Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P1856-cd

An Intelligent Algorithm for Edge Server Deployment Based on the N-1 Security Criterion

Shenghan Zhou

School of Reliability and Systems Engineering, Beihang University, Beijing, China. E-mail: zhoush@buaa.edu.cn

Jiankai Wang

School of Reliability and Systems Engineering, Beihang University, Beijing, China. E-mail: wangjk@buaa.edu.cn

Yiming Chen

School of Reliability and Systems Engineering, Beihang University, Beijing, China. E-mail: yimchen@buaa.edu.cn

Fajie Wei

School of Economics and Management, Beihang University, Beijing, China. E-mail: weifajie@buaa.edu.cn

Linchao Yang

School of Economics and Management, North China Electric Power University, Beijing, China. E-mail: yanglinchao@ncepu.edu.cn

Wenbing Chang*

School of Reliability and Systems Engineering, Beihang University, Beijing, China. E-mail: changwenbing@buaa.edu.cn

Abstracts. The study proposes an edge server deployment method based on the widely used N-1 security criterion in power systems to improve the security and reliability of edge computing systems in the event of single-point failures. The N-1 security criterion requires the system to remain operational without triggering broader system issues in the event of any single equipment failure. This paper designs redundancy mechanisms and backup server schemes to ensure that even if an edge server fails, its workload can be quickly and seamlessly transferred to a backup server, thereby avoiding negative impacts on service quality, especially in terms of latency and performance. This method effectively reduces the security risks that could arise from single-point failures in edge computing systems. Simulation results show that, compared with traditional server deployment methods, the N-1 security criterion-based approach performs significantly better in terms of system reliability, stability, and fault tolerance, substantially improving the security and service continuity of edge computing systems. Additionally, considering that the random nature of the initialization phase in traditional K-Means clustering algorithms may lead to instability in the final results and that servers may face overload issues, this study further proposes an improved K-Means algorithm. By optimizing the selection of initial cluster centres' and adjusting the clustering process, the new algorithm more effectively reduces communication latency and balances the load between servers. Experimental results indicate that the improved K-Means algorithm outperforms existing algorithms, including DBCA, K-Means, Top-K, and Random algorithms, in terms of reducing communication latency and achieving load balancing. Moreover, the deployment strategy based on the N-1 security criterion significantly enhances system robustness and security, ensuring stable system operation in the event of a single edge server failure.
Keywords: Edge server deployment, Mobile edge computing, N-1 security criterion, K-Means, Reliability and Security

1. INTRODUCTION

The rapid advancement of fifth-generation (5G) mobile communication and artificial intelligence (AI) has significantly propelled the Internet of Things (IoT), enabling applications such as autonomous driving, intelligent traffic management, and virtual reality. These applications, however, demand extensive real-

* Corresponding Author

time computing resources that exceed the capabilities of mobile devices, particularly in high-latency and high-computation scenarios. Edge computing has emerged as a distributed paradigm to address these challenges by deploying resources closer to users, reducing task offloading latency, and enhancing system performance (Ahvar et al., 2022). The effective deployment of edge servers is critical to optimizing edge computing, as strategically determining their location and number minimizes user access latency and achieves load balancing, thereby improving overall system efficiency (Tiwari et al., 2024).

While edge computing has advanced task offloading by reducing latency and processing compute-intensive tasks closer to users, the issue of server failures remains underexplored in existing research. Most studies prioritize minimizing latency and provisioning costs, overlooking disruptions to task execution, system reliability, and security risks caused by server failures. Addressing single points of failure is critical to ensuring fault tolerance and stability, particularly in critical applications. To tackle this challenge, this study incorporates the N-1 security criterion (Vaurio, 2002) from power systems into edge server deployment optimization. This approach enables seamless redirection of traffic from failed servers to pre-planned backups while adhering to delay and load constraints, thereby significantly enhancing system robustness and reliability.

