1. **研究意义**

随着万物互联时代的到来，网络边缘设备产生的数据量快速增加，带来了更高的数据传输带宽需求，同时，新型应用也对数据处理的实时性提出了更高要求，传统云计算模型已经无法有效应对，因此，边缘计算应运而生，边缘计算的基本理念是将计算任务在接近数据源的计算资源上运行。

边缘计算的一大应用场景就是智能汽车。随着机器视觉、深度学习和传感器等技术的发展，汽车的功能不再局限于传统的出行和运输工具，而是逐渐变为一个智能的、互联的计算系统，我们称这样新型的汽车为智能网联车（connected and auto nomous vehicles,CAVs）。智能网联车的出现催生出了一系列新的应用场景，例如自动驾驶、车联网以及智能交通。Intel在2016年的报告指出一辆自动驾驶车辆一天产生的数据为4TB，这些数据无法全部上传至云端处理，需要在边缘节点（汽车）中存储和计算。

清洁能源是未来的一个重要趋势。电能作为清洁能源的代表，越来越多的人选择了使用电能的电动汽车。

在这个万物互联的时代，越来越多的设备联网，汽车也不例外，这就是所谓的智能汽车。电动智能车辆将成为未来城市交通的组成部分[1]。然而，在电池技术发生重大变革之前，电池的容量始终是电动智能汽车的短板。有限的电池容量会导致“续航焦虑”的问题。

人们提出了许多解决里程问题的方法，例如考虑整车的能量, 对整车的能量进行管理[2]。车辆滑行是驾驶模式的一部分，使用滑行可以减少能耗[3]。

由于行驶过程中需要处理的数据量过大，中央处理器将消耗相当大的能量，这对行驶里程有很大影响。一些研究建议将任务发送到移动边缘计算服务器以节省能源。但如果所有数据都上传到云服务器进行处理，响应时间可能太长，无法满足低延迟要求[4]。Liu etal[5]在论文中提出了一种保证任务低延迟性能的系统。此外，目前的带宽和存储根本无法满足传输所有数据的要求。

为了解决这个问题，Yu et.al [6]将重点放在车辆边缘云中的虚拟机资源分配上。Liu et.al[7]考虑了动态需求和资源约束，并将所有任务划分为四种类型的待定列表。然后，每个列表中的任务将根据其功能卸载到不同的节点。

1. 研究的主要内容

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。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。我们建议将任务分配给附近的车辆，以最小化系统的能耗。

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。

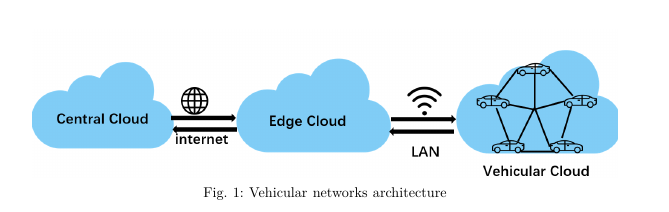
这个卸载过程形成了一个三层的结构，包括车载云、路边单元的边缘服务器和中央云处理器。边缘服务器是车量云的控制器。中央云则负责全局的调度。

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。

3．**[思路方法]** 本课题研究的基本思路、具体研究方法、研究计划及其可行性等。

**3.1 网络结构**

与类似，如图1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，它执行的任务包括复杂的计算和全局决策。边缘云是车辆云的控制器，负责创建、维护和删除车辆云。车辆云由共享其计算资源的车辆组成。

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  时间段共享资源。

**3.2 计算模型**

我们将资源共享时间定义为 .。车辆数量定义为 。

我们使用元组  来表示时间为  时，车辆  的任务大小，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  时， 时车辆  必须要在本地执行的任务， 是车辆  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 为分配矩阵，如果 ，代表车辆执行车辆所卸载的任务。注意，如果,车辆  所有的任务都在本地执行。

我们将车辆 的容量定义为 在时间为时。在时间，当车辆被选为提供者()，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：



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：



其中，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() 和CPU的频率()有如下的关系:



其中，和是待估计的参数。

最后，公式（2）中的CPU的容量在能被如下公式计算：



**3.3 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（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：



其中是CPU在时刻的频率。如果一辆车被选为了需求者（R），这辆车的频率会下降。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：



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()：



其中，是传输率。

最大的传输速率（b）可以通过香农公式计算：



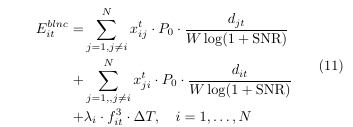
其中，SNR是信噪比，是频道的带宽。因为它和每一个场景相关，我们将他考虑为一个常数。

