Problems of Information Technology (2026), vol. 17, no. 1, 3-13

Available online at www.jpit.az

17 (1)
2026

Multi-objective resource allocation in edge computing using TF-IDF scheme

Rasim Alguliyev[a], Rashid Alakbarov[b], Ramiz Aliguliyev[c]

[a,b,c] Institute of Information Technology, B. Vahabzade str., 9A, AZ1141 Baku, Azerbaijan

[a] r.alguliev@gmail.com; [b] rashid.alakberov@gmail.com; [c] r.aliguliyev@gmail.com

[a] https://orcid.org/0000-0003-1223-7411; [b] https://orcid.org/0000-0002-7566-371X; [c] https://orcid.org/0000-0001-9795-1694

ARTICLE INFO

Keywords:

Cloud computing
Edge computing network
Resource allocation
Multi-objective optimization
Pareto front
Usage frequency
Number of users
TF-IDF scheme

ABSTRACT

Recently, the rapid increase in the number of mobile phones and IoT devices connected to networks has led to a decrease in the throughput capacity of Internet network channels and delays in delivering processed data to users. To eliminate these delays in network channels, edge computing systems are widely used. Edge computing systems enable data to be processed on computing devices close to the users, without being sent to remote cloud servers. These systems reduce network latency, allow fast real-time data processing, and enable a reduction in the costs of services. This paper proposes a model of an edge computing system based on data processing centers. Two criteria, resource usage frequency and number of users, are taken as the basis for the allocation of cloud resources in edge computing networks. For balancing these criteria, the TF-IDF scheme is used. The proposed approach is formalized as a two-objective optimization model. An algorithm for selecting the optimal solution from the Pareto-optimal front is employed.

1. Introduction

The rapid increase in the number of users connected to the Internet leads to network congestion, resulting in delays in delivering processed data to users. To address these issues–such as ensuring fast data processing, reducing delays in communication channels, and more–edge computing systems are widely utilized. Edge computing systems (e.g., edge computing, cloudlets, fog computing, etc.) place computing devices close to users, providing them with necessary computing and storage resources. Since edge computing is closer to the user, less time is required for data processing and for transmitting the results to the user. This technology allows the distance between the user and the servers hosting cloud resources to be shortened. When a user requires a specific resource, it is obtained from a device located nearby, which facilitates an increase in processing speed. The primary objective of edge computing is to enhance system efficiency by processing data in proximity to users and to reduce the volume of data transmitted to remote cloud servers. Processing applications of multiple users on remote servers and transmitting results back to them is not considered efficient. Therefore, in recent years, the use of edge computing systems has become particularly relevant for solving these problems.

Edge and cloud computing are complementary technologies, and their combined use provides significant advantages across a wide range of applications. Edge computing is presented as a paradigm that extends cloud computing platforms. In these systems, processing devices are located in areas close to users. Edge computing is an emerging

Received 23 September 2025, Received in revised form 24 November 2025, Accepted 25 December 2025
https://doi.org/10.25045/jpit.v17.i1.01

paradigm that provides users with services at relatively low costs and enables faster execution of computing tasks. By utilizing edge computing networks, Internet Service Providers (ISPs) enhance the quality of Internet services delivered to users while reducing costs. The analysis of processes within the network infrastructure and the application of mathematical-technological methods not only reduce service-related expenses but also significantly improve service quality metrics.

In centralized, traditional cloud computing systems, computing tasks are executed on remotely located cloud servers, leading to network congestion and delays in delivering results to users. Simultaneously, the provision of these services is associated with high operational costs.

In recent years, to address the aforementioned issues, edge computing systems–built upon the infrastructure of cloud computing systems–have been widely adopted. In edge computing systems, computing tasks are performed on devices located close to users. This approach reduces network latency, enables more efficient use of bandwidth, and allows for rapid real-time data processing.

Edge computing technologies facilitate the development of innovative applications across various domains by ensuring the prompt and efficient processing of data.

The implementation of a hybrid model that integrates edge and cloud computing systems allows for simultaneous utilization of the advantages of both technologies. In this model, the devices executing services are located closer to users within the edge computing environment, enabling users to experience higher quality of services (QoS).

2. Related work

Somula and Sasikala (2018) proposed to use edge computing systems (cloudlets) located closer to mobile devices to overcome the limitations of mobile device batteries and computing resources. It is noted that the limited computing and storage resources of mobile devices, as well as network-induced delays, create challenges in solving complex computational tasks. To address these issues, the execution of tasks on edge computing servers located near mobile users has been analyzed in (Sharma & Prachi, 2017; Abusaimeh, 2022). Several studies have investigated the potential of edge computing systems deployed between users and cloud servers to reduce energy consumption on mobile devices and minimize network latency. It has been shown that placing computing devices closer to users in edge computing systems ensures faster delivery of results to the end-users (Yuyi et al., 2017; Abbas et al., 2018; Mukherjee et al., 2021). In (Zhao & Zhou, 2019), the impact of executing computational tasks on virtual machines located near mobile devices using new Mobile Edge Computing (MEC) technologies on the reduction of network latency was studied. Meanwhile, Alakbarov (2021) demonstrated that executing tasks requiring high computing and storage resources on nearby cloudlets helps save energy on mobile devices and reduces delays in communication channels. In the study presented by Akomolafe and Abodunrin (2017), the execution of users' computational tasks on nearby cloudlets, the reduction of the number of communication channels between users and cloudlets to minimize latency, and the assurance of reliable network operation were investigated. In (Nayyer et al., 2018), it was proposed to utilize cloudlet-based computing clouds to eliminate delays caused by network congestion.