In this context, this paper proposes an optimization method for edge server deployment based on the N-1 security criterion. To address the limitations of the traditional K-Means algorithm, such as susceptibility to random initialization and server overload, an enhanced K-Means algorithm is introduced. The proposed approach incorporates a weighted density method for cluster center initialization and refines the cluster adjustment formula to ensure compliance with edge server load constraints, thereby improving system robustness and performance. Experiments using a dataset from Shanghai Telecom's mobile base stations demonstrate that the proposed algorithm outperforms benchmarks, including DBCA, K-Means, Top-K, and random algorithms, in reducing communication latency and achieving server load balancing.

The methods proposed in this paper provide a practical reference for the design and optimization of future edge computing systems and lay a solid foundation for the application of edge computing in scenarios such as intelligent transportation and smart manufacturing.

The remainder of this paper is organized as follows: Section 2 provides a brief overview of related work. Section 3 describes the system model and defines the problem. Section 4 presents the improved K-Means clustering algorithm. Section 5 discusses the experiments and simulations conducted. Finally, Section 6 offers the conclusions drawn from this study.

2.Related Work

Mobile Edge Computing (MEC), as an effective approach to mitigating latency issues and improving existing network architectures, has been attracting increasing attention (Shi et al.,2016).

Edge server placement is a critical task in deploying MEC architecture, requiring comprehensive research (Asghari and Sohrabi, 2022). Guo and Tang (2021) proposed the ESPHA method, combining the K-means algorithm with an improved heuristic AG algorithm to minimize access latency and load variance. Bahrami et al. (2024) introduced the binary hybrid NSGA II-MOPSO algorithm (BHNM) to approximate the Pareto front and address model-solving challenges.

Jia et al. (2017) proposed deploying cloudlets in user-dense areas to balance workloads and reduce average request waiting times. Kasi et al. (2021) formulated edge server placement as a multi-

objective constrained optimization problem, employing genetic algorithms and fractional search techniques to determine optimal deployment. Zhang et al. (2022) utilized quantum encoding to enhance edge server coverage, optimize workload balancing, and minimize waiting times in vehicular network services.

Previous studies have advanced edge server placement by addressing key issues in model formulation and algorithmic solutions within mobile edge computing. However, limited attention has been given to the reliability of edge computing networks. Most assume edge servers are fully reliable, neglecting potential failures that can disrupt tasks and introduce risks. This study focuses on optimizing edge server deployment to enhance the reliability of edge computing networks.

3. PROBLEM FORMULATION

3.1.System Model

In mobile edge computing, the edge server deployment problem can be modelled as a connected undirected graph $G = (B \cup S, E)$, composed of three layers: the cloud server layer, the edge server layer, and the user equipment layer.

We primarily focus on the edge server layer, where $B = (b_1, b_2, ..., b_j, ..., b_n)$ represents the set of base stations. The set of edge servers is $S = (s_1, s_2, ..., s_i, ..., s_k)$. The set E denotes the relationships between base stations and edge servers. When k edge servers are deployed, the base stations are divided into k subsets. Let $C = (c_1, c_2, ..., c_i, ..., c_k)$ represent these subsets, and ensure that the base stations in each subset do not intersect. Each subset has one base station equipped with an edge server, responsible for handling the tasks of all base stations within that subset.

For each base station b_j, let $l(b_j)$ and $\omega(b_j)$ represent its location and workload, respectively.

For each edge server s_i, let $l(s_i)$ and $\omega(s_i)$ represent its location and workload, respectively. In this context, $l(s_i)$ and $\omega(s_i)$ are given in terms of latitude and longitude. To accurately describe the edge server deployment problem, this paper introduces a matrix X to represent the allocation relationship between base stations and edge servers. The binary decision variable x_{ij} is an element of matrix X, where $x_{ij} \in [0,1]$. When $x_{ij} = 1$, it indicates that base station b_j is assigned to edge server s_i; otherwise, $x_{ij} = 0$.

This paper does not consider the wireless transmission delay before a mobile user's request accesses the base station, as the change in the edge server's location does not affect the delay before the base station is accessed. It focuses on the delay caused by the location and bandwidth of the edge server. Based on this, the transmission time from base station b_j to edge server s_i is then given by equation (1):

$$T_{ij} = \frac{d(s_i, b_j)}{r_{ij}} \tag{1}$$

Where $d(s_i, b_j)$ is the distance between b_j and s_i, and r_{ij} is the transmission speed.

