。



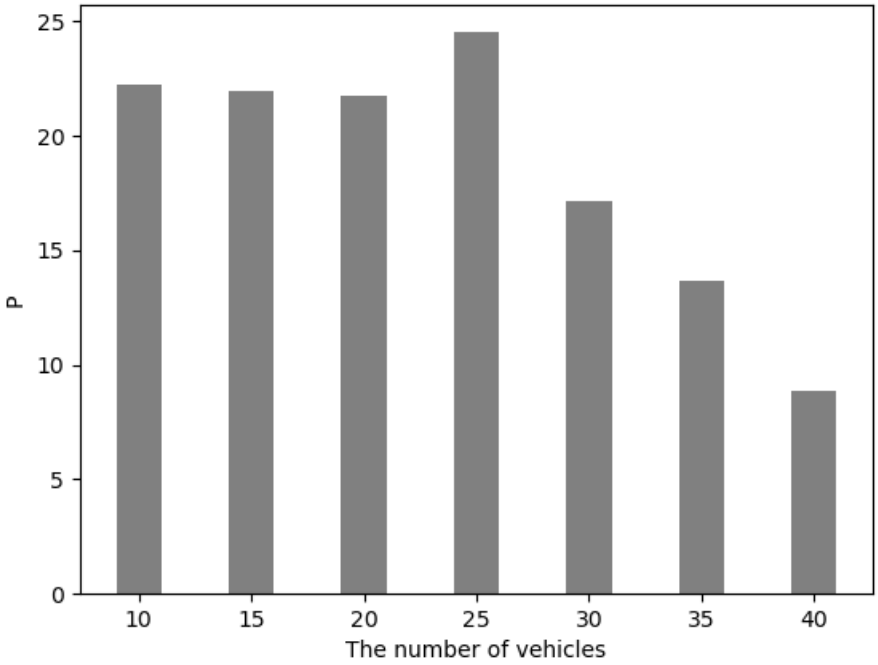
图：

在图 3 中，经过 100 次迭代以后，所有数目运行时间和迭代次数呈线性关系。然而，当迭代次数固定时，算法的运行时间和车辆之间的关系不是线性的，而是非线性的，参与资源共享的车辆的数越多，运行时间增加的越多。



图：平衡因子 FC 的值

如图 4(a)所示，随着车辆的增加，车辆能量消耗的平衡首先增加，然后减少。当车辆数目为 30 和 40 时，能耗的平衡系数反而增加。



性能度量 P 的值

如图 4(b)所示，当车辆数量在 10 到 25 辆之间时，我们的系统可以节省 20%以上的能量消耗，但是当车辆数量为 30 辆时，只有 17%的能量消耗。但当车辆数为 30 时，只能节省 17%，而当车辆数为 40 时，几乎可以节省 20%。当车辆数量为 40 辆时，几乎没有节省能量。

综合考虑性能测量、算法运行时间以及平衡因子，选择 25 辆汽车参与资源共享是非常合适的。

3.6 本章小结

本章所研究的是车联网领域的任务卸载和分配的方案，通过能耗优先的原则设计出一套卸载算法。首先，本章详细介绍了车联网中任务卸载的系统模型，并对其中的要素进行了说明，包括系统场景、车辆计算模型及车辆能耗模型。接着，作者将问题建模为数学规划形式，提出了一种基于粒子群算法的卸载与分配策略，然后设计了相关评价标准，并对实验结果进行了分析。评价标准表明，该算法能够很好地提高资源利用率，同时还兼顾了公平性。

然而，在研究过程中也发现了一些缺陷，比如，基因算法存在着编码和解码的过程，需要进行交叉和变异操作，这导致了计算量太大和计算时间过长等问题。因此，在下一步的研究中，我们将引入新的智能算法，并将其改进为适用于离散问题的算法，以加快算法的运行时间。

本章研究了一种车联网环境下能耗优先的计算卸载和分配的方案。本章首先介绍了车联网中任务卸载的系统模型，包括，系统场景，车辆计算模型、车辆能耗模型，然后将问题建模为了数学规划形式，提出了一种基于基因算法的任务卸载分配算法，然后设计了相关评价标准及分析了实验结果。评价标准显示，该算

法能够很好的提高资源的利用率，同时也很好的兼顾公平。

但在本章的研究过程中发现了一些不足：基因算法存在着编码和解码的过程，以便于进行交叉和变异操作，这带来了计算量太大以及计算时间过长的问题。因此，下一张会引出较新的智能算法并将其修改为能解决离散问题的算法，加快算法的运行时间。

第 4 章基于贪心法调整的离散侏儒猫鼬算法的任务卸载算法

通过上一章节的研究，我们完成了车联网中对任务卸载算法研究的任务，但也指出了存在计算量太大以及计算时间过长等问题。因此本章将通过使用新的智能算法并将其修改为能解决离散问题的算法来解决该问题。

4.1 侏儒猫鼬算法

和粒子群、基因算法类似，侏儒猫鼬算法（Dwarf Mongoose Optimization Algorithm，DMOA）也是一种基于动物行为的启发式优化算法，能够用来解决线性以及非线性的优化问题。该算法是由 Jeffrey O. Agushaka 等科学家于 2022 年提出的，算法的灵感源于侏儒猫鼬的捕猎行为，通过模拟侏儒猫鼬在觅食和捕猎时的行为特征来进行对问题的求解。

侏儒猫鼬算法在搜索过程中，不断尝试各种不同的位置和状态组合，计算适

应度，并据此进行调整，最终找到最优解或次优解。

像其他智能算法一样，侏儒猫鼬算法借鉴了侏儒猫鼬在觅食和捕猎时的行为特征，同时也吸收了其他算法的特征。

表 4.2 肺部 CT 图像数据集

算法名称	主要思想
粒子群算法	
训练集	1600
验证集	400
测试集	500

侏儒猫鼬是一种小型哺乳动物，以体型较小昆虫以及蚂蚁，蚁蛇等为食，它们具有机敏的行动能力和快速适应环境的能力。

侏儒猫鼬通常生活在在一个母权制的家庭群体中，由一对阿尔法夫妻（leader）领导种群生活在一起。无论在每个年龄组中，雌性都比雄性地位高，而幼崽比它们的哥哥姐姐地位高。这些群体中的劳动分工和利他主义是哺乳动物中最高的，他们根据年龄和性别，不同的猫鼬充当警卫、保姆、攻击捕食者和攻击同种入侵者。

为了模仿它们的觅食行为，优化过程建立了三个社会结构：阿尔法小组，侦察兵小组，保姆小组。

4.1.1 阿尔法小组

阿尔法小组是又进行出发去觅食（探索新的解）的行为的种群，食物的位置由下面的公式给出：

$$X_i^{t+1} = X_i^t + \text{phi} \times \text{peep} \times (X_i^t - X_{peep})$$

其中，X_i^{t+1} 是新产生的解，phi 服从于在[-1,1]之间的均匀分布。peep 是雌性首领的位置数量，X_{peep} 是选出的雌性首领的位置。

雌性首领在阿尔法小组中产生，每个个体都有可能成为雌性首领，适应度高的个体成为雌性首领的可能性更大，适应度差的个体成为雌性首领的可能性小，每个个体成为雌性首领的概率计算方式如下：

$$\alpha = \frac{fit_i}{\sum_{i=1}^{n} fit_i}$$

其中， fit_i 是第 i 个个体的适应度，n 是种群数量。

4.1.2 保姆小组

保姆通常是附属的群体成员，和幼仔呆在一起，定期轮换，让母亲带领其他成员进行日常觅食。她通常在中午和晚上回来给幼崽喂奶。也就是说，保姆小组的成员不进行觅食行为，同时，保姆的数量取决于种群规模，种群越大，保姆数量越多。

