计算机科学
COMPUTER SCIENCE

一种基于延迟与负载的最优边缘服务器放置方法

袁培燕, 马伊雯

引用本文

袁培燕, 马伊雯. 一种基于延迟与负载的最优边缘服务器放置方法[J]. 计算机科学, 2023, 50(11A): 220900260-8.

YUAN Peiyan, MA Yiwen. Optimal Edge Server Placement Method Based on Delay and Load[J]. Computer Science, 2023, 50(11A): 220900260-8.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220900260

一种基于延迟与负载的最优边缘服务器放置方法

袁培燕[1,2]　马伊雯[1]
1 河南师范大学计算机与信息工程学院　河南 新乡 453007
2 智慧商务与物联网技术河南省工程实验室　河南 新乡 453007
(peiyan@htu.cn)

摘　要　当前边缘服务器放置问题已成为边缘计算发展的关键环节。现有边缘服务器放置方法结合放置成本、网络延迟与系统能耗等指标进行优化,但大多数工作忽略了边缘服务器之间的负载均衡。文中以最小化边缘服务器服务延迟与负载均衡为优化目标,建立边缘服务器放置优化模型,根据该优化模型选择最佳放置位置,并提出了一种基于改进的元启发式算法的边缘服务器放置方案MIWOA-ESP,完成模型中多目标优化并确定基站到边缘服务器映射关系,给出最优放置与分配方案。最后,使用上海电信基站数据集进行性能分析。实验结果表明,与其他基准方案相比,所提MIWOA-ESP放置策略在网络延迟和服务器负载均衡方面具有更好性能。

关键词:边缘计算;服务器放置;延迟感知;负载均衡
中图法分类号　TP302

Optimal Edge Server Placement Method Based on Delay and Load

YUAN Peiyan[1,2] and MA Yiwen[1]
1 School of Computer and Information Engineering,Henan Normal University,Xinxiang,Henan 453007,China
2 Engineering Lab of Intelligence Business & Internet of Things,Xinxiang,Henan,453007,China

Abstract　At present,the placement of edge servers has become a key step in the development of edge computing. Existing edge server placement methods are optimized by combining placement cost,network latency and system energy consumption,but most work ignores load balancing among edge servers. The goal of this paper is to minimize the service delay and load balancing of edge servers,and an optimization model of edge computing server placement is established. According to the optimization model,the optimal placement location is selected,and an edge server placement scheme based on an improved meta-heuristic algorithm,MIWOA-ESP,is proposed. It completes the multi-objective optimization and determines the distribution relationship between the base station and the edge server,and gives the optimal placement and distribution scheme. Finally,experiments are carried out using the Shanghai Telecom base station dataset. The results show that compared with other benchmark schemes,the MIWOA-ESP placement strategy has better performance in terms of network latency and server load balancing.

Keywords　Edge computing,Server placement,Latency awareness,Load balancing

1　引言

互联网技术的快速发展和智能终端设备的普及在网络边缘产生了大量数据。传统的云计算解决方案需要将这些数据传输到远程云端进行处理[1],会给整个网络带来较大时延。与此相反,边缘计算通过将计算资源从核心网络下沉到网络边缘[2],将边缘服务器(Edge Server,ES)部署在更靠近用户的位置,用户将任务卸载到服务器执行,从而减少任务执行及往返时间,有效减少了网络拥塞和数据传输延迟。

如图1所示,用户设备通过无线信道与基站连接,基站之间利用光纤进行数据传输,通过将边缘服务器部署在不同的物理位置来满足不同业务的服务质量需求。进一步来说,边缘服务器部署问题需要在满足用户低时延、高带宽的网络需求的同时,综合考虑移动用户、边缘计算环境资源与地理范围约束,设计一种满足网络延迟与边缘服务器负载均衡目标的放置方案。

确定边缘服务器放置方案时需要考虑多方面约束,其中边缘服务器延迟与负载为放置方案的核心指标。一方面,在边缘服务器中运行的服务(如VR/AR和车辆互联网)通常为延迟敏感型和计算密集型服务,因此边缘服务器的放置位置对于移动用户的访问延迟至关重要。移动用户在通过基站访问边缘服务器时,若当前放置方案效率低下,则会导致较长的访问延迟。高效的边缘服务器放置方案应能有效改善边缘环境下各种移动应用程序的访问延迟。另一方面,放置方案中边缘服务器之间的负载均衡也是一个无法回避的问题。边缘服务器的放置位置对边缘服务器的资源利用率会产生较大

基金项目:国家自然科学基金(62072159,U1804164,61902112);河南省教育厅科学与技术基金(19A510015,20A520019,20A520020)
This work was supported by the National Natural Science Foundation of China(62072159,U1804164,61902112) and Science and Technology Foundation of Henan Educational Committee(19A510015,20A520019,20A520020).
通信作者:袁培燕(peiyan@htu.cn)

影响，若边缘计算服务器之间负载差异过大，则会出现一些边缘服务器负载过量而另一些边缘服务器资源利用率不足的问题，导致资源浪费。因此移动边缘计算中的边缘服务器部署需要同时考虑网络延迟和边缘服务器的负载平衡，在不影响用户服务质量的情况下，以尽可能低的成本部署边缘服务器。

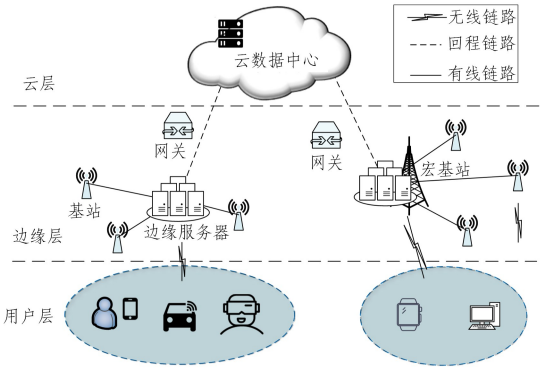


图 1 边缘服务器部署系统模型

Fig. 1 Edge server deployment system model

本文第 2 章介绍边缘服务器与云服务器放置的相关研究；第 3 章建立边缘服务器放置模型；第 4 章详细介绍本文提出的 MIWOA-ESP 算法；第 5 章利用真实数据集与基准方案进行性能分析。