Load balancing is crucial for improving system performance and reliability. We define the load of each edge server as shown in equation (2):

$$w(s_i) = \sum_{j=1}^{n} x_{ij} w(b_j) \tag{2}$$

The load variance, which measures the deviation of each edge server's load from the average load, is expressed in equation (3):

$$\sigma^2 = \frac{\sum_{i=1}^{k} (w(s_i) - \bar{w})^2}{k} \tag{3}$$

The edge server deployment problem can be formulated as the following optimization problem:

$$find\ c=(c_1,c_2,...,c_{s_k}),\ \ which\ \min \omega_1 T+\omega_2\sigma^2$$

$$\sum_{i=1}^{k} x_{ij}=1 \qquad j=(1,2,...,n)$$

$$c_i \cap c_j=\varnothing \quad i,j=(1,2,...,k)\ i\neq j$$

$$\sum_{i=1}^{k} c_i=n \qquad ,w(s_i)<w_{\max}$$

$$T_{i,j}<T_{\max} \qquad ,\omega_1+\omega_2=1,\omega_1,\omega_2\geq 0$$

Where $\omega_{\max}$ is the maximum allowable load, $T_{\max}$ is the maximum allowable transmission time.

3.2.N-1 Security Criterion

Building on the model proposed in Section 3.1, this paper integrates the N-1 security criterion in power systems into the edge server deployment strategy. This criterion ensures that if an edge server fails, the traffic it was handling can be seamlessly transferred to a pre-designated backup server, while meeting the delay and maximum load requirements. The conditions are as follows:

- Network Paths: There must be a path for transferring the traffic, i.e., the base station and the neighbouring server must have a connection and the delay constraint is satisfied.
- Neighbouring Server Capacity: The neighbouring server must have sufficient spare capacity to support the transferred traffic.

When any edge server $s_n(1\leq n\leq k,n\in Z)$ fails, the connected base stations c_{nm} require sufficient spare server capacity to handle the traffic of c_{nm}.The following conditions must be met:

$$T_{i,c_{nm}}<T_{\max}\beta\ ,\ w^{'}(s_i)<w_{\max}\ ,T_{i,c_{nm}}=\frac{d(s_i,c_{nm})}{r_{ic_{nm}}}$$

Where β is a scaling factor, $w^{'}(s_i)$ represents the updated load of the neighboring s_i serve rafter accommodating the additional traffic.

3.3. Fault model

In this study, we adopt the 0-1 failure model to describe the failure behavior of edge servers. This model assumes that an edge server can only be in one of two states at any given time: normal operation or complete failure. Complete failure means the server cannot provide any computing or communication services, while normal operation indicates that the server can fully perform its functions. To simulate the failure behavior of edge servers, we categorize failures into two types: aging failures and accidental failures.

Aging failures are caused by device aging or long-term use, and their lifetimes follow an exponential distribution defined by the probability density function $f(t)=\lambda e^{-\lambda t}$ .Using the inverse function $t=-\frac{1}{\lambda}\ln(1-U)$ (where U is a random number between 0 and 1), we generate the simulated lifetime of each edge server. During the simulation, we compare the simulated lifetime with a predefined lifetime threshold. If the simulated lifetime is shorter than the threshold, the edge server is considered to have experienced an aging failure. Accidental failures are caused by external factors such as hardware damage or network interruptions and are independent of time. When either an aging failure or an accidental failure occurs, the edge server is deemed to have failed.

4.Clustering Algorithms for the Edge Server Deployment Problem based on the N-1 Security Criterion

4.1 Improved K-means algorithm

As a typical distance-based clustering algorithm, K-Means usually uses distance as a similarity evaluation metric. Its characteristics include high

similarity between objects within the same cluster and low similarity between objects in different clusters. In the context of Edge Server (ES) deployment problems, the K-Means algorithm is also applicable to solve ES location problems. However, the traditional K-Means algorithm has the following drawbacks in the application of the model proposed in this paper:

- Random initialisation of cluster centres, leading to significant variations in clustering results
- Potential for edge servers at cluster centres to be loaded beyond their maximum capacity, leading to infeasible solutions
- Susceptibility to local optima
- Possibility of edge server failures leading to reduced system reliability

Therefore, in this paper, an improved K-means algorithm combined with the N-1 security criterion for edge server placement is used to overcome the aforementioned shortcomings and improve system reliability and stability.

