车联网边缘计算场景下基于改进型NSGA-II算法的边缘服务器部署决策

朱思峰[1]，王钰[1]，陈昊[1]，朱海[2]，柴争义[3]，杨诚瑞[1]
（1. 天津城建大学计算机与信息工程学院，天津 300384；2. 河南工程学院计算机学院，河南 郑州 451191；
3. 天津工业大学计算机科学与技术学院，天津 300387）

摘 要：车联网环境下，边缘服务器的放置位置与部署数量直接影响到边缘计算的效率。由于在宏基站或基站上部署大型边缘服务器的成本较高，可以在微基站上部署一个小型边缘服务器作为补充，并通过优化大型边缘服务器的放置位置来降低成本。为了最小化边缘服务器的部署代价和服务延迟、最大化运营商的收入和服务器负载均衡度，把边缘服务器放置问题与车联网用户应用服务放置问题联合建模为一个多目标优化问题，并提出了基于改进型 NSGA-II算法的放置方案。实验结果表明，提出的边缘服务器放置方案能够降低约 44% 的边缘服务器部署成本，降低约 14.2% 的时延，提升 24.2% 的运营商收入，具有较好的应用价值。

关键词：车联网；边缘计算；边缘服务器部署问题；多目标优化算法；NSGA-II
中图分类号：TP391
文献标志码：A
doi：10.11959/j.issn.2096−3750.2024.00382

Edge server deployment decision based on improved NSGA-II in the Internet of vehicles edge computing scenario

ZHU Sifeng[1], WANG Yu[1], CHEN Hao[1], ZHU Hai[2], CHAI Zhengyi[3], YANG Chengrui[1]
1. School of Computer and Information Engineering, Tianjin Chengjian University, Tianjin 300384, China
2. School of Computer, Henan University of Engineering, Zhengzhou 451191, China
3. School of Computer Science and Technology, Tiangong University, Tianjin 300387, China

Abstract: In the context of the Internet of vehicles, the placement and deployment number of edge servers directly affect the efficiency of edge computing. Due to the high cost of deploying a large edge server on a macro base station and a base station, it can be complemented by deploying a small edge server on a micro base station, and the cost reduction needs to be optimized by optimizing the placement of large edge servers. In order to minimize the deployment cost and service delay of the edge server, and maximize the operator's revenue and server load balance, the edge server placement problem combined with the vehicle networking user application service was modeled as a multi-objective optimization problem and a placement scheme based on improved NSGA-II algorithm was proposed. The experimental results show that the proposed scheme can reduce the deployment cost of edge servers by about 44%, the latency by about 14.2%, and improve the revenue of operators by 24.2%, which has good application value.

Key words: Internet of vehicles, edge computing, edge server placement problem, multi-objective optimization algorithm, NSGA-II

收稿日期：2023−06−13；**修回日期：**2023−12−16
通信作者：杨诚瑞，1025512113@qq.com
基金项目：国家自然科学基金项目（No. 62172457）；天津市自然科学基金重点项目（No. 22JCZDJC00600）
Foundation Items: The National Natural Science Foundation of China (No. 62172457), Key projects of Tianjin Natural Science Foundation (No. 22JCZDJC00600)

0　引言

目前，各种移动应用在网络边缘产生的数据正以指数级增加，这对移动通信网络的服务要求越来越高[1]。现在的云计算架构中，移动车联网用户直接将其计算任务卸载至云服务器中，由云服务器处理这些移动车联网用户的任务[2]。若将所有分布式数据和计算密集型任务卸载至云服务器，必将给云服务器带来严重的负担，导致云服务器的服务质量（QoS, quality of service）降低、传输延迟增高。随着车联网技术的快速发展，移动边缘计算（MEC, mobile edge computing）已经成为一种处理车联网用户任务的新模式，它将计算资源下沉至网络边缘，以提供低时延和高质量的服务。边缘服务器部署方面，现有的方案多是将边缘服务器部署在电信运营商的宏基站（MeNB, marco eNodeB）上，为车联网用户提供服务[3]。若边缘服务器无法为车联网用户提供某种服务，则需要访问计算能力与容量更强的云服务器。近年来，小蜂窝技术备受关注，可以依靠其庞大的数量无缝覆盖整个网络，从而提高系统吞吐量并且减少系统总能耗[4]。5G网络中，微基站（SeNB, small cell eNodeB）数量越来越多，已经到了随处可见的地步。预计在下一代移动通信网络中，基站的密度可达到每平方千米40~50个微基站，而大多数微基站将由企业或家庭部署并维护。微基站能够在开放车联网用户组（OSG, open subscriber group）的模式下运行，并在其作用范围内提供服务，但其服务范围与计算能力有限，无法完全为车联网用户提供其所需的实时性服务。每个边缘服务器接收不同的服务请求，这会影响部署在不同位置的边缘服务器的参数和后续服务放置[5]。为充分利用边缘服务器上的资源来提供更好的服务，不仅需要考虑边缘网络的异构性，还要考虑在现有边缘服务器部署方案的基础上进行适当的服务放置。在边缘服务部署方面，现有研究大多假设在每个宏基站上部署一个边缘服务器，每个宏基站都支持MEC，可以提供放置服务[6-11]，但这种假设与实际情况有些差距。如果在每个宏基站上都部署一个边缘服务器，则部署成本将非常高，电信运营商难以接受如此高的运营和维护成本。因此，需要优化边缘服务器的放置策略。现有研究没有考虑部署边缘服务器给电信运营商带来的收益问题，难以提升电信运营商在宏基站上部署边缘服务器的积极性。

为了提升电信运营商部署车联网用户应用服务器的积极性，本文提出了一种激励机制，由享受服务的车联网用户支付费用给电信运营商。这种激励机制可有效促进电信运营商将大型边缘服务器部署在其宏基站上，并且为边缘服务器提供长期的维护服务。电信运营商在微基站上部署的小型边缘服务器在小范围内为车联网用户提供的服务，由于小型边缘服务器大部分时间是为其附近非车联网用户提供服务且其服务能力有限，因此本文假设其处理车联网用户的服务是不收费的。此外，本文研究了一种更接近实际的服务放置场景，不仅考虑了不同服务的放置，还考虑到了每个服务的副本在边缘服务器中放置的数量，以便更有效地利用边缘服务器中的资源。为了使车联网用户能享受到高质量的服务且运营商能最大化其收益，本文以大型边缘服务器提供服务的时间成本、运营商收入和边缘服务器负载为优化目标，综合考虑大型边缘服务器放置问题。

本文研究了车联网场景下的边缘服务器放置问题，并提出了一种有效的放置方案。本文的主要贡献如下。

1) 将边缘服务器放置与车联网用户应用服务放置这两个问题结合起来，提供了相应的解决方案。

2) 提出了电信运营商的奖励机制，用以提升运营商的部署积极性。

3) 设计了一种改进的NSGA-II算法来实现对时间成本、运营商收入、边缘服务器负载和部署成本的多目标优化，并使用真实的数据对所提方案进行了仿真实验，实验结果验证了所提方案的可行性。

1　相关工作

随着5G的应用普及，移动网络边缘端产生的数据呈爆炸式增长，由于移动终端处理数据的能力十分有限，云计算应运而生。云计算模式大多需要通过广域网传输相应的数据，将移动终端的部分任务卸载至云服务器进行处理[12]，产生的网络时延与抖动会对实时交互性应用（如网络多人在线游戏）产生严重影响。下一代移动通信网络中，许多新兴应用（如车联网[13]、智能家居[14]等）对时延是极其敏感的，云计算模式显然不能满足这些新兴应用的

要求。MEC 正在成为支持上述新兴应用的主要技术。MEC 是近几年出现的一种结合 5G 架构的技术，可将其看成一种部署在数据源的微型的云服务平台[15]。在下一代移动通信网络中，在车联网场景下边缘服务器将被广泛部署在小蜂窝网络中，为在其覆盖范围的用户提供服务。在小蜂窝网络中，5G 覆盖的频率区间为 6~30 GHz，使得信号更容易穿过墙壁或其他障碍物，但美中不足的是频率高的信号衰减也会增强，从而降低其信号传输距离。作为未来的一种应用场景，具有不同计算能力的边缘服务器可以分别部署在微基站和宏基站上，用来为车联网用户提供服务。为了实现在数据源附近部署的目标，边缘服务器部署和服务放置这两方面需要被考虑[16]。

