

面向能耗优化和负载均衡的边缘服务器放置研究

付雄, 房磊, 王俊昌

引用本文

付雄, 房磊, 王俊昌. 面向能耗优化和负载均衡的边缘服务器放置研究[J]. 计算机科学, 2023, 50(6A): 220300088-5.

FU Xiong, FANG Lei, WANG Junchang. [Edge Server Placement for Energy Consumption and Load Balancing](#) [J]. Computer Science, 2023, 50(6A): 220300088-5.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于界面相似度的Android仿冒应用检测研究](#)

Study on Android Fake Application Detection Method Based on Interface Similarity
计算机科学, 2023, 50(6A): 220300114-7. <https://doi.org/10.11896/jsjcx.220300114>

[中继选择和队列稳定动态能量优化策略](#)

Dynamic Energy Optimization Strategy Based on Relay Selection and Queue Stability
计算机科学, 2023, 50(6A): 220100082-8. <https://doi.org/10.11896/jsjcx.220100082>

[基于线性规划松弛的移动边缘计算卸载模型](#)

MEC Offloading Model Based on Linear Programming Relaxation
计算机科学, 2023, 50(6A): 211200229-5. <https://doi.org/10.11896/jsjcx.211200229>

[基于“嵩山”超级计算机系统下HHL算法的模拟实现](#)

Simulation Implementation of HHL Algorithm Based on Songshan Supercomputer System
计算机科学, 2023, 50(6): 74-80. <https://doi.org/10.11896/jsjcx.220500108>

[基于多核CPU的无锁并行Semi-naive算法](#)

Lock-free Parallel Semi-naive Algorithm Based on Multi-core CPU
计算机科学, 2023, 50(6): 29-35. <https://doi.org/10.11896/jsjcx.220800050>

面向能耗优化和负载均衡的边缘服务器放置研究

付雄 房磊 王俊昌

南京邮电大学计算机学院 南京 210000

摘要 当前传统的云计算模式已经无法满足用户在低延时场景下的需求,于是移动边缘计算应运而生。为使放置于同一区域内的边缘服务器能够拥有更低的总能耗和均衡的工作负载,提出面向能耗优化和负载均衡的蚁群优化放置算法 ACO-ELP (Ant Colony Optimization Energy-consumption Load-balancing Placement)。首先通过构建功耗模型和负载均衡模型,对问题进行定义并将实际参数与算法变量进行对应匹配。在迭代过程中对蚁群算法进行优化,通过动态控制信息素的挥发和留存速率加快算法的迭代速度,同时控制信息素的最大最小值,以确保算法可以尽可能搜索全局最优解,不会陷入局部最优。最后利用上海地区的电信基站数据对算法进行了仿真评估,结果表明与基础放置算法相比,所提算法在减少服务器数量和能耗的同时显著降低了负载偏差值。

关键词: 移动边缘计算;服务器放置;蚁群算法;能耗优化;负载均衡

中图分类号 TP391

Edge Server Placement for Energy Consumption and Load Balancing

FU Xiong, FANG Lei and WANG Junchang

School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210000, China

Abstract At present, the traditional cloud computing mode can not meet the needs of users in low latency scenarios, so mobile edge computing comes into being. In order to make the edge servers placed in the same area have lower total energy consumption and balanced workload, and an ant colony optimization energy consumption load balancing placement algorithm ACO-ELP (ant colony optimization energy-consumption load-balancing placement) for energy consumption optimization and load balancing is proposed. Firstly, by constructing the power consumption model and load balancing model, the problem is defined, and the actual parameters are matched with the algorithm variables. In the iterative process, the ant colony algorithm is optimized. By dynamically controlling the volatilization and retention rate of pheromone, the iterative speed of the algorithm is accelerated, and the maximum and minimum value of pheromone is controlled to ensure that the algorithm can search the global optimal solution as much as possible and will not fall into the local optimal solution. Finally, the algorithm is simulated and evaluated with the data of Telecom base stations in Shanghai. The results show that compared with the basic placement algorithm, the algorithm not only reduces the number of servers and energy consumption, but also significantly reduces the load deviation.

