I. J. Computer Network and Information Security, **2023, 5, 13-26**
Published Online on October 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2023.05.02

Integer Programming Models for Task Scheduling and Resource Allocation in Mobile Cloud Computing

Rasim M. Alguliyev
Institute of Information Technology/Azerbaijan National Academy of Sciences, Baku, AZ1141, Azerbaijan
E-mail: r.alguliev@gmail.com
ORCID iD: https://orcid.org/0000-0003-1223-74111

Rashid G. Alakbarov*
Institute of Information Technology/Azerbaijan National Academy of Sciences, Baku, AZ1141, Azerbaijan
E-mail: rashid.alakberov@gmail.com
ORCID iD: https://orcid.org/0000-0002-7566-371X
*Corresponding Author

Received: 01 November 2022; Revised: 13 February 2023; Accepted: 20 May 2023; Published: 08 October 2023

Abstract: In traditional mobile cloud computing, user tasks are uploaded and processed on a cloud server over the Internet. Due to the recent rapid increase in the number of mobile users connected to the network, due to overload of the Internet communication channels, there are significant delays in the delivery of data processed on cloud servers to the user. Furthermore, it complicates the optimal scheduling of the tasks of many users on cloud servers and the delivery of results. Scheduling is an approach used to reduce the tasks execution time by ensuring a balanced distribution of user tasks on cloud servers. The goal of scheduling is to ensure selection of appropriate resources to handle tasks quickly, taking into account user requirements. Whereas the goal of cloud service providers is to provide users with the required resources through performing effective scheduling so that both the user and the service provider can benefit. The article proposes a scheduling model to reduce processing time, network latency, and power consumption of mobile devices through optimal task placement in the cloudlet network in a mobile cloud computing environment.

Index Terms: Mobile Cloud Computing, Task Scheduling, Resource Allocation, Integer Programming Model, Matlab 2022a.

1. Introduction

Cloud computing is an important platform for users in terms of cost and convenience of Internet services. The services are offered on a paid basis. Cloud services are easily accessed by users via the Internet. Such computing model provides mobile users with the access to computing and memory resources of distributed data processing centers and enables them to use these resources at a lower cost through simple management. Cloud computing is a new type of distributed computing system. It has large computing resources and allows mobile users to use these resources based on a payment strategy. Users and businesses can solve their problems using cloud services without the need to buy additional computing equipment to solve complex problems. Cloud users can rent computing and memory resources when needed. Conditions must be created so that cloud resources could be used to meet the users' requests. Implementation of user software on cloud servers minimizes the execution time of tasks and reduces the power consumption of mobile devices.

Recently, mobile devices (tablets, smartphones and laptops) are widely used as communication and computing platforms. Additionally, various software applications are used on mobile devices (for example, watching movies, online shopping, interactive games, online social networking services, e-mail services, visiting web browsers, etc.). Extensive use of these services by mobile users leads to rapid depletion of energy sources (batteries) of mobile devices. The use of recently developed applications requires large computing and memory resources on mobile devices. Due to their small size, mobile devices have limited computing and memory resources. Therefore, the problems related to the use of software applications that require large computing and memory resources arise.

These problems can be solved by using a new paradigm, i.e., mobile cloud technology. Cloud technology provides solutions to users' problems on cloud servers with sufficient computing and memory resources. Mobile cloud computing (MCC) is used to provide mobile devices with computing and memory resources. MCC is a new paradigm created with the integration of mobile network and cloud computing, which eliminates the limitations on computing and memory resources in mobile devices and reduces the power consumption of mobile devices [1]. Cloud computing technology offers users unlimited resources to solve high-performance problems that require large memory resources.

MCC is a network infrastructure that stores and processes the users' data beyond the mobile devices, i.e., on cloud servers. Cloud computing performs the user tasks on cloud servers. On the other hand, traditional cloud computing solves the tasks on cloud servers, consequently the Internet is overloaded, resulting in delays in delivering results to the user. When the tasks are solved on remote servers, mobile phones run out of battery quickly.

Peripheral computing (edge computing, cloudlet, etc.) systems are used to eliminate these problems. Peripheral computing places the processing devices (computers) close to users. This article discusses a model for deploying remote cloud resources in cloudlets located close to users. Cloudlet is a set of devices (servers, laptops, tablets, laptops, etc.) connected to the base stations of the mobile network, which provides faster task processing and delivery of results to users. Cloudlet is an intermediate cloud layer in the three-level MCC architecture, namely mobile devices, cloudlet and cloud. Since the cloudlets are located between mobile users and cloud servers, they deliver results faster to users. Thus, building cloudlet-based mobile cloud computing becomes a topical issue to overcome these problems.

