doi:10.3772/j.issn.1002-0470.2021.04.001

车联网中基于服务的虚拟网络功能放置算法①

韩淑君②[*,**] 李 俊③[*] 董 谦[*,**,***] 马宇翔[****] 宋留静[*,**]

([*]中国科学院计算机网络信息中心 北京 100190)

([**]中国科学院大学 北京 100049)

([***]佛山科学技术学院电子信息工程学院 佛山 528000)

([****]河南大学计算机与信息工程学院 开封 475004)

摘 要 为满足车联网中海量数据的采集、传输以及对这些数据的快速处理的需求,可采用移动边缘计算(MEC)技术。本文考虑移动边缘计算中基站连接方式和物理资源的特点,对边缘服务器的部署问题进行了分析,以部署成本和网络时延为优化目标,划分基站集群,并使用整数线性规划(ILP)建立模型。为了获得运行效率更高的边缘服务器部署方案,本文使用分支定界算法和启发式贪婪算法获得优化模型的近似最优解。实验评估结果显示,分支定界算法和启发式贪婪算法最高可以把边缘服务器部署算法运行时间减少37.6%。此外,本文分析了用户服务器请求数量和用户服务优先级对算法运行时间和边缘服务器运行成本的影响。

关键词 车联网;移动边缘计算(MEC);网络功能虚拟化(NFV);软件定义网络(SDN);服务质量(QoS)

0 引 言

近年来,汽车行业正经历着关键性、巨大的变革。国际研究及顾问机构 Gartner 预测,到 2020 年底全球联网车将超过 2.5 亿辆[1],2023 年时,汽车业将成为 5G 物联网解决方案最大市场商机,占比达到 53%,其中嵌入式汽车模组是 5G 的主要使用案例。随着车联网技术不断发展和应用,很多新兴的车载服务和应用程序等陆续被开发并应用,通过给汽车配备各种传感器和通信模块,采集汽车自身的各种行驶参数,整理分析后上传到通信网络基站,请求各种服务,例如自动驾驶服务、智能交通服务、娱乐办公服务等[2]。

这些新出现的车载应用需要通过传感器不断获取、分析、处理车辆本体以及周围环境的信息,所要求的计算能力和速度都很高,仅依靠车辆终端很难独立完成。除此之外,车辆网系统能够被应用的前提是对安全性的保障,这就要求车辆能够根据周围环境信息快速做出反应,对响应的速度有严格的要求。有研究表明,自动驾驶对服务响应延时的要求约为 5 ~ 10 ms[3],实时智能导航对网络来回时延的要求要小于 20 ms[4]。未来车联网是典型的“低时延、高带宽、高可靠”的应用场景。显然,现有移动设备和移动网络都不能满足这些场景要求。

移动云计算(mobile cloud computing,MCC)是借助于互联网,充分发挥后台云计算和处理能力的一种技术。在移动云计算中,移动设备执行任务不再依靠终端的计算能力,而是通过核心网络将任务上传到云计算平台,利用平台强大的计算能力来执行任务。

① 国家重点研发计划(2017YFB1401500),国家自然科学基金(61672490),河南省重点研发与推广专项(202102210352)和河南省青年人才托举工程(2020HYTP008)资助项目。

② 女,1986 年生,博士生;研究方向:网络体系结构,网络功能虚拟化;E-mail:hanshujun@cstnet.cn

③ 通信作者,E-mail:lijun@cnic.cn

(收稿日期:2020-04-03)

移动边缘计算(mobile edge computing,MEC)是为了避免移动承载网络被管道化,有效提升移动网络带宽价值而研究提出的一种新的技术。该技术基于 5G 演进架构,从传统的云计算演化而来,能够做到将原有的移动接入网与互联网业务深度融合。移动边缘计算可以在网络侧增加计算、存储等功能,将计算能力下沉到移动边缘节点,利用开放式平台植入应用,并与业务服务器之间进行信息交互,为移动边缘入口的服务创新提供了无限可能。

移动边缘计算可向行业提供高效、快速并有差异化的服务,与移动云计算相比,移动边缘计算有着近距离、超低时延、超高能效、超高可靠性等与车联网服务要求匹配的特点,是 5G 网络的一项关键技术[5]。

软件定义网络(software defined network,SDN)[6]和网络功能虚拟化(network functions virtualization,NFV)[7]作为支撑移动边缘计算的两项关键技术,重新解释了车联网服务的定义、编排以及路由规则。SDN 控制器从全局角度感知网络状态,结合 NFV 中对车载服务的编排,在物理层网络中确定虚拟网络功能的放置位置,以及服务流量的路径。这种网络实现方式,不仅满足了车联网服务对网络性能的高要求,也实现了网络资源的最大化利用。

本文主要考虑如何将具有计算、网络控制、计费、存储和安全认证等能力的各种虚拟网络功组合成不同的、能够满足车辆中车载系统和各类应用程序的边缘服务器,并将这些边缘服务器部署在基站中。边缘服务器中部署虚拟网络功能,需要综合考虑基站中计算、存储等资源分配情况,以及网络延时和故障恢复等网络服务质量(quality of service,QoS)。