2 相关工作

目前，已有学者基于上述问题开展相关研究。其中，部分研究者假设所有基站呈相互连接的状态，并将基站之间的距离作为衡量延迟参数，将延迟作为优化目标来确定边缘服务器放置位置[3]。考虑到在真实环境中，网络延迟并不完全取决于传输距离，文献[4]对移动边缘计算中的 5G 环境进行了建模分析，以最小化时延开销和能量开销为目的，提出了基于等效带宽的 ES 放置方法。

文献[5]指出，放置方案对资源效率有很大的影响。Santoyo-Gonzalez 等[5]设计了一个影响效率的参数列表，以评估边缘服务器在 5G 方案中的放置性能。文献[6]讨论了容量有限的 ES 放置问题，提出了 PACK 算法，最小化 ES 与用户之间的时延。文献[7]以优化商业指标为目标，提出了一种基于资源需求预测的 ES 放置算法。文献[8]利用社交网络信息放置边缘服务器。文献[9]考虑分布式边缘计算环境下的网络鲁棒性，提出了一种基于整数规划的方法，用以放置边缘服务器并改善用户的体验。文献[10]通过优化边缘服务器的放置来节省系统能源消耗，采用贪婪算法来优化边缘服务器放置的成本。文献[4,11]研究了边缘服务器的能耗，通过 CPU 利用率和边缘服务器的功率之间的线性关系计算边缘服务器的能耗，并利用粒子群优化算法最小化边缘服务器的能源消耗。

文献[12]提出一种基于 k 均值的边缘服务器部署算法，该算法基于部署预算和计算需求统计数据，确定边缘节点的数量以及不同类型的边缘服务器的最佳数量。但该放置方案中未全面考虑系统延迟的影响。文献[13]提出了一种新型的元启发式算法来部署边缘计算服务器，该方法将搜索空间划分为一定数量的子空间，并评估每个子空间的计算潜力，进而确定在收敛过程中是增加还是减少子空间的计算资源。文献[14]通过改进的 Top-K 算法综合考虑了基站与边缘服务器的距离、基站集群中基站的权重比、边缘服务器的覆盖范围、计算任务的上限等因素，旨在降低任务的访问时延和边缘服务器的部署成本，平衡边缘服务器之间的负载，提高用户体验质量(QoE)和服务质量。文献[15]提出一种在超密集网络中部署边缘服务器的最优部署和分配策略，基于排队理论和矢量量化技术，优化边缘服务器的数量和位置，确定移动用户分配，最大限度地降低服务商的成本，保证服务的完成时间。文献[16]提出了一个涉及访问延迟和能源消耗的利润模型，并基于此模型设计了一种基于粒子群优化(PSO)的算法来优化运营商利润。文献[17]将 PSO 与遗传算法结合使用，最大程度地减少用户设备和边缘服务器的总能源消耗。针对 PSO 覆盖搜索空间较小、无法解决高维数或复杂目标函数的问题，文献[18]提出将 PSO 算法与鲸鱼优化算法(WOA)中对数螺旋更新机制相结合，确定智能家居和智能环境中优化的运动传感器位置，提高了运动传感器的覆盖率、检测精度和运行成本。文献[19]利用 WOA 算法执行对电池储能系统(BESS)放置的优化，降低了配电网中的功率损耗。文献[20]使用 WOA 优化光纤无线(FiWi)中光网络单元的放置位置，缩短了光网络单元与其相关无线路由器之间的平均通信距离，为部署高能效 FiWi 网络提供了最佳途径。

尽管大多数边缘服务器放置工作都将访问延迟作为优化目标或目标函数约束，但大部分研究都忽略了边缘服务器负载差异对访问延迟与系统能耗的影响。本文中将边缘服务器放置问题分为两个步骤解决：

1)根据基站覆盖范围内邻近基站数量与任务量确定边缘服务器的位置和数量；

2)以最小化时延与负载为目标，确定基站和边缘服务器的映射关系。

基于上述步骤，本文工作包括以下 3 点：

1)建立边缘计算服务器放置问题的优化模型，将边缘服务器放置问题建模为多个决策变量的混合整数规划问题，在多个约束条件的限制下，对边缘服务器延迟和负载差异进行优化。

2)提出一种多策略改进的鲸鱼优化算法来确定基站到边缘服务器的最佳分配关系。在算法中，通过改进收敛因子、加入惯性权重与融合差分进化思想，用改进的元启发式算法求解模型的最优放置方案，确保访问延迟与边缘服务器负载差异最小化。

3)使用实际数据集验证本文提出的算法，并分析了关键参数对算法性能的影响。实验结果表明，与其他基准放置方案相比，本文方法具有更优异的性能。

3 系统模型与问题建模

本文在图 1 所示的移动边缘环境下，根据部署算法确定边缘服务器的最佳位置、最优数量以及基站(Base Station, BS)和边缘服务器的关联性。由于该边缘服务器部署(Edge Server Placement, ESP)问题属于 NP 问题[21]，首先对 ESP 系统模型进行设计，表 1 列出了系统模型中涉及的符号变量。

表 1　相关符号说明表

Table 1　Related symbol descriptions

符号	说明
B	基站集合
S	边缘服务器集合
k_{ij}	基站 b_i 是否被边缘服务器 s_j 覆盖
z_{ij}	边缘服务器 s_j 是否放置在基站 b_i 处
R_i	基站 b_i 收集到的移动设备发出的需要转发给边缘计算服务器 s_j 的任务量
$\overline{B_{ij}}$	等效带宽
TR	电磁波在信道的传输速率
$\lambda_{\max}$	边缘服务器可以处理的最大任务量
λ_j	边缘服务器 s_j 接收到的任务量
γ	任务平均处理时间
t_{bi}	经由基站 b_i 转发的工作负载
w_i	基站一跳内的任务量总和

3.1　问题描述

在 ESP 中，ES 可以部署在移动网络的不同物理位置上，如宏基站、多制式基站汇聚点或无线网络控制器。大量文献研究表明，基于宏基站的 ES 部署方式对于降低用户访问时延及缓解移动网络拥塞效果最为显著。因此，本研究中 ES 与基站共享站址。

图 2 举例说明了 3 个 ES 与 11 个 BS 之间的分配情况。本文要解决的问题是如何从图中的 b_1-b_{11} 的 11 个基站中选择出放置 3 个边缘服务器 s_1，s_2 和 s_3 的位置，并将 11 个基站分别分配到边缘服务器 s_1，s_2 和 s_3 管辖范围内，每个基站通过访问这 3 个边缘服务器获取网络服务。以边缘服务器 s_2 为例，s_2 负责处理 b_3-b_64 个基站所转发的用户请求。模型中每一个边缘服务器的放置方案都尽量满足负载均衡化、延迟最小化两个目标。