The computer equipment used to build cloudlet networks comprises various technical capabilities. It is complicated to choose a cloudlet with high computing and memory resources according to the complexity rate of the user application. Task scheduling (efficient use of resources) in cloudlets is an NP-hardness task that can be solved using heuristic algorithms. The proposed scheduling strategy can reduce the power consumption and network latency of mobile devices by uploading the applications to appropriate cloudlets.

Thus, a model (strategy) is proposed to implement a balanced loading of workload in geographically distributed cloudlets to minimize the task processing time and network delays.

2. Related Works

[2] proposes an algorithm for faster task processing, efficient use of resources, and reduced latency. [3] introduces a two-step algorithm for scheduling virtual machines (VMs) in a cloud computing environment to balance the workload on existing virtual machines and minimize processing time. The problem of task scheduling for computing in peripheral clouds are analyzed and the Petrel algorithm is proposed to solve the tasks [4]. Some studies analyze the issues of reducing power consumption and network latency in mobile devices of peripheral computing systems built between users and cloud servers [5,6]. [7] presents an adaptive scheduling algorithm to optimize energy consumption in mobile devices, taking into account time constraints. A comparative analysis of differences in power consumption and the network delays during the implementation of applications in cloudlets or cloud servers is performed [8]. [9] presents the mechanisms for scheduling the placement of the main and auxiliary parts of the tasks in the peripheral and in the clouds. [10] reviews the issue of selecting a cloudlet according to the application format. [11] analyzes the problem of optimal distribution of tasks in a cloud environment. A queue model is proposed to reduce response time and improve the use of cloud resources. An optimization model is proposed to reduce power consumption and network latency in mobile devices [12, 13]. [14] proposes a strategy for task-oriented resource adaptation scheduling (LA-RATS) in the MCC system for latency-sensitive mobile applications, taking into account the changing QoS requirements of users and the cost of the cloudlet provider. The distribution of mobile users' tasks among cloudlets in mobile cloud computing is explored [15]. [16] offers a strategy to use peripheral cloud computing with a hierarchical architecture to reduce network latency during peak hours. [17] proposes a strategy for using a cloudlet network to reduce task processing time and network latency. The use of mobile cloud computing to solve complex problems in mobile devices is considered [18, 19]. Some studies indicate the reduction in energy consumption and network latency of mobile devices when using cloud-based mobile computing networks [20, 21]. The use of peripheral computing systems is highlighted to reduce energy consumption and latency in mobile devices [22, 23]. [24] offers using the game theory to ensure that the main and auxiliary parts of the task are uploaded in remote cloud and edge cloud, respectively, in order to ensure fast task processing. The article [25] proposes a VM selection algorithm that provides a minimum task execution time. In [26], an author proposes a method for task scheduling that meets cost requirements in a heterogeneous cloud environment using two heuristic algorithms. [27] offers a workload scheduling model based on a genetic algorithm, taking into consideration the service costs in a cloud environment. The article [28] presents a scheduling mechanism for selecting virtual machines that provide fast task processing. [29] considers a scheduling for tasks placement in cloudlets located close to the user. [30] presents a task scheduling mechanism to optimize energy consumption in a cloud environment. This mechanism allows for more efficient use of resources by finding a correlation between the solution time and energy consumption. The study [31] analyzes the delays in delivering results to users when solving problems on remote cloud servers. [32] proposes mechanisms for the optimal distribution of energy-saving issues in computing resources in MCC. [33] offers an algorithm for selecting virtual machines to provide task solutions in cloud with minimal energy consumption.

3. The Proposed Optimization Models

Taking into account the above, a mathematical model is proposed to provide energy consumption on mobile devices, to upload the problem into the cloud, to get results fast, to reduce network delays and to balance load distribution in cloudlets.

The following notations are introduced:

- $\mathbb{U} = \{U_i | i = 1, \dots, n\}$ – the set of mobile users;
- $\mathbb{R} = \{R_j | j = 1, \dots, m\}$ – the number of cloud resources used by mobile users;
- $\mathbb{V} = \{V_j | j = 1, \dots, m\}$ – the volume of cloud resources;
- $\mathbb{F} = \|f_{ij}\|$ – the frequency matrix of the use of cloud resources by users;
- $\mathbb{D} = \|d_{ij}\|$ – the geographical distance matrix from users to cloud resources;
- $\mathbb{S} = \{S_k | k = 1, \dots, K\}$ – the set of servers;

where V_j denotes the volume of resource R_j; f_{ij} - the usage frequency of the user U_i of the resource R_j, d_{ij} - the physical distance from the user U_i to the resource R_j. Here, it is assumed that $m \gg n \gg k$.