由于不同运营商建设基站时选择的基站位置和硬件资源配置的差异性,使得一些基站不适合部署边缘服务器。这种情况下,如何选择部署边缘服务器的基站,使之既能够满足用户服务处理的需求,也可以最大程度提高对各种边缘服务器的使用,降低部署成本,就成为移动边缘计算中一个具有挑战性的问题。

本文的主要贡献如下。

(1) 根据联网车中各种应用程序的特点,使用优先级对用户服务进行细粒度的划分,兼顾了网络时延和部署成本。

(2) 设计边缘服务器集群式的管理方式,集群中边缘服务器处理所有集群中相同的用户服务请求,提高网络资源利用率,降低边缘服务器部署成本。

(3) 使用整数线性规划(integer linear programming, ILP)对边缘服务器集群中网络资源分配进行数学建模,并使用分支定界算法(branch and bound, B&B)和贪婪算法(greedy)对数学模型进行求解,使得数学模型在求解过程中兼顾了优化目标和求解效率。

1 模型描述与算法设计

1.1 相关工作

本节概述现有移动边缘服务器对资源分配方案及分配效果,探讨基于服务的虚拟网络功能放置算法的合理性。

现有的边缘服务器资源分配方案通常情况下是按照移动边缘计算系统特点进行设计的。移动边缘计算系统可以分为单用户 MEC 系统、多用户 MEC 系统以及异构无线网络的 MEC 系统。

在单用户 MEC 系统中,由于用户数量少,只要确定了用户任务的卸载方案,就可以选择合适的边缘服务器进行数据的处理和计算,并把计算结果返回给用户,这种场景资源分配计算比较简单,但是应用的场合不多。在多用户 MEC 系统中,需要同时考虑无线资源和服务器资源,一般会应用到 MEC 服务器调度、多用户协作等机制。由于无线传输有限的频谱限制了上传任务的速率,所以要作为一种资源与服务器中的资源进行联合分配,分配计算通常很复杂。

对于异构无线网络的 MEC 系统,国内外学者开展了较多研究。Lei 等人[8]提出了包括中央云和多个边缘服务器的异构 MEC 系统。Zhao 等人[9]针对多用户情况下的服务器选择问题开展研究,基于边缘服务器通信距离较短、中央云服务器的计算能力

通常未被充分发挥的情况,设计了相应模型,有效提高了系统中任务卸载的成功率。但是作为实际应用,文章中设计的网络结构显得较为单薄。Ge 等人[10]提出了拥塞博弈算法,解决了多用户系统中卸载任务的计算量与选定的边缘服务器之间的相关性问题。Dinh 等人[11]设计了一种基于板顶松弛算法的模型,建立了计算任务卸载框架,确定了中央处理器(central processing unit, CPU)频率缩放和任务卸载决策。文献[8-11]都是边缘服务器的资源管理方案,但是均将基站中的所有硬件资源作为边缘服务器可以使用的资源,没有考虑基站本身的其他任务对资源的使用需求。

Ceselli 等人[12]提出的 HAF(heaviest-AP first)算法根据负载对无线接入点(access point)进行排序,从中选择节点作为边缘服务器。Liang 等人[13]提出了一种基于 K-means 的位置动态感知算法,将用户请求使用 K-means 算法划分集群,将边缘服务器部署在集群中心的位置,减少网络延迟时间。但是文献[12,13]没有考虑用户服务的移动性。

1.2 基于服务的虚拟网络功能放置模型

1.2.1 三级 MEC 系统介绍

在移动边缘计算中,用户应用程序多种多样,为了满足这些用户请求的服务,边缘计算服务器使用 NFV 技术部署虚拟网络功能,通过对用户服务进行分析,确定服务需要占用的计算、存储等资源。但是由于移动边缘服务器本身也存在资源有限、越靠近边缘计算能力和存储空间越少的情况,本文采用三级 MEC 系统进行部署,如图 1 所示。

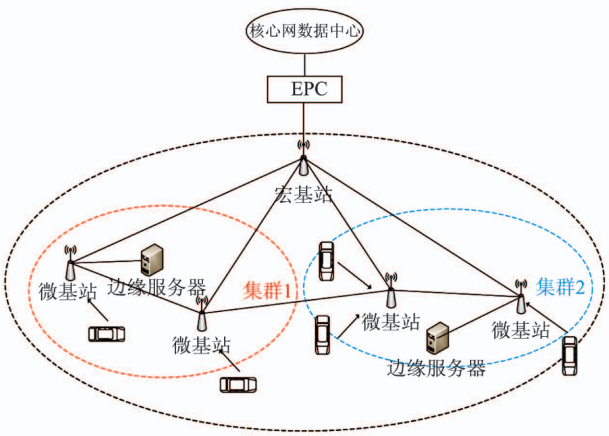

图 1 三级 MEC 系统示意图

图 1 中包含了 2 种类型的基站,分别是微基站(micro site)和宏基站(macro site)。这两种基站都可以作为移动边缘服务器的部署基站,完成接入用户服务请求。