在算法中，当一个个体很久没有进行觅食（更新）时，就说明不是陷入了局部最优，就是这个解太差了，更新的可能比较小，将其放入保姆小组中，并重新进行初始化。

4.1.3 侦察兵小组

侦察兵寻找下一个睡觉的土堆（睡眠丘），因为猫鼬不会回到先前睡觉的土堆，这保证了探险的有效性。在该算法中，侦察兵小组和阿尔法小组是一个小组，首先会进行阿尔法小组的跟新行为，然后再进行侦察兵小组的更新行为。

睡眠丘的计算公式如下：

$$sm_i = \frac{fit_{i+1} - fit_i}{\max\left\{\left| fit_{i+1}, fit_i \right|\right\}}$$

睡眠丘的平均值反映了该轮迭代的移动范围，φ 计算公式如下：

$$\varphi = \frac{\sum_{i=1}^{n} sm_i}{n}$$

CF 是表示侦察兵小组移动能力的参数：

$$CF = \left(1 - \frac{iter}{T}\right)^{\left(2\frac{iter}{T}\right)}$$

其中，*iter* 为当前迭代次数，*T* 为最大迭代次数，当迭代次数增加时，其值会减小，也就是说，随着迭代次数增加，侦察兵小组移动能力会减小。

侦察兵小组的更新方式如下：

$$X_{i+1} = \begin{cases} X_i - CF * \text{ phi } * \text{ rand } * \left[X_i - \overrightarrow{\mathbf{M}}\right] & \text{if } \varphi_{i+1} > \varphi_i \\ X_i + CF * \text{ phi } * \text{ rand} * \left[X_i - \overrightarrow{\mathbf{M}}\right] & \text{else} \end{cases}$$

其中，phi 服从于[0,1]之间的均匀分布，$\overrightarrow{\mathbf{M}} = \sum_{i=1}^{n} \frac{X_i \times sm_i}{X_i}$

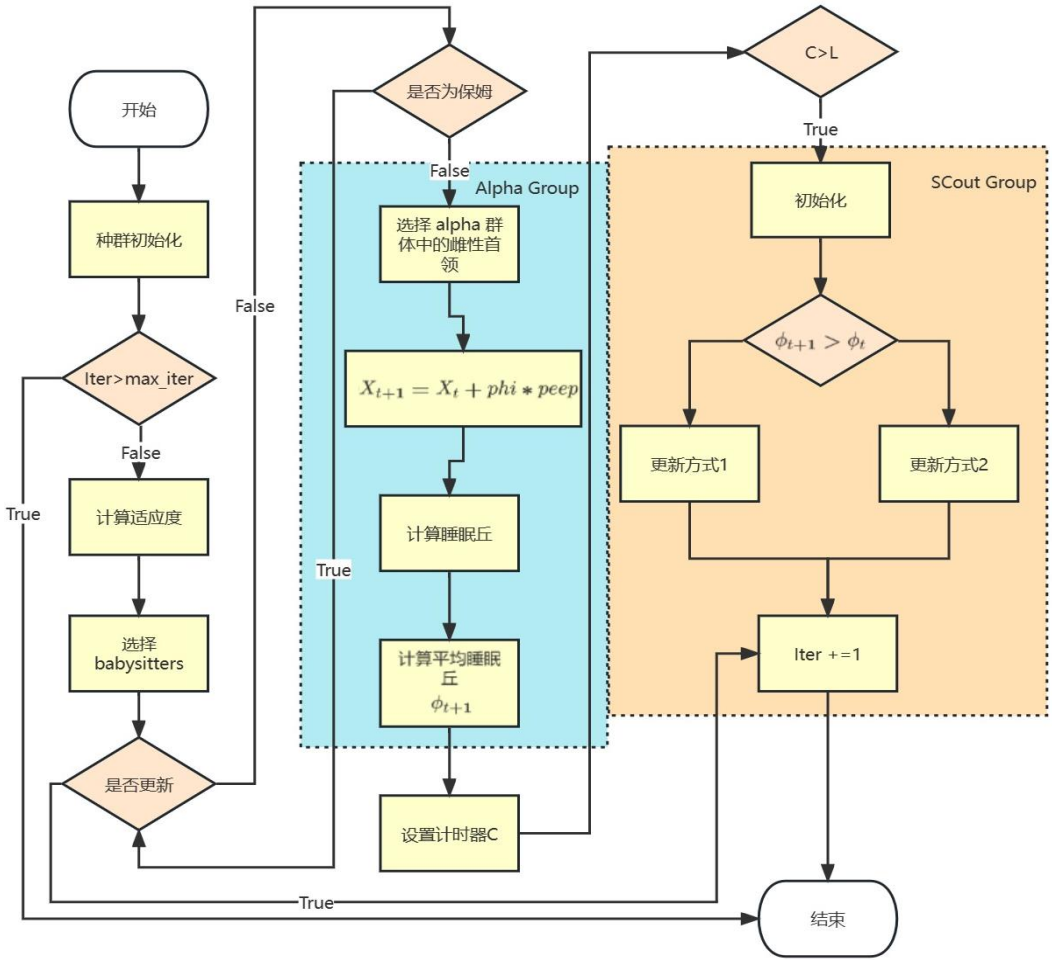
4.1.4 算法的流程

优化从阿尔法小组出发(探索空间)觅食开始，将保姆小组留在巢穴中。一旦找到觅食地点，阿尔法小组就一直觅食到中午（保留更新值）。

当他们返回交换保姆时，按照保姆交换标准，很久没觅食的成员会成为保姆。一旦保姆被交换，他们就不会回到先前觅食的地方，以避免过度放牧。

侦察兵会发现一个新的觅食地点，并通知雌性首领，带领家族到达新的地点。

可以看出，侏儒猫鼬算法中种群的主体部分（阿尔法小组或者侦察兵小组）一次迭代中会进行两次更新行为。



图：侏儒猫鼬算法流程图

显而易见，侏儒猫鼬算法在进行每一步的更新时，使用了类似于梯度的更新方式：$X_i^{t+1} = X_i^t + \text{phi} \times \text{peep} \times (X_i^t - X_{peep})$，因此该算法只能解决连续形式的问题，不能解决离散形式的问题，需要我们将其改造为像基因算法一样的离散型算法。

4.2 离散算法相关的知识

本节介绍了想要将算法修改为离散算法的相关知识。

4.2.1 优化问题的描述形式

我们的问题和并行机器调度问题（parallel machines scheduling problem）有同质性，并行机器调度问题是指在一些并行处理的机器上调度执行一组作业的问题。该问题与日常生产生活紧密联系，通常出现在工业车间生产、物流快递配送、计算机任务分配等领域，可以帮助企业有效地安排生产计划、降低成本、提高效率，也可以在计算机系统中动态地调度任务，提高系统总体性能。

该问题的形式化描述如下：有一个工件（任务）集合 $J=\{J_1, J_2,\ldots J_n\}$，其中，任务共有 n 个。还有一个机器的集合即 $M=\{M_1, M_2,\ldots,M_m\}$，共有 m 个机器，所有的机器都是一样的，并且每台机器一次只能处理一个任务。每一项任务应该必须在其中一台机器上进行，在所有机器上运行的时间不会改变。任何一台机器处理任务 i 所需的时间由 p_i 给出。调度中分配给机器 M_j 的作业的子集用 S_j 表示。作业一旦开始处理，就必须在不中断作业的情况下完成，即，任务一旦开始做那就必须做完。

此外，作业之间没有优先关系，也就是说，在任何时间，作业可以不受限制地随意分配到任意一台机器上。研究了以最小化最后一个工件离开系统的时间为目标的最优工件分配问题，即最大完工时间准则。