The activity of mobile users (α_i) is defined as follows:

$$\alpha_i = \frac{\sum_{j=1}^{m} f_{ij}}{\sum_{p=1}^{n} \sum_{q=1}^{m} f_{pq}}, \; i = 1, \dots, n \tag{1}$$

Then, from Eq. (1) follows that:

$$\sum_{i=1}^{n} \alpha_i = 1 \tag{2}$$

Similarly, the usage rate of cloud resources, in other words, the relative usage frequency of cloud resources (β_j) can be calculated as follows:

$$\beta_j = \frac{\sum_{i=1}^{n} f_{ij}}{\sum_{p=1}^{n} \sum_{q=1}^{m} f_{pq}}, \; j = 1, \dots, m \tag{3}$$

From Eq. (3) follows that:

$$\sum_{j=1}^{m} \beta_j = 1 \tag{4}$$

As can be seen, there are two types of distances between the set of users $\mathbb{U} = \{U_j | i = 1, \dots, n\}$ and the set of cloud resources $\mathbb{R} = \{R_j | j = 1, \dots, m\}$: semantic $\mathbb{F} = \|f_{ij}\|$ and geographical $\mathbb{D} = \|d_{ij}\|$.

Preliminary analysis shows that since the characteristics (values) of the distances $\mathbb{F}$ and $\mathbb{D}$ are not taken into account when designing the network infrastructure:

- additional costs arise for users;
- uploading users' cloud resources is time consuming;
- energy resources of users' mobile devices are quickly depleted;
- on the other hand, Cloud Service Providers (CSP) are not able to provide the declared (agreed) quality indicators, the network infrastructure is overloaded, etc.;
- consequently, CSP is forced to lose customers (users), face economic losses, and so on.

In these networks, CSP must take certain measures in order to improve the quality of services provided, as well as to meet the users' needs. These measures can be primarily extensive and intensive. Extensive measures include the implementation of technological projects requiring large investments, such as the purchase of new equipment, increasing the Internet speed, the creation of "hot", online cloud resources backups, etc. Intensive measures may include spending relatively little financial resources and analyzing the processes running in the network infrastructure to reveal the problems that can be significantly improved by solving through mathematical and technological methods.

A conceptual model of cloud computing systems is shown in Figure 1.

Numerous hypotheses and approaches can be put forward in this field. As one of such intensive measures, a certain optimization model based on matrices $\mathbb{F}$ and $\mathbb{D}$, which characterize the physical distances between users $\mathbb{U} = \{U_j | i = 1, \dots, n\}$ and cloud resources $\mathbb{R} = \{R_j | j = 1, \dots, m\}$ respectively, can be built. In this regard, in the region which users $\mathbb{U}$ belong to, based on the investment policy of CSP, it is possible to create hierarchically low-level cloud service servers (CSS), $\mathbb{S} = \{S_k | k = 1, \ldots, K\}$.