Some studies, such as (Jia et al., 2015), explored the formation of network infrastructure based on the placement of cloudlets near access points of wireless city-scale networks, and investigated the prediction of which base stations should host cloudlets. In (Ceselli et al., 2017), structural issues of cloudlet-based networks were examined, including the establishment of cloudlet networks in wireless city-scale networks, as well as determining the locations and number of cloudlets. In (Alguliyev et al., 2023; Alguliyev & Alakbarov, 2023), methods and algorithms were proposed for the optimal placement of cloud resources within cloudlets, considering user activity, the usage frequency of cloud resources, and the physical distance between users and cloud resources. In (Madni et al., 2019), a Multi-Objective Cloud Search Optimization (MOCSO) algorithm was presented to solve the problem of optimal resource planning in cloud computing environments. The primary goal of resource planning was stated as reducing user costs and enhancing system performance, while also emphasizing the importance of minimizing execution time to maximize cloud providers' revenue. In (El-Sayed et al., 2017), the impact of migrating workloads from centralized cloud servers to servers closer to users in edge computing systems on processing speed and network latency reduction was analyzed. The study extensively examined edge computing systems and emphasized their advantages over traditional cloud computing systems.

In (Rajak & Shukla, 2018), detailed information and analysis were provided on the architectures and models of cloud and mobile cloud computing systems. Additionally, the service models and application areas of cloud and mobile cloud computing systems were explored, highlighting both their strengths and limitations. In (Nandal et al., 2021), it was emphasized that, against the backdrop of exponential growth observed in cloud computing environments, users' demand for additional services and enhanced results increases, turning load balancing into a critical issue. The study analyzed workload distribution across multiple nodes to optimize system performance. Various load balancing algorithms were presented to ensure more efficient use of resources, with comparative analyses based on metrics such as average response time, processing cost, and data service time. In (Mukherjee et al., 2024), the significance of efficient resource management in cloud computing systems was highlighted as a key factor in providing high-quality service to users. Various approaches to managing cloud resources were analyzed. In addition to cloud computing, resource management in edge and fog computing systems was discussed, with challenges in managing cloud resources being emphasized. In (Mukherjee et al., 2024), the issue of optimally placing computing devices within the network in edge computing systems to reduce their energy consumption was investigated. To achieve energy savings, an energy-aware cloud placement method called ECPM was proposed. Experimental results demonstrated that the proposed method effectively reduces energy consumption. In (Tian & Zhong, 2022), three edge computing paradigms–fog computing, multi-access edge computing (MEC), and cloudlet systems–were examined in terms of their application areas, architectures, and differences. Additionally, their usage requirements, benefits, and mechanisms were discussed in the context of IoT and 5G wireless systems. The study also addressed challenges in edge computing systems, their potential solutions, and future development directions. In (Alakbarov, 2021), the execution of tasks requiring high computing and storage resources on cloudlets located close to users was analyzed as a means to save energy on mobile devices and reduce delays in communication channels.

In (Malik et al., 2021), a system approach called EFFORT was proposed for offloading workloads to cloud servers. The approach was shown to partially address energy consumption issues on smartphones and difficulties related to rapid data processing. Experimental studies demonstrated that the EFFORT approach outperforms existing methods in terms of energy consumption and data processing speed.

3. Problem statement

The primary problem addressed in this paper is the organization and optimization of telecommunication services (Internet services) in a manner that ensures faster processing of computational tasks, minimizes network latency, and reduces the costs incurred by users while accessing services. Overall, the aim is to decrease nationwide service-related expenses, limit the outflow of foreign currency, and contribute to the development of the national digital economy. To achieve this, it is necessary to establish a more efficient network environment between users and the Internet.

To create the aforementioned network environment, the use of edge computing technologies is proposed. The resulting network infrastructure is located closer to users, and the formation of the edge computing system leverages data processing centers (DPCs). The main function of this network environment is to store cloud resources (e.g., web pages, images, videos, music files, applications, etc.) that are most frequently accessed by users in DPCs located nearby. This approach reduces the cost of telecommunication services within the country, thereby minimizing expenses both at the governmental level and for individual citizens.

At the same time, the creation of such infrastructure requires additional financial investments, and the allocated budget is limited. Due to budgetary constraints, the number of DPCs that can be established within the network infrastructure is also limited. Therefore, a technological project must be developed within the available budget, determining the number of DPCs to be established, as well as the volume of their computing and storage resources. Given the limited financial and technical resources, it is not feasible to place all cloud resources used by national users on the international Internet entirely within the established DPCs. In this context, considering the technical capabilities of the network infrastructure, it is appropriate to select the most frequently used cloud resources and place them in the memory resources of the DPCs. Bringing cloud resources into the national network infrastructure allows for a reduction in international telecommunication costs. Moreover, it is necessary to select not only the resources intensively used by individual users but also those cloud resources that have high usage frequency and a large number of users nationwide.

In the newly established network infrastructure, both newly deployed DPCs and the unused memory resources of existing DPCs within the country are intended to be utilized. Due to the limited computing and storage resources of the established network, it is not possible to host all cloud resources available on the international network within the local infrastructure. Therefore, the implementation of an optimal selection mechanism is required. The proposed edge computing system is formed between Network Service Provider (NSP), ISP, and mobile users. The model for placing cloud resources with high usage frequency and a large number of users in DPCs located close to users within the Internet environment is illustrated in Figure 1.

As shown in the figure, a national network infrastructure based on DPCs has been established close to users. The created network infrastructure provides the following capabilities:

- Increased access speed – ensuring faster access for users to cloud resources;
- Enhanced security – enabling higher levels of information security within the system;
- Proximity of cloud resources to users – reducing costs incurred during service delivery;
- Other additional advantages, etc.

Within the country, users access cloud resources through the NSP. Information regarding the cloud resources accessed by users is collected in log files on the provider’s servers. Analysis of these log files allows identification of which cloud resources are most frequently requested by users.