最小最大完工时间的混合整数规划公式如下:

$$\min \max_{1\le j\le m} \sum_{i=1}^{n} p_i x_{ij}$$

$$\text{s.t.} \sum_{j=1}^{m} x_{ij} = 1 \quad i = 1,\ldots,n$$

$$x_{ij} \in \{0,1\}$$

公式~(ref{pmpmin1})表明，我们应该耗时最久的那台机器尽可能地小，因为所有的作业完成才算是完成了任务。公式~(ref{pmpst1})表明每个作业必须分配

到一台机器上。如果 $x_{ij}=1$，意味着作业 i 被分配到 j 的机器上。

然而，工件问题这个形式的空间复杂度是 $O(mn)$，而且因为每个变量都是二进制变量，所以时间复杂度是 $O(2^{mn})$。我们将并行机器的调度表述为另一种形式：

$$\min \max_{1\le j\le m} \sum_{1\le i\le n, x_i=j} p_i$$

$$\text{s.t.} \quad x_i \in \{0,1,\ldots,m\}, \quad i=1,\ldots,n$$

其中，工件问题该形式的空间复杂度为 $O(n)$，时间复杂度为 $O(m^n)$。

工件问题每个变量之间的取值有相关性，如，一个两个变量的值交换，很大的概率就会导致最优解变成次优解甚至是劣解。

后文中，称呼每一个 x_i 为变量，所有变量的解集合即 $X=(x_1,x_2,...x_n)$ 称为该问题的一个解。

在后文设计离散形式的算法时，解的表示是一个关键的步骤，解具有与问题域相关的必要的信息。例如上文中使用基因算法解决问题，基因算法为了表示解，将解重新进行了编码，用二进制字符串的形式表示解。为了建立算法和问题之间的联系，对于有 n 个任务的问题，对应的解的维度也是 n。因此，在后文进行离散算法表示时，并行机器调度问题的解是用一个长度等于作业数量的数组表示的。

图：解的表示

如图所示，一共有六个任务，第 1，2，5 号任务放在第一台机器上处理，第 3，4，6 号任务放在编号为 2 的机器上进行处理

4.2.2 各变量之间的相关性

本节通过两个实验来验证各个变量之间的相关性。第一个实验是验证解中某

些变量的取值发生改变对目标函数的影响，第二个实验是验证两个解进行交叉行为对目标函数的影响。第二个实验实际是第一个实验的更进阶的版本，第一个实验交换的是同一个解中的某些变量，第二个实验交换的是不同解中的连续的变量。

这两个实验都说明了解变量之间的相关性，为了消除变量之间的相关性，我们生成了解以后，还需要对解的变量进行调整。

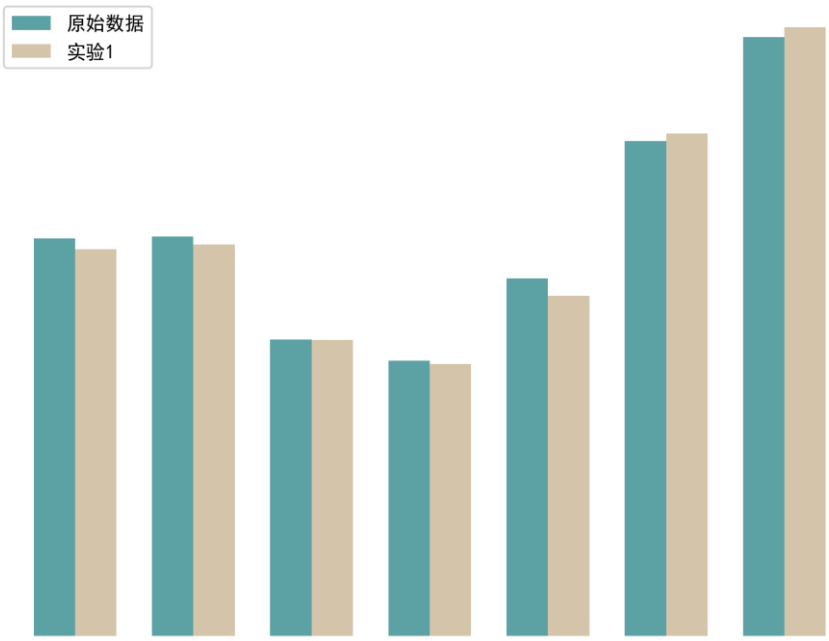
4.2.3 实验一及其结果

实验一：验证解中某些变量的取值发生改变对目标函数的影响。

（1） 随机生成初始解，且保证解的合法性，计算目标函数。

（2） 对每个解的某些位置的变量，交换他们的取值。

（3） 计算完成操作后的解所对应的目标函数，生成图像。



图：

在图中，绿色是没有更新的解所对应的函数值，黄色柱状图是更新解以后所对应的变量值，可以看出，如果仅仅只是交换两个变量值的更新方式，更新和不更新没有什么显著的区别。

4.2.4 实验二及其结果

实验二：两个解进行交叉行为对目标函数的影响。

（1） 随机生成初始解，且保证解的合法性，计算目标函数。

（2） 解两两配对，进行交叉操作。

（3） 计算完成操作后的解所对应的目标函数，生成图像。

乘法运算法($\overset{+}{O}$)，解中每一个等于变量进行，值为另一个解相对应变量的值，然后进行交叉操作。

下面将解释什么是交叉操作。交叉操作是基因算法的一个解更新行为，是指在两个父代个体（解）之间交换染色体上的一部分基因信息（变量）并生成新的子代个体（解）。作为进化算法的重要步骤之一，用于产生更好的后代解，并增加群体的多样性。

交叉操作通常需要满足如下两个条件：首先，确定交叉方式，即交叉方式有多种形式，例如单点交叉、多点交叉、均匀交叉和线性交叉等，本文选择了双点交叉的方式。其次，保证基因信息的完整性和正确性，即交叉操作需要保证新生成的解是有效的解，因此需要保证每个基因在子代中都不缺失或重复。

下面讲解双点交叉的操作，在双点交叉（Two-point Crossover）中，需要随机选择两个交叉点的位置，然后将两个父代个体的这两个交叉点之间的基因序列进行互换，从而产生新的子代个体。

下面以一个长度为 6 的染色体为例，说明双点交叉的具体操作过程：假设有两个父代个体分别是：

$$S_1 = [0,1,0,1,0,1]$$

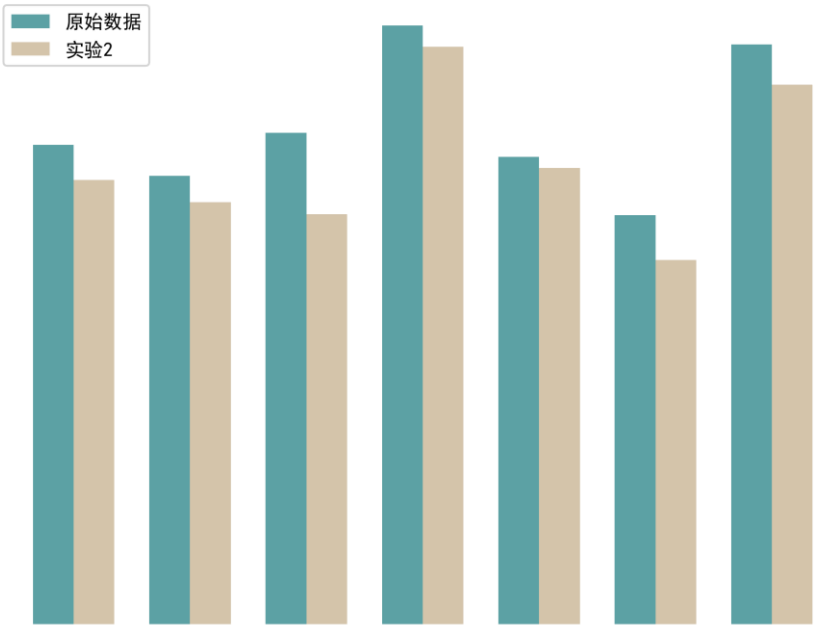
$$S_2 = [1,0,1,0,1,0]$$

我们随机选择两个交叉点位置 k1=2 和 k2=5，然后将两个父代个体在这两个位置之间的基因信息进行互换，得到新的子代个体 S_3 和 S_4：

$$S_3 = [0,0,1,0,1,1]$$

$$S_4 = [1,1,0,1,0,0]$$

子代个体中来自父代个体的基因信息（变量）在交叉点之前或之后是完全一致的，但在交叉点之间则发生了互换。



图：

在图中，绿色是没有本地执行所对应的函数值，黄色柱状图是进行交叉操作以后所对应的变量值，可以看出，虽然比实验一效果要好，但是这种交叉这种更新策略效果并不明显，且这几次实验之间的差值比较大，更加说明了变量之间有相关性。