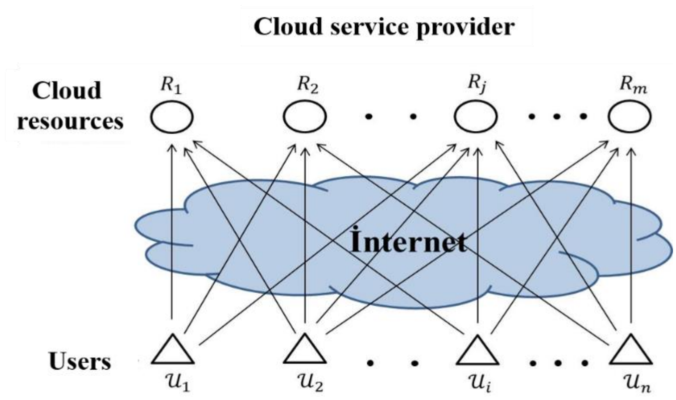

Fig.1. Conceptual model of cloud computing systems

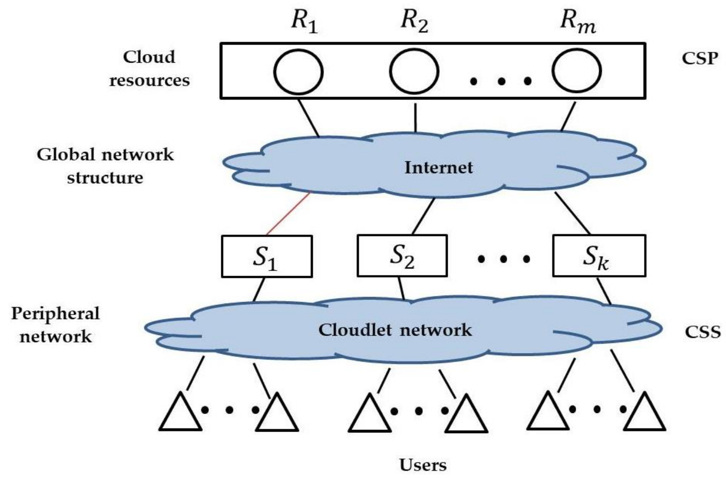

Fig.2. Cloudlet-based mobile cloud computing system architecture

Then, based on the matrices $\mathbb{F}$ and $\mathbb{D}$, more distant resources most frequently used by certain user groups can be distributed among the servers S_k, i.e., cloudlets to serve those groups through an optimal strategy so that the load on the network infrastructure significantly reduces, the requests for resources are timely ensured, the energy resources of mobile devices are not quickly depleted due to delays. It should be noted that, in this case, the number of cloud servers K, the amount of finance allocated by CSP are determined within the terms of the services provided to users. Thus, as a result of the intensive measures described above, the CSP infrastructure becomes as presented in Figure 2.

It should be noted that each server S_k has a limited amount of v_k memory, and cloud resources $\mathbb{R}$ cannot be allocated entirely on certain server. In other words, the volume of cloud resources is much larger than the storage capacity of each server:

$$\sum_{j=1}^{m} V_j \gg \vartheta_k, k = 1, \ldots, K \tag{5}$$

On the other hand, the storage capacity of servers is not greater than the sum of the total cloud resources:

$$\sum_{k=1}^{K} \vartheta_k \le \sum_{j=1}^{m} V_j \tag{6}$$

Now let’s view the building of an optimization model to ensure cloud resources $\mathbb{R}$ and users $\mathbb{U}$ to be distributed among cloud servers $\mathbb{S}$. Preliminary analysis shows that a pseudo-Boolean variable programming model can be used along with other methods to solve this type of task.

In this regard, based on the essence of the problem, let’s include the Boolean variables x_{jk} and y_{ik} into the following interpretation:

$$x_{jk} = \begin{cases} 1, \text{if resource } R_j \text{ is allocated at server } S_k \\ 0, \quad\quad \text{otherwise} \end{cases} \tag{7}$$

$$y_{ik} = \begin{cases} 1, \text{if user } U_i \text{ is assigned to server } S_k \\ 0, \quad \text{otherwise} \end{cases} \tag{8}$$

where $i = 1, \dots, n; j = 1, \dots, m; k = 1, \dots, K$.

Now, let's move on to formulate a pseudo-Boolean optimization model. Before formalizing the model, let's normalize the matrices $\mathbb{F}$ and $\mathbb{D}$, $\boldsymbol{\Lambda} = \|\lambda_{ij}\|$ and $\boldsymbol{\Delta} = \|\delta_{ij}\|$, respectively, as follows:

$$\lambda_{ij} = \frac{f_{ij}}{\sum_{q=1}^{m} f_{iq}}, i = 1, \dots, n; j = 1, \dots, m \tag{9}$$

$$\sum_{j=1}^{m} \lambda_{ij} = 1, i = 1, \dots, n \tag{10}$$

$$\delta_{ij} = \frac{d_{ij}}{\sum_{q=1}^{m} d_{iq}}, \ i = 1, \dots, n; j = 1, \dots, m \tag{11}$$

$$\sum_{j=1}^{m} \delta_{ij} = 1, i = 1, \dots, n \tag{12}$$

where λ_{ij} and δ_{ij} are normalized values of the elements f_{ij} and d_{ij} of the matrices $\mathbb{F}$ and $\mathbb{D}$, respectively.

Thus, taking into account the resource R_j, the usage frequency β_j and the distance δ_{ij}, the server(s) should also be placed in such a way that:

$$\beta_j(1 - \delta_{ij}) \to \max \tag{13}$$

Figure 2 shows that $\delta_{ij} = \delta_{ik} + \delta_{kj}$, in other words, the distance (δ_{ij}) between the user (U_i) and the resource (R_j) is equal to the sum of the distance (δ_{ik}) between the user (U_i) and the server (S_k) and the distance (δ_{kj}) between the server and the resource. Taking into account that the distance δ_{ik} is much shorter than the distance δ_{kj} ($\delta_{ik} \ll \delta_{kj}$), then the expression (13) can be replaced by:

$$\beta_j(1 - \delta_{kj}) \to \max \tag{14}$$

Thus, given the definition (7), the first objective function of the optimization model can be written as follows:

$$f_1 = \sum_{j=1}^{m} \beta_j \sum_{k=1}^{K} (1 - \delta_{kj}) x_{jk} \to \max \tag{15}$$

The second part of the model includes the assignment of users to servers. Given the activity α_i and the frequency λ_{ij}, the user should be assigned to the server as follows:

$$\alpha_i \lambda_{ij} \to \max \tag{16}$$

Using definitions (7) and (8), the second objective function of the optimization model can be written as follows:

$$f_2 = \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{m} \sum_{k=1}^{K} \lambda_{ij} x_{jk} y_{ik} \to \max \tag{17}$$