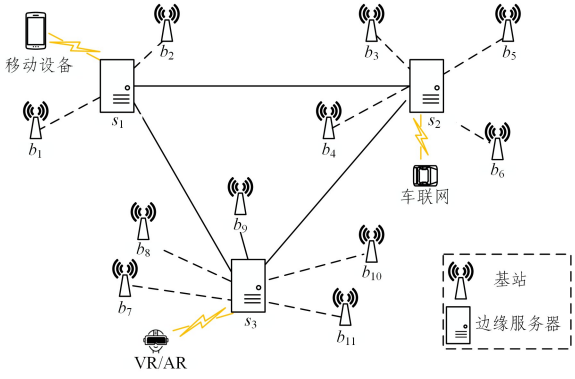


图 2　ES 与 BS 分配示意图

Fig. 2　Example of ES and BS distribution

因此可将 ESP 问题描述为：在给定的一组 ES、BS 以及每个基站负责的用户集合中寻找一种最佳的 ES 放置与 BS 分配方案，确定 ESP 问题中 ES 服务时延和负载差异最小放置方案。目标函数的数学模型表示如下：

$$\min Delay[E](a_i)\&\&Workload\_Balance[E](a_i)$$

其中，数学模型中 $Delay[E](a_i)$ 与 $Workload\_Balance[E](a_i)$ 分别代表当前放置方案 a_i 中的服务时延与负载差异值。

3.2　系统模型

边缘计算系统由大量移动终端设备（如智能手机、可穿戴设备和 VR/AR 等）、基站与边缘服务器组成。本文将多智能终端与边缘服务器的网络以网络图 $G=(V,E)$ 表示，G 中点的集合表示为 $V=B\cup S$，其中基站（BS）的集合定义为 $B=\{b_1,b_2,\cdots,b_n\}$，n 表示基站的总数。每个基站负责通信覆盖范围内的一个区域，通过已经部署的相互关联的基站可以实现基站之间直接或多跳方式通信。边缘服务器（ES）集合定义为 $S=\{s_1,s_2,\cdots,s_k\}$。k 表示边缘服务器数量，$E=\langle b_i,s_j\rangle$ 是无向图 G 中基站和边缘服务器之间连接的边的集合。

引入二元决策变量 $h_{ij}\in\{0,1\}$ 来表示 ES 与 BS 的覆盖关系，$h_{ij}=1$ 表示基站 b_i 被边缘服务器 s_j 覆盖，$h_{ij}=0$ 表示基站 b_i 不在边缘服务器 s_j 覆盖范围内；二元决策变量 $z_{ij}\in\{0,1\}$ 表示边缘计算服务器与基站放置关系，$z_{ij}=1$ 表示边缘服务器 s_j 放置在基站 b_i 处，反之未放置。对于所有的 i 和 j，满足 $1\leqslant i\leqslant n$，$1\leqslant j\leqslant k$。

3.2.1　时延模型

边缘服务器 s_j 服务延迟由 3 部分组成：1）BS 转发任务到 ES 时产生的转发延迟 D_j^f；2）BS 将任务发送给 ES 时的传输延迟 D_j^t；3）当前任务队列在 BS 上处理的时延 D_j^p。

基站 b_i 接收到用户 u_i 发送的任务后通过附近基站转发至最近的边缘服务器 s_j 进行处理，将基站 b_i 收集到的移动设备发出的需要转发给边缘计算服务器 s_j 的任务量记为 R_i，由此转发过程产生的延迟表示为 f_{ij}，f_{ij} 的值可通过当前需要转发的任务量与转发路径等效带宽计算，其计算式如式（1）所示：

$$f_{ij}=\frac{\sum_{i=1}^{m}R_i}{\overline{B_{ij}}} \tag{1}$$

其中，$\overline{B_{ij}}$ 为从基站 b_i 到最优的边缘计算服务器 s_j 通信链路的等效带宽，当通信链路为一跳时，其值为两个基站之间的标准带宽 B_{ij}；当通信链路为多跳时，$\overline{B_{ij}}=\frac{B_{ij}}{h}$，其中 h 为基站 b_i 到边缘服务器 s_j 的跳数。由式（1）可知，当前边缘服务器 s_j 的总转发延迟可根据二元决策变量 z_{b_i,s_j} 与覆盖范围内基站的转发延迟计算得出，如式（2）所示：

$$D_j^f=\sum_{i\in s_j}z_{ij}\cdot f_{ij} \tag{2}$$

基站将任务转发至边缘计算服务器时产生的传输延迟在本模型中被定义为传输的信道长度与电磁波在信道传输速率的比值。任意基站 b_i 可以直接通过链路连接请求边缘服务器获取网络服务，将基站所在位置表示为 (bx_i,by_i)，边缘计算服务器 s_j 位置表示为 (sx_i,sy_i)，将基站与边缘服务器之间链路长度记为 dis_{ij}，则当基站 b_i 与边缘计算服务器 s_j 为一跳传输时两者之间链接的链路长度可根据计算其欧氏距离得出，如式（3）所示：

$$dis_{ij}=\sqrt{(bx_i-sx_i)^2+(by_i-sy_i)^2} \tag{3}$$

传播时延 D_j^t 的计算式如式（4）所示：

$$D_j^t=\frac{\sum_{i,j=0}^{m}dis_{ij}}{TR} \tag{4}$$

其中，TR 为电磁波在信道传输速率。

基站 b_i 转发用户任务到边缘服务器 s_j，若此时边缘服务器 s_j 空闲，则立即处理该任务；否则，此任务将排队等待处理。在真实的边缘计算环境中，有些任务可能需要边缘服务器的多个处理阶段来完成服务过程，因此将边缘服务器视为单服务系统。假设等待边缘服务器空闲的过程遵循一个 $M/E_k/1/\infty$ 队列，即系统中有 k 个边缘服务器的服务台串联服务，系统服务时间服从 k 阶 Erlang 分布，应用该排队模型拟合边缘服务器的服务系统。该系统无容纳上限且遵循先到