边缘服务器部署问题已经得到了广泛的研究[17-18]。与云计算模式不同，MEC 将边缘服务器放置在基站上[19]。边缘服务器的放置问题引起了国内外学者的关注：Cao 等[20]对部署异构边缘服务器进行了研究，提出了一种由离线和在线阶段组成的方法，以优化整个基站和单个基站的预期响应时间；Wang 等[21]考虑了在边缘服务器出现故障时，容错服务器可以及时替换故障服务器，使用一种灰狼遗传算法来优化边缘服务器的部署成本；Zhao 等[22]把智能城市中的边缘服务器放置问题抽象为多目标优化问题，在多种约束条件下，使用改进的多目标非支配排序遗传算法实现了最佳工作负载平衡和系统延迟。上述文献从不同角度和不同优化目标考虑了边缘服务器放置问题，然而美中不足的是，在现实生活中的车联网场景下边缘服务器需要处理的服务是多种多样的，这些文献只考虑了在单一服务的情况下边缘服务器的放置问题，而没有考虑在边缘服务器放置方案下的服务放置问题。

目前，车联网环境中的服务放置问题引起了一些学者的关注：文献[23]研究了服务的负载分配和放置问题，以尽量减少边缘服务器资源的限制和其他冲突的目标造成的违反服务水平协议（SLA, service level agreement）的情况；文献[24]研究了联合优化边缘服务器的接入网络选择和服务放置的问题，目的是通过明智地平衡接入延迟、通信延迟和服务切换成本，以具有成本效益的方式提高服务质量；文献[25]设计了一种基于不可分流的启发式算法，为每一类服务部署计算资源；文献[26]提出了一种并列的分类法，对云端服务放置方法和算法的相关研究进行分类，提供了对现有方法的统计和技术分析，并讨论了评价因素和属性；文献[27]针对数据分析应用，在计算和存储资源的限制下，联合优化 MEC 网络中的服务放置和请求路由；文献[28]提出了一种边缘云协作智能制造的微服务放置机制，其中提出了一种由精确数据驱动的 E2E 延迟估计方法支持的延迟感知边缘云协作放置的微服务放置算法；文献[29]综合考虑了车辆的移动性和变化的需求以及不同类型服务请求的动态性，提出了一种基于深度强化学习的动态服务放置框架；文献[30]介绍了目前对雾/边缘计算中的服务放置问题进行的研究；文献[31]设计了一个边缘服务器部署和服务放置的联合优化模型，使所有边缘服务器的部署受益最大化；文献[32]开发了一种新的算法，称为 PACK，用于将服务器放置作为一个有容量限制的位置分配问题，PACK 最大限度地缩短了服务器及其相关接入点之间的距离，同时考虑了负载平衡的容量限制，并支持服务器之间的工作负载共享。

上述文献在研究服务放置问题时，都是假设在每个宏基站上部署一个大型边缘服务器，将带来巨大的服务器部署代价。上海市的 5G 网络，宏基站的密度已经达到了每平方千米 8.2 个[33]，如果在每个基站上都部署一个大型边缘服务器，那部署成本将是短期内无法承受的。与上述文献不同，本文将边缘服务器放置和边缘服务放置进行联合优化，在同时分布着小蜂窝网络和宏蜂窝网络的场景下，提出了一种考虑部署成本、负载均衡、运营商受益等多个目标的优化部署方案。

2　系统模型和问题定义

2.1　系统模型

本文考虑的云边协同计算系统，如图 1 所示。该系统中有多个宏基站和微基站、多个边缘服务器和一个云服务器。

本文假设所有车联网用户都在基站的覆盖范围之内，车联网用户通过无线信道与基站通信。在一个时间段内一个车联网用户可能被多个基站所覆盖，但他只能连接其中一个基站。本文假设该系统中有 M 个微基站 $\mathrm{SeNB}=\{\mathrm{SeNB}_1,\mathrm{SeNB}_2,\cdots,\mathrm{SeNB}_M\}$，微基站 $\mathrm{SeNB}_i(1\leqslant i\leqslant M)$ 被抽象为一个二元组 $\mathrm{SeNB}_i=(n_i,d_{i,p})$，其中，$n_i$ 表示微基站 SeNB_i 信号

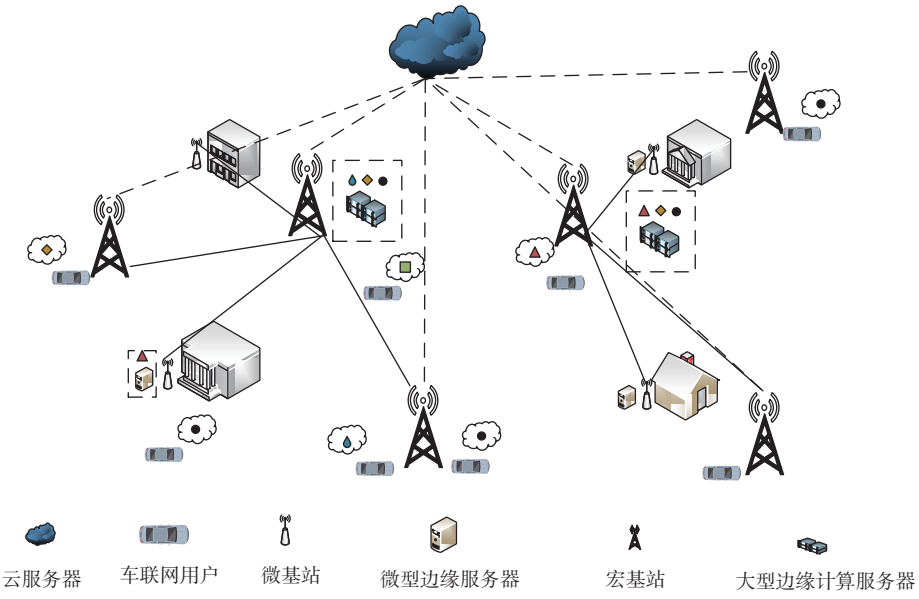

图1　云边协同计算系统

覆盖范围内的用户数为n_i个，$d_{i,p}$表示微基站SeNB_i到微基站SeNB_p的距离。每个微基站由家庭或建筑商部署在建筑中，假设每个建筑内都部署微基站，在每个微基站上部署与其匹配的小型边缘服务器$\mathrm{SES}=\{\mathrm{SES}_1,\mathrm{SES}_2,\cdots,\mathrm{SES}_M\}$，小型边缘服务器$\mathrm{SES}_i(1\leqslant i\leqslant M)$被抽象为一个五元组$\mathrm{SES}_i=(r_i^s,C_i,\mathrm{rep}_i^r,\mu_i,d_i^{\mathrm{Cloud}})$。其中，$r_i^s$表示小型边缘服务器$\mathrm{SES}_i$收到的服务请求，$C_i$表示小型边缘服务器$\mathrm{SES}_i$的存储容量，$\mathrm{rep}_i^r$表示放置在小型边缘服务器$\mathrm{SES}_i$中$S_r$的服务副本数量，$\mu_i$表示小型边缘服务器$\mathrm{SES}_i$的平均服务率，$d_i^{\mathrm{Cloud}}$表示小型边缘服务器$\mathrm{SES}_i$到远程云服务器之间的距离。假设该系统中有$N$个宏基站$\mathrm{MeNB}=\{\mathrm{MeNB}_1,\mathrm{MeNB}_2,\cdots,\mathrm{MeNB}_N\}$，宏基站$\mathrm{MeNB}_j(1\leqslant j\leqslant N)$被抽象为一个二元组$\mathrm{MeNB}_j=(n_j,d_{j,q})$，其中，$n_j$表示宏基站$\mathrm{MeNB}_j$信号覆盖范围内的车联网用户数量，$d_{j,q}$表示宏基站$\mathrm{MeNB}_j$到宏基站$\mathrm{MeNB}_q$的距离。而宏基站由电信运营商自行部署，考虑到部署成本，只有部分宏基站上部署与其匹配的大型边缘服务器，假设有K个大型边缘服务器$\mathrm{LES}=\{\mathrm{LES}_1,\mathrm{LES}_2,\cdots,\mathrm{LES}_K\}$，大型边缘服务器$\mathrm{LES}_k(1\leqslant k\leqslant K)$被抽象为一个六元组$\mathrm{LES}_k=(n_k,r_k^s,C_k,\mathrm{rep}_{k,r},\mu_k,d_k^{\mathrm{Cloud}})$。其中，$n_k$表示与大型边缘服务器$\mathrm{LES}_k$相连的宏基站服务范围内的车联网用户总数。$r_k^s$表示大型边缘服务器$\mathrm{LES}_k$收到的服务请求。$C_k$表示大型边缘服务器$\mathrm{LES}_k$的存储容量。$\mathrm{rep}_{k,r}$表示放置在大型边缘服务器$\mathrm{LES}_k$中$S_r$的服务副本数量。$\mu_k$表示大型边缘服务器$\mathrm{LES}_k$的平均服务率。$d_k^{\mathrm{Cloud}}$表示大型边缘服务器$\mathrm{LES}_k$到远程云服务器之间的距离。