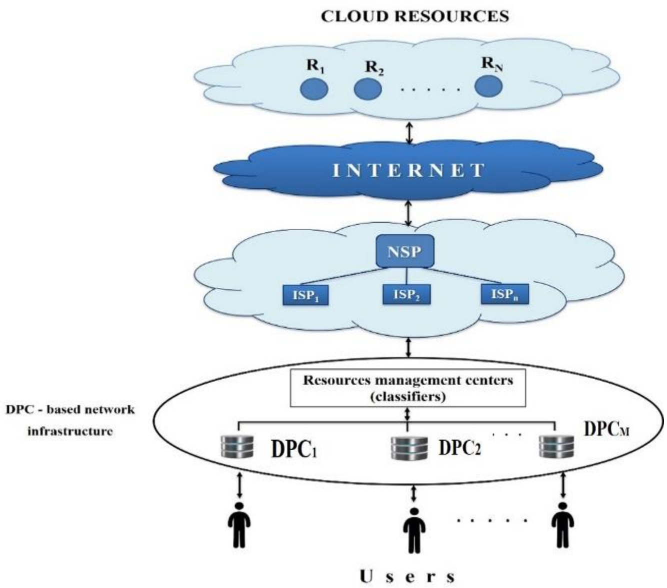

Fig. 1. Placement of cloud resources in DPCs within the Internet environment

Based on monitoring results, cloud resources with high usage frequency are placed in the memory resources of the newly established network infrastructure. Consequently, these resources are loaded from the memory of processing centers that are closer to users within the national network infrastructure. This approach reduces service costs and enables faster access to data.

Through the analysis of log files collected by the NSP, it is possible to identify the cloud resources most frequently accessed by users within the country. Monitoring conducted over defined time intervals (hourly, daily, monthly, yearly, etc.) allows for the calculation of usage frequency for each cloud resource. If the usage frequency of a particular cloud resource exceeds a minimum threshold set by the administrator, the resource is considered active. Cloud resources whose activity level falls below the defined threshold are not recommended to be placed in the memory resources of the national network infrastructure.

Active resources identified by the NSP are regarded as relevant for the country. Based on the analysis of these resources, statistical tables are generated, and a separate statistical database is created for each active resource. By analyzing daily, weekly, monthly, and yearly histograms of active resources, it is possible to determine which resources are most important for users nationwide.

For each active cloud resource, annual reports are used to generate histograms reflecting monthly usage frequencies. Analyzing these histograms allows for the determination of the usage dynamics of each resource. This approach provides an assessment of resource usage both for individual months and over the course of the entire year. Subsequently, detailed analysis of resource utilization throughout the year is conducted.

According to the established conditions, it is not possible to place all cloud resources whose activity levels exceed the defined threshold in the memory resources of the national network infrastructure. The primary reason for this limitation is the restricted memory resources of the national infrastructure, which are formed in accordance with the allocated budget. Therefore, active cloud resources must be analyzed, and the resources most frequently accessed by users should be selected for placement within the national network infrastructure.

Placing cloud resources with high activity levels and a large number of users in the memory resources of edge computing systems is considered appropriate. This approach ensures the principle of social equity by prioritizing resources based on the number of users.

For example, consider two active cloud resources. Assume that both resources exceed the defined activity threshold and have similar activity levels. However, over the course of a year, the first resource was accessed by 1,000 users, while the second resource was accessed by 3,000 users. In this case, placing the first resource in the memory resources of the national network infrastructure is not recommended. Instead, the second cloud resource, which has a high activity level and a larger number of users (3,000 users), should be placed in the system's memory resources, representing a more appropriate choice.

The interaction between users and cloud resources is organized as follows:

- During a defined time interval (e.g., a week, a month, a year), log files in the Resource Management Center (RMC) are analyzed to identify the most frequently accessed cloud resources by Internet users nationwide.
- The memory requirements of the accessed cloud resources are compared with the available memory resources of the established network.
- The available memory resources of the network are significantly smaller than the total volume of used cloud resources. The budget allocated for network deployment does not allow for placing all used cloud resources in the memory storage system of DPCs.
- Therefore, the most frequently used active cloud resources (those exceeding the access threshold set by the network administrator) are selected and placed in the memory storage systems of DPCs.
- Once active resources are identified, the number of users accessing each resource is analyzed. If the number of users is below a minimum threshold defined by the administrator, placement of that resource in the national network infrastructure is not required.
- Active resources are selected to prioritize those accessed by a larger number of users.
- Resource management is carried out based on two criteria: usage frequency and number of users. Active cloud resources selected according to these criteria are placed in DPCs, ensuring the principle of social equity (prioritizing resources with higher user counts).
- Active resources placed in DPCs are re-evaluated after a certain period (e.g., weekly, monthly, yearly). If any resource no longer meets the predefined usage frequency and user count criteria, it is considered a passive resource and removed from the DPC memory storage system.
- The RMC identifies new active cloud resources and places them in the newly freed memory resources.

Thus, the proposed model functions as a dynamic system, optimizing the placement of cloud resources in DPCs based on user behavior.

4. Model for optimal allocation of cloud resources in edge computing network

Considering the above, a mathematical model is proposed to ensure the optimal allocation of cloud resources with high usage frequency and a large number of users within the memory resources of the national network infrastructure.