先服务的原则，构造以到达率为自变量、返回值为排队时间的函数，如式(5)所示：

$$f(\lambda)=\frac{\rho}{\mu(1-\rho)}-\frac{\rho(k-1)}{2k\mu(1-\rho)} \tag{5}$$

其中，λ 与 μ 分别为任务到达率和边缘服务器服务速率，$\rho=\frac{\lambda}{\mu}$ 表示服务强度。因此，边缘服务器的处理延迟可以表示为：

$$D_j^p=f(\lambda_j)+\frac{1}{\mu}+\gamma \tag{6}$$

式(6)中 $\lambda_j=\sum_{b_i\in s_j}\sum_{u_l\in b_i}k_{ij}\times R_l$ 是边缘计算服务器 s_j 接收到的任务量，$\lambda_{\max}$ 是边缘服务器可以处理的最大任务量，γ 为任务平均处理时间。

综上所述，边缘服务器平均服务延迟为：

$$D\_S[E]=\frac{\sum_{j=0}^{k}D_j^f+D_j^t+D_j^p}{k} \tag{7}$$

3.2.2 负载模型

将边缘服务器 s_j 的工作负载表示为 L_{s_j}，经由基站 b_i 转发的工作负载用 l_{b_i} 表示。可将当前边缘服务器放置方案中服务器平均负载 $\bar{w}$ 表示为：

$$\bar{w}=\frac{\sum_{i=1}^{k}s_j}{k} \tag{8}$$

其中，$L_{s_j}=\sum_{i=0}k_{ij}\times l_{b_i}$，将工作负载标准差作为边缘计算服务器 s_j 工作负载的衡量指标，即工作负载差异值 W 越小，表示当前的边缘计算服务器放置方案中边缘服务器之间的负载越均衡。W 可表示为：

$$W\_B[E]=\sqrt{\frac{\sum_{j=1}^{k}(L_{s_j}-\bar{w})^2}{k}} \tag{9}$$

3.3 问题求解

由上述 ESP 系统模型可知，ESP 为典型的多目标优化问题。本文中使用加权法将多目标优化问题转为单目标问题。由于 ESP 中服务时延与负载均衡模型中目标函数值量纲不同，首先为了将不同数量级大小的数据变成可以相互进行数学运算的具有相同数量级的可比性数据，将归一化后的数据与原始数据分别表示为 $V_{\text{nor}}(a_i)$，$V_{\text{nor}}(a_i)$，采用 max-min 方法将服务时延与负载差异数据映射至(0,1)区间进行归一化，如式(10)所示。

$$V_{\text{nor}}(a_i)=\frac{V(a_i)-\min\limits_{1\leqslant j\leqslant\mathbb{K}}\{V(a_j)\}}{\max\limits_{1\leqslant j\leqslant\mathbb{K}}-\min\limits_{1\leqslant j\leqslant\mathbb{K}}\{V(a_j)\}} \tag{10}$$

基于归一化后数据，对原 ESP 优化目标设置加权系数 α 与 β，α 与 β 的值位于区间[0,1]，满足关系为 $\alpha+\beta=1$，因此 ESP 数学模型可表示为：

$$\min\ \alpha D\_S[E]_{\text{nor}}+\beta W\_B[E]_{\text{nor}} \tag{11}$$

$$\text{s.t. } s_i\cap s_j=\varnothing \tag{11a}$$

$$\sum_{i=1}^{k}h_{ij}=\{0,1\},1\leqslant j\leqslant k \tag{11b}$$

$$\sum_{i=1}^{n}z_{ij}=1,1\leqslant j\leqslant n \tag{11c}$$

约束条件(11a)表示放置方案中任意两个 ES 覆盖的基站范围没有交叉；(11b)表示一个基站只能被分配给一个边缘服务器，即 ES 覆盖的基站是全部基站的集合，所有 BS 都被边缘服务器覆盖；(11c)表示 ES 必须放置在基站位置上。

3.4 优化方法

为了有效地解决 ESP 问题，应充分考虑用户移动性带来的负载动态变化，当基站负载发生变化时，需验证边缘服务器放置方案是否更改。因此本文提出了一种基于多策略改进的鲸鱼优化边缘服务器放置算法 MIWOA-ESP。可知，ESP 问题可通过两个步骤解决：1)在大量基站中选择边缘服务器放置位置；2) 基于 ESP 问题目标函数约束确定 BS 分配关系。

基于上述步骤，MIWOA-ESP 算法首先在最小化服务时延与负载差异基础上计算每个 BS 放置权重系数，基于 BS 覆盖范围约束将 BS 划分为 K 类，K 取值为 ES 数量。基于初始边缘服务器放置位置与分配关系，使用多策略改进的鲸鱼优化算法进行优化。

4 MIWOA-ESP 算法

4.1 选择边缘服务器放置位置

在边缘服务器放置问题中，优化目标为最小化负载与延迟，因此在选择放置位置时需考虑每个基站的任务量与到附近跳数为一跳(即在基站传输范围内)的基站个数。

基站任务量决定了基站的负载程度，一跳基站个数意味着当前基站位置作为边缘服务器放置位置时，覆盖范围内的基站时延最小。因此使用 TOP-K 算法思想[22]，设置基站放置权重系数 ξ_i，选择边缘服务器满足优化目标函数要求的任务量最大且到当前基站为一跳的基站个数最多的位置放置。

首先计算基站 b_i 为一跳的基站数量记为 σ_i，并计算任务量总和记为 w_n，将总和标准化。得到将基站数与任务量归一化后的加权总和，并获得了将边缘服务器放置在每个基站上的权重系数 ζ_i 度量，计算式如式(2)所示：

$$\zeta_i=\eta\times w_i-(1-\eta)\times\sigma_i \tag{12}$$

其中，η 是基站的任务量加权因子。选择具有较小 ζ_i 值的 k 个基站放置边缘服务器并确定边缘服务器的位置矩阵。

4.2 编码方案

基于确定的边缘计算服务器放置位置，第二步确定基站到边缘服务器的分配矩阵。由上节分析可知，ESP 问题是一个离散优化问题，传统的二进制编码和实数编码难以有效描述 ESP 问题，并且难以与迭代过程相结合。为了高效解决 ESP 问题，首先使用改进的编码方案提升算法寻优速率。由于边缘服务器覆盖基站的分配问题属于一到多的映射关系问题，每个基站的位置也是边缘服务器的潜在放置位置，因此使用二维矩阵进行编码，建立 $k*(n+1)$ 矩阵 $\boldsymbol{Y}$。

矩阵每一行代表边缘服务器，第一列表示要部署的边缘服务器的数量，此列中每个标记表示该标号边缘服务器已放置至其取值的基站位置。矩阵 $\boldsymbol{Y}$ 中后 n 列表示为基站的分配方案。由于物理位置限制以及部署的边缘服务器负载问题，每个基站的覆盖范围是固定的且连接的边缘服务器必须在其覆盖范围内。