本文假设不论是宏基站还是微基站都最多只能部署一个边缘服务器，每个微基站上由企业或者家庭部署一个微型的边缘服务器，而宏基站上是否部署边缘服务器未知，这也是本文的优化变量之一。每个边缘服务器应与其对应的基站处于同一位置。基站可以通过其他基站连接到大型边缘服务器，并忽略中继基站的转发时间。此外，由于资源有限，宏基站上部署的边缘服务器可能无法处理所有的服务请求，它将无法处理的请求通过基站转发给中心云。本文假设中心云远离所有的边缘服务器且具有各种服务，可以处理所有的服务请求。

宏基站分布不均匀，不同的大型边缘服务器负责数量不同的宏基站，每个大型边缘服务器所收到的服务请求不同，这会影响部署在不同位置的边缘服务器参数及边缘服务器上的服务部署方案。此外，边缘服务器之间频繁的服务迁移会导致成本大幅增加，因此本文不考虑边缘服务器之间的服务迁移。为了充分利用部署在微基站上小型边缘服务器和部署在宏基站上大型边缘服务器的资源，更好地为车联网用户提供服务，需要在现有的边缘服务器放置方案的基础上进行适当的服务放置。

假设系统中有R个服务请求$S=\{S_1,S_2,\cdots,S_R\}$，服务$S_r(1\leqslant r\leqslant R)$被抽象为一个二元组$S_r=(i_r,c_r)$，

其中，i_r表示服务S_r提供其副本的收入，c_r表示服务S_r所占用的存储容量。当车联网用户请求服务时，车联网用户首先需要与其连接的基站通信并发送其服务请求。然后，基站将请求转发给边缘服务器。边缘服务器处理请求后，会向车联网用户提供相关服务；否则，它会要求云中心处理请求并提供服务。

为了使电信运营商积极地将大型边缘服务器部署在宏基站上，本文提出一种对电信运营商的奖励机制，即大型边缘服务器为车联网用户提供的服务是按其服务种类进行收费，不同种类的服务对应不同的价格。而对于由家庭或企业部署在微基站上的小型服务器本文假设其提供的服务是不收费的。从电信运营商的角度出发，其在进行部署服务时希望尽可能地提高自己的利润，如图1所示，车联网用户需要的服务各不相同，在边缘服务器上部署不同的服务，电信运营商会收到不同收益。本文假设电信运营商为用户提供“红色三角”服务收取1元，提供“蓝色水滴”服务收取2元，提供“黄色菱形”服务收取2.5元，提供“黑色圆形”服务收取3元，提供“绿色方块”服务收取3.5元。

此外，本文根据车联网用户请求服务的历史，利用算出的每个车联网用户的平均服务请求率λ_S，反映出车联网用户服务请求在单位时间中的变化趋势，并利用其进行服务放置方案的设计。

在边缘服务器放置方面，本文用$\boldsymbol{X}=\{x_1,x_2,\cdots,x_N\}$表示边缘服务器放置的决策向量，其中，$x_j=1$表示在宏基站$\mathrm{MeNB}_j$上部署一个大型边缘服务器，$x_j=0$表示不在宏基站$\mathrm{MeNB}_j$上部署一个大型边缘服务器。

2.2　问题模型

如上所述，所有宏基站和微基站的位置都是固定的，而大型边缘服务器由于部署成本问题只能部署在一部分宏基站上，其中，$x_j=1$表示在宏基站MeNB_j上部署一个大型边缘服务器，$x_j=0$表示不在宏基站MeNB_j上部署一个大型边缘服务器。本文需要在N个宏基站上部署K个大型边缘服务器，表示为

$$\sum_{i=1}^{N}x_j=K \tag{1}$$

由于大型边缘服务器的数量远小于宏基站的数量，宏基站和微基站将无法在小型边缘服务器中处理的任务转发至相应的大型边缘服务器。本文将$a_{i,k}$表示基站SeNB_i是否被大型边缘服务器LES_k覆盖，将$a_{j,k}$表示宏基站MeNB_j是否被大型边缘服务器LES_k覆盖。若$a_{i,k}=1$，则微基站SeNB_i被大型边缘服务器LES_k覆盖，若$a_{j,k}=1$，则宏基站MeNB_j被大型边缘服务器LES_k覆盖。大型边缘服务器与微基站关系$a_{i,k}$和大型边缘服务器与宏基站关系a_k^j表示为

$$\sum_{k=1}^{K}a_{i,k}=1 \tag{2}$$

$$\sum_{k=1}^{K}a_{j,k}=1 \tag{3}$$

本文假设每个车联网用户对服务S_r的平均请求率为λ_r，在部署小型边缘服务器后，被小型边缘服务器覆盖的用户首先要与其连接的微基站进行通信，然后微基站与小型边缘服务器通信以请求相关服务。假设微基站SeNB_i覆盖车联网用户数为n_i，对于小型边缘服务器SES_i，其接收到的服务请求S_r的数量$r_{i,r}$表示为

$$r_{i,r}=n_i\times\lambda_r \tag{4}$$

小型边缘服务器SES_i其存储大小为C_i，而服务S_r所占用的存储容量为c_r，$\mathrm{rep}_{i,r}$表示放置在小型边缘服务器SES_i上服务S_r的副本，则放置在小型边缘服务器上SES_i所有服务副本所占用的存储容量大小不能超过小型边缘服务器上SES_i存储大小，表示为

$$\sum_{r=1}^{R}\mathrm{rep}_{i,r}\times c_r\leqslant C_i \tag{5}$$

为了请求服务，假设每个车联网用户只能连接一个宏基站或微基站，车联网用户首先与其连接的基站通信，然后基站向其部署的边缘服务器请求相关服务。如果与微基站连接的车联网用户请求无法被小型边缘服务器处理，则微基站将其转发给与之相对应的大型边缘服务器。对于宏基站MeNB_j，假设其覆盖车联网用户数为n_j，每个车联网用户对服务S_r的平均请求率为λ_r，放置在小型边缘服务器SES_i中的服务副本数量为$\mathrm{rep}_{i,r}$，对于大型边缘服务器LES_k，其接收到的服务请求S_r的数量$r_{k,r}$表示为

$$r_k^r=\sum_{i=1}^{M}a_{i,k}\times[\max(r_{i,r},\mathrm{rep}_{i,r})-\mathrm{rep}_{i,r}]+\sum_{j=1}^{N}a_{i,r}\times n_j\times\lambda_r \tag{6}$$

大型边缘服务器LES_k的存储大小为C_k，而服务S_r所占用的存储容量为c_r，$\mathrm{rep}_{k,r}$表示放置在大型

边缘服务器LES_k上服务S_r的副本，则放置在大型边缘服务器上所有服务副本所占用的存储容量大小不能超过其存储大小，表示为

$$\sum_{r=1}^{R}\mathrm{rep}_{k,r}\times c_r \leqslant C_k \tag{7}$$

若放置在大型边缘服务器上的服务所占用的存储容量超过其存储大小，则大型边缘服务器所对应的基站将其超出部分转发给云服务器进行处理，当然与此同时云服务传输时延会相应增加。

除了边缘服务器放置和边缘服务放置外，电信运营商的利润也是本文考虑的一个问题。假设对于服务S_r的服务请求，如果大型边缘服务器为其提供服务副本$\mathrm{rep}_{k,r}$，则电信运营商收取费用i_r，因此电信运营商提供服务的收入$\mathrm{income}_{\mathrm{TO}}$表示为

$$\mathrm{income}_{\mathrm{TO}}=\sum_{r=1}^{R}\sum_{k=1}^{K}r_{k,r}\times i_r \tag{8}$$