Keywords Edge computing, Server placement, Ant colony algorithm, Energy consumption, Load balancing

1 引言

随着增强/虚拟现实(AR/VR)以及自动驾驶等技术的不断发展,移动用户对于实时计算的需求不断增加。而传统移动云计算网络需要移动用户通过基站将请求卸载到远程云服务器并代替其执行任务^[1-2],但这种方式必然带来较长的延时。移动边缘计算^[3-4]作为一种在需要低延时的场景下的解决方案为众人所知晓。

近几年,移动边缘计算已经受到人们的广泛关注,其中边缘服务器放置(Edge Server Placement, ESP)问题是首先要解决的问题。通常边缘服务器以分布式方式放置^[5-6],以便它们能够覆盖不同的地理区域。在解决 ESP 问题时,各研究的目标不同,例如 Li 等和 Cao 等^[7-8]最小化移动用户的访问延时,

Cui 等^[9]讨论了用户覆盖和网络健壮性的问题,证明了求解该问题的最优解是 NP-hard 的,并提出了一套基于整数规划的优化方法来解决该问题。在移动边缘计算的卸载^[10-11]的研究中, Tao 等^[12]研究了移动边缘计算在多信道无线干扰环境下的多用户计算卸载问题。他们将移动设备用户之间的分布式计算卸载决策问题表述为一个多用户计算卸载博弈,设计了一种实现纳什均衡的分布式计算卸载算法。

关于负载均衡, Kyryk 等^[13]研究了 CDN 体系结构和全球概念内容分发网络(Content Delivery Network, CDN)系统的主要原理,并研究了边缘计算网络中负载均衡问题的求解算法。Li 等^[14]深入研究了有关能量感知的边缘服务器放置问题,将问题描述为多目标优化问题,并系统设计了一个基于粒子群优化的能量感知边缘服务器布局算法来寻找最优解。

基金项目:国家自然科学基金(51977113);江苏省重点研发计划(社会发展)项目(BE2017743)

This work was supported by the National Natural Science Foundation of China(51977113) and Primary Research & Development Plan (Social Development) of Jiangsu Province(BE2017743).

通信作者:付雄(fux@njupt.edu.cn)

结果表明,采用该算法可以在保证访问时延一定的前提下,降低边缘服务器的总能耗。

本文提出了一个基于蚁群优化算法的最小化功耗和最优资源匹配度放置策略(ACO-ELP)。首先建立了移动边缘计算的系统模型,对比其他研究工作,本文分别对能耗和负载进行分析建模,再通过归一化方法对提出的模型进行处理并将其表述为多目标优化问题,从而在一定范围内选出最适合放入这个集群的基站,达到最小化边缘服务器能耗和数量的目的。最后使用蚁群优化算法(Ant Colony Optimization, ACO)对定义的问题进行求解。

2 系统模型和问题表述

2.1 系统模型

如图1所示,在移动边缘计算中一般将网络分为3层架构,分别为数据中心层、基站服务器层和移动设备层。本文的研究重点在基站服务器层,它主要由 n 台边缘服务器和 m 个基站组成。定义边缘计算服务器集合 $S = \{s_1, \dots, s_i, \dots, s_n\}$,基站集合 $B = \{b_1, \dots, b_j, \dots, b_m\}$,其中为了控制部署成本,基站数量 m 一般由该区域用户密集程度决定,同时边缘服务器数量 $n \leq m$ 。

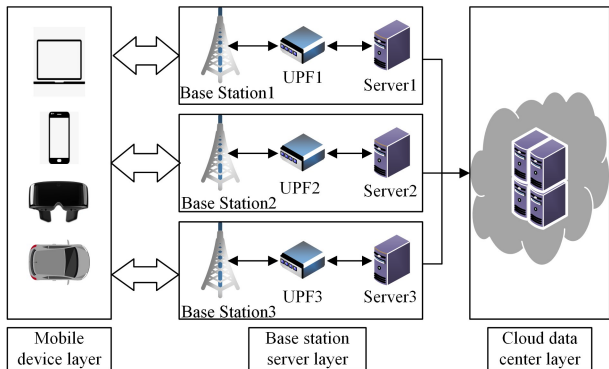


图1 移动边缘计算架构图

Fig. 1 Diagram of Mobile edge computing architecture

由于服务器的延时与服务器和基站距离成正比关系,随着延时的降低成本会迅速增加^[15],且本文的优化重点不在于服务器的延时情况,所以假设边缘计算服务器在覆盖范围 R 内的访问延时均为可接受的,定义服务器 i 与基站 j 的欧氏距离为:

$$D(s_i, b_j) = \sqrt{(l_{i_x} - l_{j_x})^2 + (d_{i_y} - d_{j_y})^2} \quad (1)$$

其中, $l_{i_x}, d_{i_y}, l_{j_x}, d_{j_y}$ 分别表示服务器 i 和基站 j 的经纬度经高斯投影坐标正算后得到的纵横坐标。

2.2 能耗模型

在边缘计算服务器对外提供服务时,有许多因素影响着服务器的能耗,如CPU、内存、硬盘等。在这些因素中CPU是最主要的能耗设备^[16],表示可以通过CPU利用率来线性地表示出服务器的功耗。因此本文通过CPU利用率来表示服务器资源的利用率,从而得到服务器的功耗。

当边缘计算服务器处于空闲状态时,依然会有基本能耗,约占服务器满负荷运行能耗的60%^[17],因此如果有服务器处于低利用率甚至空闲状态时会浪费大量能量。为了降低边缘计算中能量的损耗,提高边缘服务器的利用率,本文对边缘服务器的布局方案进行优化,提出的能耗模型如下:

$$E_i = \min \int_{t_1}^{t_2} P_i(t) dt \quad (2)$$

$$P_i(t) = P_{idle} + (P_{max} - P_{idle}) \times \omega_i / \omega_{max} \quad (3)$$

其中, E_i 表示服务器 i 的功耗, P_i 表示服务器 i 的功率, P_{idle} 表示服务器在空闲状态下工作时的功率, P_{max} 表示服务器处于满负荷工作时的功率, ω_{max} 表示边缘服务器CPU工作负载的上限, ω_i 表示服务器 i 的负载。

$$\omega_i = \sum_{j \in B} t_j * \alpha_{ij} \quad (4)$$

其中, t_j 表示基站 j 的负载, α_{ij} 为一个二元变量,其含义为:

$$\alpha_{ij} = \begin{cases} 1, & \text{if } i \text{ choose } j \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

那么所有服务器的平均功耗可表示为:

$$E_{total} = \sum_{i=1}^n E_i \quad (6)$$

2.3 负载均衡模型

在边缘移动计算中,常用 $T(l)$ 表示边缘服务器之间的工作负载均衡,其负载均衡模型可以表示为:

$$T(l) = \max \omega_i - \min \omega_j, (i, j \in S) \quad (7)$$

要使 $T(l)$ 最小即要使整个系统中所有边缘服务器的最大值和最小值之差最小。首先,找到基于所有基站的每个边缘服务器的平均工作负载 \bar{W}_i 。

$$\bar{W}_i = \frac{\sum_{j \in B} t_j}{Num_S} \quad (8)$$

其中, Num_S 表示边缘服务器数量。最后,计算平方差之和,即负载偏差值 W_i :

$$W_i = (\sum_{j \in B} t_j * \alpha_{ij} - \bar{W}_i)^2 \quad (9)$$

2.4 问题表述

根据上述模型,最小化功耗和均衡工作负载的边缘服务器放置问题可以表述为:

$$\min \mu E_{total} + (1 - \mu) W_i \quad (10)$$

$$\text{s. t. } D(s_i, b_j) \leq R \quad (11)$$

$$\omega_i \geq 0 \quad (12)$$

其中, μ 为权重参数,本文中 μ 值取0.5。

3 基于蚁群算法的边缘服务器放置算法

3.1 编码方案

为了更加明确基站与服务器之间的对应关系,本文采用二维矩阵的方式进行编码。如式(13)所示矩阵为 $n \times m$:

$$H = \begin{bmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{n1} & \cdots & h_{nm} \end{bmatrix} \quad (13)$$

其中, n 表示边缘服务器数,即集群数; m 表示基站总数;初始环境下 H 为空,处于未赋值状态。 $h_{ij} = C (C \neq 0, 1 \leq C \leq n)$ 表示边缘服务器 i 放置于基站 C 上且基站 j 属于边缘服务器 i 所在集群($i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\}$)。

通过上述表述可以看出矩阵 H 有以下特征:一行至少有一个位置的值不为0;一行不为0的值都相同;各行不为0的值均不相同;每列有且仅有一个位置的值不为0。例如: $b_{23} = 4$ 表示3号基站在集群2中,集群2中边缘服务器放置于基站4。