在图 1 中,宏基站通过运营商蜂窝网络接入互联网,与远端数据中心进行通信。微基站与宏基站之间使用有线网络连接[14]。网络中宏基站的集合可以表示为 $M = \{1,2,3,\cdots,m\}$,每个宏基站中资源量表示为 $R_i = \{r_i^{cpu}, r_i^{cache}\}, 1 \leqslant i \leqslant m$。一个宏基站连接的 n 个微基站表示为 $N = \{1,2,3,\cdots,n\}$,每个微基站中资源量表示为 $R_{ij} = \{r_{ij}^{cpu}, r_{ij}^{cache}\}$, $1 \leqslant i \leqslant m$, $1 \leqslant j \leqslant n$, 表示第 i 个宏基站连接的第 j 个微基站。宏基站中划分的微基站集群表示为集合 $C = \{1,2,3,\cdots,c\}$,因为集群划分与实际用户的接入密度有关,所以集合 C 中的集群数量是会随着用户数量的变化而变化的,但是在每次部署边缘服务器的过程中,因为网络中基站集群划分已经完成,所以 c 值变为一个常数,只有在重新划分基站集群时才会再次发生变化。

需要说明的是,如果一个宏基站连接的所有微基站属于不同的集群,那么宏基站同时属于不同的集群。如图 1 所示,宏基站下存在两个微基站组成的集群,那么宏基站上可以同时部署分属于这两个集群的边缘服务器。但是连接于不同宏基站的微基站不能划分为同一个基站集群。

为了更好地服务于用户,本文没有采用传统的以用户为基本服务单位的策略,而是以用户服务为基本服务单位,使用向量记录每个用户服务的优先级和服务类型,其中服务类型包含处理用户服务需要消耗的服务器资源信息,比较常见的资源信息是指部署边缘服务器需要的 CPU 和缓存的信息,用户服务集合表示为 $S = \{1,2,3,\cdots,k\}$,每个用户服务表示为 $s_h = \{p_h, r_{s_h}^{cpu}, r_{s_h}^{cache}, price_h^{micro}, price_h^{macro}\}$, $1 \leqslant h \leqslant k$。集合中优先级 p_h 表示的是用户服务到达时,由哪一级移动边缘服务器进行处理,是一个二进制变量,表示用户服务是否部署在接入的微基站本地的边缘服务器上进行处理。服务优先级的确定是根据用户服务对于边缘服务器处理时延的要求进行划分的。

原则上,对时延敏感的用户服务优先在用户接入的微基站中的边缘服务器中进行处理,当服务器不能处理该用户服务时,考虑在微基站所在的集群中寻找合适的边缘服务器,最后才考虑到微基站连接的宏基站中进行处理。

1.2.2 模型分析

在考虑边缘服务器的放置位置时,需要综合考虑部署成本和边缘服务器对基站资源的使用以及网络时延,由此可以得出算法的优化目标。

$$\min \sum_{l=1}^{m}\sum_{i=1}^{n}\sum_{h=1}^{k} x_{ih}^{l} \times price_{h}^{micro} + \sum_{l=1}^{m}\sum_{j=1}^{c}\sum_{h=1}^{k} y_{jh}^{l} \times price_{h}^{macro} \tag{1}$$

$$\min \sum_{i=1}^{n}\sum_{h=1}^{k} (t_{ih}^{cpu} + t_{ih}^{up} + t_{ih}^{down}) + \sum_{j=1}^{c}\sum_{h=1}^{k} (t_{jh}^{cpu} + t_{jh}^{up} + t_{jh}^{down}) \tag{2}$$

式(1)中,最小化所有边缘服务器的部署成本,其中包括分别部署在微基站和宏基站上的所有用户服务的成本总和。使用矩阵 $\boldsymbol{X}_{nk}^{l}(1 \leqslant l \leqslant m)$ 记录每个微基站上边缘服务器的部署情况,矩阵的行数为微基站的数量 n,列数为网络服务类型数量 k,其中矩阵中的元素 $x_{ih}^{l}(1 \leqslant i \leqslant n, 1 \leqslant h \leqslant k)$ 表示第 l 个宏基站上第 i 个微基站是否部署了用户服务类型为 h 的服务。使用矩阵 $\boldsymbol{Y}_{ck}^{l}(1 \leqslant l \leqslant m)$ 记录每个宏基站上边缘服务器的部署情况,其中矩阵元素 $y_{ij}^{l}(1 \leqslant i \leqslant c, 1 \leqslant j \leqslant k)$ 表示属于第 i 个集群的用户服务类型为 j 的服务是否部署在第 l 个宏基站上。参数 x_{ij}^{l} 和 y_{ij}^{l} 都是二进制变量。在最后的边缘服务器部署方案中,矩阵 $\boldsymbol{X}_{nk}^{l}$ 和 $\boldsymbol{Y}_{ck}^{l}$ 都有 m 个,对应于移动网络中 m 个宏基站和连接在这 m 个宏基站的所有微基站的边缘服务器的部署情况。