4.3 基于贪心法调整的离散侏儒猫鼬算法

本节提出了修改后的基于贪心法调整的离散侏儒猫鼬算法。

4.3.1 基于贪心法的调整策略

为了克服变量之间存在的相关性，基因算法有着解码和编码两个过程，来配合交叉操作，使得交叉操作更具有随机性，克服了变量之间的相关性。

我们不使用基因算法，而是想将侏儒猫鼬算法改造为能够求解离散形式问题

的原因是，基因算法的解码和编码两个过程十分浪费时间，所以，为了加快侏儒猫鼬算法的运行时间，我们在此使用基于贪心法的调整策略。

我们采用的贪心法是处理时间最短法（Shortest Processing Time，SPT），该算法是将作业调整到当前剩余处理时间最短的机器上。SPT 算法的执行步骤如下：首先，将所有待处理的任务按照所需处理时间从小到大排序。其次，依次将任务分配给选择当前剩余处理时间最短的机器。直到所有任务都被分配并完成。

下面是 SPT 的一个实例：假设有 6 个任务需要调度，它们的处理时间分别为 3，4，5，4，3 和 7，有两台机器可用。首先，按照所需处理时间从小到大对任务进行排序：time=[3, 3, 4, 4, 5, 7]。

接下来，依次将任务分配给可用的处理器，使得处理时间最短的任务尽量早地被处理。第一批任务的处理时间分别为 3、3，将这两个任务分配给两台处理器中处理时间较短的那台。此时，两台处理器的处理时间分别为 3 和 3。现在剩余的任务列表为 [4, 4, 5, 7]。

接着，在两台处理器中选择处理时间最短的处理器（即第一个处理器），并为其分配下一个处理时间最短的任务，即处理时间为 4 的任务。第一个处理器的处理时间增加到了 7。现在剩余的任务列表为 [4, 5, 7]。

然后，选择处理时间最短的处理器（即第二个处理器），并为其分配下一个处理时间最短的任务，即处理时间为 4 的任务。第二个处理器的处理时间增加到了 7。现在剩余的任务列表为 [5, 7]。

继续选择处理时间最短的处理器（即第一个处理器），并为其分配下一个处理时间最短的任务，即处理时间为 5 的任务。第一个处理器的处理时间增加到了 12。现在剩余的任务列表为 [7]。

最后，选择处理时间最短的处理器（即第二个处理器），并为其分配最后一个任务，即处理时间为 7 的任务。第二个处理器的处理时间增加到了 14，此时所有任务都被分配并完成。

因此，使用 SPT 算法进行调度后，两台处理器的处理时间分别为 12 和 14

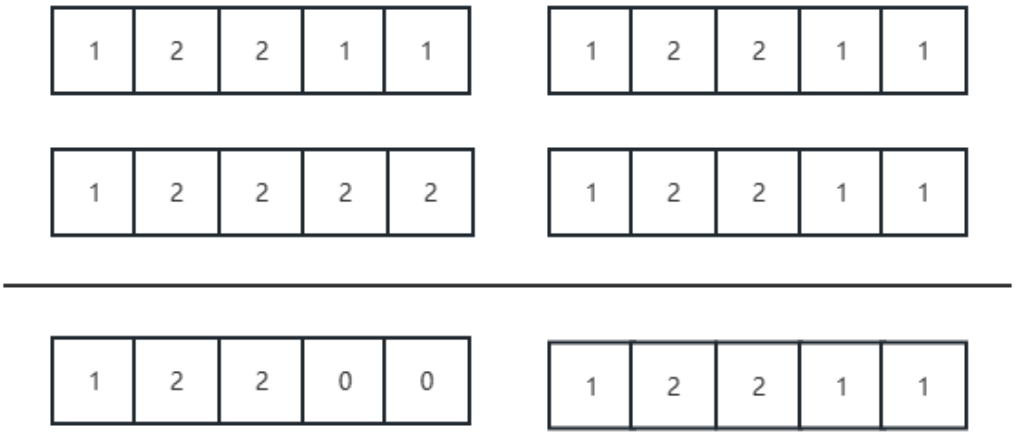
4.3.2 重定义运算规则

为了更好的用数学语言描述每个解更新的过程，以及适应离散形式的更新形式，重新定义运算规则如下：

4.3.3 双目减法运算符($\overline{\mathrm{O}}$)

双目减法运算符($\overline{\mathrm{O}}$)，解中每一个变量相应元素的值是否不同。如果是，则获得其值，如果不是，则置为 0。也就是说，获得两个解中相同的部分，不同的部分置为 0。

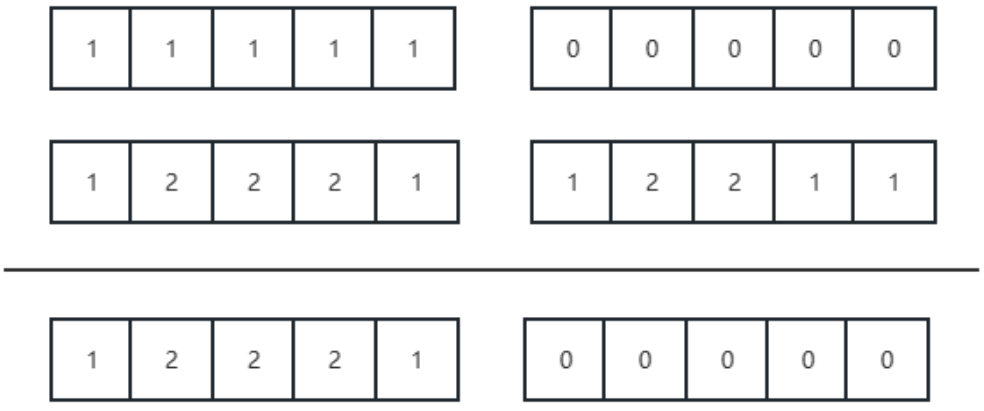
$$X_1 \overline{\mathrm{O}} X_2 = \begin{cases} X_{1i} & X_{1i} = X_{2i} \\ 0 & X_{1i} != X_{2i}, i = 1,2...n \end{cases}$$



图：减法运算符号

4.3.4 双目乘法运算法($\overset{\times}{O}$)

双目乘法运算法($\overset{\times}{O}$)，每个变量的取值分别相乘，前一个变量的取值是 n 重伯努利分布，其值只有 0 和 1。



图：乘法运算符号

4.3.5 单目运算符$\odot X_1$

单目运算符$\odot X_1$，对X_1中所有为 0 的元素进行 SPT 调整。

一共两台机器，编号分别为 1 和 2，$time=[3,4,5,4,3,7]$，$X_1=[0,1,2,2,0,0]$，对值为 0 的元素进行 SPT 调整， 值为 k（$k\neq 0$）的元素的意思：该任务的调整到第 k 台机器上。调整步骤如下：