In this section, we use the weighted density method to select the initial base stations as the initial cluster centres for the K-Means algorithm. The weighted density method allows us to obtain initial centres that are geographically evenly distributed, have lower loads, and have higher density. Step 1: Perform max-min normalisation on the base station load to facilitate better clustering analysis. Step 2: Calculate the density of each base station based on its geographical location. The density is calculated based on the number of base stations within a certain neighbourhood. Step 3: Select the point with the highest weighted density as the first centre. The weighted density is a function of both density and normalised load, ensuring that base stations with lower loads and higher densities are prioritised. Step 4: Select the remaining centres by calculating the minimum distance of each base station from the centres already selected. The weighted distance is then calculated, multiplied by the density to give a new weighted density and the base station with the highest weighted density is selected as the new centre.

Considering the maximum load constraints of the edge servers in this paper, the clustering results obtained by the K-Means algorithm may lead to scenarios where the edge servers at the cluster centres are overloaded, making the solution infeasible. To address this issue, this paper proposes a location update formula:

$$x_{new} = x + r_1(x_{random} - x) + r_2(x_{best} - x)$$

The variables r_1 and r_2 are random factors, both in the range $[0,1]$. This update formula aims to adjust the cluster centers to avoid overload and ensure feasible solutions.

4.2. Edge server deployment method based on N-1 security criteria

Based on the clustering algorithm, an edge server deployment plan can be developed. To enhance system robustness, the N-1 security criterion should be followed. In the event of a single edge server failure, affected base stations must be reassigned to a backup server. If the backup server meets the delay and maximum load constraints, the scheme is considered feasible under the N-1 criterion. This approach ensures smooth network operation in case of a server failure, improving the overall reliability and security of the network.

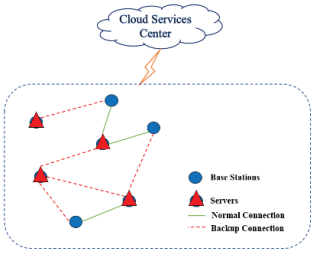

Figure 1. Edge server deployment based on N-1 security guidelines

As shown in Figure 1, blue circles represent base stations, red triangles represent base stations with edge servers, green solid lines represent normal connections between base stations and edge

servers, and red dashed lines represent connections to backup edge servers. If the normal edge server fails, base stations will connect to their designated backup edge servers to continue operations.

5.Performance Evaluation

In our experiments, we validated these methods using real data from China Telecom in Shanghai. The dataset includes the locations of over 1800 mobile base stations in Shanghai and service request time information from the mobile smart terminals they serve. We focus exclusively on the load balancing of edge servers and the average access delay of the entire edge system. All experiments and simulations were performed on the same general-purpose computer. The experiments and simulations were conducted on a general-purpose laptop equipped with a 12th Gen Intel(R) Core(TM) i7-12700H processor (2.3 GHz) and 16 GB of RAM. All programs and simulations were implemented using Python 3.11.3, along with the required libraries and packages

5.1. Performance Evaluation of the Improved K-Means Algorithm

In this section, we will evaluate the performance of the improved K-Means algorithm in traditional edge server deployments using the above data and compare it with other algorithms.

To evaluate the performance of the improved K-Means algorithm, the number of edge servers in this paper is set between 50 and 200. The performance of the algorithm is evaluated with different numbers of edge servers. To avoid randomness in the experiments, all results are averaged over multiple runs. We compare our algorithm with the Top-K, Random, K-means (Ye et al.,2023) and DBCA (Li et al.,2022) algorithms in terms of delay and load balancing.

Figure 2 illustrates the variation of the total objective function value, which is composed of the total delay and load variance, as the number of edge servers increases for different algorithms. The experimental results show that the improved K-Means algorithm consistently outperforms other algorithms in terms of the total objective function value across different numbers of edge servers. This suggests that our improved algorithm offers superior overall performance in reducing both total delay and load variance.