式(2)中,最小化所有服务处理的网络时延。优化目标中统计了一个宏基站以及连接的所有微基站中部署所有边缘服务器处理网络服务的总时间。使用变量 t_{ih}^{cpu} 表示网络服务类型为 h 的任务在基站中使用 CPU 计算时消耗的时间,用户上传数据和服务请求的时间 t_{ih}^{up},以及向边缘服务器返回处理结果的时间 t_{ih}^{down}。需要对下标 i 进行说明:当边缘服务器部署在宏基站中,下标 i 表示第 i 个集群,其取值范围是 $1 \leqslant i \leqslant c$;当边缘服务器部署在微基站中,下标 i 表示第 i 个微基站,其取值范围是 $1 \leqslant i \leqslant n$。其中,边缘服务器处理用户服务的时间主要是指 CPU 计算和处理数据的时间,这与分配给用户服务的 CPU 资源有密切关系。用户上传数据和服务请求的时间主要指用户将数据传输到边缘服务器的时间,与用户接入后分配的无线信道资源有关。另外如果用户服务不是在接入微基站的边缘服务器本地进行处理,还需要计算该微基站将数据传输到处理该用户请求的边缘服务器所在的微基站或者宏基站的时间;与此同时,也就可以计算边缘服务器处理完成后返回给用户结果的时间。

在完成优化目标的过程中,边缘服务器的部署方案还受到了基站中物理资源的限制。

$$\sum_{h=1}^{k} x_{jh}^{i} \times r_{s_h}^{cpu} \leqslant r_{ij}^{cpu} \quad 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n \tag{3}$$

$$\sum_{h=1}^{k} x_{jh}^{i} \times r_{s_h}^{cache} \leqslant r_{ij}^{cache} \quad 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n \tag{4}$$

$$\sum_{j=1}^{c}\sum_{h=1}^{k} y_{jh}^{i} \times r_{s_h}^{cpu} \leqslant r_{i}^{cpu} \quad 1 \leqslant i \leqslant m \tag{5}$$

$$\sum_{j=1}^{c}\sum_{h=1}^{k} y_{jh}^{i} \times r_{s_h}^{cache} \leqslant r_{i}^{cache} \quad 1 \leqslant i \leqslant m \tag{6}$$

$$x_{ih}^{l} + y_{jh}^{l} \leqslant 1$$
$$1 \leqslant l \leqslant m, 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant c, 1 \leqslant h \leqslant k \tag{7}$$

式(3)~式(7)表示在部署边缘服务器的过程中应该满足的约束条件,是针对一个宏基站及其连接的所有微基站。式(3)和式(4)表示连接在第 i 个宏基站的第 j 个微基站上部署边缘服务器的资源约束,部署在这个微基站上的所有类型的边缘服务所需要的 CPU 和存储资源的总量,不能超过该微基站物理资源的量。同样的,式(5)和式(6)表示的是,部署在第 i 个宏基站上的所有集群中边缘服务器所需要的 CPU 和存储资源的总量,不能超过该宏基站物理资源的总量。式(7)表示在一个集群中,部署相同类型的边缘服务器的个数不超过 1 个。随着车联网技术的发展和用户应用程序种类的增加,处理用户请求需要的资源类型也会变多,而不是仅限于 CPU 和存储,因此可以按照模型中对 CPU 和存储资源的约束条件增加对新资源的约束。

通过分析系统特点,建立数学模型,可以得到一个使用整数线性规划求得最优解的边缘服务器的部署方案。把边缘服务器看作体积大小不同的货物,基站看作体积总量有限的箱子,那么求解边缘服务器部署方案的过程就是求解一个装箱问题,这是已经被证明了的 NP 难问题[15],尤其是问题规模不断增加的时候,求解该问题的最优解将是一个很复杂的过程。在实际求解过程中,通常会考虑以满足条件的局部最优解或者全局近似最优解作为替代方案。

1.3 基于服务的虚拟网络功能放置算法

本文中采用了两种解决方法,分支定界法[16]和启发式算法,计算所有满足约束条件的矩阵 $\boldsymbol{X}_{nk}$ 和矩阵 $\boldsymbol{Y}_{ck}$,并且找出能使式(1)和式(2)取值最小的一组值。使用分支定界法的目的则是找到与整数线性规划最优解相似的近似最优解,使用启发式算法的目的是可以快速地找到符合约束条件的局部最优解。