Let's introduce the following notations:

- $\mathrm{U}(T) = \{U_i(t) | i \in I\}$ – a set of users, where $I = \{1,2,\dots,n\}$ is the set of users' indices;
- $\mathrm{R}(T) = \{R_j(t) | j \in J\}$ – a set of resources, where $J = \{1,2,\dots,m\}$ is the set of resources' indices;
- $\mathrm{S} = \{S_q | q \in K\}$ – a set of servers, where $K = \{1,2,\dots,k\}$ is a set of servers' indices;
- $\mathrm{v}(T) = \{v_j(t) | j \in J\}$ – volume of resources;
- $\mathrm{V} = \{V_q | q \in K\}$ – memory capacity of servers;
- $M_{ij}(T)$ – number of requests of the ith user (U_i) to the j-th resource (R_j) during the time period T;
- $M_i(T)$ – total number of requests of the i-th user (U_i) to all resources (during the time period T): $M_i(T) = \sum_{j=1}^{m} M_{ij}(T)$;
- $n_j(T)$– the number of users applying to the j th resource (R_j) (during time period T),

where n denotes the number of users, m is the number of resources, and k is the number of servers. v_j denotes the volume of the jth resource (R_j), and V_q is the storage capacity of the qth server (S_q).

To calculate the importance degree of the jth resource (R_j) for the ith user (U_i)), the TF-IDF scheme is used:

$$w_{ij}(T) = \begin{cases} f_{ij}(T)\log\left(\frac{n(T)}{n_j(T)}\right) & \text{if } n_j(T) \neq 0 \\ 0 & \text{if } n_j(T) = 0 \end{cases} \quad (1)$$

here

$$f_{ij}(T) = \frac{M_{ij}(T)}{M_i(T)},\ i \in I, j \in J \quad (2)$$

is the relative frequency of the jth resource (R_j) by the ith user (U_i).

Thus, the problem is formulated as follows: the resources to be placed in the memory resources of the constructed system should be those that have both a high relative processing frequency ($f_{ij}(T)$)) and are accessed by a large number of users, i.e., for which $n_j(T)$ is maximized. In other words, the value of $\log\left(\frac{n(T)}{n_j(T)}\right)$ should be minimized.

For simplicity, the time variable T will be omitted in the rest of the paper without confusion. Taking the above into account, the problem is formalized as follows:

$$\log(n/n_j) \to \min \quad (3)$$

$$f_{ij} \to \max \quad (4)$$

Considering the limited memory capacity of the servers, the following condition must be satisfied:

$$\textstyle\sum_{j\in J} x_j v_j \leq \sum_{q\in K} V_q \quad (5)$$

here

$$x_j = \begin{cases} 1 & \text{if resource } R_j \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Equations (3)-(4) are regarded as a multi-objective optimization problem. Multi-objective optimization is the process of simultaneously optimizing two or more conflicting objective functions. In such problems, different objectives may conflict with each other; that is, improving one objective may lead to the deterioration of another. When solving multi-objective optimization problems, there is no guarantee that a single solution will optimize all objective functions simultaneously.

Therefore, the concept of Pareto optimality is applied in solving such problems. A solution is called Pareto optimal if none of the objective functions can be improved without worsening at least one of the other objective functions. In other words, a solution is considered Pareto optimal if there exists no other solution that improves some objectives without degrading others. In multi-objective optimization, the Pareto front represents the set of Pareto optimal solutions. When these solutions are plotted graphically, the resulting curve is known as the Pareto frontier (Multi-objective optimization with Pareto front, 2025; Gunantara, 2018).

Considering the above, the following iterative algorithm is proposed to solve the formulated problem:

Step 1. Let $p = 1$. Initially, the set of resources is taken as $\mathrm{R}_0 = \mathrm{R}$ and the set of selected resources is defined as $J_0 = \{\emptyset\}$.

Step 2. For each $i \in I$, a set of Pareto solutions of problem (3)-(4) is found, and resources corresponding to this set of solutions are selected. Let's denote the set of selected resources by P_p and the set of their indices by J_p: $\mathbf{P_p} = \{R_r | r \in J_p\}$.

The number of selected resources is denoted by $m_p = |J_p|$, where $|J_p|$ is the number of elements of the set J_p.

Step 3. Let $J_p = \bigcup_{r=p-1}^{p} J_r$. The following condition is checked for the selected resources:

$$\sum_{j \in J_p} v_j \leq \sum_{q \in K} V_q \qquad (7)$$

Step 4. $\mathrm{R}_p = \mathrm{R}_{p-1} \setminus \mathrm{P}_p$ (eliminating Pareto solutions from the set of resources). If condition (7) is provided, then we accept $m := m - m_p$ and $p := p + 1$ and proceed to Step 2, otherwise the process stops.

The final Pareto optimal set of resources (P) will be the union of the set of resources (P_p) found at each step:

$$\mathrm{P} = \bigcup_{r=1}^{p} \mathrm{P}_r$$

Example. Suppose the number of users is 6 ($n = 6$), the number of resources is 20 ($m = 20$), and the number of servers is 3 ($k = 3$). Table 1 presents the number of resource requests by users. Table 2 shows the number of resources and memory capacity of servers.

Table 1. Number of resource requests by users (M_{ij})

User	Resource																			
	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	R_{16}	R_{17}	R_{18}	R_{19}	R_{20}
U_1	0	40610	0	50608	0	48383	0	28780	0	0	0	72433	0	6313	0	29531	42354	0	20538	72086
U_2	0	98004	51829	0	51110	0	30596	0	11385	20773	43840	0	0	52309	76127	71165	0	48048	0	56386
U_3	87541	0	0	94647	0	0	25140	33621	0	0	0	49166	68323	0	95443	87702	56600	56407	17632	14451
U_4	19506	91015	1658	70219	0	74839	0	0	0	0	203	0	0	10383	97284	79598	0	51911	15388	67866
U_5	76378	93925	31859	39897	0	0	57077	38945	60468	0	58616	0	92767	51963	0	0	0	0	57417	0
U_6	60657	60533	22052	0	0	0	32241	0	0	0	0	40147	2277	0	31926	0	0	0	0	0

Table 2. Number of resources and memory capacity of servers

Resource ($v_j, j \in J$)																			
R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	R_{16}	R_{17}	R_{18}	R_{19}	R_{20}
22	58	75	64	35	67	34	78	71	26	32	83	44	32	47	28	10	11	42	52

Servers ($V_q, q = 1,2,3$)		
V_1	V_2	V_3
275	236	151

Based on the data provided in Table 1, the calculated values of the parameters f_{ij}, n_j and $\log(n/n_j)$ are presented in Table 3.