（1）首先，将两个数组打包，得到$\begin{matrix}[3,4,5,4,3,7]\\[0,1,2,2,0,0]\end{matrix}$

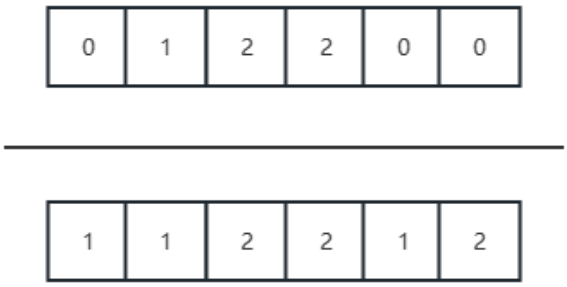
（2）将任务按照所需处理时间从小到大进行排序，得到$\begin{matrix}[3,3,4,4,5,7]\\[0,0,1,2,2,0]\end{matrix}$。

（3）1 号机器的处理时间为 4，2 号机器的处理时间为 9，将所需处理时间最短的且对应分配机器值为 0 的任务，即任务处理时间为 3 的任务调度到 1 号机器上，1 号机器处理时间变成 7。

（4）1 号机器的处理时间为 7，2 号机器的处理时间为 9，将所需处理时间最短的且对应分配机器值为 0 任务，即任务处理时间为 3 的任务调度到 1 号机器上，1 号机器处理时间变成 10。

（5）1 号机器的处理时间为 10，2 号机器的处理时间为 9，将所需处理时间最短的且对应分配机器值为 0 任务，即任务处理时间为 7 的任务调度到 2 号机器上，1 号机器处理时间变成 16。

（6）将分配的方案映射回到没有排序的数组中得到$[1,1,2,2,1,2]$。即$\odot X_1=[1,1,2,2,1,2]$。



图：单目运算符

图：

4.3.6 离散侏儒猫鼬算法伪代码

Algorithm 1: Genetic Algorithm

输入: 种群数量 n, 目标函数 $f(x)$, 最大迭代次数max_iter, 变量的维度 ndim

输出: x_{best}

初始化种群数量 n
初始化目标函数 $f(x)$
初始化种群 X
初始化最大迭代次数max_iter
初始化保姆的数量 bs
n=n-bs
设置保姆交换参数 L
for *iter=1:max_iter* **do**
  计算种群的适应度
  设置计时器 C
  通过下列公式, 找到 alpha 群体中的雌性首领 X_α

$$\alpha = \frac{fit_i}{\sum_{i=1}^{n} fit_i}$$

  对种群中的每一个个体, 通过下列公式, 计算候选的觅食位置

$$X_i^{iter+1} = \mathrm{RVS} \overset{\times}{\mathrm{O}} X_i^{iter} \overset{-}{\mathrm{O}} X_\alpha$$

  其中, $\mathrm{RVS} = \prod_{i=1}^{ndim} Bernoulli(a_i; p)$
  评估每个新值 X_i^{iter+1}
  用下列公式计算睡眠丘陵

$$sm_i = \frac{fit_i^{iter+1} - fit_i^{iter}}{\max\left\{\left|fit_i^{iter+1}, fit_i^{iter}\right|\right\}}$$

  用下列公式计算睡眠丘的平均值

$$\varphi = \frac{\sum_{i=1}^{n} sm_i}{n}$$

  用下列公式计算移动向量

$$\overrightarrow{M} = \sum_{i=1}^{n} \frac{X_i \times sm_i}{X_i}$$

  如果 $C \geq L$, 交换保姆并重新初始化
  用如下公式进行更新

$$\boldsymbol{X}_{i+1} = \begin{cases} X_i - CF * \text{ rand } * \left[X_i - \overrightarrow{M}\right] & \text{if } \varphi_{i+1} > \varphi_i \\ X_i + CF * \text{ rand } * \left[X_i - \overrightarrow{M}\right] & \text{else} \end{cases}$$

  更新 X_{bset}
end
return X_{best}

图：基于贪心法调整的离散侏儒猫鼬算法流程图

4.4 仿真实验

4.4.1 实验环境及评价标准

名称	型号或版本
处理器	Intel(R) Core(TM) i7-7700HQ
显卡	NVIDIA GeForce RTX 1050
RAM	8.0GB
操作系统	Windows10 专业版
Python	3.8
Numpy	1.19
Matplotlib	3.3

4.4.2 评价标准

群的多样性：主要有两种衡量种群的多样性的方法。第一类是基于距离的度量方法（distance-based measurement，DMB），通过计算个体之间的欧氏距离来衡量种群间的多样性。DBMs 有几种不同的形式，如所有个体之间的平均空间距离、个体与群中心位置之间的平均距离、个体之间的最大距离等[23]、[24]、[25]。但是总体而言，距离越远代表着种群多样性越丰富，距离越小代表着的种群多样性越差。

第二种方法是基于频率的测量方法(frequency-based measurement,FBM)。

在 FBM 中，通过统计具有代表性的个体的频率来评价种群多样性[26]，[27]。在[28]中，利用平均基因频率来确定种群多样性。此外，熵度量是属于 FBM 的另一种方法，因为熵使用了基因出现的概率/适应度。在[29]，[30]中可以找到关于群体智能的 FBM 的详细研究。

在本文中，我们使用了一种度量方法，即将种群中每个解的各个维度的方差之和的二分之一次方作为多样性 D 的衡量标准。下面是计算多样性 D 的公式：

$$D=\sqrt{\sum_{j=1}^{m}D_j^2},\ D_j=\frac{1}{n}\sum_{i=1}^{n}\left(x_{ij}-\bar{x}_j\right)^2$$

其中，m 是解的维度，n 是种群的数量，x_{ij} 是是第 i 个粒子的第 j 个维数。

[23] R. K. Ursem, “Diversity-guided evolutionary algorithms,” in Proc. 7th Int. Conf. Parallel Problem Solving Nature, 2002, pp. 462–474.
[24] R. Morrison and K. De Jong , “Measurement of population diversity,” in Artif. Evol., vol. 2310, pp. 31–41, 2002.
[25] O. Olorunda and A. P. Engelbrecht, “Measuring exploration/exploitation in particle swarms using swarm diversity,” in Proc. IEEECongr. Evol. Comput. World Congr. Comput. Intell., 2008, pp. 1128–1134.
[26] M. M. Gouvea and A. F. R. Araujo, “Diversity control based on population heterozygosity dynamics,” in Proc. IEEE Congr. Evol. Comput. World Congr. Comput. Intell., 2008, pp. 3671–3678.
[27] B. Wyns, P. De Bruyne, and L. Boullart, “Characterizing diversity in genetic programming,” in Proc. Genetic Program., pp. 250–259, 2006.
[28] R. J. Collins and D. R. Jefferson, “Selection in massively parallel genetic algorithms,” in Proc. 4th Int. Conf. Genetic Algorithms, 1991, pp. 249–256.
[29] J. Rosca, “Entropy-driven adaptive representation,” in Proc. Workshop Genetic Program. Theory Real-World Appl., 1995, pp. 23–32.
[30] Y. Shi and R. C. Eberhart, “Population diversity of particle swarms,” IEEE Congr. Evol. Comput. World Congr. Comput. Intell., 2008, pp. 1063–1067.