1.3.1 分支定界法

分支定界法作为一种常用的精细算法,可以找到与全局最优解比较接近的近似解,通常分为分支和定界 2 个部分。

将移动网络划分多个区域,每个区域包括为一个宏基站及其连接的所有微基站,分支定界算法计算每个区域的最小值,全部区域计算完成后就可以得到全局的边缘服务器部署方案。

图 2 是基于分支定界思想算法流程图。需要对流程图作如下说明。

(1)参数 h 表示边缘服务器的服务类型,它的取值范围是 $1 \leqslant h \leqslant k$。

(2)计算边缘服务器在微基站中的部署优化目标时,使用参数 i' 表示优化目标取值为最小值时,对应的微基站的位置。

(3)在每次计算优化目标的值时,使用现有的部分确定取值的参数求解,并且使用线性松弛求得优化目标的最小值。

1.3.2 启发式算法

本文设计的启发式算法是为了快速找到能满足约束条件的一组解,把所有基站按照资源负载从小到大排列,采用遍历满足用户服务资源请求的节点,找到满足目标优化条件的解即可。

```
BEGIN
初始化: Bound = ∞ ; X^l_nk = 0; Y^l_ck = 0;
{
    把所有基站按照资源负载从小到大排列,放入数组 T
    = [t_0, t_1, ⋯, t_n];
    for (1 ≤ h ≤ k)
    {
        取用户服务 s_h 的优先级 p_h;
        if (p_h = 0)&
        (用户服务资源需求小于 Master 剩余资源)
        {
            该服务部署在集群 Master 上;
            对应集群 Master 的微基站是第 i 个微基站,
        x^l_ik = 1, 1 ≤ i ≤ n;
             h = h + 1;
            continue;
        }
        else
        {
            找出剩余物理资源满足服务资源需求的所
        有基站 t_l, 0 ≤ l ≤ n;
            for (0 ≤ l ≤ n)
            {
            计算函数值 bound_l;
            找出最小值 bound_l';
            if (l' = 0)
            {
                y^l_jh = 1, 1 ≤ j ≤ c;
            }
            else
            {
                找出 t_l' 对应的微基站下标 i,1 ≤ i ≤ n;

                x^l_ih = 1;
            }
        }
             h = h + 1;
            continue;
        }
        else
        {return false;}
    }
}
END
```

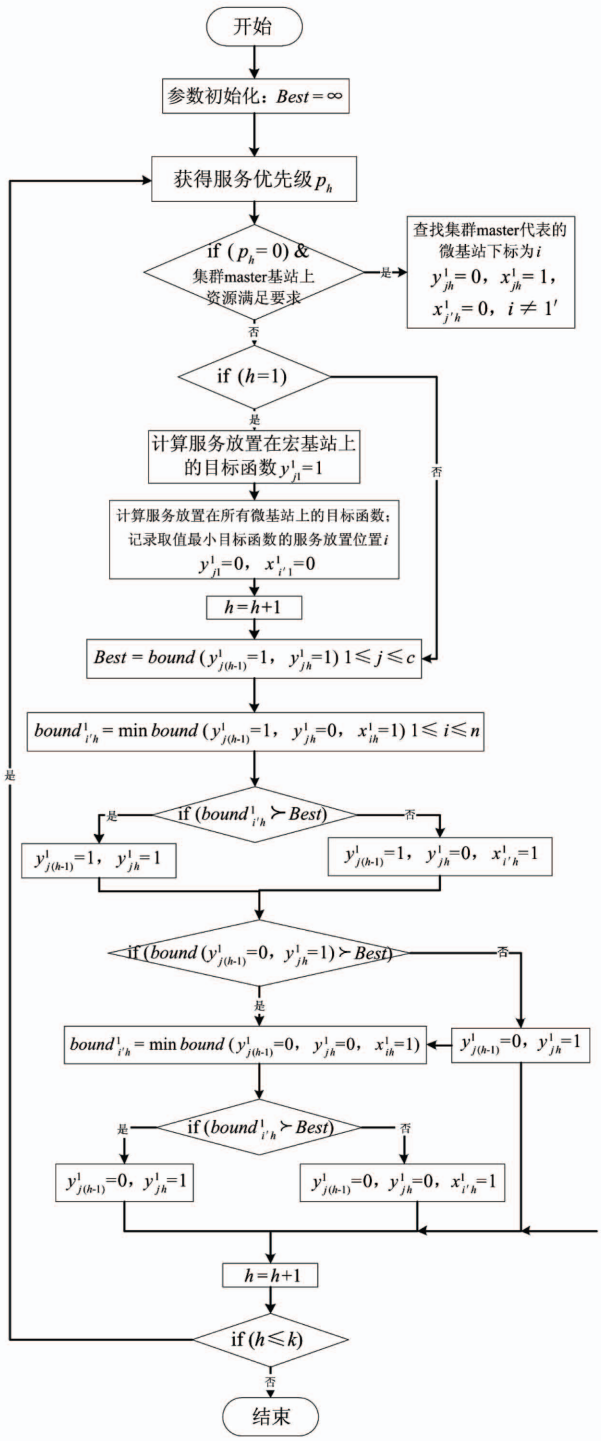

图 2　基于分支定界思想算法流程图

2　性能评估

由于本文存在两个不同计量单位的优化目标，所以优化过程属于多目标优化，并且这两个优化目标之间存在一定的约束关系。为了进一步简化计算过程和减少优化目标，使用标准化的方法对数据进行处理，使用 ε-约束[17]的方法，将多目标优化问题简化为单目标优化问题。在优化过程中，分别以两个优化目标为单目标优化问题的优化目标，将另一个优化目标作为约束条件对优化过程进行限制。