Step 1. $p = 1, J_0 = \{\emptyset\}$.

Step 2. A set of Pareto solutions is found using for Table 3. To this end, first, a set of Pareto solutions is found for each pair ($\log(n/n_j)$, f_{ij}) ($i \in I, j \in J$):

- $i = 1$: $P_{11} = \{R_2, R_{12}, R_{20}\}$; Figure 2 presents the set of Pareto solutions for the pair $\log(n/n_j)$ with $U_1(f_{1j})$. As the graph shows, the set of Pareto solutions found is $P_{11} = \{R_2, R_{12}, R_{20}\}$. Analogously, with other users, the set of Pareto solutions is found for pair $\log(n/n_j)$ and $U_1(f_{1j})$.
- $i = 2$: $P_{12} = \{R_2\}$;
- $i = 3$: $P_{13} = \{R_2, R_{15}\}$;
- $i = 4$: $P_{14} = \{R_2, R_{15}\}$;
- $i = 5$: $P_{15} = \{R_2\}$;
- $i = 6$: $P_{16} = \{R_1, R_2\}$.

The set of Pareto solutions P_{1i} will be the union of sets $i = 1, \ldots, 6$:

$\mathrm{P}_1 = \bigcup_{i=1}^{6} P_{1i} = \{R_1, R_2, R_{12}, R_{15}, R_{20}\}$.

$J_1 = \{1, 2, 12, 15, 20\}$, $m_1 = |J_1| = 5$.

Step 4.

$J := J \setminus J_p = J \setminus J_1 = \{3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 16, 17, 18, 19\}$.

Since condition (7) is satisfied, the number of resources is updated as follows: $m := 20 - m_1 = 20 - 5 = 15$, $p := 1 + 1 = 2$, and the algorithm proceeds to **Step 2**.

Table 3. f_{ij}, n_j, $\log(n/n_j)$ (case $p = 1$)

Resource	f_{ij} $(i \in I; j \in J)$						n_j	$\log(n/n_j)$
	U_1	U_2	U_3	U_4	U_5	U_6		
R_1	0	0	0.1275	0.0336	0.1158	0.2428	4	0.1761
R_2	0.0987	0.1602	0	0.157	0.1425	0.2423	5	0.0792
R_3	0	0.0847	0	0.0029	0.0483	0.0883	4	0.1761
R_4	0.1229	0	0.1378	0.1211	0.0605	0	4	0.1761
R_5	0	0.0836	0	0	0	0	1	0.7782
R_6	0.1175	0	0	0.1291	0	0	2	0.4771
R_7	0	0.05	0.0366	0	0.0866	0.1291	4	0.1761
R_8	0.0699	0	0.049	0	0.0591	0	3	0.3010
R_9	0	0.0186	0	0	0.0917	0	2	0.4771
R_{10}	0	0.034	0	0	0	0	1	0.7782
R_{11}	0	0.0717	0	0.0004	0.0889	0	3	0.3010
R_{12}	0.176	0	0.0716	0	0	0.1607	3	0.3010
R_{13}	0	0	0.0995	0	0.1407	0.0091	3	0.3010
R_{14}	0.0153	0.0855	0	0.0179	0.0788	0	4	0.1761
R_{15}	0	0.1245	0.139	0.1678	0	0.1278	4	0.1761
R_{16}	0.0717	0.1164	0.1277	0.1373	0	0	4	0.1761
R_{17}	0.1029	0	0.0824	0	0	0	2	0.4771
R_{18}	0	0.0786	0.0821	0.0895	0	0	3	0.3010
R_{19}	0.0499	0	0.0257	0.0265	0.0871	0	4	0.1761
R_{20}	0.1751	0.0922	0.021	0.117	0	0	4	0.1761

At this stage, the set of Pareto-optimal resources identified in the previous iteration, $P_1 = \{R_1, R_2, R_{12}, R_{15}, R_{20}\}$ is removed from the resource set. Consequently, the new resource set is formed as $R_1 = R_0 \setminus P_1$.

After performing this operation, the parameters f_{ij}, n_j and $\log\left(\frac{n}{n_j}\right)$ are recalculated for the remaining resources. The computed values for the $p = 2$ case are presented in Table 4.

Table 4. Values of the parameters f_{ij}, n_j and $\log(\frac{n}{n_j})$ (for $p = 2$ case)

Resource	f_{ij} $(i \in I; j \in (J \setminus J_1))$						n_j	$\log(n/n_j)$
	U_1	U_2	U_3	U_4	U_5	U_6		
R_3	0	0.0847	0	0.0029	0.0483	0.0883	4	0.1761
R_4	0.1229	0	0.1378	0.1211	0.0605	0	4	0.1761
R_5	0	0.0836	0	0	0	0	1	0.7782
R_6	0.1175	0	0	0.1291	0	0	2	0.4771
R_7	0	0.05	0.0366	0	0.0866	0.1291	4	0.1761
R_8	0.0699	0	0.049	0	0.0591	0	3	0.3010
R_9	0	0.0186	0	0	0.0917	0	2	0.4771
R_{10}	0	0.034	0	0	0	0	1	0.7782
R_{11}	0	0.0717	0	0.0004	0.0889	0	3	0.3010
R_{13}	0	0	0.0995	0	0.1407	0.0091	3	0.3010
R_{14}	0.0153	0.0855	0	0.0179	0.0788	0	4	0.1761
R_{16}	0.0717	0.1164	0.1277	0.1373	0	0	4	0.1761
R_{17}	0.1029	0	0.0824	0	0	0	2	0.4771
R_{18}	0	0.0786	0.0821	0.0895	0	0	3	0.3010
R_{19}	0.0499	0	0.0257	0.0265	0.0871	0	4	0.1761