在大型边缘服务器放置方面，车联网用户请求到达边缘服务器延迟最小化是本文的优化目标之一。由于车联网用户或微基站请求传输到宏基站的时延远小于其对大型边缘服务器的请求时延，因此前者可以忽略。假设大型边缘服务器LES_k负责多个宏基站和微基站，其到大型边缘服务器LES_k的距离不同。本文设定$d_{j,k}$为宏基站MeNB_j到大型边缘服务器LES_k的距离，$d_{i,k}$为微基站SeNB_i到大型边缘服务器LES_k的距离，这些距离的平均值被视为服务请求传输到大型边缘服务器LES_k所需的距离d_k^S表示为

$$d_k^S=\frac{\sum_{j=1}^{N}a_{j,k}\times d_{j,k}+\sum_{i=1}^{M}a_{i,k}\times d_{i,k}}{\sum_{j=1}^{N}a_{j,k}+\sum_{i=1}^{M}a_{i,k}} \tag{9}$$

车联网用户的服务请求S_r到达大型边缘服务器的时延$t_{k,r}$与服务请求传输到大型边缘服务器LES_k所需的平均距离d_k^S成正比，本文假设服务的单位传输成本为tdc_s，则服务S_r到边缘服务器LES_k的传输延迟成本$\mathrm{cost}_{j,r}$表示为

$$\mathrm{cost}_{j,r}=\frac{\sum_{j=1}^{N}a_{j,k}\times d_{j,k}+\sum_{i=1}^{N}a_{i,k}\times d_{i,k}}{\sum_{j=1}^{N}a_{j,k}+\sum_{i=1}^{M}a_{i,k}}\times \mathrm{tdc}_s\times r_{k,r} \tag{10}$$

所有服务请求传输到边缘服务器的传输延迟成本$\mathrm{cost}_k^{\mathrm{trans}}$表示为

$$\mathrm{cost}_k^{\mathrm{trans}}=\sum_{r=1}^{R}\mathrm{cost}_{k,r} \tag{11}$$

边缘服务器在处理服务请求时存在计算时延。本文假设每个边缘服务器对服务请求的处理模型为一个M/M/1排队模型[34]。假设大型边缘服务器LES_k在单位时间可以处理的请求数为μ_k，边缘服务器在单位时间需要处理的请求数不得大于大型边缘服务器LES_k在单位时间可以处理的请求数μ_k，表示为

$$\sum_{r=1}^{R}\min\{\mathrm{rep}_k^r,\mathrm{rep}_{k,r}\}\leqslant\mu_k \tag{12}$$

假设cdc_s为计算服务的延迟成本单位，如M/M/1排队模型所述，大型边缘服务器LES_k处理所有服务的时延$\mathrm{cost}_k^{\mathrm{comp}}$表示为

$$\mathrm{cost}_k^{\mathrm{comp}}=\sum_{r=1}^{R}\frac{\mathrm{cdc}_s\times\min\{r_{k,r},\mathrm{rep}_{k,r}\}}{\mu_k-\min\{r_{k,r},\mathrm{rep}_{k,r}\}+1} \tag{13}$$

对于大型边缘服务器即使在存储容量和处理能力方面已经远超终端，但仍不足以应付5G时代服务请求的激增，而大型边缘服务器无法处理的服务请求应被转发至云服务器进行处理。在本文中，需要转发至云服务器的服务请求S_r数量r_r^{Cloud}表示为

$$r_r^{\mathrm{Cloud}}=r_{k,r}-\min\{r_{k,r},\mathrm{rep}_{k,r}\} \tag{14}$$

假设大型边缘服务器LES_k到云服务器的距离为d_k^{Cloud}，服务的单位传输成本为tdc_s，则将所有大型边缘服务器LES_k无法处理的服务传输至云服务器的传输延迟成本$\mathrm{cost}_{\mathrm{trans}}^{\mathrm{Cloud}}$表示为

$$\mathrm{cost}_{\mathrm{trans}}^{\mathrm{Cloud}}=\sum_{r=1}^{R}(r_r^{\mathrm{Cloud}}=r_{k,r}-\min\{r_{k,r},\mathrm{rep}_{k,r}\})\times d_k^{\mathrm{Cloud}}\times\mathrm{tdc}_s \tag{15}$$

基于上述分析，大型边缘服务器提供服务的时间成本cost_k表示为

$$\mathrm{cost}_k=\mathrm{cost}_k^{\mathrm{trans}}+\mathrm{cost}_k^{\mathrm{comp}}+\mathrm{cost}_k^{\mathrm{Cloud}} \tag{16}$$

在大型边缘服务器的负载平衡方面，本文用阿特金森指数[22]衡量基站分配不平衡的程度。阿特金森指数数值越小，基站分配越公平，阿特金森指数数值越大，基站分配越不公平。由于在该场景下有K个大型边缘服务器，则大型边缘服务器的平均工作量ϖ表示为

$$\varpi=\frac{\sum_{k=1}^{K}\sum_{r=1}^{R}\min\{r_{k,r},\mathrm{rep}_{k,r}\}}{K} \tag{17}$$

本文假设ε是对于不平等的厌恶，用阿特金森指数表达的负载平衡LB表示为

$$\mathrm{LB}=1-\left\{\frac{1}{R}\sum_{r=1}^{R}\left[\frac{\left(\min\left\{r_{k,r},\mathrm{rep}_{k,r}\right\}\right)^{1-\varepsilon}}{\varpi}\right]^{\frac{1}{1-\varepsilon}}\right\} \tag{18}$$

如果在每个基站上都部署一个大型边缘服务器，那部署成本将是电信运营商短期内无法承受的。因此，部署成本问题也是本文要考虑的问题。假设在宏基站 MeNB_j 部署大型边缘服务器的部署成本为 γ_j，则部署大型边缘服务器的总成本 γ 表示为

$$\gamma=\sum_{j=1}^{N}x_j\times\gamma_j \tag{19}$$

综上，本文建立多目标优化模型表示为

$$\begin{aligned}&\mathrm{P}:\max\mathrm{income}_{\mathrm{TO}}(X),\max\mathrm{LB}(X),\min\mathrm{cost}_k(X),\min\gamma(X)\\&\text{s.t.}\quad \mathrm{C1}:\sum_{k=1}^{K}a_{i,k}=1\\&\qquad\mathrm{C2}:\sum_{k=1}^{K}a_{j,k}=1\\&\qquad\mathrm{C3}:\sum_{r=1}^{R}\mathrm{rep}_{i,r}\times c_r\leqslant C_i\\&\qquad\mathrm{C4}:\sum_{r=1}^{R}\mathrm{rep}_{k,r}\times c_r\leqslant C_k\\&\qquad\mathrm{C5}:\sum_{r=1}^{R}\min\left\{r_{k,r},\mathrm{rep}_{k,r}\right\}\leqslant\mu_k\end{aligned} \tag{20}$$

其中，P 表示最小化大型边缘服务器提供服务的时间成本和部署成本，最大化运营商收入和负载均衡度作为需要优化的目标函数。C1 表示微基站 SeNB_i 必被某个大型边缘服务器覆盖；C2 表示宏基站 MeNB_j 必被某个大型边缘服务器覆盖；C3 表示放置在小型边缘服务器上所有服务副本所占用的存储容量大小不能超过其存储大小；C4 表示放置在大型边缘服务器上所有服务副本所占用的存储容量大小不能超过其存储大小；C5 表示边缘服务器在单位时间需要处理的请求数不得大于大型边缘服务器 LES_k 在单位时间可以处理的请求数 μ_k。

3　改进型 NSGA-II 放置策略

式(20)中的模型有两个变量需要求解，一个是大型边缘服务器的放置位置，另一个是服务的放置。只有当所有的大型边缘服务器位置固定后，$a_{i,k}$、$a_{j,k}$、$r_{k,r}$ 才能被确定，然后对 $\mathrm{rep}_{i,r}$ 和 $\mathrm{rep}_{k,r}$ 进行优化。

本文首先需对大型边缘服务器的放置问题进行求解，枚举法是解决该问题的最好办法，但如果将每个大型边缘服务器的位置和其所处理的用户服务全部枚举，时间复杂度太高，因此暴力算法不适合解决此问题，本文需要使用一种更加有效的算法。