本文使用 Matlab 作为仿真环境，对比整数线性规划、分支定界法和启发式算法这 3 种算法的算法性能，优化目标等参数。

2.1　实验环境与设置

根据文献[2]，城市中车辆密度约为 1000 ~ 3000 辆/km^2；郊区车辆密度为 500 ~ 1000 辆/km^2；高速公路上车辆密度为 100 ~ 500 辆/km^2。微基站信号覆盖范围为 50 ~ 200 m，宏基站信号覆盖范围则可以针对不同的应用场景进行调整。城市环境中车辆密度高，基于安全性和防止阻塞的需求，联网车辆的服务请求对网络响应时间比较敏感，本文将城市环境划分为边长为 500 m 的正方形区域进行分析，这个区域内车辆数大概在 250 ~ 750 辆之间。在这个区域内，部署一个宏基站和 24 个微基站[18]，这些微基站的覆盖范围存在重复，本文中假设车辆会接入距离自己最近的一个基站，区域中车辆密度服从泊松分布。

2.2　实验场景与结果分析

2.2.1　优化边缘服务器部署成本

以边缘服务器部署成本为优化目标，将处理用户服务的网络时延作为约束条件，随着接入到基站集群中用户服务数量的增加，基站上资源负载会变得越来越大，设计基站资源负载率 β，表示集群对新增用户服务的处理能力。β 越小，表示集群可以处理的新增用户服务数越多，处理用户服务的时延就会相应地变长。

当 β 值到达阈值后，说明目前集群中部署的边缘服务器剩余的处理能力不能满足新的用户服务请求，这就需要减少集群半径的数值，将当前集群重新划分为 2 个或者多个新的集群，并在集群中重新部署边缘服务器，以处理更多的用户请求。

本文将仿真区域划分为 3 部分，每部分包含 8 个微基站，作为可以部署边缘服务的物理基站。边缘服务器资源负载率 β 和集群划分半径的算法讨论将在后续的工作中进行。

由图3可以看出,3种部署算法的处理时间变化趋势基本一致,均会随着用户服务请求数量增加而增加,当用户服务请求数量超过600个时,算法的处理时间将有大幅升高,并随着用户服务请求数量的持续增加,算法处理时间的增加幅度不断升高。

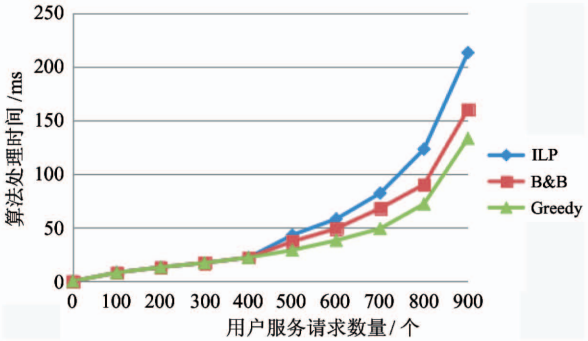

图3 3种部署算法运行时间对比

在3种部署算法处理时间的对比分析中,在用户服务请求数量小于400个时,3种算法的处理时间相差很少,变化曲线基本重合。从用户服务请求数量为500个开始,3种算法的处理时间曲线有了明显差异,其中贪婪算法(greedy)处理时间最短,分支定界算法(B&B)处理时间居中,ILP算法处理时间最长。这种差异一直持续到仿真结束,且处理时间差距逐渐拉大,在用户服务请求数量达到900个后,3种算法的处理时间差距最大。贪婪算法处理时间为133 ms,分支定界算法处理时间为160 ms,而ILP算法的处理时间则达到了213 ms。

综上所述,在用户服务请求数量大于500个后,贪婪算法与分支定界算法的处理时间明显小于线性整数规划,其中贪婪算法处理时间最高可减少37.56%,分支定界算法的处理时间最高可减少24.88%。相比分支定界算法,贪婪算法的处理时间最高也可减少16.88%。

在计算边缘服务器部署成本时,为了便于统计,假设每种边缘服务器在微基站的部署成本相同,部署成本为1,在宏基站的部署成本也一样,部署成本为5。这种设计是为了鼓励将边缘服务器部署在微基站上,尽可能减少处理数据需要的网络时延。

从图4中可以看出,在3种部署算法的对比分析中,用户服务请求数量小于400个时,3种算法的部署成本相差很少,变化曲线基本重合。从用户服务请求数量为500个开始,ILP算法的部署成本曲线较其他两种算法有了较为明显的差异,而贪婪算法和分支定界算法的部署成本在整个仿真过程中均基本处于一致。在用户服务请求数量达到900个后,3种算法的部署成本出现了最大幅度的差异。贪婪算法部署成本为51,分支定界算法部署成本为52,而ILP算法的部署成本则为44。