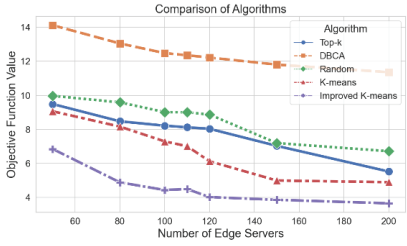

Figure 2. Objective function values for different number of edge servers.

As can be seen from the curve of the improved K-Means algorithm in Figure 2, the total objective function value decreases as the number of edge servers increases. However, once the number of edge servers reaches 100, the curve tends to stabilise with minimal fluctuations in the total objective function value. Therefore, considering the cost of deploying edge servers, selecting 100 edge servers appears to be a reasonable option.

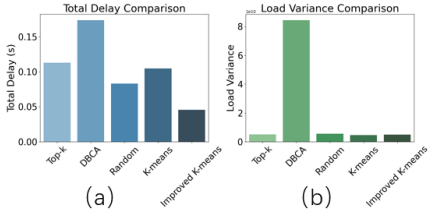

Figure 3. Total Delay and load Variance at 100 Edge Servers for Different Algorithms.

Figure 3(a) shows the variation curve of the total objective function composed of total delay and load variance. However, in real-world scenarios, it is challenging to simultaneously minimise load balancing and access delay. To evaluate the performance of the improved algorithm in terms of both total delay and load balancing, we use the example of 100 edge servers and compare the total delay and load variance of different

algorithms. Figure 3(b) shows that the improved algorithm significantly outperforms other algorithms in terms of total delay, demonstrating superior performance in delay reduction. Figure 4 shows that the improved K-Means algorithm performs similarly to the other algorithms in terms of load balancing, with only the DBCA method having a significantly higher load variance than the others.

5.2. Edge Server Deployment under the N-1 Security Criterion

This simulation compares edge server deployment methods using the N-1 security criterion with traditional methods in terms of reliability and stability. The N-1 security constraints (Section 3.2) were integrated into an improved K-means algorithm (Section 4). Although this approach increases the total objective function value, the improvement in system reliability justifies the trade-off. Monte Carlo simulations are then performed to compare system reliability under two scenarios: traditional deployment (without N-1 security standards) and deployment based on the N-1 security criterion.

In a traditional Edge Server (ES) deployment, if one ES fails, all base stations under that ES cannot operate normally. Therefore, system reliability is defined as the ratio of the number of base stations operating normally to the total number of base stations. In an N-1 ES deployment, if one ES fails, the base stations can continue to operate normally via backup edge servers. Only if both the primary and backup ESs fail will the base stations be unable to operate normally. In this Monte Carlo simulation, the experiment was run 5000 times, assuming that each server operates independently. The final system reliability is the average of all the simulation results.

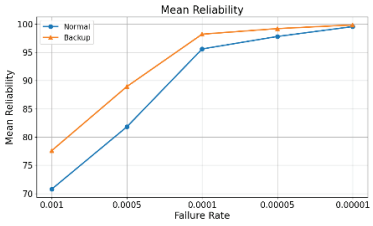

Figure 4. Mean System Reliability Under Different Failure Rates of Edge Servers.

The Figure 4 gives the system reliability under different failure rates for the traditional edge server deployment method and the edge server deployment method based on the N-1 security criterion.

In Figure 4, the horizontal axis represents the edge server failure rate, while the vertical axis represents the mean system reliability. It can be seen that the system reliability of the N-1 based deployment method consistently exceeds that of the traditional method across different failure rates.

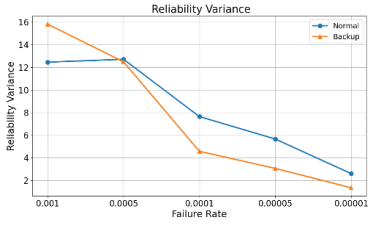

Figure 5. Variance of System Reliability Under Different Failure Rates of Edge Servers.