边缘服务器放置问题从本质上来讲是一个 NP-hard 问题，并且随着基站和边缘服务器的数量增加，问题的求解空间将急剧增加，这使得精确的求解方法难以胜任。从上述模型来看，该问题是一个多目标问题，可以使用多目标进化算法获得一组最优解，成为帕累托解集。文献[22]利用 NSGA-II 算法来求解边缘服务器放置问题，其基本思路是：1) 随机生成初始种群；2) 对初始种群进行快速的非支配排序，计算种群拥挤度，然后通过选择、交叉和变异 3 种操作生成新的种群；3) 对种群进行快速非支配排序并计算拥挤度，将新的种群与原种群进行合并转化为新的种群，并使用精英策略将精英个体选入下一代；4) 返回 2)，直至迭代结束。但 NSGA-II 用在边缘服务器放置问题上，仍有局限性：基本的 NSGA-II 使用二进制编码方法来编码染色体，但是边缘服务器放置问题变量较多，如果采用二进制编码方法，会产生大量的二进制变量，不利于算法对变量的更新。本文采用了改进型 NSGA-II 来解决边缘服务器放置问题，这些改进包括编码方法、选择和交叉操作。

3.1　多目标优化问题与帕累托解集

本文定义了一个多目标优化问题，可表示为

$$\begin{cases}\max\mathrm{income}_{\mathrm{TO}}(X),\max\mathrm{LB}(X),\min\mathrm{cost}_k(X),\min\gamma(X)\\x\in\Omega\end{cases} \tag{21}$$

其中，$X=(x_1,x_2,\cdots,x_N)$ 为决策变量，表示边缘服务器的放置，$\mathrm{cost}_k(X)$ 表示用户任务传输时延，$\gamma(X)$ 为大型边缘服务器的部署成本，$\mathrm{LB}(X)$ 表示大型边缘服务器的负载。$\mathrm{cost}_r(X)$，$\gamma(X)$，$\mathrm{LB}(X)$ 构成了一个三目标优化问题。与单目标优化不同，多目标优化问题不能在每次迭代过程中根据优化目标选择最优解。根据帕累托的定义，在算法的每次迭代过程中将生成一组具有帕累托性质的解。

3.2　编码

在本文中，把问题候选解 $X=\{x_1,x_2,\cdots,x_i,\cdots,x_N\}$ 编码为染色体 $X=x_1x_2\cdots x_i\cdots x_n$，染色体基因 x_i 采用二进制编码方式 x_i 的取值表示基站 i 放置边缘

服务器的决策结果。$x_i = 1$表示第i个基站放置边缘服务器；$x_i = 0$表示第i个基站不放置边缘服务器。

3.3　非支配排序算法

快速非支配排序算法用于获得所有解的帕累托水平。对于大小为N的种群，本文首先根据帕累托支配计算每个个体的两个参数n_p和S_p。其中n_p表示支配个体p的其他支配个体的数量，S_p表示由个体p支配的个体集合。然后，可以使用算法1获得所有个体的帕累托水平。

算法1　快速非支配排序

输入　个体的所有目标函数的集合

输出　帕累托水平

根据第3.2节决定每个个体的n_p和S_p；

将参数$n_p = 0$的个体放入F_1（F_1是具有帕累托水平为1的个体）；

$j \leftarrow 1$；

while 得到所有个体的帕累托水平 **do**

　for $\forall i \in F_1$

　　for $\forall l \in S_i$

　　$n_l = n_l - 1$;

　　if $n_l = 0$ **then**

　　将个体l放入F_{l+1};

　　$j \leftarrow j + 1$;

　　end for

　end for

end while

return 所有个体的帕累托水平

3.4　拥挤度距离

在每次迭代中，为了确保种群的多样性，NSGA-II使用拥挤度策略来选择均匀分布在帕累托前沿的个体[29]。n_d表示个体n的拥挤程度，可以反映个体n周围其他个体的密度，其计算过程如算法2所示。

算法2　计算拥挤度距离

输入　种群P,种群各目标函数值的集合

输出　所有个体的拥挤度距离

使参数$n_d = 0$，$n \in \{1, 2, \cdots, N\}$;

for $i \leftarrow 1$ to M ***do***

　将第i个目标函数值f_i从小到大排序，并且把$f_i^{\max}$表示为f_i的最大值，$f_i^{\min}$表示为f_i的最小值；

　for $l \leftarrow 1$ to $(N - 1)$ **do**

$$l \leftarrow \frac{f_i(l+1) - f_i(l-1)}{f_i^{\max} - f_i^{\min}},\ (f_i(l+1)\text{是第}l+1\text{个个体的第}i\text{个目标函数值});$$

$n_d \leftarrow n_d + l_d$;

　end for

end for

return 所有个体的拥挤度距离

经过快速的非支配排序和拥挤计算，种群中的每个个体都获得了两个属性，分别为帕累托水平n_{rank}和拥挤度n_d。利用这两个属性，本文可以确定任何两个个体的优劣。

3.5　初始种群

本文按照以下步骤进行种群的初始化。

步骤1：设计一个$N \times m$的向量来保存种群（N为种群中个体数量，m为基站数量），然后转到步骤2。

步骤2：根据3.2节中的编码方法和规则初始化染色体并加入群体，然后转到步骤3。

步骤3：重复步骤2，直到产生一个个体并退出循环。

3.6　选择和交叉操作

选择和交叉操作对遗传算法的寻优精度有重要影响。为了保证择优的准确性，在选择操作中，本文每次都选择与优秀染色体交叉，选择系统开销较小的染色体作为优秀的父亲染色体，并与母亲染色体交叉。使用这种选择策略可以保存到目前为止在迭代过程中发现的优秀基因，从而使染色体继续朝着更好的方向进化。对于交叉操作，本文使用多点交叉策略。在每次交叉期间，首先根据策略选择一个交叉节点，然后交换位于交叉节点之后的父染色体和母染色体中的所有基因。

本文根据非线性式选择交叉节点的位置。使用这种方法，在算法的早期阶段，交叉节点位于染色体的起始处，此时发生交叉的基因数量很高，并且算法具有很强的搜索能力。在随后的迭代中，交叉节点的位置以更快的速度向染色体尾部移动，发生交叉的基因数量越来越少，这有助于算法收敛到某个值。

交叉节点位置的确定方式，可表示为

$$\text{cp} = \left\lceil m \times \frac{l^2}{L^2} \right\rceil - 1 \tag{22}$$

其中，cp是交点节点的位置，l是当前迭代次数，L

是最大迭代次数。

在确定了父亲染色体和母亲染色体后，本文根据以下步骤来进行选择与交叉操作。

步骤 1：根据式(22)确定交叉节点的位置，并转至步骤2。

步骤 2：确定母染色体，然后转到步骤3。

步骤 3：交换位于交叉节点之后的父染色体和母染色体中的基因。

3.7　变异操作

变异操作可以为算法提供前所未有的解决方案，并增加种群多样性。结合第3.2节中提出的染色体编码方法和规则，本文使用的突变策略是随机破坏整个染色体上的基因排列。利用这种突变，可以有效地确保种群的多样性。变异操作之后按照第3.2节进行规则检查。

多项式变异操作形式，可表示为

$$x_i' = x_i + \delta \cdot (u_i - l_i) \tag{23}$$

其中，δ表示为

$$\begin{cases} 2\rho + (1-2\rho)(1-\delta^1)^{\frac{1}{\eta_m+1}}, \rho \leqslant 0.5 \\ 1 - \left[2(1-\rho) + 2(\rho - 0.5)(1-\delta^2)\right], \rho > 0.5 \end{cases} \tag{24}$$

其中，$\delta^1 = (x_i - l_i)/(u_i - l_i)$，$\delta^2 = (u_i - x_i)/(u_i - l_i)$，$\rho$为一个[0, 1]区间内的随机数，$\eta_m$是分布指数，$x_i$表示一个父代个体，$u_i$、$l_i$分别表示对应维度决策空间的上下限。

3.8　精英策略

改进的NSGA-II在迭代过程中使用精英策略为下一次迭代选择染色体，所选择的染色体称为精英个体。精英策略首先将父母染色体和后代染色体结合起来，形成$2N$大小的群体。然后，它使用拥挤度比较，以确定染色体的优劣势关系，并最终确定进入下一次迭代的精英染色体。具体步骤如下。