在本文提出的编码方案中，编码矩阵 $\boldsymbol{Y}$ 每一行都可被视为边缘服务器，每一列代表基站。且每个放置决策可以通过二维数组唯一识别，例如矩阵 $\boldsymbol{Y}$ 中第二行表示将边缘服务器 s_2 放置在基站 b 处，并将 b_{n-1} 等基站分配至 s_2。基于 ESP 问题优化目标函数约束限制，在对 ES 放置方案进行编码时，应满足以下编码规则：1)$\boldsymbol{Y}$ 矩阵中第 1 列任何两行标记值不同；2)$\boldsymbol{Y}$ 矩阵中的后 n 列数据每列累加和为 1。

$$\boldsymbol{Y}=\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \cdots \\ k \end{array}\begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & \cdots & n-1 & n \end{array} \\ \begin{bmatrix} 2 & 1 & 0 & \cdots & 0 & 1 \\ 15 & 0 & 0 & \cdots & 1 & 0 \\ 4 & 0 & 1 & \cdots & 0 & 0 \\ & \cdots & & \cdots & & \cdots \\ 5 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \end{array}$$

4.3 鲸鱼优化算法

鲸鱼优化算法通过模拟座头鲸的捕食行为进行迭代寻优。其优点在于操作简单、参数少以及跳出局部最优的能力强。WOA 优化进度包括 3 种策略:包围猎物、气泡网络狩猎方法和寻找猎物。

4.3.1 包围猎物

代表候选解决方案集的鲸鱼种群的位置可以表示为:

$$X=\begin{bmatrix} x_{1,j} & \cdots & x_{1,D} \\ \vdots & \ddots & \vdots \\ x_{N,j} & \cdots & x_{N,D} \end{bmatrix}_{N\times D} \tag{13}$$

其中,N 表示鲸鱼种群的大小,D 是优化问题的维度。$X_i=(x_{i,1},x_{i,2},\cdots,x_{i,D})$。由于事先未知最佳信息,因此 WOA 算法将当前的最佳解决方案视为猎物或迭代过程中的近似最佳。座头鲸能识别猎物的位置并围着它们转。由于最优位置在搜索空间中的位置是未知的,WOA 算法假设当前的最佳候选解是目标猎物或接近最优解。在定义了最佳候选解之后,其他候选位置将尝试向最佳位置移动并更新其位置。此行为由式(14)表示。一旦获得了最佳鲸鱼个体(当前的最佳解决方案),种群中的其他鲸鱼将根据式(14)更新其对猎物的位置以及迭代。

$$\begin{aligned} D&=|C\cdot X^*(t)-X(t)| \\ X(t+1)&=X^*(t)-\boldsymbol{A}\cdot D \end{aligned} \tag{14}$$

其中,t 是当前的迭代编号,$X(t)$代表第 t 次迭代,$X^*(t)$表示到目前为止鲸鱼种群达到的最佳鲸鱼位置,如果在迭代中找到更好的位置,则将更新 $X^*(t)$。$\boldsymbol{A}\cdot D$ 代表用于更新鲸鱼的位置的步长大小。$\boldsymbol{A}$ 和 $\boldsymbol{C}$ 作为系数向量由式(15)计算:

$$\begin{aligned} \boldsymbol{A}&=2a\boldsymbol{r}_1-a \\ \boldsymbol{C}&=2\boldsymbol{r}_2 \end{aligned} \tag{15}$$

其中,$\boldsymbol{r}_1$ 和 $\boldsymbol{r}_2$ 是分布在[0,1]中的随机向量;$a=2-2t/t_{\max}$是调整开发与探索之间关系的收敛因子;$t_{\max}$ 表示最大迭代次数。

4.3.2 气泡网络狩猎方法

WOA 的气泡网络攻击方法在可行域中当前最佳解决方案附近进行了精细的搜索,根据座头鲸的狩猎行为(以螺旋运动游向猎物),其数学模型被定义为:

$$\begin{aligned} D'&=X^*(t)-X(t) \\ X(t+1)&=D'\cdot e^{bl}\cdot\cos(2\pi l)+X^*(t) \end{aligned} \tag{16}$$

其中,D'表示当前鲸鱼个体与最佳解决方案之间的距离。$l\in[-1,1]$是一个随机值,b 是定义对数螺线形状的常数。当 $|A|<1$ 时,鲸鱼向猎物发起攻击,座头鲸在一个缩小的包围圈内绕着猎物游动,同时沿着螺旋形的路径游动。当前的鲸鱼种群随机执行气泡网络攻击方法或环绕猎物,在当前最佳解决方案附近执行局部搜索,该搜索描绘了 WOA 算法的局部搜索阶段。假设有 50%的可能性选择收缩包围机制或螺旋模型来更新优化过程中鲸鱼的位置,则当前鲸鱼种群的位置更新机制可以描述为:

$$X(t+1)=\begin{cases} X^*(t)-\boldsymbol{A}\cdot D, & p<0.5 \\ X^*(t)+D'\cdot e^{bl}\cdot\cos(2\pi l), & p\geqslant 0.5 \end{cases} \tag{17}$$

其中,$p\in[0,1]$是一个随机值。

4.3.3 搜索猎物

搜索猎物是 WOA 算法的重要组成部分,以实现全局搜索。当$|A|\geqslant 1$,执行此随机搜索策略。当前的鲸鱼种群随机选择一个搜索个体作为参考以更新其他鲸鱼的位置。

该策略试图在整个搜索空间中找到更好的解决方案,迫使鲸鱼偏离猎物,借此找到更合适的猎物,这样可以加强算法的探索能力,使 WOA 算法能够进行全局搜索。数学模型如下:

$$\begin{aligned} D&=|\boldsymbol{C}\cdot X_{\text{rand}}(t)-X(t)| \\ X(t+1)&=X^*(t)-\boldsymbol{A}\cdot D \end{aligned} \tag{18}$$

$X_{\text{rand}}(t)$是当前种群中随机选择的鲸鱼。因此,WOA 的优化进度除以$|a|$后决定进入局部搜索阶段或全局搜索阶段。如果$|A|\geqslant 1$,进行全局搜索阶段,否则执行局部搜索。

鲸鱼算法优点众多,但不适用于解决 ESP 问题,并且接近全局最优时存在搜索能力不足和容易陷入局部最优的问题。为了解决这些问题,本文提出一种多策略改进的鲸鱼优化放置方法。