通过这种编码方式可以更容易发现约束之间的冲突,也能更加明确基站与服务器之间的对应关系,通过3个值即可很清楚地了解服务器集群和基站的多种信息。

3.2 信息素

信息素的更新是蚁群算法的核心,蚂蚁通过信息素间接地进行交流。当一只蚂蚁完成了解决方案的寻找,问题空间

中所有路径上的信息素都会蒸发一些,然后在蚂蚁选择走过的路径上释放一些信息素。信息素更新公式为:

$$\tau_{ij} = (1 - \rho) \times \tau_{ij} + \rho \times \Delta\tau_{ij} \quad (14)$$

其中, $\Delta\tau_{ij}$ 表示当 s_i 选择了 b_j 后,在更新信息素时信息素的增加量。定义为式(10)的倒数即:

$$\Delta\tau_{ij} = \begin{cases} \frac{1}{\mu E_{\text{total}} + (1 - \mu) \omega_i}, & \text{if } i \text{ choose } j \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

确保了当负载均衡且能耗越小时, $\Delta\tau_{ij}$ 的取值越大。

3.3 概率传递函数

$$P_{ij}^k(t) = \begin{cases} \frac{[\gamma_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum [\gamma_{ik}(t)]^\alpha \cdot [\eta_{ik}]^\beta}, & \text{if } D(s_i, b_j) \leq R \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

$$\eta_{ij} = \frac{1}{E_{\text{total}}} \cdot \frac{1}{\omega_i} \quad (17)$$

为了防止只能得到局部最优解我们需动态调整 α 和 β 的值。在一般的 ACO 中 α 和 β 的值是固定不变的,而本文中通过动态调整使 α 和 β 随迭代次数的变化而变化。在算法搜索初期,以资源匹配度和服务器与基站欧氏距离为依据来引导算法的搜索;在搜索后期,调整为以信息素浓度进行搜索。这样可以保证解的最优性和全局性。 α 和 β 的迭代关系如下:

$$\alpha = \begin{cases} \alpha_1 \in (0.1, 0.5), & 0 < G \leq G_1 \\ \alpha_2 \in (0.5, 0.9), & G_1 < G \leq G_{\text{max}} \end{cases} \quad (18)$$

$$\beta = \begin{cases} \beta_1 \in (0.5, 0.9), & 0 < G \leq G_1 \\ \beta_2 \in (0.1, 0.5), & G_1 < G \leq G_{\text{max}} \end{cases} \quad (19)$$

本文中 $G_1 = \frac{1}{2} G_{\text{max}}$ 。

3.4 最大最小蚂蚁系统

由于一次循环后信息素只更新到最优解,容易造成搜索停滞,导致蚂蚁总是选择同一条路径,因信息素积累过高从而陷入局部最优。为了避免这种情况发生,本文在算法中融合了最大最小蚂蚁系统的思想。

各条路径上的 $\tau \in [\tau_{\min}, \tau_{\max}]$, 当 $\tau > \tau_{\max}$ 则令 $\tau = \tau_{\max}$; 当 $\tau < \tau_{\min}$ 则令 $\tau = \tau_{\min}$ 。

在每次迭代后,只更新最优解对应的路径上的信息素浓度。

本文算法将初始时刻的信息素浓度设置为 τ_0 , 上限 $\tau_{\max} = 1.5 \tau_0$, 下限 $\tau_{\min} = 0.5 \tau_0$, 即:

$$\tau = \begin{cases} \tau_0 \\ \tau_{\max} = 1.5 \tau_0 \\ \tau_{\min} = 0.5 \tau_0 \end{cases} \quad (20)$$

3.5 基于能耗和负载均衡的边缘服务器放置算法

基于蚁群算法的边缘服务器放置算法的基本思想是:用欧氏距离表示每个边缘服务器和基站的延时,在通信延时可以接受的情况下让蚁群随机分布到服务器上并选择与之相连接的基站,通过式(6)及式(9)计算出该次迭代的最终值,通过式(14)更新各条连接之间的信息素值并确保信息素的值在最大和最小值之间,最后得到该次迭代的值并保留结果集合。至迭代完成后找出最优结果并返回结果集合。伪代码如算法 1 所示。

算法 1 基于蚁群算法的边缘服务器放置算法(ACO-ELP)