1) 将父代P_t和子代Q_t全部个体合成为一个统一的种群$R_t = P_t \cup Q_t$，R_t的个体数为$2N$；

2) 将种群R_t快速非支配排序并计算每一个体的局部拥挤距离，依据等级的高低逐一选取个体，直到个体数量达到N时就形成了新的父代种群P_{t+1}；

3) 在此基础上开始新一轮的选择、交叉和变异，形成新的子代种群Q_{t+1}。

3.9　改进的NSGA-II算法

本文设计的改进型NSGA-II算法如算法3所示。

算法 3　具有精英策略的改进多目标非支配排序遗传算法

输入　网络图G，种群数量N，宏基站数据集，最大迭代次数L，交叉率M_r，变异率C_r，用户数据集，微基站数据集，服务数据集

输出　BL，cost_k，$\gamma(X)$，$\text{income}_{\text{TO}}(X)$

根据第3.5节初始化种群

while 迭代次数<L **do**

根据式(16)、式(18)和式(19)计算所有染色体的目标函数值；

执行步骤1，以执行快速非支配的种群排序；

执行步骤2，计算拥挤度距离；

根据第3.6节确定父染色体；

while 生成种群规模为N的新种群 **do**

确定母染色体；

产生一个随机数R；

根据式(23)确定交叉位置；

if $R < C_r$ **then**

根据第3.6节进行交叉操作；

else

不产生交叉；

生成一个随机数R；

if $R < M_r$ **then**

根据第3.7节执行变异操作；

else

不执行变异操作；

一条新染色体的产生；

end while

根据式(16)、式(18)和式(19)计算新种群的目标函数值；

对新种群执行步骤1从而进行快速非支配排序；

对新种群执行步骤2从而计算个体的拥挤度距离；

原始种群与新种群的合并；

执行精英策略；

end while

得出最优个体编码；

得出最优个体的负载、传输延迟成本、运营商收入和部署成本。

3.10　算法稳定性与收敛性

反向世代距离（IGD, inverted generational dis-

tance）是一个用于评价多目标优化算法性能的指标，它与算法的收敛性和稳定性之间存在密切关系。

IGD可以用来度量算法生成的解集合与真实前沿之间的距离，包括距离的平均值。当一个算法具有较好的收敛性时，它通常能够在较短的时间内生成接近真实前沿的解，因此IGD值会较小。较小的IGD值表示较好的收敛性。

IGD也可以用来评价算法生成的解集合与真实前沿的距离，包括距离的平均值。如果一个算法在多次独立运行中能够产生相似的解集合，那么它通常被认为具有较好的稳定性。稳定的算法应该在不同运行中产生类似的IGD值。

本文对改进的NSGA-II算法的IGD值进行仿真，IGD值分析如图2所示。

从图2可知，随着迭代次数的增加，算法逐渐趋于收敛和稳定。

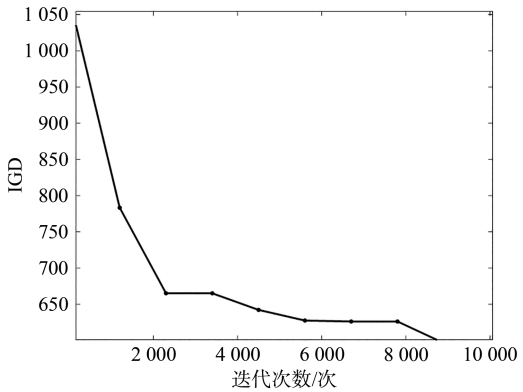

图2　IGD值分析

4　实验结果及分析

本文将所提出的解决方案应用于实际数据集，并将所提方案的实验结果（包括传输成本、负载和部署成本）与其他方案的结果进行比较，解决天津市西青区张家窝镇的边缘服务器的放置问题。

4.1　数据集描述

为了证明改进的NSGA-II算法的优越性，本文使用真实的数据集来进行仿真，该数据集包含来自天津市西青区张家窝镇部分区域的宏基站和微基站地理位置信息（经度和纬度）、用户接入信息与腾讯天津云数据处理中心的地理位置数据（经度和纬度）。车联网用户信息数据集是通过与一家汽车制造商建立合作关系得到的。该制造商配备了一批车辆，这些车辆都安装了先进的车载通信设备，包括GPS接收器、OBD-II接口以及与车辆内部系统连接的传感器。这些设备能够实时收集车辆性能、位置、速度、引擎状态等信息。原始数据集包含大量在本文仿真之前必须要进行处理的无用信息。在处理与过滤无用信息之后，本文总共获得了其中的150个基站的有用信息，还有3 000个车联网用户的信息和微基站的服务器地理位置信息（经度和纬度），其位置信息如图3所示。

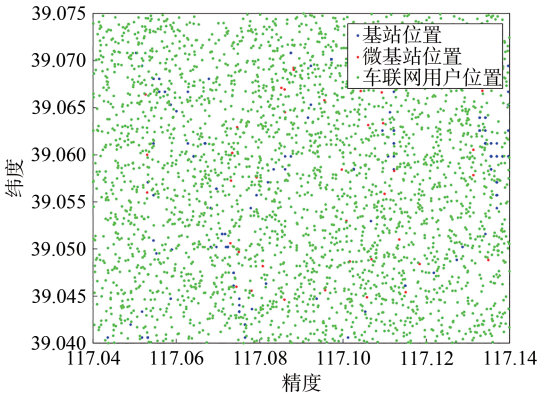

图3　位置信息

4.2　对比算法

为了验证算法的优越性，本文使用以下4种经典的多目标算法来进行对比。

1) NSGA-II算法

它采用了非支配排序算法，计算复杂度比NSGA大大降低，并采用拥挤度和拥挤算子，使其中个体能拓展到整个帕累托域，保持了种群多样性，引入了精英策略，扩大了采样空间，防止最佳个体丢失，提高了算法的运算速度与鲁棒性。

2) NSGA-III算法

NSGA-II与NSGA-III具有类似框架，但NSGA-III引入了广泛分布的参考点来维持种群的多样性。

3) MOEA/D算法

将分解方法引入MOEA算法，形成了MOEA/D算法。MOEA/D把一个多目标优化问题分解为若干个标量优化子问题，并同时对其进行优化，对非分解MOEA的适应度分配和多样性维护等问题可能会变得更容易处理。MOEA/D将MOP分解为标量子问题，它通过进化出一系列解决方案来同时解决这些子问题。在每一代中，总体由算法运行开始为每个子问题找到的最佳解决方案组成。这些子问题之间的邻域关系是根据它们的权重系数向量之间

的距离来定义的。两个相邻子问题的最优解应该非常相似。每个子问题在MOEA/D中仅使用其相邻子问题的信息进行优化。

4) SPEA-II算法

SPEA-II将非支配解存储在一个不断更新的种群中，根据一个个体独自支配它的非支配解的个数计算适应度值，根据帕累托支配关系保存种群多样性，并且为了减少非支配解集并不破坏它的特征，加入了聚类分析过程。

4.3　实验设计

本文在使用Intel Core i7-11800H CPU（2.3 GHz和32 GB RAM）的计算机上运行了实验。算法用MATLAB R2021a实现。为了评估改进的NSGA-II算法的性能，本文设计了以下实验。

由于大型边缘服务器由电信运营商统一采购，因此型号必须统一。为了确保真实性，本文采用的边缘服务器型号为Think System SR658，CPU：5218（2.3 GHz/16核32线程/22 M/125 W）×2，部署成本为686 000元。而小型边缘服务器由于是由企业或家庭部署并维护，因此本文选取了市面上最常见的4种小型边缘服务器，小型边缘服务器的参数见表1。

表1　　小型边缘服务器的参数

服务器类型	内核处理频率	内核数量	计算能力	部署成本
HPE DL388	2.4 GHz	10	24.0 GHz	21 510元
Dell R750XS	2.1 GHz	12	25.2 GHz	18 599元
Lenovo SR588	2.4 GHz	10	24.0 GHz	16 999元
Inspur NF5280M5	1.9 GHz	6	11.4 GHz	14 699元

将大型边缘服务器放置好后，本文将对服务进行放置，服务种类和服务价格见表2。

表2　　服务种类和服务价格

服务种类	A	B	C	D	E
服务价格/元	1.0	2.0	2.5	3.0	3.5

4.4　对比分析

为了验证所提出的改进的NSGA-II算法的性能，本文进行了大量实验。在本文中，边缘服务器接收所有车联网用户请求服务的时间消耗被视作评估边缘服务器放置问题的重要指标，需要对这一指标进行详细的分析。