在[31]中，Wineberg 和 Oppacher 得出结论，尽管这两种测量方法有不同的形式，但它们都使用了一个群体中所有可能的染色体对和生物体之间的距离概念。对于 FBM，通常是计算每个个体的频率。特别是对于持续性的问题，它很少出现在两个相同的人身上。因此，本文没有使用 FBM 来度量种群多样性，而是使用 DBM 来度量种群多样性。根据[30]，提出了多种类型的 DBM。

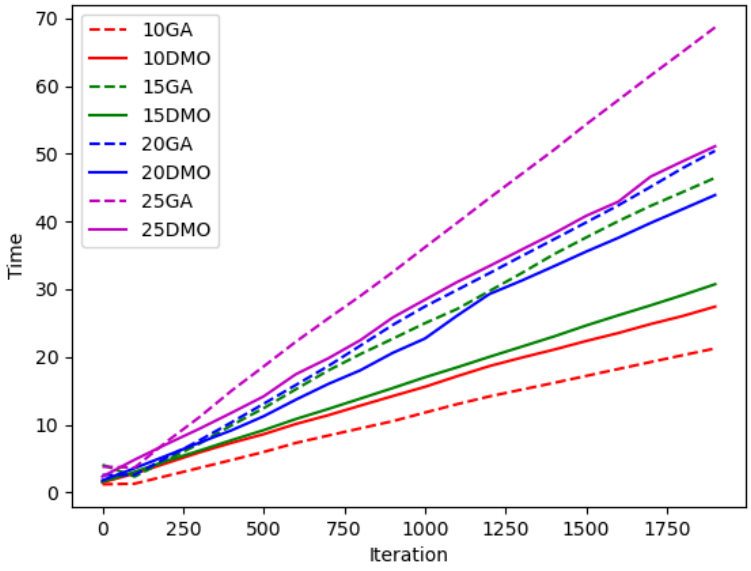
其他评价标准和第三章一样

4.4.3 算法参数设置

参数名称	参数值
Peep	2
nBabysitter	3%
npop	40
nBabysitter	3%

本实验选择了 4 种种类的参与资源共享的车辆数目，分别是{10，15，20，25}。在算法中，种群数量被设置为 40。分别用基因算法和侏儒猫鼬算法进行比较。

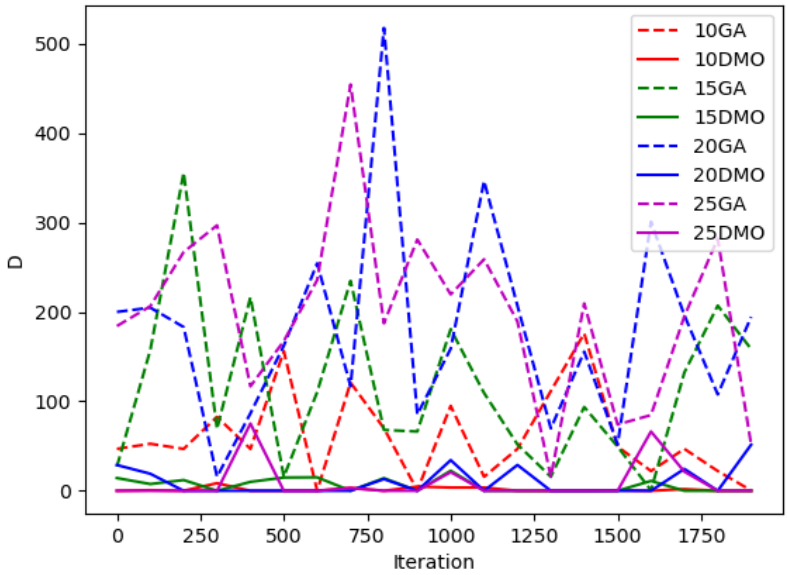
4.5 实验结果与分析



图：

如图所示，虚线是基因算法求解该问题的时间，实线是侏儒猫鼬算法的

运行时间，标签前面的数字代表着车辆的数目。可以看出，侏儒猫鼬算法的运行时间明显比基因算法的运行时间要短，且当车辆数目为 15 和 20 的时候，运行时间大约要少三分之一，当车辆数目为 10 和 25 时，运行时间能节省约四分之一。



图：

可以看出，基因算法的种群差异度变化幅度很大，这就说明，每次算法的运行过程中还在产生新的解，因此导致种群差异度很大，这可以充分保证侏儒猫鼬算法不会跌入到很差解中。

图：

4.6 小结

本章在上一章的基础上，探索了新的方法来解决第三章提出的问题，解决了基因算法有着编码解码过程造成时间消耗大的问题，通过改进侏儒猫鼬算法，将其改造为离散侏儒猫鼬算法，实验结果表明和基因算法相比，我们的算法运行时间明显缩短。

第 5 章结论与展望

5.1 本文工作总结

随着国产汽车公司比亚迪公司的崛起，电动汽车越来越受到人们的追捧，其中很重要的一个原因就是可玩的智能系统。虽然电动汽车的电池续航里程越来越大，但是对比人们对于里程的渴望而言是远远不够的，如何在有限的容量中让汽车的行驶里程变得更大，仍然需要解决，而任务卸载是其中一个有希望能够解决该问题的方法，将任务卸载到边缘端或者是其他的汽车上，通过合理的调度来降低所有车辆的能耗和。这就需要我们来设计合理的算法来进行调度，但是该问题的模型复杂，且并不能通过凸优化的方法来解决，因此需要其他的方法来解决。智能算法是作为求解复杂问题的佼佼者，我们通过智能算法来求解该问题。本文采用基因算法以及基于贪心法调整的侏儒猫鼬算法来解决本文的复杂问题。本文的总结如下：

（1） 研究了车联网中任务卸载的问题模型，将能耗模型分为两部分来描述，将其描述为数学规划的形式。

（2） 本文使用了基因算法来求解该问题，但是从本质上来讲，并没有对基因算法进行创新。

（3） 本文在侏儒猫鼬算法的基础上提出了基于贪心法调整的侏儒猫鼬算法，核心思想是通过贪心法（处理时间最短法）来实现对种群进行更新。

随着实际情况的不断变化，我国防疫相关政策也做出了改变。由全城静默、动态清零到全面放开。在全民经历了一波病毒的洗礼后，当今社会看似歌舞升平，新冠病毒已经离我们而去。但实际上，新冠病毒一直在变异，我们只是获得了新冠病毒亚变异体中某些分支病毒的抗体。人们仍旧面临着再次被感染的风险。鉴于抗原试剂检验的滞后性，对于带有基础病以及老年人来说，及早发现及早治疗是保证个人生命安全的最优解。而我国老年化人口众多，医疗系统应对起来较为吃力，为了提高医生的工作效率，释放医疗系统的部分压力，通过人工智能对新冠病毒进行诊断是非常有必要的。本文采用 Inception-Resnet 网络架构和改进 U-Net 网络架构建立模型，通过实验数据验证了模型的性能。本文的总结如下：

（1）研究了卷积神经网络中各种模型，分析了它们的核心特点、实现方式、本身优缺点。通过归纳总结，形成了在基于深度学习视角下对于图像识别这一部

分较为完整的论述。

（2）本文在图像分类部分选取 Inception-Resnet 网络，该网络模型结合了 GoogLeNet 以及 ResNet 两个模型的优点，因此在数据集上各项结果都优于后两者。但本质上来说，该网络并没有从底层逻辑上创新。

（3）本文在图像分割方面选取的改进 U-Net 模型，核心思想是想从编码器、损失函数等方面对模型进一定的改进，迁移学习的思想能使得我们能在固定计算硬件等条件下更好的提高模型的准确率。注意力机制的效果也是类似，由于浅层特征图中提取了多余且冗长的信息，因此通过标注重要信息的方式提高模型的性能。

5.2 未来研究方向

本文虽然最终完成了新冠病毒肺炎识别系统的搭建，但存在以下问题需要完善：

（1）本文实验结果较好，其可能原因是本文选取的图像都较为清晰，即数据本身较好。在实际应用过程由于现实各种因素，可能得到的图像不会这么清晰，其结果较实验结果可能会略有差别，未来可通过使用更大、更全的数据作为样本对模型进行一定调参。