输入: data set of base stations; site. csv; base station set B; edge server set S; edge server coverage R; number of ants K; algorithm itera-

tion number T

输出: the global optimal solution set PL

1. For $T=1, 2, \dots, 200$ do
2. For each edge do
3. Set initial pheromone τ_0
4. End for
5. For $k=1, 2, \dots, K$ do
6. update the encoding matrix H of dimension $[n \times m]$
7. randomly put the ant on the server node i
8. While the size of $B \neq \text{null}$
9. choose an unassigned base station j
10. If i meets the placement conditions
11. load j into H[i] and delete from B
12. update the load balancing utilization of i
13. update the energy consumption of i
14. Else
15. H[i][j] the first j-value($j \in H[i]$)
16. End if
17. put the ant randomly on the next server node that has not been selected
18. End while
19. update the value of pheromone τ and make $\tau \in [\tau_{\min}, \tau_{\max}]$
20. End for
21. End for
22. compare all of the solution sets and return the optimal solution set PL.

该算法的时间复杂度为 $O(Tmn)$, 其中 T 为最大迭代次数, n 为边缘服务器的数量, m 为基站的数量。通过对边缘服务器位置不断搜索并更新比较边缘服务器的运行功耗和工作负载,最终得到最低的边缘服务器运行功耗和工作负载的偏差值组合。

4 实验结果及分析

本节将使用真实的数据集来评估基于最小化功耗和负载均衡的边缘服务器放置算法的性能,并将其与多种算法进行比较。实验结果表明,本文提出的算法在降低边缘服务器总能耗和平衡负载方面表现突出。

4.1 数据集处理

本文实验所用的数据集为上海电信基站数据集,其中包含百万条由位置和用户上传日志等组成的数据。如果一个用户访问了互联网,我们可以了解它何时开始、何时结束以及连接到哪个基站。

我们在数据集中选择上海地区的基站数据集^[18]为本次实验的数据,因为上海是一个人口密集的城市,可以很好地体现出算法的优劣势。将数据集进行预处理,保留 3000 个有效基站数据,并剔除与本次实验无关的参数,只保留用户上传日志、基站的经纬度、基站负载等参数。

4.2 实验准备

本文设计了两组实验:1)保持边缘计算器的覆盖范围改变基站的数量;2)保持基站数量改变覆盖范围。从而观察基站的总能耗和负载偏差值。

本次实验选择 HP ProLiant ML350 作为边缘服务器。配置为 Xeon E5506 CPU 主频为 2.13 GHz 和 32 GB 内存,满载功率为 460 W,根据 Dayarathna^[17] 的调查结果将服务器

空闲状态的功率设为满负载功率的 60%。所有的实验都是在同一台计算机上进行的,其使用 Inter Core (TM) i7-7700HQ CPU 2.80GHz 处理器和 16 GB 的 RAM,操作系统为 64 位 windows 10 professional。仿真平台使用的是 EdgeCloudSim^[19],这是一个在 CloudSim 基础上增添了边缘计算场景的优秀仿真平台。

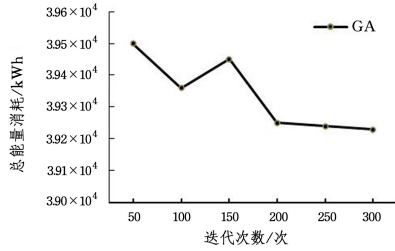


图2 总能量消耗随迭代次数变化图

Fig. 2 Diagram of total energy consumption varies with iteration times

如图 2 所示,实验中设置基站数为 1500,蚁群数量为 100,以确定最优迭代次数 Q 。由实验可得迭代 200 次后迭代次数对总能量的影响很小,所以本实验选择的最优迭代次数为 200 次。

图 3 给出了 4 种算法下相对于边缘服务器覆盖半径,用于边缘服务器放置的平均边缘服务器数。

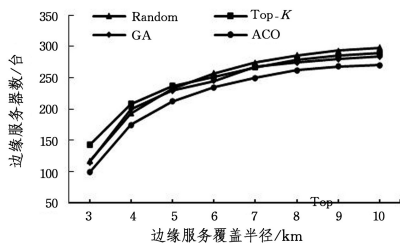


图3 边缘服务覆盖半径与服务器数量对应图

Fig. 3 Diagram of edge service coverage varies with the number of servers

由图 3 可知,ACO 算法可以更有效地分配资源,使得资源匹配度更高,从而有效地减少边缘服务器的数量,表明了 ACO 算法在节能方面表现优秀。后续实验中的边缘服务器数量参照此对应图设置,并根据基站数量同比增加或减少。