4.4.1　5种方案的Pareto前沿面

5种方案生成的Pareto前沿面如图4所示。

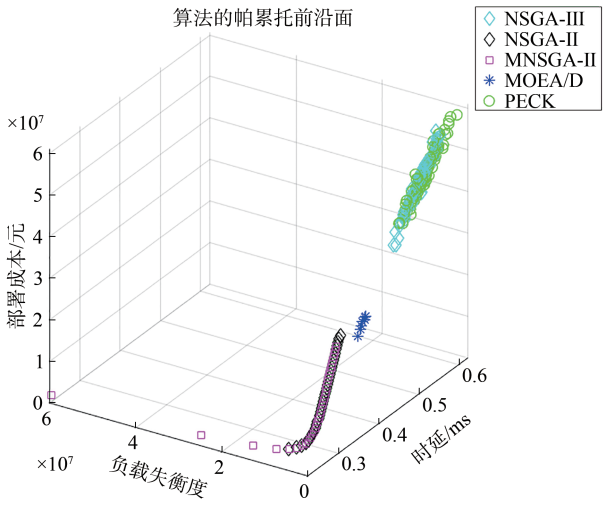

图4　5种方案生成的Pareto前沿面

与NSGA-III、NSGA-II、MOEA/D和SPEA-II方案相比，改进的NSGA-II方案生成的最优解集Pareto前沿面更接近于中心点。这表明将改进的NSGA-II方案应用于高效节能卸载策略优化的时候，其收敛性和仿真实验结果多样性方面要更加优秀一些，这是因为改进的NSGA-II方案进行多目标优化时种群分布更加均匀，优势个体更多，且具有更强的全局搜索能力。改进后的NSGA-II方案更符合车联网场景下服务器放置问题的低时延、低部署成本、高负载均衡度和高运营商收入。

4.4.2　平均时延的比较

本文使用非支配排序，选出5种方案中前10种最优的放置策略。5种方案的平均时延对比如图5所示。

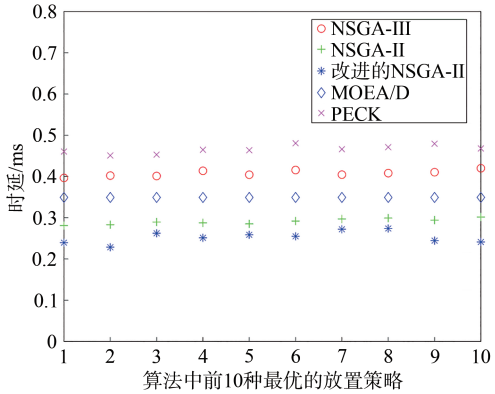

图5　5种方案的平均时延对比

与NSGA-III、NSGA-II、MOEA/D和PECK方案相比，改进的NSGA-II方案生成的平均时延最短。这表明将改进的NSGA-II方案应用于下一代

移动通信网络中的边缘服务器放置问题的时候，因为车联网场景下服务器放置问题中的平均时延与车联网用户距请求服务的平均距离成正比，相较NSGA-III、NSGA-II、MOEA/D和PECK，其在降低时延的效果上更优秀一些。

4.4.3　负载均衡度比较

本文使用非支配排序，选出5种方案中前10种最优的放置策略。5种方案的负载均衡度对比如图6所示。

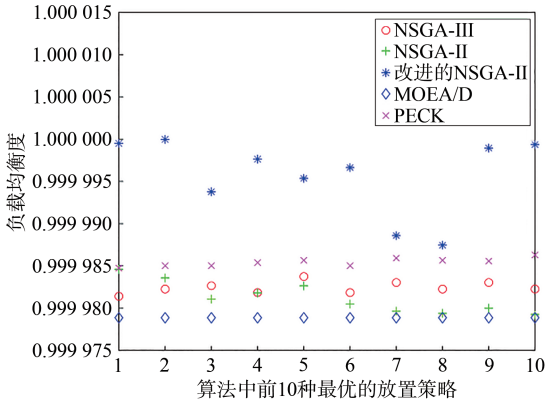

图6　5种方案的负载均衡度对比

与NSGA-III、NSGA-II、MOEA/D和PECK方案相比，改进的NSGA-II方案生成的负载均衡度最大。这表明将改进的NSGA-II方案应用于下一代移动通信网络中的边缘服务器放置问题的时候，因为车联网场景下服务器放置问题中的负载均衡度与宏基站部署边缘服务器服务的个数差成正比，相较NSGA-III、NSGA-II、MOEA/D和PECK，其在提升负载均衡度的效果上更优秀一些。

4.4.4　部署成本的比较

本文使用非支配排序，选出5种方案中前10种最优的放置策略。5种方案的部署成本如图7所示。

与NSGA-III、NSGA-II、MOEA/D和PECK方案相比，改进的NSGA-II方案生成的大型边缘服务器部署成本最低。这表明将改进的NSGA-II方案应用于下一代移动通信网络中的边缘服务器放置问题的时候，因为车联网场景下服务器放置问题中的部署成本与宏基站部署边缘服务器的个数成正比，相较NSGA-III、NSGA-II、MOEA/D和PECK，其在降低大型边缘服务器部署成本的效果上更优秀一些。

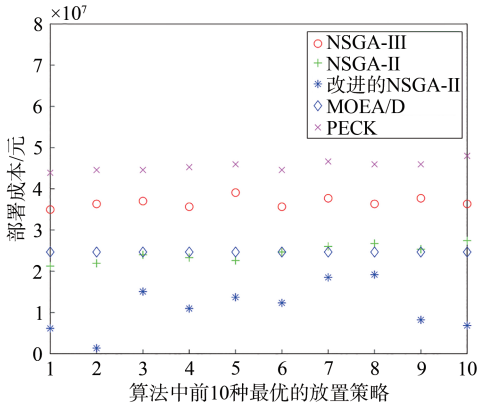

图7　5种方案的部署成本

4.4.5　运营商收入

本文使用非支配排序，选出5种方案中前10种最优的放置策略。5种方案的运营商收入如图8所示。

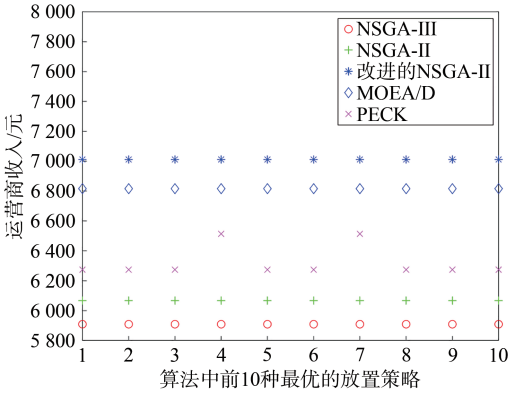

图8　5种方案的运营商收入

与NSGA-III、NSGA-II、MOEA/D和PECK方案相比，改进的NSGA-II方案生成的运营商收入最大。这表明将改进的NSGA-II方案应用于下一代移动通信网络中的边缘服务器放置问题的时候，因为车联网场景下服务器放置问题中的运营商收入与宏基站部署边缘服务器提供服务能力成正比，相较NSGA-III、NSGA-II、MOEA/D和PECK，其在提升运营商收入的效果上更优秀。

5　结束语

本文研究了在车联网场景下的边缘服务器放置问题，它可为某些地区运营商放置大型边缘服务器提供一些参考。本文首先介绍了在下一代移动通信网络中与边缘服务器放置相关的问题；然后将其建模为具有运营商收入最大化目标、大型边缘服务器配置代价最小化目标、卸载时延最小化目标、负载均衡度最小化目标的多目标优化问题；最后提出了

一种改进的多目标非支配排序精英策略遗传算法来求解。基于真实数据集的实验结果表明，基于改进的NSGA-II的边缘服务器放置方案，在优化运营商收入、大型边缘服务器配置代价、卸载时延和负载均衡度方面表现良好。边缘服务器的能量消耗也是边缘服务器放置过程中不可忽略的因素，下一步工作将研究边缘服务器放置的能量消耗。此外，在现实场景中，移动的车联网用户的位置不是固定的，而是变化的，因此考虑车联网用户的移动性也是未来工作的重点。

参考文献：

[1] 张依琳, 梁玉珠, 尹沐君, 等. 移动边缘计算中计算卸载方案研究综述[J]. 计算机学报, 2021, 44(12): 2406-2430.
ZHANG Y L, LIANG Y Z, YIN M J, et al. Survey on the methods of computation offloading in mobile edge computing[J]. Chinese Journal of Computers, 2021, 44(12): 2406-2430.