The final objective function can be written as a weighted linear combination of the functions (15) and (17):

$$\begin{aligned} \mathcal{F}_1 &= \gamma f_1 + (1-\gamma) f_2 = \\ &= \gamma \sum_{j=1}^{m} \beta_j \sum_{k=1}^{K} (1 - \delta_{kj}) x_{jk} + (1-\gamma) \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{m} \lambda_{ij} \sum_{k=1}^{K} x_{jk} y_{ik} \to \max \end{aligned} \tag{18}$$

where $0 \le \gamma \le 1$ denotes the weight factor.

On the other hand, the objective function can also be defined as a multiplication of the functions (15) and (17):

$$\mathcal{F}_2 = \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{m} \beta_j \lambda_{ij} \sum_{k=1}^{K} (1 - \delta_{kj}) x_{jk} y_{ik} \to \max \tag{19}$$

Given the problem statement and the servers' characteristics, the optimum of this objective function must be found within the following constraints.

The sum of the volume of cloud resources to be placed on the server cannot exceed the total storage capacity of the server:

$$\sum_{j=1}^{m} V_j x_{jk} \le \vartheta_k, k = 1, \dots, K \tag{20}$$

Each cloud resource should be allocated on only one server:

$$\sum_{k=1}^{K} x_{jk} = 1,\ j = 1, \dots, m \tag{21}$$

Each server should host at least one cloud resource and all resources cannot be allocated on one server:

$$1 \le \sum_{j=1}^{m} x_{jk} < m,\ k = 1, \dots, K \tag{22}$$

Each user should be assigned only to one server:

$$\sum_{k=1}^{K} y_{ik} = 1,\ i = 1, \dots, n \tag{23}$$

Each server should have at least one user assigned and all users cannot be assigned to the same server:

$$1 \le \sum_{i=1}^{n} y_{ik} < n,\ k = 1, \dots, K \tag{24}$$

Table 1. Frequency matrix ($\mathbb{F}$), users' activity (α_i) and frequency of resource use β_j

$\mathbb{F} = \lVert \boldsymbol{f_{ij}} \rVert$, $(i = 1, \dots, 7; j = 1, \dots, 30)$								
$\mathbb{R} = \{R_j \mid j = 1, \dots, 30\}$	$\boldsymbol{\beta} = \{\beta_j \mid j = 1, \dots, 30\}$	$\mathbb{U} = \{U_i \mid i = 1, \dots, 7\}$						
		U_1B	U_2	U_3	U_4	U_5	U_6	U_7
R_1	$\beta_1 = 0.0222$	25	75	25	23	38	26	59
R_2	$\beta_2 = 0.0389$	90	56	54	92	68	55	59
R_3	$\beta_3 = 0.0368$	32	31	99	61	85	85	55
R_4	$\beta_4 = 0.0336$	42	64	97	83	72	19	33
R_5	$\beta_5 = 0.0340$	56	28	33	81	64	85	68
R_6	$\beta_6 = 0.0405$	80	68	64	36	68	89	89
R_7	$\beta_7 = 0.0359$	81	41	64	94	64	30	64
R_8	$\beta_8 = 0.0328$	93	35	11	17	89	84	71
R_9	$\beta_9 = 0.0276$	90	85	45	16	33	45	22
R_{10}	$\beta_{10} = 0.0324$	58	64	77	51	51	76	18
R_{11}	$\beta_{11} = 0.0306$	37	25	38	34	72	77	90
R_{12}	$\beta_{12} = 0.0399$	94	27	89	36	83	87	71
R_{13}	$\beta_{13} = 0.0377$	95	55	45	63	38	69	95
R_{14}	$\beta_{14} = 0.0421$	87	51	46	85	97	47	100
R_{15}	$\beta_{15} = 0.0282$	24	47	22	74	90	25	62
R_{16}	$\beta_{16} = 0.0269$	52	50	61	16	24	99	26
R_{17}	$\beta_{17} = 0.0386$	91	63	46	82	43	72	73
R_{18}	$\beta_{18} = 0.0328$	69	42	32	86	31	97	43
R_{19}	$\beta_{19} = 0.0241$	44	74	31	28	49	49	19
R_{20}	$\beta_{20} = 0.0277$	55	24	15	92	92	26	33
R_{21}	$\beta_{21} = 0.0360$	31	37	47	92	71	64	97
R_{22}	$\beta_{22} = 0.0353$	34	56	76	72	25	78	89
R_{23}	$\beta_{23} = 0.0412$	57	81	90	63	67	81	63
R_{24}	$\beta_{24} = 0.0340$	80	75	88	28	33	75	36
R_{25}	$\beta_{25} = 0.0390$	89	70	66	85	60	82	24
R_{26}	$\beta_{26} = 0.0231$	53	18	32	55	71	29	24
R_{27}	$\beta_{27} = 0.0297$	17	39	24	74	94	60	54
R_{28}	$\beta_{28} = 0.0325$	56	57	17	81	73	79	33
R_{29}	$\beta_{29} = 0.0325$	97	29	12	58	27	73	100
R_{30}	$\beta_{30} = 0.0334$	18	98	66	90	41	50	44
		$\boldsymbol{\alpha_i}$ $(i = 1, \dots, 7)$						
		α_1= 0.1498	α_2= 0.1284	α_3= 0.1240	α_4= 0.1516	α_5= 0.1487	α_6= 0.1569	α_7= 0.1406