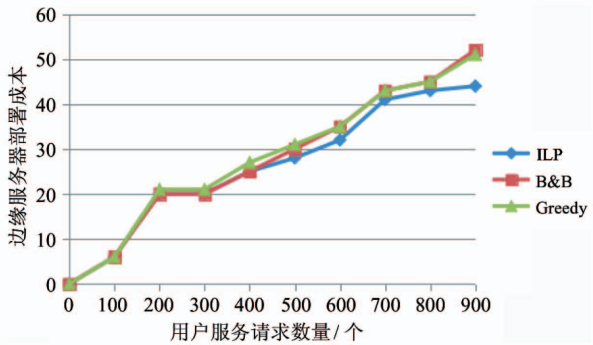

图4 3种部署算法部署成本对比

综上所述,在用户服务请求数量大于500个后,贪婪算法与分支定界算法的部署成本高于线性整数规划,其中贪婪算法部署成本最高增加了15.91%,分支定界算法的部署成本最高增加了18.18%。而分支定界算法和贪婪算法的部署成本基本一致。

2.2.2 用户服务优先级的影响

由于对用户服务的优先级进行了划分,本文对于网络时延敏感的用户服务,优先将其部署在微基站上,而相对不敏感的用户服务既可以部署在集群中其他微服务上,也可以部署在宏基站上。当用户服务请求中对网络时延敏感的请求数量增加时,会将边缘服务器部署位置下拉到车辆接入的微基站,进而也会增加部署边缘服务器的成本。为了更明显地显示出用户优先级的影响,仿真过程中使用的用户服务请求数量为900个。

从图5可以看出,用户服务优先级对算法处理时间的影响变化趋势基本一致,均随着高优先级用户服务比例的增高,算法处理时间持续下降,且算法处理时间的下降幅度基本一致,每种算法的下降曲线呈近似线性。

在3种部署算法用户优先级对处理时间影响的

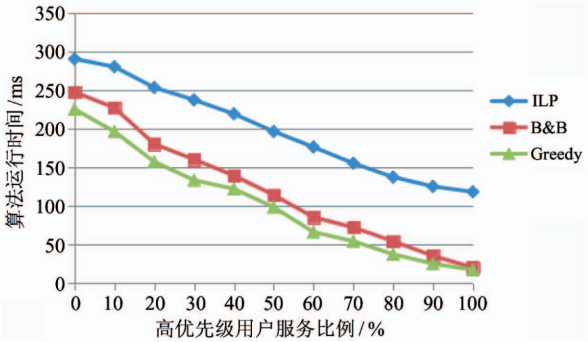

图 5　用户服务优先级对算法运行时间的影响

对比分析方面，在高优先级用户服务比例较小时，贪婪算法的处理时间最短，分支定界算法处理时间居中，ILP 算法处理时间最长。但在高优先级用户服务请求数量达到总用户请求数量的 90% 后，贪婪算法和分支定界算法的处理时间基本一致，与 ILP 算法处理时间的差距达到最大。由于贪婪算法和分支定界算法都是首先考虑了用户服务优先级，所以算法处理时间会比线性整数规划需要的时间短，算法运行时间明显减少。

由图 6 中可以看出，3 种算法在用户服务优先级对部署成本的影响变化趋势基本一致，均随着高优先级用户服务比例的增高，部署成本持续增加，且在高用户优先级比例达到 40% 之后，部署成本的增加值幅度有了明显加大，但 3 种算法的部署成本一直处于近似相同的状态。高优先级用户服务比例达到 100% 时，3 种算法的部署成本分别为 170、175 和 180，差异很小。

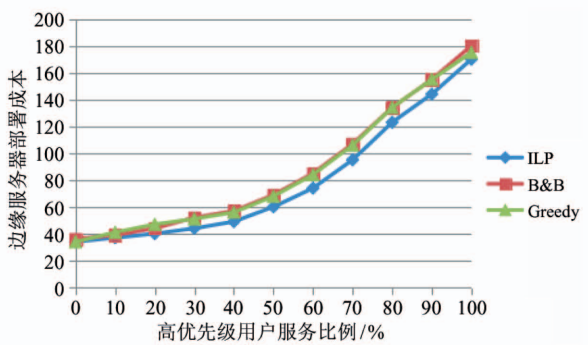

图 6　用户服务优先级对部署成本的影响

3　结　论

本文分析了移动边缘计算体系的结构，对基站进行集群划分，并细化用户服务的等级，在满足移动网络对边缘服务器部署资源需求的前提下，以边缘服务器部署成本和用户服务处理时延作为优化目标，建立 NP 难的整数线性规划模型，对问题进行优化求解；提出分支定界算法和启发式贪婪算法，并对这 3 种算法进行运行时间、部署成本等多个维度的比较。仿真结果显示，分支定界算法和启发式贪婪算法将运行时间减少了 37.6%。后续研究工作将主要集中在如何提高基站资源使用效率以及如何选择微基站集群半径等方面。

参考文献

[1] van der Meulen R, Rivera J. A quarter billion connected vehicles will enable new in-vehicle services and automated driving capabilities[EB/OL]. https://www.gartner.com/en/newsroom/press-releases/2015-01-26-gartner-says-by-2020-a-quarter-billion-connected-vehicles-will-enable-new-in-vehicle-services-and-automated-driving-capabilities :Gartner, 2015