[2] ZHANG L X, ZHOU L Q, SALAH A. Efficient scientific workflow scheduling for deadline-constrained parallel tasks in cloud computing environments[J]. Information Sciences, 2020, 531: 31-46.

[3] MAO Y Y, ZHANG J, LETAIEF K B. Dynamic computation offloading for mobile-edge computing with energy harvesting devices[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(12): 3590-3605.

[4] GE X H, TU S, MAO G Q, et al. 5G ultra-dense cellular networks[J]. IEEE Wireless Communications, 2016, 23(1): 72-79.

[5] SHEN B W, XU X L, QI L Y, et al. Dynamic server placement in edge computing toward Internet of Vehicles[J]. Computer Communications, 2021, 178: 114-123.

[6] ZHAO X H, SHI Y, CHEN S Z. MAESP: mobility aware edge service placement in mobile edge networks[J]. Computer Networks, 2020, 182: 107435.

[7] OUYANG T, RUI L, XU C, et al. Adaptive user-managed service placement for mobile edge computing: an online learning approach[C]//Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2019: 1468-1476.

[8] XU J, CHEN L X, ZHOU P. Joint service caching and task offloading for mobile edge computing in dense networks[C]//Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2018: 207-215.

[9] TRAN T X, CHAN K, POMPILI D. COSTA: cost-aware service caching and task offloading assignment in mobile-edge computing [C]//Proceedings of the 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). Piscataway: IEEE Press, 2019: 1-9.

[10] HE T, KHAMFROUSH H, WANG S Q, et al. It's hard to share: joint service placement and request scheduling in edge clouds with sharable and non-sharable resources[C]//Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS). Piscataway: IEEE Press, 2018: 365-375.

[11] SANTOS F , IMMICH R , MADEIRA E R M .Multimedia services placement algorithm for cloud-fog hierarchical environments[J]. Computer communications, 2022.191:78-91.

[12] LI C L, SONG M Y, YU C C, et al. Mobility and marginal gain based content caching and placement for cooperative edge-cloud computing[J]. Information Sciences, 2021, 548: 153-176.

[13] 韩牟, 杨晨, 华蕾, 等. 面向移动边缘计算车联网中车辆假名管理方案[J]. 计算机研究与发展, 2022, 59(4): 781-795.
HAN M, YANG C, HUA L, et al. Vehicle pseudonym management scheme in Internet of vehicles for mobile edge computing[J]. Journal of Computer Research and Development, 2022, 59(4): 781-795.

[14] 张珠君, 范伟, 朱大立. 面向智能家居的区块链轻量级认证机制[J]. 软件学报, 2022, 33(7): 2699-2715.
ZHANG Z J, FAN W, ZHU D L. Lightweight blockchain authentication mechanism for smart home[J]. Journal of Software, 2022, 33(7): 2699-2715.

[15] LUO Y Z, DING W R, ZHANG B C. Optimization of task scheduling and dynamic service strategy for multi-UAV-enabled mobile-edge computing system[J]. IEEE Transactions on Cognitive Communications and Networking, 2021, 7(3): 970-984.

[16] HADŽIĆ I, ABE Y, WOITHE H C. Server placement and selection for edge computing in the ePC[J]. IEEE Transactions on Services Computing, 2019, 12(5): 671-684.

[17] JIA M K, CAO J N, LIANG W F. Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks[J]. IEEE Transactions on Cloud Computing, 2017, 5(4): 725-737.

[18] FAN Q, ANSARI N. On cost aware cloudlet placement for mobile edge computing[J]. IEEE/CAA Journal of Automatica Sinica, 2019, 6(4): 926-937.

[19] WANG S G, ZHAO Y L, XU J, et al. Edge server placement in mobile edge computing[J]. Journal of Parallel and Distributed Computing, 2019, 127: 160-168.

[20] CAO K, LI L Y, CUI Y G, et al. Exploring placement of heterogeneous edge servers for response time minimization in mobile edge-cloud computing[J]. IEEE Transactions on Industrial Informatics, 2021, 17(1): 494-503.

[21] WANG Z M, ZHANG W Y, JIN X M, et al. An optimal edge server placement approach for cost reduction and load balancing in intelligent manufacturing[J]. The Journal of Supercomputing, 2022, 78(3): 4032-4056.

[22] ZHAO X B, ZENG Y, DING H W, et al. Optimize the placement of edge server between workload balancing and system delay in smart city[J]. Peer-to-Peer Networking and Applications, 2021, 14(6): 3778-3792.

[23] MAIA A M, GHAMRI-DOUDANE Y, VIEIRA D, et al. An

improved multi-objective genetic algorithm with heuristic initialization for service placement and load distribution in edge computing[J]. Computer networks, 2021,194(20): 108146.1-108146.15.
MAIA A M, GHAMRI-DOUDANE Y, VIEIRA D, et al. An improved multi-objective genetic algorithm with heuristic initialization for service placement and load distribution in edge computing[J]. Computer Networks, 2021, 194: 108146.

[24] GAO B, ZHOU Z, LIU F M, et al. An online framework for joint network selection and service placement in mobile edge computing[J]. IEEE Transactions on Mobile Computing, 2022, 21(11): 3836-3851.

[25] ZHANG Z H, WU G W, REN H Z. Multi-attribute-based QoS-aware virtual network function placement and service chaining algorithms in smart cities[J]. Computers & Electrical Engineering, 2021, 96: 107465.

[26] HENG L, YIN G F, ZHAO X F. Energy aware cloud-edge service placement approaches in the Internet of Things communications[J]. International Journal of Communication Systems, 2022, 35(1): e4899.1-e4899.23.

[27] YUAN B B, GUO S T, WANG Q Y. Joint service placement and request routing in mobile edge computing[J]. Ad Hoc Networks, 2021, 120: 102543.

[28] WANG Y M, ZHAO C, YANG S S, et al. MPCSM: microservice placement for edge-cloud collaborative smart manufacturing[J]. IEEE Transactions on Industrial Informatics, 2021, 17(9): 5898-5908.

[29] TALPUR A, GURUSAMY M. DRLD-SP: a deep-reinforcement-learning-based dynamic service placement in edge-enabled Internet of vehicles[J]. IEEE Internet of Things Journal, 2022, 9(8): 6239-6251.

[30] SALAHT F A, DESPREZ F, LEBRE A. An overview of service placement problem in fog and edge computing[J]. ACM Computing Surveys, 53(3): 65.

[31] ZHANG X L, LI Z J, LAI C, et al. Joint edge server placement and service placement in mobile-edge computing[J]. IEEE Internet of Things Journal, 2022, 9(13): 11261-11274.

[32] LÄHDERANTA T, LEPPÄNEN T, RUHA L, et al. Edge computing server placement with capacitated location allocation[J]. Journal of Parallel and Distributed Computing, 2021, 153: 130-149.

[33] 黄景源 . 上海 5G 基站建设密度全国排名第一，将持续推进 700 MHz 频段补充完善 5G 网络覆盖[J]. 界面新闻, 2022.
HUANG J Y. Shanghai's 5G base station construction density ranks first in the country, and will continue to promote the 700 MHz frequency band to supplement and improve the 5G network coverage[J]. Jiemian News, 2022.

[34] TIAN R L, WANG Y L. Optimal strategies and pricing analysis in M/M/1 queues with a single working vacation and multiple vacations[J]. RAIRO-Operations Research, 2020, 54(6): 1593-1612.

［作者简介］

朱思峰（1975– ），男，博士，天津城建大学计算机与信息工程学院教授，主要研究方向为车联网、移动边缘计算和多目标优化算法等。

王钰（1996– ），男，天津城建大学计算机与信息工程学院硕士生，主要研究方向为车联网、边缘计算和多目标优化算法。

陈昊（1979– ），男，博士，天津城建大学计算机与信息工程学院副教授，主要研究方向为无线通信、通信系统接口、功率控制等。

柴争义（1976– ），男，博士，天津工业大学计算机科学与技术学院教授，主要研究方向为车联网、移动边缘计算和多目标优化算法等。

朱海（1977– ），男，博士，河南工程学院计算机学院教授，主要研究方向为车联网、移动边缘计算和多目标优化算法等。

杨诚瑞（1996– ），男，天津城建大学计算机与信息工程学院硕士生，主要研究方向为车联网、移动边缘计算和多目标优化算法等。