It is obvious that, (18), (20)-(24) and also (19)-(24), the binary nonlinear integer programming models belong to the class of NP-complete problem. Therefore, it is practically impossible to find a global optimal solution to the problem with the help of classical algorithms at large values of n, m and k. Therefore, we can only talk about suboptimal solutions. Then the question may arise which of these models will provide the best solution. The next section discusses the comparison of models on example.

To solve the binary nonlinear integer programming models, (18), (20)-(24) and (19)-(24), we have used Optimization Toolbox of MATLAB 2022a.

4. Experiment

This section compares models (18), (20) - (24) and (19) - (24). Assume that the number of cloud resources is30 ($m = 30$), the number of users is 7 ($n = 7$) and the number of servers is 3 ($k = 3$). Table 1 shows the frequency matrix of users $\mathbb{F} = \|f_{ij}\|$ using cloud resources, and the last rows of the table shows the values of user activity (α_i) using cloud resources and column 2 of the table lists the values of the frequency of use of cloud resources β_j. Table 2 shows the normalized frequency matrix of cloud resources usage.

Table 2. Normalized frequency matrix (**Λ**)

$\boldsymbol{\Lambda} = \|\lambda_{ij}\|, i = 1, \dots, 7; j = 1, \dots, 30$							
$\mathbb{R} = \{R_j \mid j = 1, \dots, 30\}$	$\mathbb{U} = \{U_i \mid i = 1, \dots, 7\}$						
	U_1	U_2	U_3	U_4	U_5	U_6	U_7
R_1	0.0137	0.0479	0.0165	0.0124	0.0210	0.0136	0.0344
R_2	0.0493	0.0358	0.0357	0.0498	0.0375	0.0288	0.0344
R_3	0.0175	0.0198	0.0655	0.033	0.0469	0.0444	0.0321
R_4	0.0230	0.0409	0.0642	0.0449	0.0397	0.0099	0.0193
R_5	0.0307	0.0179	0.0218	0.0438	0.0353	0.0444	0.0397
R_6	0.0438	0.0435	0.0423	0.0195	0.0375	0.0465	0.0519
R_7	0.0443	0.0262	0.0423	0.0509	0.0353	0.0157	0.0373
R_8	0.0509	0.0224	0.0073	0.0092	0.0491	0.0439	0.0414
R_9	0.0493	0.0543	0.0298	0.0087	0.0182	0.0235	0.0128
R_{10}	0.0317	0.0409	0.0509	0.0276	0.0281	0.0397	0.0105
R_{11}	0.0203	0.016	0.0251	0.0184	0.0397	0.0403	0.0525
R_{12}	0.0515	0.0173	0.0589	0.0195	0.0458	0.0455	0.0414
R_{13}	0.0520	0.0351	0.0298	0.0341	0.0210	0.0361	0.0554
R_{14}	0.0476	0.0326	0.0304	0.0460	0.0535	0.0246	0.0583
R_{15}	0.0131	0.0300	0.0146	0.0400	0.0496	0.0131	0.0362
R_{16}	0.0285	0.0319	0.0403	0.0087	0.0132	0.0518	0.0152
R_{17}	0.0498	0.0403	0.0304	0.0444	0.0237	0.0376	0.0426
R_{18}	0.0378	0.0268	0.0212	0.0465	0.0171	0.0507	0.0251
R_{19}	0.0241	0.0473	0.0205	0.0152	0.0270	0.0256	0.0111
R_{20}	0.0301	0.0153	0.0099	0.0498	0.0507	0.0136	0.0193
R_{21}	0.0170	0.0236	0.0311	0.0498	0.0392	0.0335	0.0566
R_{22}	0.0186	0.0358	0.0503	0.0390	0.0138	0.0408	0.0519
R_{23}	0.0312	0.0518	0.0595	0.0341	0.0370	0.0423	0.0368
R_{24}	0.0438	0.0479	0.0582	0.0152	0.0182	0.0392	0.0210
R_{25}	0.0487	0.0447	0.0437	0.0460	0.0331	0.0429	0.0140
R_{26}	0.0290	0.0115	0.0212	0.0298	0.0392	0.0152	0.0140
R_{27}	0.0093	0.0249	0.0159	0.0400	0.0518	0.0314	0.0315
R_{28}	0.0307	0.0364	0.0112	0.0438	0.0403	0.0413	0.0193
R_{29}	0.0531	0.0185	0.0079	0.0314	0.0149	0.0382	0.0583
R_{30}	0.0099	0.0626	0.0437	0.0487	0.0226	0.0261	0.0257