对于每一辆车，接收信息所消耗的能量是：



对于每一辆车,发送信息所需要的能量是：

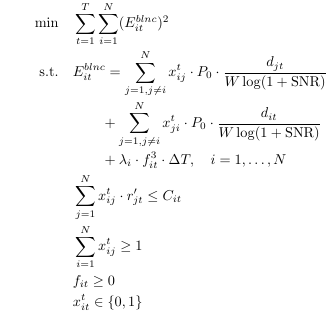


定义是车辆i在时刻所消耗的能量总和: 可以被这么计算

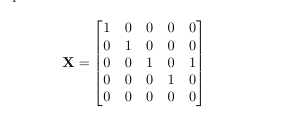
我们算法的设计目标是:最小化所有车辆能量消耗的平方。这个目标既能够考虑最小化能量消耗，又能考虑公平。也就是说平衡了能量消耗和公平。公式表示如下文：



该问题最终被建模为如下形式：



正如前面所提到的，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。

**3.4一个例子**

这是一个分配矩阵的例子，车辆的集合是｛A，B，C，D，E｝，车辆A，B，D没有没有分配自己的任务，或者负载其他车辆的任务。因此他们没有产生传输能耗。车辆C执行了车辆E的任务。每一列的和都大于等于1， 这满足了公式（18）这个约束。

**3.5 约束**

为了保证需求者和提供者之间的服务的质量（QoS）,下面的约束必须被满足:



其中，是车辆i和j之间的距离。

**3.6 具体研究计划**

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。

为了更好的描述计算资源和判断是否能够完成任务，我们使用CPU的频率来刻画计算资源。并且通过频率来计算小号的能量。

**3.6 研究计划与可行性**

2022年11月-2023年2月 阅读相关领域文献，如机制设计、资源分配，学习运筹学、经济学等相关知识，了解常用的启发式算法，构建任务卸载模型，并设计相关参数的计算公式，并从数学和经济学方面对问题进行建模。

2023年3月-2023年7月 进行仿真环境配置：使用云服务器作为我们模型中虚拟服务提供商。将建模的问题使用启发式的方法解决，也就是确定问题的解决办法。

2023年8月-2023年10月 将问题解决办法用编程语言实现，并进行仿真实验。分析实验数据，将实验结果使用图标的方式进行描述，进行论文的撰写。

2023年11月-2023年12月 将得到的实验数据生成图表添加到我们的论文中，并且进一步完善我们的论文以进行投稿。

4．**[创新之处]** 在学术思想、学术观点、研究方法等方面的特色和创新。

由于行驶过程中需要处理的数据量过大，中央处理器将消耗相当多的能量，这对行驶里程有很大影响。一些研究建议 (1) 将任务发送到移动边缘计算服务器以节省能源。但如果所有数据都上传到云服务器进行处理，响应时间可能太长，无法满足低延迟要求。(2)将重点放在车辆边缘云中的虚拟机资源分配上。(3)考虑了动态需求和资源约束，并将所有任务划分为四种类型的待定列表。然后，每个列表中的任务将根据其功能卸载到不同的节点。

他们的工作没有考虑将任务卸载到其他车辆中, 具有一定的局限性。我们的创新之处如下:

（1）我们考虑将任务卸载到附近的其他车辆中，这个过程由边缘服务器控制。

（2）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，以频率为基础计算能耗。为了更好的描述计算资源和判断是否能够完成任务，我们使用CPU的频率来刻画计算资源。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。

7. 论文工作进度安排

7. 论文工作进度安排

6. 论文前期已经完成的研究工作，和可能存在的相关问题

7. 论文工作进度安排

7. 论文工作进度安排

所提研究计划为一年，预期安排如下：

8．参考文献

[1] Zachary P Cano et al. “Batteries and fuel cells for emerging electric vehicle markets”. In: Nature Energy 3.4 (2018), pp. 279–289.

[2] T. Constantijn J. Romijn et al. “A Distributed Op- timization Approach for Complete Vehicle Energy Management”. In: IEEE Transactions on Control Systems Technology 27.3 (2019), pp. 964–980. doi: 10.1109/TCST.2018.2789464.

[3] Dalibor Barta et al. “Possibility of increasing vehicle energy balance using coasting”. In: Advances in Sci- ence and Technology. Research Journal 12.1 (2018).

[4] Virgilios Passas et al. “V2MEC: Low-Latency MEC for Vehicular Networks in 5G Disaggregated Archi- tectures”. In: 2021 IEEE Wireless Communications and Networking Conference (WCNC). IEEE. 2021, pp. 1–6.