（2）由于新冠病毒不停的变异，未来新的毒株会造成什么样的影响无法预测，新的毒株对于肺部的影响也可能和之前的毒株造成的 CT 图像不同，该系统存在无法识别新的变异毒株引起的病症的可能性。因此未来可通过深度学习继续发展，创造出性能更好的模型解决此问题。

（3）本文最终完成的新冠病毒 CT 图像识别模型还未曾应用到手机 APP 程序上，这样在操作上不够便利。未来应该考虑尽量减轻模型的量级，以便能适用在手机 APP 上。

备用

图片备用

图：

表 4.2 肺部 CT 图像数据集

分类	正常 CT 图像	普通肺炎 CT 图像	新冠肺炎 CT 图像
总计	2500	2500	2500
训练集	1600	1600	1600
验证集	400	400	400
测试集	500	500	500

表格备用

表 4.1 注意力机制执行步骤

注意力机制执行步骤
第一步：对 g 做 1*1 卷积得到 1×256×64×64 特征图
第二步：对 x^l 做 1*1 卷积得到 1×256×64×64 特征图
第三步：为了突出特征，将第一步和第二步的结果相加
第四步：对第三步结果进行 ReLU 函数激活

第五步：对第四步结果做 Conv(256,1)卷积,将 256 通道降到 1 通道。得到 1×1×64×64 的特征图

第六步：对第 5 步结果进行 sigmoid，使得值落在（0，1）区间，值越大，越是重点。这个该值也就是注意力权重。

第七步：对图片使用 resampler 使得图片大小一样

第八步：最后和 x' 相乘，把注意力权重赋到 low-level feature 中。

与其他深度神经网络一样，FNS 通常是基于随机梯度下降(SGD)损失函数[30]

表 4.2 肺部 CT 图像数据集

分类	正常 CT 图像	普通肺炎 CT 图像	新冠肺炎 CT 图像
总计	2500	2500	2500
训练集	1600	1600	1600
验证集	400	400	400
测试集	500	500	500

表 4.3 不同分割算法结果比较

算法	Dice	Jaccard	Precision
CNN	0.937	0.866	0.945
FCN	0.943	0.898	0.949
改进版 UNet	0.951	0.935	0.965

表 4.2 肺部 CT 图像数据集

分类	正常 CT 图像	普通肺炎 CT 图像	新冠肺炎 CT 图像
总计	2500	2500	2500
训练集	1600	1600	1600
验证集	400	400	400
测试集	500	500	500

表 3-1 表 肺部 CT 图像数据集

分类	正常肺部 CT 图像	普通肺炎 CT 图像	新冠肺炎 CT 图像
总计	3500	3500	3500
训练集	2500	2500	2500
验证集	500	500	500
测试集	500	500	500

表 1：仿真参数

参数名称	**值/分布**
f_i	[700, 1900]
v_i	7.683
θ	-4558.52
λ	0.00125
ΔT	1
r_{it}	U[820, 10000]

r_{it}'	U[200, r_{it}]
P_0	0.2
W	10 MHz
SNR	677

无边框表格

文字备用

这个卸载过程形成了一个三层的结构，包括车载云、路边单元的边缘服务器和中央云处理器。边缘服务器是车辆云的控制器。中央云则负责全局的调度。

影音娱乐、路径规划、辅助驾驶等任务。

在移动边缘计算中，位于边缘端的边缘设备承担了计算任务，那么，拥有计算能力的车辆同样可以视为边缘端，车辆云（Vehicular Cloud，VC）是汽车组成的资源系统。城市地区有大量的车辆，将空闲的车辆资源加以利用，可以为移动设备提供巨大的资源和价值，通过整合车辆空闲的计算资源形成车辆云，为其他车辆或者移动用户提供服务，可以大大提高服务和应用的质量。同时，在汽车移动的过程中，有些车辆的位置是重合的，将某些车辆作为服务端，就不会有超

出服务范围的问题。

智能网联车辆包含使用无线通信议的设备，车辆能以较低的通信延迟接发计算任务，充分利用车辆上空闲的计算、存储资源，在保证车辆安全行驶的前提下，降低任务处理延迟，提高用户的驾驶体验。车辆云中车辆的角色不是固定的，当车辆资源空闲时可以作为服务提供者，接受来自其他车辆或是移动设备的任务请求；当车辆需要卸载任务时，可以作为服务请求者，将任务卸载至其他拥有空闲计算资源的车辆，来实现资源的整合和资源利用率的提高。

1.2.1 任务卸载的评价指标

对车联网中任务卸载的研究中，主要研究两个指标，节约能耗占比与时延。当节约能耗占比越大时，说明越多的任务分配到边缘或云，意味着发送的数据量越大，时延越高。

2.3 车联网中的任务卸载

任务卸载是指将移动设备（如智能手机、车载终端等）上的计算任务分解成若干个子任务，其中一部分在本地完成，另一部分则通过网络卸载到云端或者其他设备上进行计算，最终将结果返回到本地设备，从而提高移动设备的计算性能和能耗效率的一种技术。

根据不同的任务卸载方式和卸载目标，可以将任务卸载分为以下几种类型：

（1）本地卸载：任务完全在本地设备上完成，不需要使用其他远程设备或云端资源；

（2）远程卸载：将部分任务卸载到远程设备或云端上进行计算，通过网络返回计算结果；

（3）协同卸载：多个本地设备相互协作，将部分任务卸载到其他本地设备上进行计算，以达到计算效率和能耗的均衡优化；

（4）分布式卸载：将任务分解成多个子任务，通过分布式计算的方式在多个设

备上并行计算，最后合并结果并返回。

本文中选择了协同卸载的卸载方式

根据卸载的目标，任务卸载还可以分为以下两类：

（1）计算卸载：主要目的是将计算密集型任务卸载到云端或其他远程设备上，减少本地设备的计算负担，提高计算性能和能耗效率；

（2）通信卸载：主要目的是将数据密集型任务卸载到云端或其他远程设备上进行处理，减少本地设备的数据传输负担，提高通信性能和能耗效率。

车联网中任务卸载主要应用于以下几个方面：

（1）智能驾驶：智能驾驶需要大量的计算和数据处理能力，通过任务卸载技术，可以将一部分计算和数据处理任务卸载到云端或其他设备上，提高计算性能和能耗效率；

（2）车辆监控：车辆监控需要实时收集和处理大量的数据，并进行数据分析和决策，通过任务卸载技术，可以将一部分数据处理任务卸载到云端或其他设备上，提高数据处理效率；

（3）车辆诊断：车辆诊断需要进行大量的数据分析和处理，通过任务卸载技术，可以将一部分数据处理任务卸载到云端或其他设备上，提高诊断效率和精度；

（4）车联网服务：车联网服务需要进行大量的数据计算和处理，通过任务卸载技术，可以将一部分计算任务卸载到云端或其他设备上，提高服务性能和能耗效率。

边缘计算是指在网络边缘中处理数据的一种计算方式。由于移动智能设备的普及、数据量的爆发式增长和低延迟的需求，边缘计算日益成为解决云计算缺点的方案之一。与传统的云计算相比，边缘计算更注重将服务部署在更接近用户的地方，可以实现更低的时延和更高的带宽，同时也减少了核心网络的负载，还十分有利于数据安全以及隐私保护。

早期的云计算技术是通过集中式计算，将用户的计算任务卸载到资源丰富的远程云中进行处理。这种方式使得数据不能做到及时响应，在一些需要毫秒级响应的应用中不能发挥很好的作用。

通过将服务部署在更靠近移动用户的接入侧，边缘计算可以实现低时延和高带宽，减轻核心网络的负载。边缘计算也可以通过与物联网技术结合，实现一些智能化的应用，如自动驾驶、智慧城市等。美国自然基金委和英特尔在 2016 年开始合作讨论建设无线边缘网络。同一年，中国也兴起了移动边缘计算的研究，由科研院所、高等院校、企业、行业协会等单位共同发起，边缘计算产业联盟在北京揭牌成立。

1.2.2 计算卸载

参考文献

[1] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[2]Ciresan D, Giusti A, Gambardella L, et al. Deep neural networks segment neuronal membranes in electron microscopy images[J]. Advances in neural information processing systems, 2012, 25.