Analogously to Table 3, the set of Pareto solutions is found for Table 4:

- $i = 1$: $P_{21} = \{R_4\}$;
- $i = 2$: $P_{22} = \{R_{16}\}$;
- $i = 3$: $P_{23} = \{R_4\}$;
- $i = 4$: $P_{24} = \{R_{16}\}$;
- $i = 5$: $P_{25} = \{R_{13}, R_{19}\}$;
- $i = 6$: $P_{26} = \{R_7\}$.

The set of Pareto solutions P_{2i} will be the union of sets $i = 1, \ldots, 6$:

$\mathrm{P}_2 = \bigcup_{i=1}^{6} P_{2i} = \{R_4, R_7, R_{13}, R_{16}, R_{19}\}$.

The final set of Pareto solutions will be the union of $\mathrm{P}_1 = \{R_1, R_2, R_{12}, R_{15}, R_{20}\}$ and $\mathrm{P}_2 = \{R_4, R_7, R_{13}, R_{16}, R_{19}\}$: $\mathrm{P} = \mathrm{P}_1 \cup \mathrm{P}_2 = \{R_1, R_2, R_4, R_7, R_{12}, R_{13}, R_{15}, R_{16}, R_{19}, R_{20}\}$.

Checking condition (7) for these resources:

$$\sum_{j \in J_2} v_j \leq \sum_{q \in K} V_q, \qquad (7)$$

where

$J_2 = \{1, 2, 4, 7, 12, 13, 15, 16, 19, 20\}$.

Thus,

$$\sum_{j \in J_2} v_j = v_1 + v_2 + v_4 + v_7 + v_{12} + v_{13} + v_{15} + v_{16} + v_{19} + v_{20} \leq \sum_{q \in K} V_q = V_1 + V_2 + V_3$$

22+58+64+34+83+44+47+28+42+52≤275+236+151

474≤662.

As seen, condition (7) is met. Then, we accept $m := 15 - m_2 = 15 - 5 = 10$ and $p := 2 + 1 = 3$ and proceed to **Step 2**.

Firstly, the set of Pareto solutions $\mathbf{P}_2 = \{R_4, R_7, R_{13}, R_{16}, R_{19}\}$ found in the previous step from the resource set **R** is excluded: $\mathbf{R}_2 = \mathbf{R}_1 \setminus \mathbf{P}_2$. Table 5 presents the result obtained after this operation.

Table 5. $f_{ij}, n_j, \log(n/n_j)$ (case $p = 3$)

Resources	f_{ij} $(i \in I; j \in (J \setminus J_2))$						n_j	$\log(n/n_j)$
	U_1	U_2	U_3	U_4	U_5	U_6		
R_3	0	0.0847	0	0.0029	0.0483	0.0883	4	0.1761
R_5	0	0.0836	0	0	0	0	1	0.7782
R_6	0.1175	0	0	0.1291	0	0	2	0.4771
R_8	0.0699	0	0.049	0	0.0591	0	3	0.3010
R_9	0	0.0186	0	0	0.0917	0	2	0.4771
R_{10}	0	0.034	0	0	0	0	1	0.7782
R_{11}	0	0.0717	0	0.0004	0.0889	0	3	0.3010
R_{14}	0.0153	0.0855	0	0.0179	0.0788	0	4	0.1761
R_{17}	0.1029	0	0.0824	0	0	0	2	0.4771
R_{18}	0	0.0786	0.0821	0.0895	0	0	3	0.3010

Analogously, the set of Pareto solutions is found for Table 5:

- $i = 1$: $P_{31} = \{R_6, R_8, R_{14}\}$;
- $i = 2$: $P_{32} = \{R_{14}\}$;
- $i = 3$: $P_{33} = \{R_3, R_{17}, R_{18}\}$;
- $i = 4$: $P_{34} = \{R_6, R_{14}\}$;
- $i = 5$: $P_{35} = \{R_9, R_{11}, R_{14}\}$;
- $i = 6$: $P_{36} = \{R_3\}$.

The set of Pareto solutions will be the union of sets P_{3i} $(i = 1, \ldots, 6)$:

$\mathrm{P}_3 = \bigcup_{i=1}^{6} P_{3i} = \{R_3, R_6, R_8, R_9, R_{11}, R_{14}, R_{17}, R_{18}\}$.

The final Pareto solution set will be the combination of sets $\mathrm{P}_1 = \{R_1, R_2, R_{12}, R_{15}, R_{20}\}$, $\mathrm{P}_2 = \{R_4, R_7, R_{13}, R_{16}, R_{19}\}$ and $\mathrm{P}_3 = \{R_3, R_6, R_8, R_9, R_{11}, R_{14}, R_{17}, R_{18}\}$:

$\mathrm{P} = \mathrm{P}_1 \cup \mathrm{P}_2 \cup \mathrm{P}_3 = \{R_1, R_2, R_{12}, R_{15}, R_{20}\} \cup \{R_4, R_7, R_{13}, R_{16}, R_{19}\} \cup \{R_3, R_6, R_8, R_9, R_{11}, R_{14}, R_{17}, R_{18}\}=$

$$= \{R_1, R_2, R_3, R_4, R_6, R_7, R_8, R_9, R_{11}, R_{12}, R_{13}, R_{14}, R_{15}, R_{16}, R_{17}, R_{18}, R_{19}, R_{20}\}$$

Checking condition (7) for these resources:

$$\sum_{j \in J_3} v_j \leq \sum_{q \in K} V_q.$$

Here

$J_3 = \{1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$.