4.3 实验结果

4.3.1 对比算法和评价指标

本文从总能量消耗和负载偏差值两个方向通过与其他放置算法比较来评价所提算法性能。其他放置算法如下:

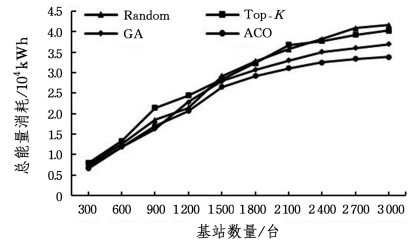
(1)随机算法:该算法通过随机选择基站来放置边缘服务器,并将满足约束条件的基站分配给边缘服务器。

(2)Top-K 算法:该算法是将边缘服务器放置在前 K 个工作负载最大的基站区域内,并为其分配满足约束条件的基站。

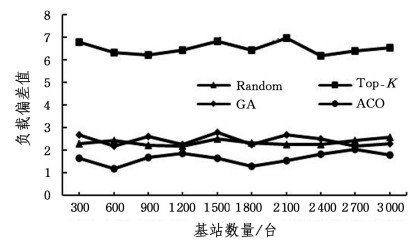
(3)遗传算法:该算法是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型,是一种通过模拟自然进化过程搜索最优解的方法。其主要特点是直接对结构对象进行操作,不存在求导和函数连续性的限定;具有内在的隐并行性和更好的全局寻优能力;采用概率化的寻优方法,不需要确定的规则就能自动获取和指导优化的搜索空间,自适应地调整搜索方向。

4.3.2 结果对比

本文在上海电信基站数据集上评估不同方法的性能。本次实验中将基站数量 m 从 300 逐步增加到 3000,边缘服务器的覆盖半径为 5 km,蚁群数量为 100,最大迭代次数为 200。通过实验给出了 4 种算法在不同基站数量、不同边缘计算服务覆盖范围和不同用户请求次数下的总能量消耗以及负载偏差值的对比。



(a) 总能量消耗随基站数量变化图

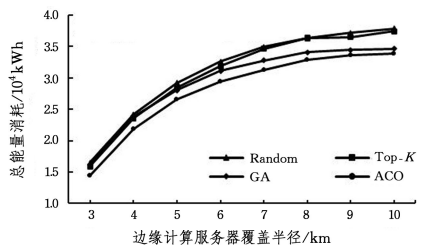


(b) 负载偏差值随基站数量变化图

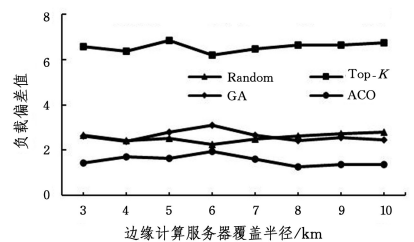
图4 总能量消耗和负载偏差值随基站数量变化图

Fig. 4 Diagram of total energy consumption and load deviation varies with the number of base stations

通过图 4(a)可以看出当基站数量不足 1500 时,基站数量不足以服务整个区域,每个服务器几乎都是满功耗运行,不能体现算法的优劣势。当基站数超过 1500 之后 ACO 算法产生的总能耗明显低于其他两种基础算法。在图 4(b)中,负载偏差值基本没有受到基站数量的影响,其中 Top-K 算法的负载偏差值最大,这是由于其算法特性将服务器按区域的工作负载放置,导致任务集中在一部分服务器上导致的。因此我们将基站数量设置为 1500,结果如图 5 所示。



(a) 总能量消耗随服务器覆盖半径变化图



(b) 负载偏差值随服务器覆盖半径变化图

图5 总能量消耗和负载偏差值随用户请求次数变化图

Fig. 5 Diagram of total energy consumption and load deviation value varies with server coverage radius

图 5(a)说明 ACO 算法在巨量用户请求时对总能量的消耗也有一定的减少作用。图 5(b)中,随着用户请求次数的增加 Top-K 算法的负载偏差值有所增大,ACO 算法由于从整体上平衡负载所以偏差值反而有所下降。