Table 3 shows the distance matrix (($\mathbb{D}$) between servers and cloud resources. Table 4 shows the normalized distance matrix (($\boldsymbol{\Delta}$)).

Table 3. Distance matrix ($\mathbb{D}$)

$\mathbb{D} = \|d_{qj}\|, q = 1, 2, 3;\ j = 1, \dots, 30$			
$\mathbb{R} = \{R_j \mid j = 1, \dots, 30\}$	$S = \{S_q \mid q = 1,2,3\}$		
	S_1	S_2	S_3
R_1	8369	5935	4814
R_2	4516	2390	6828
R_3	3796	2770	7462
R_4	7273	2676	7387
R_5	5927	6123	3590
R_6	7868	7085	8281
R_7	9184	5153	8909
R_8	4646	6613	6708
R_9	5914	1552	6853
R_{10}	7632	9909	3210
R_{11}	4863	1954	5605
R_{12}	1979	3757	4968
R_{13}	9289	4015	9909
R_{14}	3941	2583	4550
R_{15}	5181	1039	8109
R_{16}	4517	4393	5786
R_{17}	7910	9264	3816
R_{18}	9742	4114	1763
R_{19}	5022	4871	1882
R_{20}	5297	2520	6884
R_{21}	6447	9697	5579
R_{22}	4204	8628	5381
R_{23}	8400	9878	2884
R_{24}	8765	1754	3131
R_{25}	1754	1471	4637
R_{26}	3488	9875	5569
R_{27}	7648	1178	2053
R_{28}	9763	7268	8792
R_{29}	4904	4994	9297
R_{30}	2314	8070	4814

Table 5 illustrates the volume of cloud resources and storage capacity of servers

Table 6 illustrates solutions of optimization tasks (18), (20)-(24) for different values of the parameter γ.

Table 7 illustrates solutions of optimization tasks (19)-(24).

Such a strategy is used to determine the best solutions out of the ones found, as well as to compare optimization models. The value of the objective function of another model is calculated according to the solution given by one model. In this regard, let's first label the solutions found as follows. Let's denote the solutions of the first model corresponding to the values of 0.5, 0.6 and 0.4 of γ as $(x^{0.5}, y^{0.5})$, $(x^{0.6}, y^{0.6})$ and $(x^{0.4}, y^{0.4})$. Let's denote the solution of the second model as (x, y).

The value of the objective functions is calculated for each solution found. Table 8 illustrates the results.

Table 4. Normalized distance matrix (**Δ**)

$\Delta = \lVert \delta_{qj} \rVert, q = 1, 2, 3;\ j = 1, \ldots, 30$			
$\mathbb{R} = \{R_j \mid j = 1, \ldots, 30\}$	$\mathbb{S} = \{S_q \mid q = 1,2,3\}$		
	S_1	S_2	S_3
R_1	0.0464	0.0392	0.0282
R_2	0.0250	0.0158	0.0400
R_3	0.0210	0.0183	0.0437
R_4	0.0403	0.0177	0.0433
R_5	0.0328	0.0404	0.021
R_6	0.0436	0.0468	0.0485
R_7	0.0509	0.034	0.0522
R_8	0.0257	0.0436	0.0393
R_9	0.0328	0.0102	0.0401
R_{10}	0.0423	0.0654	0.0188
R_{11}	0.0269	0.0129	0.0328
R_{12}	0.0110	0.0248	0.0291
R_{13}	0.0514	0.0265	0.0580
R_{14}	0.0218	0.0170	0.0266
R_{15}	0.0287	0.0069	0.0475
R_{16}	0.0250	0.0290	0.0339
R_{17}	0.0438	0.0611	0.0223
R_{18}	0.0540	0.0271	0.0103
R_{19}	0.0278	0.0321	0.011
R_{20}	0.0293	0.0166	0.0403
R_{21}	0.0357	0.0640	0.0327
R_{22}	0.0233	0.0569	0.0315
R_{23}	0.0465	0.0652	0.0169
R_{24}	0.0485	0.0116	0.0183
R_{25}	0.0097	0.0097	0.0272
R_{26}	0.0193	0.0652	0.0326
R_{27}	0.0424	0.0078	0.012
R_{28}	0.0541	0.0480	0.0515
R_{29}	0.0272	0.0330	0.0544
R_{30}	0.0128	0.0533	0.0358