Thus,

$$\sum_{j\in J_3} v_j = v_1 + v_2 + v_3 + v_4 + v_6 + v_7 + v_8 + v_9 + v_{11} + v_{12} + v_{13} + v_{14} + v_{15} + v_{16} + v_{17} + v_{18} + v_{19} + v_{20} \leq \sum_{q\in K} V_q = V_1 + V_2 + V_3$$

22+58+75+64+67+34+78+71+32+83+44+32+47+28+10+11+42+52 ≤ 275+236+151

850≤662

Since condition (7) is not satisfied, the resource selection process is terminated.

In the next stage, the selected resources are deployed on servers according to the previously determined set of Pareto-optimal solutions.

The following procedure is proposed for this purpose:

Initially, the resources included in the set P_1 are sequentially placed in the memory resources of servers, starting from the first server. During the placement process, the memory capacity of each server is taken into account. If the total volume of the resources being placed exceeds the memory capacity of the first server, the remaining resources are allocated to the memory of the next server.

Once all resources in P_1 have been fully deployed, the process continues in the same manner with the resources in P_2. Subsequently, the resources in P_3 are sequentially placed in the servers' memory resources (V_q). The placement process continues until condition (7) is satisfied.

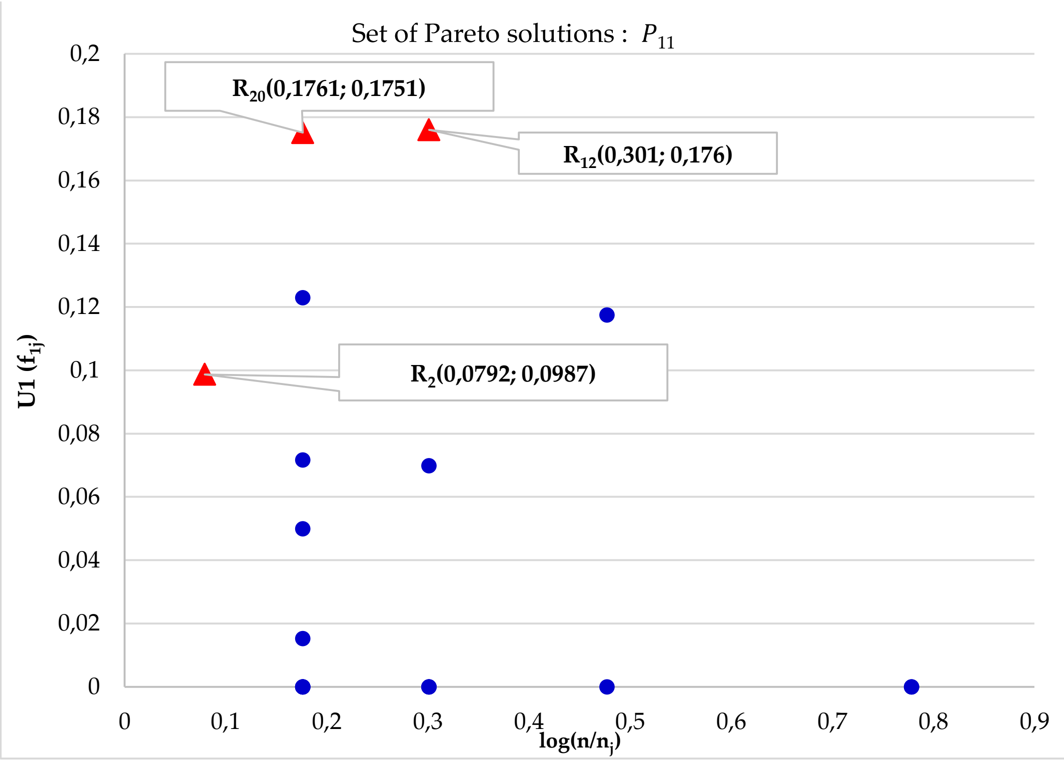

Fig. 2. Set of Pareto solutions for the pair $\log(n/n_j)$ with $U_1(f_{1j})$

5. Conclusion

In recent years, edge computing systems have been widely employed to eliminate delays in the communication channels of cloud computing networks. Edge computing systems enable tasks to be processed on computing devices located close to the users.

This approach reduces network latency, ensures rapid real-time data processing, and minimizes the costs associated with providing services.

In this paper, a model of an edge computing system based on DPCs is proposed. Additionally, a solution is presented for placing active cloud resources–those with high usage frequency and a large number of users–within the memory resources of the national network infrastructure.

The problem under consideration constitutes a multi-objective optimization task. Multi-objective optimization involves the simultaneous optimization of two or more potentially conflicting criteria–in this case, usage frequency and number of users. The Pareto optimality approach has been applied to solve such problems.