[3]Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Pa rt III 18. Springer International Publishing, 2015: 234-241.

[4]Lessmann N, Sánchez C I, Beenen L, et al. Automated assessment of CO-RADS and chest CT severity scores in patients with suspected COVID-19 using artificial intelligence[J]. Radiology, 2020.

[5]Zheng Y Y, Ma Y T, Zhang J Y, et al. COVID-19 and the cardiovascular system[J]. Nature reviews cardiology, 2020, 17(5): 259-260.

[6]Bai H X, Wang R, Xiong Z, et al. Artificial intelligence augmentation of radiologist performance in distinguishing COVID-19 from pneumonia of other origin at chest CT[J]. Radiology, 2020, 296(3): E156-E165.

[7]Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks [C]//International conference on machine learning. PMLR, 2019: 6105-6114.

[8]Wang S H, Govindaraj V V, Górriz J M, et al. Covid-19 classification by FGCNet wit h deep feature fusion from graph convolutional network and convolutional neural network [J]. Information Fusion, 2021, 67: 208-229.

[9]Hua K L, Hsu C H, Hidayati S C, et al. Computer-aided classification of lung nodules on computed tomography images via deep learning technique[J]. OncoTargets and therapy, 2015: 2015-2022.

[10]David C, Nöhammer B, Solak H H, et al. Differential x-ray phase contrast imaging us ing a shearing interferometer[J]. Applied physics letters, 2002, 81(17): 3287-3289.

[11]Martinez-Murcia F J, Górriz J M, Ramírez J, et al. Convolutional neural networks for

neuroimaging in Parkinson’s disease: Is preprocessing needed?[J]. International journal of n eural systems, 2018, 28(10): 1850035.

[12]Avci D, Leblebicioglu M K, Poyraz M, et al. A new method based on adaptive discre te wavelet entropy energy and neural network classifier (ADWEENN) for recognition of ur ine cells from microscopic images independent of rotation and scaling[J]. Journal of medic al systems, 2014, 38: 1-9.

[13]Wei G, Li G, Zhao J, et al. Development of a LeNet-5 gas identification CNN structu re for electronic noses[J]. Sensors, 2019, 19(1): 217.

[14]Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutiona l neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[15]Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image rec ognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[16]Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentatio n[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 201 5: 3431-3440.

[17]Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical imag e segmentation[C]//Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, P art III 18. Springer International Publishing, 2015: 234-241.

[18]He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Procee dings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[19]Falk T, Mai D, Bensch R, et al. U-Net: deep learning for cell counting, detection, and morphometry[J]. Nature methods, 2019, 16(1): 67-70.

[20]Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[C]//Proceedings of the AAAI conference on artificial in telligence. 2017, 31(1).

[21]Kim B C , Choi J S , Suk H I . Multi-Scale Gradual Integration CNN for False Posi tive Reduction in Pulmonary Nodule Detection[J].2018,24(7):10581

[22]康牧. 图像处理中几个关键算法的研究[D].西安电子科技大学,2009.

[23]Hasana MK, Dahala L, Samarakoonb PN, Tushara F, Martía R. DSNet: Automatic dermoscopic skin lesion segmentation. IEEE Int Conf Comput Adv Bio Med Sci 2020:103738. InPress.

[24]Tran TT, Pham VT, Shyu KK. Image segmentation using fuzzy energy-based active contour with shape prior. J Vis Commun Image Represent 2014;25:1732–45.

[25]Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks [C]//International conference on machine learning. PMLR, 2019: 6105-6114.

[26]Xu X, Wen Y, Zhao L, et al. CARes-UNet: Content-aware residual UNet for lesion segmentation of COVID-19 from chest CT images[J]. Medical Physics, 2021, 48(11): 7127-7140.

[27]Hu J, Shen L, Sun G. Squeeze-and-Excitation networks. Proc Conf Comput Vis Pattern Recognit (CVPR) 2018:7132–41.

[28]Oktay O, Schlemper J, Folgoc LL, Lee M, Heinrich M, Misawa K, et al. Attention U Net: learning where to look for the pancreas. Proc 1st Conf Med Imaging with Deep Learn 2018. p. Available

[29]Pham V T, Tran T T, Wang P C, et al. EAR-UNet: A deep learning-based approach forsegmentation of tympanic membranes from otoscopic images[J]. Artificial Intelligence in Medicine, 2021, 115: 102065.

[30]王驰. 面向随机森林与随机梯度下降的分解学习算法研究[D].哈尔滨工业大学,2021.DOI:10.27061/d.cnki.ghgdu.2021.001948.

[31]Izmailov P, Podoprikhin D, Garipov T, Vetrov D, Wilson A. Averaging weights leads towider optima and better generalization. 34th Conference on Uncertainty in Artificial Intelligence 2018:876–85.

[32]Tang P, Liang Q, Yan X, Xiang S, Sun W, Zhang D, et al. Efficient skin lesion segmentation using separable-Unet with stochastic weight averaging. Comput Methods Programs Biomed 2019;178:289–301.

[33]吴云峰. 基于深度学习的肺炎医学CT图像分类算法研究[D].福建中医药大学,2021.DOI:10.27021/d.cnki.gfjzc.2021.000011.

[34]代宇涵. 基于深度学习的肺部CT辅助诊断算法研究[D].重庆邮电大学,2021.DOI:10.27

675/d.cnki.gcydx.2021.000334.

致谢

这三年时光匆匆流逝，如白驹过隙般，回首昨日仿佛还在 2020 年 9 月的那个夏天。仔细回想这段记忆，它的确是我想要拥有的，也无比庆幸当年自己坚持考研的那个决定。不知不觉已经到了写致谢的时候，直到落笔这一刻心中思绪万千而又不知所措，想写些东西出来但又不知道该从何说起。就闲谈一些自己这三年时光的经历、感受。

首先很感激我的导师 XX 老师。每个老师可以教给学生的东西很多，XX 老师并没有逼着我们去读博，去走学术研究这条路。而是告诉我，自己想要做什么，那就去做，朝着自己的目标前进。正是 XX 老师给我的这种支持，让我能顺利找到一份满意的工作。研二的时候参加数学建模竞赛，跟着师兄、学姐一起学到了很多东西，而这个宝贵的机会也恰恰是 XX 老师给我的。此外感谢 XXX 院长允许我参加他的学术讨论班，让我的见识有了更进一步的提高。

感谢我的父母能支持我读研，他们披星戴月，日夜奔波，让我能在如此安逸、舒适的环境下度过我这充实的三年求学时光。感谢我的室友梁瀚文、邹欣卫、潘渊，隔壁寝室的唐弋超和戎思宇以及像半个室友一样的黄治国，你们在日常生活和学习中给予了我太多太多的帮助。感谢同组的冀伟、李孟航、孙佳丽、于群，和你们相处的时光总是让人放松、惬意。

最后，我非常感谢我的女朋友龙家英，其实要说感谢或许也有点不当，请原谅我的辞藻不够描述此刻的心情。很庆幸和她相识、相恋。我们携手走过春华，抓过夏蝉，采过秋实，赏过冬雪。我希望未来我的一生波澜壮阔也好，风平浪静也罢，只要携手一生的人是她就足矣。

研究生这三年对我来说非常宝贵，但已经走到了尾声。新的征程即将开启，愿所有我爱的人和爱我的人，大家都得偿所愿，幸福安康。