结束语 本文研究了在移动边缘计算场景下,如何放置边缘服务器以达到最小化功耗和优化负载均衡的问题。首先提出了移动边缘计算的系统模型,通过归一化方法对提出的能耗模型和负载均衡进行处理并将其表述为多目标优化问题。然后提出了基于蚁群的优化算法,并在上海电信的基站数据集上验证了本文算法在减少服务器数量、降低能量消耗和提高负载均衡等方面有着较好的效果。

后续研究将考虑在真实的移动边缘计算环境中进行验证,进一步验证相关仿真的结果,并优化相关调度的改进。

参 考 文 献

- [1] LIU M Y, TU Q N, WANG Y, et al. Research Status of Mobile Cloud Computing Unloading Technology and Application in Power Grid[J]. *Electric Power Information and Communication Technology*, 2021, 19(1): 49-56.
- [2] DINH H, LEE C, NIVATO D, et al. A survey of mobile cloud computing: Architecture, applications, and approaches[J]. *Wireless Commun. Mobile Comput.*, 2013, 13(18): 1587-1611.
- [3] YU Y F, REN C M, RUAN L F, et al. Analysis on the Development of Mobile Edge Computing Technology[J]. *Telecommunications Network Technology*, 2016(11): 59-62.
- [4] MACH P, BECVAR Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading[J]. *IEEE Communications Surveys & Tutorials*, 2017, 19(3): 1628-1656.
- [5] GUO F Y, TANG B. Mobile Edge Server Placement Method Based on User Delay Perception[J]. *Computer Science*, 2021, 48(1): 103-110.
- [6] WANG S, ZHAO Y, XU J, et al. Edge server placement in mobile edge computing[J]. *Journal of Parallel and Distributed Computing*, 2018, 127(1): 160-168.
- [7] LI H, DONG M, OTA K, et al. Pricing and repurchasing for big data processing in multi-clouds[J]. *IEEE Transactions on Emerging Topics in Computing*, 2016, 4(2): 266-277.
- [8] CAO K, LI L, CUI Y, 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.
- [9] CUI G, HE Q, CHEN F, et al. Trading off between User Coverage and Network Robustness for Edge Server Placement, [J]. *IEEE Transactions on Cloud Computing*, 2020, 10(3): 2178-2189.
- [10] HU H, JIN F L, LANG S Q. Overview of Computing Offload Technology in Mobile Edge Computing Environment[J]. *Computer Engineering and Applications*, 2021, 57(14): 15.
- [11] XIANG H, XU X, ZHENG H, et al. An adaptive cloudlet placement method for mobile applications over GPS big data[C]// 2016 IEEE Global Communications Conference(GLOBECOM). IEEE, 2016: 1-6.
- [12] TAO M, OTA K, DONG M, Foud. Integrating fog and cloud for 5G-enabled V2G networks[J]. *IEEE Network*, 2017, 31(2): 8-13.
- [13] KYRYK M, PLESKANKA N, PLESKANKA M, et al. Load Balancing Method in Edge Computing[C]// 2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering(TCSET). IEEE, 2020.
- [14] LI Y, WANG S. An Energy-Aware Edge Server Placement Algorithm in Mobile Edge Computing[C]// 2018 IEEE International Conference on Edge Computing(EDGE). San Francisco (US); IEEE, 2018: 66-73.
- [15] FIEDLER M, HOSSFELD T, TRAN-GIA P. A generic quantitative relationship between quality of experience and quality of service[J]. *IEEE Network*, 2010, 24(2): 36-41.
- [16] GUPTA V, NATHUJI R, SCHWAN K. An analysis of power reduction in datacenters using heterogeneous chip multiprocessors [J]. *ACM Sigmetrics Performance Evaluation Review*, 2013, 39(3): 87-91.
- [17] DAYARATHNA M, WEN Y G, FAN R, et al. Data Center Energy Consumption Modeling: A Survey[J]. *IEEE Communications Surveys and Tutorials*, 2016, 18(1): 732-794.
- [18] WANG S, ZHAO Y, HUANG L, et al. QoS prediction for service recommendations in mobile edge computing[J]. *Journal of Parallel and Distributed Computing*, 2019, 127: 134-144.
- [19] SONMEZ C, OZGOVDE A, ERSOY C. EdgeCloudSim: An environment for performance evaluation of Edge Computing systems [C]// The 2nd International Conference on Fog and Mobile Edge Computing(FMEC 2017). IEEE, 2017: 39-44.



FU Xiong, born in 1979, professor. His main research interests include parallel and distributed computing, and cloud computing.