[2] 5G-Infrastructure-Association (5G PPP). 5G automotive vision [EB/OL]. https://5g-ppp.eu/wpcontent/uploads/2014/02/5G-PPP-White-Paper-on-Automotive-VerticalSectors.pdf:5G PPP, 2015

[3] Huawei. Immersive AR and VR experiences with mobile broadband [EB/OL]. http://www.huawei.com/minisite/hwmbbf16/insights/HUAWEI-WHITEPAPER-VR-AR-Final.pdf: Huawei, 2016

[4] Liu Y, Lee M J, Zheng Y. Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system [J]. *IEEE Transactions on Mobile Computing*, 2015, 15(10): 2398-2410

[5] 田辉，范绍帅，吕昕晨，等. 面向 5G 需求的移动边缘计算[J]. 北京邮电大学学报，2017，40(2)：1-10

[6] Lantz B, Heller B, McKeown N. A network in a laptop: rapid prototyping for software-defined networks[C]//Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, Monterey, USA, 2010: 1-6

[7] Guerzoni R. Network functions virtualisation: an introduction, benefits, enablers, challenges and call for action, introductory white paper[C]//SDN and OpenFlow World Congress, Darmstadt, Germany, 2012: 5-7

[8] Lei L, Zhong Z, Zheng K, et al. Challenges on wireless heterogeneous networks for mobile cloud computing[J]. *IEEE Wireless Communications*, 2013, 20(3): 34-44

[9] Zhao T, Zhou S, Guo X, et al. A cooperative scheduling

scheme of local cloud and Internet cloud for delay-aware mobile cloud computing[C]// 2015 IEEE Globecom Workshops (GC Wkshps), San Diego, USA, 2015: 1-6

[10] Ge Y, Zhang Y, Qiu Q, et al. A game theoretic resource allocation for overall energy minimization in mobile cloud computing system[C]// Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design, Redondo Beach, USA, 2012: 279-284

[11] Dinh T Q, Tang J, La Q D, et al. Offloading in mobile edge computing: task allocation and computational frequency scaling[J]. *IEEE Transactions on Communications*, 2017, 65(8): 3571-3584

[12] Ceselli A, Premoli M, Secci S. Mobile edge cloud network design optimization[J]. *IEEE/ACM Transactions on Networking*, 2017, 25(3): 1818-1831

[13] Liang T Y, Li Y J. A location-aware service deployment algorithm based on k-means for cloudlets[J]. *Mobile Information Systems*, 2017: doi: 10.1155/2017/8342859

[14] Jafari A H, López-Pérez D, Song H, et al. Small cell backhaul: challenges and prospective solutions[J]. *EURASIP Journal on Wireless Communications and Networking*, 2015, 2015(1): 206

[15] Khuller S, Raghavachari B, Young N. Designing multi-commodity flow trees[J]. *Information Processing Letters*, 1994, 50(1): 49-55

[16] Karp R M, Zhang Y. Randomized parallel algorithms for backtrack search and branch-and-bound computation[J]. *Journal of the ACM*, 1993, 40(3): 765-789

[17] Haimes Y Y, Lasdon L S, Wismer D A. On a bicriterion formation of the problems of integrated system identification and system optimization[J]. *IEEE Transactions on Systems, Man and Cybernetics*, 1971 (3): 296-297

[18] Chen M, Hao Y. Task offloading for mobile edge computing in software defined ultra-dense network[J]. *IEEE Journal on Selected Areas in Communications*, 2018, 36(3): 587-597

Service-based virtual network function placement algorithm in Internet of vehicles

Han Shujun[*,**], Li Jun[*], Dong Qian[*,**,***], Ma Yuxiang[****], Song Liujing[*,**]

([*]Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190)

([**]University of Chinese Academy of Sciences, Beijing 100049)

([***]School of Electronic Information Engineering, Foshan University, Foshan 528000)

([****]School of Computer and Information Engineering, Henan University, Kaifeng 475004)

Abstract

In order to meet the needs for the collection and transmission of massive data in the Internet of vehicles and the fast processing of these data, mobile edge computing (MEC) technology can be used. This paper considers the characteristics of base station connection methods and physical resources in mobile edge computing, analyzes the deployment of edge servers, optimizes deployment costs and network delays, divides base station clusters, and builds models using integer linear programming (ILP). In order to obtain a more efficient edge server deployment scheme, this paper uses branch and bound algorithm and heuristic greedy algorithm to obtain the approximate optimal solution of the optimization model. The experimental evaluation results show that the branch and bound algorithm and the heuristic greedy algorithm can reduce the running time of the edge server deployment algorithm by up to 37.6%. Moreover, the paper analyzes the impact of the number of user server requests and user service priority on the algorithm running time and edge server running cost.

Key words: Internet of vehicles, mobile edge computing (MEC), network functions virtualization (NFV), software defined network (SDN), quality of service (QoS)