4.4 多策略改进的鲸鱼优化算法

4.4.1 设定适应度函数

本文将边缘计算服务器放置方案目标函数定义为最小化时延与负载,将负载与时延通过加权法归一化后的系统开销用作鲸鱼优化算法的适应度函数,如式(19)所示:

$$F(X)=0.5\sum_{j=1}^{k}D\_S(X)+0.5\sum_{j=1}^{k}W\_B(X) \tag{19}$$

4.4.2 改进非线性收敛因子

在处理优化问题时,理想的优化过程具有高精度、快速收敛的特点以及强大的全局搜索能力。通过控制式(15)中 A 的值可以调整全局和局部搜索阶段。即 A 的值越小,本地搜索能力就越强。与原算法中线性减少的策略相比,非线性策略将增强算法的优化能力。

显然,A 的线性减少策略不符合这一期望。基于 sin 函数特征,提出了非线性收敛因子 a,其更新公式如下:

$$a=2-2\sin\left(\frac{t}{t_{\max}}\pi+\varphi\right) \tag{20}$$

其中,t 为当前迭代次数;μ 和 φ 是其表达式相关参数,定值选取 $\mu=0.5$,$\varphi=0$。

4.4.3 加入自适应权重

受粒子群优化算法的启发,在鲸鱼的位置更新中加入一个随迭代次数变化的惯性权重 w。惯性权重 w 控制 MI-WOA-ESP 算法在搜索过程前期最优鲸鱼位置对当前个体位置调整的影响,提升算法在前期的全局搜索能力;随着迭代次数的增加,逐渐提升最优鲸鱼位置的影响力,使得其他鲸鱼能够快速收敛到最优鲸鱼的位置,加快整个算法的收敛速度。根据鲸鱼优化算法中更新次数的变化,选用迭代次数 t 构成的自适应惯性权值如下:

$$w(t)=\exp\left(1-\frac{T_{\max}+t}{T_{\max}-t}\right) \tag{21}$$

加入的惯性权值 w 在[0,1]之间非线性变化,由于指数函数的非线性特性,在算法前期,鲸鱼以较大的螺旋形状搜寻目标,鲸鱼尽可能地去探索全局最优解,提升算法的全局最优

搜寻能力;在算法后期,鲸鱼以小螺旋形状搜寻目标,提升算法的寻优精度。因此权值在算法初期较小,但变化速度稍快;在算法后期其值较大,但变化速度会减缓,充分保证了算法的收敛性。

改进后的鲸鱼优化算法位置更新公式如式(22)所示:

$$X(t+1)=\begin{cases} w(t)X^*(t)+D\cdot e^{bl}\cos(2\pi l), & p\geqslant 0.5 \\ w(t)X^*(t)-\boldsymbol{A}\cdot|\boldsymbol{C}\cdot X^*(t)-X(t)|, & p<0.5 \end{cases} \tag{22}$$

引入自适应权重之后,在位置更新时会根据迭代次数的增加动态调节权重大小,使得最优鲸鱼位置 $X^*(t)$ 在不同时刻对鲸鱼个体的指导能力不同。随迭代次数的增加,鲸鱼群的会集中向最优位置方向靠近,此时权值 w 较大会使鲸鱼位置移动速度加快,进而加快算法收敛速度。

4.4.4 引入差分变异策略

在鲸鱼优化算法迭代后期,鲸鱼在更新位置时,以当前最优位置作为本次迭代的目标。在整个迭代环节中,最优位置只有在出现优于它的位置时才会更新,因此实际更新次数并不多,这会导致算法搜寻效率不高。鲸鱼位置将快速聚集在最优解附近,导致鲸鱼种群趋同性严重,算法停滞不前,进而增大算法陷入局部最优值的概率。为解决此问题,融合差分进化思想至鲸鱼优化算法中,通过随机选择两个鲸鱼个体计算差值与全局最优个体进行变异操作产生新的个体,并对新个体引入反向学习策略,评估比较保留适应度更好的个体,从而改善种群的多样性,提升算法局部最优值的逃逸能力。新个体的数学模型可表示为:

$$X_{new}^t=xb^t+\lambda(X_{r1}^t-X_{r2}^t) \tag{23}$$

其中,X_{r1}^t,X_{r2}^t 是随机选择的鲸鱼位置;λ 为缩放因子,其值为[0,1]的随机数。xb^t 为全局最优值。

对于生成的邻域位置,采用贪婪的策略判断是否保留,如式(24)所示:

$$X^*(t)=\begin{cases} \widetilde{X}(t), & f(\widetilde{X}(t))<f(X^*(t)) \\ X^*(t), & f(X^*(t))\leqslant f(\widetilde{X}(t)) \end{cases} \tag{24}$$

其中,$f(\widetilde{X}(t))$ 为 $\widetilde{X}(t)$ 位置时的适应度值,如果生成的位置适应度值比原全局最优的适应度值更小,则更新全局最优位置。反之,最优位置保持不变。

4.5 MIWOA-ESP 算法描述

求解 ESP 问题的具体过程如算法 1 所示。

算法 1 负载均衡的低时延边缘服务器放置算法

输入:基站数据集,L,Max_delay,N,G

输出:边缘计算服务器放置矩阵 **Y**

1. INITIALIZE;
2. 计算基站权重系数 ζ_i;
3. 计算适应度函数值,寻找最佳适应度函数值 fg,令 X^* 为最优搜索单元;
4. WHILE iteration number<L do
5. FOR v≤−1 to N then
6. UPDATE a,A,C,l,p;
7. CALCULATE the current iteration inertia weight w(t);
8. IF(p<0.5)
9. IF(|A|<1)
10. 使用式(22)更新当前搜索单元的位置;
11. ELSE IF(|A|≥1)
12. 式(23)的差分进化策略产生一个搜索单元并与原搜索单元比较;
13. 若新的搜索单元更优,则更新原个体的位置和适应度函数值;
14. END IF
15. ELSE IF(p≥0.5)
16. 更新当前搜索单元的位置;
17. END IF
18. END FOR
19. 检查修正;
20. 计算所有搜索单元的适应度;
21. 更新 X^*
22. t=t+1
23. END WHILE
24. RETURN **Y**