References

Abbas, N., Zhang, Y., Taherkordi, A., & Skeie, T. (2018). Mobile edge computing: A survey. IEEE Internet of Things Journal, 5(1), 450-465. https://doi.org/ 10.1109/JIOT.2017.2750180

Abusaimeh, H. (2022). Computation Offloading for Mobile Cloud Computing Frameworks and Techniques. TEM Journal, 11(3), 1042-1046. https://doi.org/10.18421/TEM113-08

Akomolafe, O. P., Abodunrin, M. O. (2017). A Hybrid Cryptographic Model for Data Storage in Mobile Cloud Computing, International Journal of Computer Network and Information Security, 9(6), 53-60. https://doi.org/10.5815/ijcnis.2017.06.06

Alakbarov, R. G. (2021). Challenges of mobile devices' resources and in communication channels and their solutions. International Journal of Computer Network and Information Security, 13(1), 39-46. https://doi.org/10.5815/ijcnis.2021.01.04

Alguliyev, R. M., & Alakbarov, R. G. (2023). Integer programming models for task scheduling and resource allocation in mobile cloud computing. International Journal of Computer Network and Information Security, 15(5), 13-26. https://doi.org/10.5815/ijcnis.2023.05.02

Alguliyev, R. M., Aliguliyev, R. M., & Alakbarov, R. G. (2023). Constrained k-means algorithm for resource allocation in mobile cloudlets. Kybernetika, 59(1), 88-109. https://doi.org/10.14736/kyb-2023-1-0088

Andriulo, F. C., Fiore, M., Mongiello, M., Traversa, E., & Zizzo, V. (2024). Edge computing and cloud computing for ınternet of things: A review. Informatics, 11(4), 71. https://doi.org/10.3390/informatics11040071

Ceselli, A., Premoli, M., & Secci, S. (2017). Mobile edge cloud network design optimization. IEEE/ACM Transactions on Networking, 25(3), 1818-1831. https://doi.org/10.1109/TNET.2017.2652850

El-Sayed, H., Sankar, S., Prasad, M., Puthal, D., Gupta, A., Mohanty, M., & Lin, C. T. (2017). Edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment. IEEE Access, 6, 1706-1717. https://doi.org/10.1109/ACCESS.2017.2780087

Gunantara, N. (2018). A review of multi-objective optimization: Methods and its applications. Cogent Engineering, 5(1), 1-16. https://doi.org/10.1080/23311916.2018.1502242

Jia, M., Cao, J., & Liang, W. (2015). Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks. IEEE Transactions on Cloud Computing, 5(4), 725-737. https://doi.org/10.1109/TCC.2015.2449834

Madni, S. H. H., Latiff, M. S. A., Ali, J., & Abdulhamid, S. I. M. (2019). Multi-objective-oriented cuckoo search optimization-based resource scheduling algorithm for clouds. Arabian Journal for Science and Engineering, 44(4), 3585-3602. https://doi.org/10.1007/s13369-018-3602-7

Malik, S. U., Akram, H., Gill, S. S., Pervaiz, H., & Malik, H. (2021). EFFORT: Energy efficient framework for offload communication in mobile cloud computing. Software: Practice and Experience, 51(9), 1896-1909. https://doi.org/10.1002/spe.2850

Mukherjee, A., De, D., & Buyya, R. (2024). Cloud Computing Resource Management. In: A. Mukherjee, D. De, R. Buyya, R. (Eds) Resource Management in Distributed Systems. Studies in Big Data (pp. 17-37). https://doi.org/10.1007/978-981-97-2644-8_2

Mukherjee, A., De, D., Ghosh, S. K., & Buyya, R. (2021). Introduction to mobile edge computing. In A. Mukherjee, D. De, S.K. Ghosh, R. Buyya. (Eds) Mobile Edge Computing (pp. 3-19). https://doi.org/10.1007/978-3-030-69893-5_1

Multi-objective optimization with Pareto front. https://www.d3view.com/multi-objective-optimization-with-pareto-front/ Accessed 7 October 2025

Nandal, P., Bura, D., Singh, M., & Kumar, S. (2021). Analysis of different load balancing algorithms in cloud computing. International Journal of Cloud Applications and Computing (IJCAC), 11(4), 100-112. https://doi.org/10.4018/IJCAC.2021100106

Nayyer, M. Z., Raza, I., & Hussain, S. A. (2018). A survey of cloudlet-based mobile augmentation approaches for resource optimization. ACM Computing Surveys (CSUR), 51(5), 1-28. https://doi.org/10.1145/3241738

Rajak, N., & Shukla, D. (2018). Comparative study of cloud computing and mobile cloud computing. International Journal of Engineering Sciences & Research Technology (IJESRT), 7(3), 734-739.

Sharma, M., & Prachi, B. (2017). Study On Mobile Cloud Computing, It's Architecture, Challenges and Various Trends. International Research Journal of Engineering and Technology (IRJET), 4(6), 1168-1177.

Shen, C., Xue, S. & Fu, S. (2019). ECPM: an energy-efficient cloudlet placement method in mobile cloud environment. EURASIP Journal on Wireless Communications and Networking 2019, 141. https://doi.org/10.1186/s13638-019-1455-8

Somula, R. S., & Sasikala, R. (2018). A survey on mobile cloud computing: mobile computing+ cloud computing (MCC= MC+ CC). Scalable Computing: Practice and Experience, 19(4), 309-337. https://doi.org/10.12694/scpe.v19i4.1411

Tian, L., & Zhong, X. (2022). A case study of edge computing implementations: Multi-access edge computing, fog computing and cloudlet. Journal of Computing and Information Technology, 30(3), 139-159. https://doi.org/10.20532/cit.2022.1005646

Tsai, Y., Chang, D., & Hsu, T. (2022). Edge Computing Based on Federated Learning for Machine Monitoring. Applied Sciences, 12(10), 5178. https://doi.org/10.3390/app12105178

Yuyi, M., You, C., Zhang, J., Huang, K., & Letaief, K. (2017). A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials, 19(4), 2322-2358. https://doi.org/10.1109/COMST.2017.2745201

Zhao, M., & Zhou, K. (2019). Selective Offloading by Exploiting ARIMA-BP for Energy Optimization in Mobile Edge Computing Networks. Algorithms, 12(2), 48. https://doi.org/10.3390/a12020048