[5] Chen-Feng Liu et al. “Dynamic Task Offloading and Resource Allocation for Ultra-Reliable Low-Latency Edge Computing”. In: IEEE Transactions on Com- munications 67.6 (June 2019). Conference Name: IEEE Transactions on Communications, pp. 4132– 4150. issn: 1558-0857. doi: 10.1109/TCOMM.2019. 2898573.

[6] Rong Yu et al. “Toward cloud-based vehicular net- works with efficient resource management”. In: Ieee Network 27.5 (2013), pp. 48–55.

[7] Chunhui Liu et al. “Adaptive Offloading for Time- Critical Tasks in Heterogeneous Internet of Vehi- cles”. In: IEEE Internet of Things Journal 7.9 (2020), pp. 7999–8011. doi: 10.1109/JIOT.2020.2997720.

[8] Kyong Hoon Kim, Rajkumar Buyya, and Jong Kim. “Power Aware Scheduling of Bag-of-Tasks Applica- tions with Deadline Constraints on DVS-enabled Clusters”. In: Seventh IEEE International Sympo- sium on Cluster Computing and the Grid (CCGrid ’07). 2007, pp. 541–548. doi: 10 . 1109 / CCGRID . 2007.85.

[9] R. Ge, Xizhou Feng, and K.W. Cameron. “Performance-constrained Distributed DVS Scheduling for Scientific Applications on Power- aware Clusters”. In: SC ’05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing. 2005, pp. 34–34. doi: 10.1109/SC.2005.57.

[10] E. N. (Mootaz) Elnozahy, Michael Kistler, and Ramakrishnan Rajamony. “Energy-Efficient Server Clusters”. In: Power-Aware Computer Systems. Ed. by Babak Falsafi and T. N. Vijaykumar. Berlin, Hei- delberg: Springer Berlin Heidelberg, 2003, pp. 179– 197. isbn: 978-3-540-36612-6.

[11] J. Li and J.F. Martinez. “Dynamic power- performance adaptation of parallel computation on chip multiprocessors”. In: The Twelfth International Symposium on High-Performance Computer Architecture, 2006. 2006, pp. 77–87. doi: 10.1109/HPCA.2006.1598114.

[12] C. Piguet, C. Schuster, and J.-L. Nagel. “Optimizing architecture activity and logic depth for static and dynamic power reduction”. In: The 2nd Annual IEEE Northeast Workshop on Circuits and Systems, 2004. NEWCAS 2004. 2004, pp. 41–44. doi: 10 . 1109 / NEWCAS.2004.1359011.

[13] Jeffrey R. Sampson. “Adaptation in Natural and Artificial Systems (John H. Holland)”. In: SIAM Review 18.3 (1976), pp. 529–530. doi: 10 . 1137 / 1018105. eprint: https://doi.org/10.1137/1018105. url: <https://doi.org/10.1137/1018105>.

[14] Vincent Roberge, Mohammed Tarbouchi, and Gilles Labonté. “Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning”. In: IEEE Transactions on industrial informatics 9.1 (2012), pp. 132–141.

[15] Yourim Yoon and Yong-Hyuk Kim. “An Efficient Genetic Algorithm for Maximum Coverage Deploy- ment in Wireless Sensor Networks”. In: IEEE Trans- actions on Cybernetics 43.5 (2013), pp. 1473–1483. doi: 10.1109/TCYB.2013.2250955.

[16] Anand Kannan et al. “Genetic algorithm based feature selection algorithm for effective intrusion detection in cloud networks”. In: 2012 IEEE 12th In- ternational Conference on Data Mining Workshops. IEEE. 2012, pp. 416–423.

[17] R Sivaraj and T Ravichandran. “A review of selec- tion methods in genetic algorithm”. In: International journal of engineering science and technology 3.5 (2011), pp. 3792–3797.

[18] Cortex-A57. url: https : / / developer . arm . com / Processors/Cortex-A57 (visited on 07/20/2022).

[19] Tayebeh Bahreini, Marco Brocanelli, and Daniel Grosu. “VECMAN: A Framework for Energy-Aware Resource Management in Vehicular Edge Comput- ing Systems”. In: IEEE Transactions on Mobile Computing (2021), pp. 1–1. doi: 10 . 1109 / TMC . 2021.3089338.

[20] Zhigang Xu et al. “DSRC versus 4G-LTE for con- nected vehicle applications: A study on field experi- ments of vehicular communication performance”. In: Journal of advanced transportation 2017 (2017).