5 实验与分析

本章通过 3 个实验评估负载均衡的低时延边缘服务器放置算法的性能,并将其与其他算法进行比较。

5.1 实验设置

本文实验部分基于上海电信基站数据集模拟构建接近现实的模拟环境。数据集由总计 3233 个基站的位置和用户的互联网访问日志组成[23],如图 3 所示[3]。首先对数据集进行预处理,表 2 列出了随机选择的处理后的部分基站数据信息。将此数据集中基站的工作负载的时间和空间分布用作输入,以提供用户工作负载的时间和空间分布。

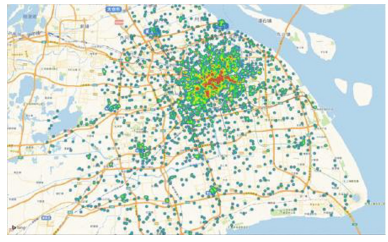


图 3 上海的 3233 个基站的分布图

Fig. 3 Distribution map of 3233 base stations in Shanghai

表 2 部分基站信息

Table 2 Partial base station information

ID	用户数	经度	纬度	任务到达率	工作负载时间
40	1824	30.8324	121.4397	0.28602	24589
70	1242	26.1393	103.0786	0.60824	3755547
315	606	30.9785	121.4477	0.34189	1702826
650	283	31.3506	121.5891	0.78426	998393
2767	330	46.7775	131.8122	0.50826	1808387

原始数据集不包含有关基站之间跃点数的信息,为了找到基站之间的跳数,本文设定两基站之间的距离小于 5 km,则两个基站之间存在直接连接。否则,没有直接连接。根据连接情况可计算出两个基站之间的跳数。基于数据集中提供的基站位置生成网络拓扑。将相邻的基站链接在一起以形成单跳关系。使用 Floyd 算法来生成 G 拓扑。

5.2 算法收敛性分析

在进行算法收敛性分析时,设置基站数量为 700,边缘服务器数量为 30,迭代次数为 10:100,步长为 10。

如图 4 所示,适应度函数设置为平均服务时延与负载

差异最小化。由适应度函数变化折线图可知，MIWOA-ESP 算法更快地降低了系统开销，并在大约 20 次迭代时算法适应度函数值降低至局部最优，此时寻得局部最优解。约 40 次迭代后算法适应度函数保持恒定，跳出局部最优解，寻得 ESP 最佳方案。

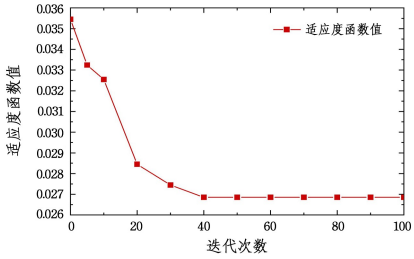


图 4　不同迭代次数下 MIWOA-ESP 算法适应度函数值变化情况

Fig. 4　Variation of MIWOA-ESP algorithm fitness function values with different numbers of iteration

5.3　实验对比算法

本节将提出的 MIWOA-ESP 算法与文献[11，13，24]中 3 种典型的放置算法 *K*-means，TOP-*K* 与 Random 进行对比以评估本文算法的时延与负载差异，进一步论证本文算法的优越性。基于文中研究内容，设置实验测试指标为当前方案平均时延、负载差异与综合性能。通过 3 个实验场景完成算法对比部分。

5.3.1　对比实验

将基站数量设为固定值 1000，动态增加边缘服务器 *k* 的数量。比较不同边缘计算服务器数量时，3 种放置方案中计算时延与负载均衡程度。边缘服务器平均时延方面，实验结果如图 4 所示。

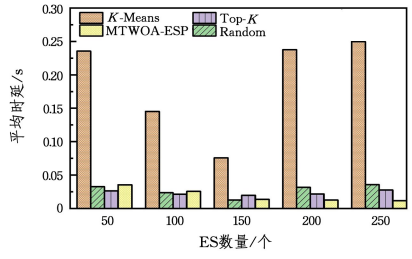


图 5　不同 ES 数量下放置方案平均时延对比图

Fig. 5　Comparison of average latency of placement schemes with different number of ES

根据图 5 可知，基站数量固定时，随着 ES 的动态加入，基站可选的边缘服务器数量增多，缩短了基站到边缘服务器的传输距离，因此边缘服务器的服务时延呈现出下降趋势。

其中 MIWOA-ESP 平均时延最小，其次为 Top-*K*、Random 和 *K*-Means。在 4 种放置方案中，除 MIWOA-ESP 之外，其他算法均在 ES 数量为 150 即 ES 与 BS 数量比例为 15∶100 时表现最好，当 ES 数量达到 250 时，MIWOA-ESP 放置方案表现最好，其他 3 种放置方法性能反而较 $ES=150$ 时有所下降。其原因为本文提出的 MIWOA-ESP 使用精英反向学习策略初始化种群并且通过加入非线性收敛因子与动态权重对搜索单元进行改进，加快了算法收敛速度，同时避免求解陷入局部最优。

综合访问平均距离及负载标准差两个属性，取每个属性所占权重比例为 1∶1 将其设置为适应度函数，边缘服务器 ES 与基站 BS 数量比保持为15∶100。不同基站数量与边缘服务器数量下，放置方案中 ESP 系统负载标准差如图 6 所示。

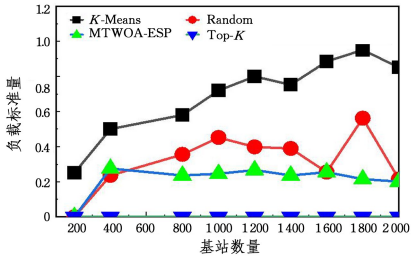


图 6　ES 与 BS 数量比为 15∶100 时放置方案负载标准差对比图

Fig. 6　Comparison of standard deviation of the placement scheme load for ES to BS number ratio of 15∶100

由图 6 可知，各边缘服务器访问的负载标准差随着基站 BS 与边缘服务器 ES 的数量逐渐增多而减小。4 种放置方案中，TOP-*K* 算法思想为计算各基站负载值并选择负载值最大的前 *K* 个基站放置 ES，其中 *k* 即为当前比例下的 ES 数量，因此算法的负载差异始终保持最小值。根据实验结果分析可得，除 TOP-*K* 算法外，MIWOA-ESP 算法负载差异程度最小，低于 Random 与 *K*-means 方法。