Although the mean system reliability of both methods converge as the edge server failure rate decreases, the challenge of improving reliability increases as the reliability approaches1. Furthermore, under ideal conditions with a failure rate of 0, the system reliability of both methods should theoretically be equal. In Figure 5, the horizontal axis represents the failure rate of edge servers, while the vertical axis represents the variance of system reliability. It can be observed that, across different failure rates, the system reliability variance of the edge server deployment method based on the N-1 security criterion is consistently higher than that of the traditional method. Therefore, the edge server deployment

scheme based on the N-1 security criterion exhibits stronger risk resilience and enhanced security.

6.Conclusion

This paper proposes an edge server placement method based on the N-1 security criterion to address issues with the traditional K-Means algorithm, such as random initialization and server overload. Experimental results demonstrate that this algorithm outperforms DBCA, K-Means, Top-K, and Random algorithms in reducing communication latency and balancing load. To handle potential edge server failures, this paper introduces the N-1 security criterion for power systems in edge server deployment. Comprehensive experimental data shows that the N-1 security criterion deployment strategy not only improves the overall reliability of the system, but also reduces the security risk potential and provides a more stable service assurance.

Future research will further integrate the N-1 security criterion with optimization algorithms to enhance system reliability. Additionally, we will focus on the issue of task offloading between edge servers in dynamically changing computational demand scenarios. The design of online algorithms for dynamic task offloading will be a key area of future work.

Acknowledgement

This research was funded by the National Natural Science Foundation of China (Grant No.72371013 & 71971013) and the Fundamental Research Funds for the Central Universities (YWF-23-L-933). The study was also sponsored by the Teaching Reform Project of Beihang University.

References

E. Ahvar, A. -C. Orgerie and A. Lebre, "Estimating Energy Consumption of Cloud, Fog, and Edge Computing Infrastructures," in *IEEE Transactions on Sustainable Computing*, vol. 7, no. 2, pp. 277–288, 1 April-June 2022.

Vaibhav Tiwari, Chandrasen Pandey, Abisek Dahal, Diptendu Sinha Roy, Ugo Fiore. "A Knapsack-based Metaheuristic for Edge Server Placement in 5G Networks with Heterogeneous Edge Capacities," in Future Generation Computer Systems, vol. 153, pp. 222–233, 2024.

J. K. Vaurio, "Treatment of General Dependencies in System Fault-Tree and Risk Analysis," in *IEEE Transactions on Reliability*, vol. 51, no. 3, pp. 278–287, Sept. 2002.

W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges," in IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, Oct. 2016.

A. Asghari and M. K. Sohrabi, "Multiobjective Edge Server Placement in Mobile-Edge Computing Using a Combination of Multiagent Deep Q-Network and Coral Reefs Optimization," in IEEE Internet of Things Journal, vol. 9, no. 18, pp. 17503–17512, 15 Sept. 2022.

GUO Fei-yan, TANG Bing. "Mobile Edge Server Placement Method Based on User Latency-aware," in Computer Science, vol. 48, no. 1, pp. 103–110, 2021.

B. Bahrami, M. R. Khayyambashi and S. Mirjalili, "Multiobjective Placement of Edge Servers in MEC Environment Using a Hybrid Algorithm Based on NSGA-II and MOPSO," in IEEE Internet of Things Journal, vol. 11, no. 18, pp. 29819–29837, 15 Sept. 2024.

M. Jia, J. Cao and W. Liang, "Optimal Cloudlet Placement and User to Cloudlet Allocation in Wireless Metropolitan Area Networks," in *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 725–737, 1 Oct.-Dec. 2017.

S. K. Kasi et al., "Heuristic Edge Server Placement in Industrial Internet of Things and Cellular Networks," in *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10308–10317, 1 July 2021.

J. Zhang, J. Lu, X. Yan, X. Xu, L. Qi and W. Dou, "Quantified Edge Server Placement With Quantum Encoding in Internet of Vehicles," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9370–9379, July 2022.

Ye, H., Cao, B., Liu, J. et al. "An Edge Server Deployment Method Based on Optimal Benefit and Genetic Algorithm," in *Journal of Cloud Computing*, vol. 12, article no. 148, 2023.

Li, W., Chen, J., Li, Y. et al. "Mobile Edge Server Deployment Towards Task Offloading in Mobile Edge Computing: A Clustering Approach," in *Mobile Networks and Applications*, vol. 27, pp. 1476–1489, 2022.