固定基站数量为 2000，比较不同基站数量下边缘服务器放置方案计算时延与负载均衡程度，观察其性能变化。

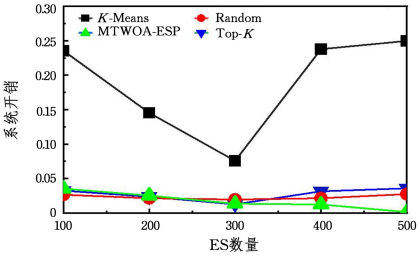


图 7　不同方案与 ES 数量下综合性能对比图

Fig. 7　Comprehensive performance comparison of different schemes with different ES quantities

由图 7 可知，在解决 ESP 问题时，若不考虑经济成本，边缘服务器放置比例越大，即边缘服务器数量越多，ES 密度越大时，各基站访问边缘服务器系统开销最小。

结束语　本文对边缘服务器放置问题进行了研究与分析，针对以往边缘服务器放置中仅将传输距离作为考虑因素，未考虑时延与服务器负载程度问题，建立优化的时延与负载均衡模型。通过分析 ESP 的问题特点，将时延与负载均衡作为放置问题的优化目标，提出了一种多方面改进的鲸鱼优化算法，改善了传统鲸鱼优化算法容易陷入局部最优及收敛速度慢等问题。通过与现有边缘服务器放置问题解决方案对比可知，本文算法的时延与负载均衡程度最佳。但目前仍有多方面的工作需要改进：本文使用的数据集未考虑用户移动性，而真实的网络环境中，用户数量与任务量是实时变化的。下一步工作中将注重考虑用户移动性。由于边缘服务器属资源消耗型，其能耗对总体性能并无太大影响，但从运营商角度来说，能耗问题无法忽视。在今后的工作中，将考虑从运营商角度建立能耗与负载模型，并在此基础上完成算法设计。

参 考 文 献

[1]　KEMP R，PALMER N，KIELMANN T，et al. Cuckoo：A Com-

putation Offloading Framework for Smartphones[M]// Mobile Computing, Applications, and Services: Vol. 76. Berlin, Heidelberg: Springer, 2012: 59-79.

[2] LEWIS G, ECHEVERRIA S, SIMANTA S, et al. Tactical Cloudlets: Moving Cloud Computing to the Edge[C]// 2014 IEEE Military Communications Conference. Baltimore, MD, USA: IEEE, 2014: 1440-1446.

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

[4] LI B, HOU P, WANG K, et al. Deployment of edge servers in 5G cellular networks[J]. Transactions on Emerging Telecommunications Technologies, 2020, 33(8): e3937.

[5] SANTOYO-GONZALEZ A, CERVELLO-PASTOR C. Edge Nodes Infrastructure Placement Parameters for 5G Networks [C]// 2018 IEEE Conference on Standards for Communications and Networking(CSCN). Paris, France: IEEE, 2018: 1-6.

[6] 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.

[7] XIAO K, GAO Z, WANG Q, et al. A Heuristic Algorithm Based on Resource Requirements Forecasting for Server Placement in Edge Computing[C]// 2018 IEEE/ACM Symposium on Edge Computing(SEC). Seattle, WA, USA: IEEE, 2018: 354-355.

[8] MANASVI G, CHAKRABORTY A, MANOJ B S. Social Network Aware Dynamic Edge Server Placement for Next-Generation Cellular Networks[C]// 2020 International Conference on Communication Systems & Networks(COMSNETS). Bengaluru, India: IEEE, 2020: 499-502.

[9] CUI G, HE Q, XIA X, et al. Robustness-oriented k Edge Server Placement[C]// 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing(CCGRID). Melbourne, Australia: IEEE, 2020: 81-90.

[10] ZENG F, REN Y, DENG X, et al. Cost-Effective Edge Server Placement in Wireless Metropolitan Area Networks[J]. Sensors, 2018, 19(1): E32.

[11] 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, CA: IEEE, 2018: 66-73.

[12] LI B, WANG K, XUE D, et al. K-Means Based Edge Server Deployment Algorithm for Edge Computing Environments[C]// 2018 IEEE Smart World, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation(SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). 2018: 1169-1174.

[13] CHIU T L, CHEN P H, CHEN H, et al. An effective metaheuristic algorithm for the deployment problem of edge computing servers[C]// 2019 IEEE International Conference on Systems, Man and Cybernetics(SMC). 2019: 1995-2000.

[14] QIN Z, XU F, XIE Y, et al. An improved Top-K algorithm for edge servers deployment in smart city[J]. Transactions on Emerging Telecommunications Technologies, 2021, 32(8): e4249.

[15] LI B, HOU P, WU H, et al. Optimal edge server deployment and allocation strategy in 5G ultra-dense networking environments [J]. Pervasive and Mobile Computing, 2021, 72: 101312.

[16] LI Y, ZHOU A, MA X, et al. Profit-aware edge server placement [J]. IEEE Internet of Things Journal, 2021, 9(1): 55-67.

[17] BI J, YUAN H, DUANMU S, et al. Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization[J]. IEEE Internet of Things Journal, 2020, 8(5): 3774-3785.

[18] NASROLLAHZADEH S, MAADANI M, POURMINA M A. Optimal motion sensor placement in smart homes and intelligent environmentsusing a hybrid WOA-PSO algorithm[J/OL]. Journal of Reliable Intelligent Environments, 2021: 1-13. https://www.mdpi.com/1424-8220/19/1/32.

[19] WONG L A, RAMACHANDARAMURTHY V K, WALKER S L, et al. Optimal placement and sizing of battery energy storage system for losses reduction using whale optimization algorithm [J]. Journal of Energy Storage, 2019, 26: 100892.

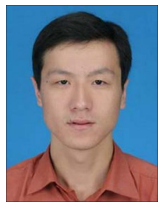
[20] BHATT U R, DHAKAD A, CHOUHAN N, et al. Fiber wireless(FiWi) access network: ONU placement and reduction in average communication distance using whale optimization algorithm[J]. Heliyon, 2019, 5(3): e01311.

[21] GUO Y, WANG S, ZHOU A, et al. User allocation-aware edge cloud placement in mobile edge computing[J]. Software: Practice and Experience, 2020, 50(5): 489-502.

[22] SUCIU D, RE C. Efficient top-K query evaluation on probabilistic data: U. S. Patent 7814113 B2[P]. 2010-12-10.

[23] WANG Z, ZHANG W, JIN X, 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.

[24] SHARMA A, JALAL A S. Clustering based hybrid approach for facility location problem[J]. Management Science Letters, 2017, 7(12): 577-584.



YUAN Peiyan, born in 1978, Ph.D, professor, is a member of China Computer Federation. His main research interests include edge computing and group intelligence perception.