Table 5. Volume of cloud resources ($\mathbb{V}$) and storage capacity of servers (v_k)

Volume of cloud resources, $\mathbb{V} = \{V_j \mid j = 1, \ldots, 30\}$																													
V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}	V_{12}	V_{13}	V_{14}	V_{15}	V_{16}	V_{17}	V_{18}	V_{19}	V_{20}	V_{21}	V_{22}	V_{23}	V_{24}	V_{25}	V_{26}	V_{27}	V_{28}	V_{29}	V_{30}
5	24	17	25	8	8	23	24	31	20	25	17	7	8	26	9	20	8	21	29	15	14	15	10	5	19	26	23	23	34
Storage capacity of servers, $v_k, k = 1, 2, 3$																													
v_1									v_2											v_3									
118									137											169									

Table 6. Solutions of the model (18), (20)-(24) for different values of the parameter γ

	$\gamma = 0.5$; $\mathcal{F}_1^{0.5} = \mathcal{F}_1 = 124.0661$				$\gamma = 0.6$; $\mathcal{F}_1^{0.6} = \mathcal{F}_1 = 100.3885$				$\gamma = 0.4$; $\mathcal{F}_1^{0.4} = \mathcal{F}_1 = 141.0612$			
	$x_{jk}^{0.5}$				$x_{jk}^{0.6}$				$x_{jk}^{0.4}$			
	$k = 1$	$k = 2$	$k = 3$		$k = 1$	$k = 2$	$k = 3$		$k = 1$	$k = 2$	$k = 3$	
$j = 1$	1	0	0	$x_{11} = 1$	0	0	1	$x_{13} = 1$	0	0	1	$x_{13} = 1$
$j = 2$	0	1	0	$x_{22} = 1$	0	1	0	$x_{22} = 1$	0	1	0	$x_{22} = 1$
$j = 3$	1	0	0	$x_{31} = 1$	1	0	0	$x_{31} = 1$	1	0	0	$x_{31} = 1$
$j = 4$	1	0	0	$x_{41} = 1$	0	0	1	$x_{43} = 1$	0	0	1	$x_{43} = 1$
$j = 5$	1	0	0	$x_{51} = 1$	1	0	0	$x_{51} = 1$	1	0	0	$x_{51} = 1$
$j = 6$	1	0	0	$x_{61} = 1$	1	0	0	$x_{61} = 1$	1	0	0	$x_{61} = 1$
$j = 7$	1	0	0	$x_{71} = 1$	1	0	0	$x_{71} = 1$	1	0	0	$x_{71} = 1$
$j = 8$	1	0	0	$x_{81} = 1$	1	0	0	$x_{81} = 1$	1	0	0	$x_{81} = 1$
$j = 9$	1	0	0	$x_{91} = 1$	1	0	0	$x_{91} = 1$	1	0	0	$x_{91} = 1$
$j = 10$	1	0	0	$x_{10,1} = 1$	1	0	0	$x_{10,1} = 1$	1	0	0	$x_{10,1} = 1$
$j = 11$	0	0	1	$x_{11,3} = 1$	1	0	0	$x_{11,1} = 1$	1	0	0	$x_{11,1} = 1$
$j = 12$	1	0	0	$x_{12,1} = 1$	1	0	0	$x_{12,1} = 1$	1	0	0	$x_{12,1} = 1$
$j = 13$	1	0	0	$x_{13,1} = 1$	1	0	0	$x_{13,1} = 1$	1	0	0	$x_{13,1} = 1$
$j = 14$	1	0	0	$x_{14,1} = 1$	1	0	0	$x_{14,1} = 1$	1	0	0	$x_{14,1} = 1$
$j = 15$	1	0	0	$x_{15,1} = 1$	1	0	0	$x_{15,1} = 1$	1	0	0	$x_{15,1} = 1$
$j = 16$	1	0	0	$x_{16,1} = 1$	1	0	0	$x_{16,1} = 1$	1	0	0	$x_{16,1} = 1$
$j = 17$	1	0	0	$x_{17,1} = 1$	1	0	0	$x_{17,1} = 1$	1	0	0	$x_{17,1} = 1$
$j = 18$	1	0	0	$x_{18,1} = 1$	0	0	1	$x_{18,3} = 1$	0	0	1	$x_{18,3} = 1$
$j = 19$	1	0	0	$x_{19,1} = 1$	1	0	0	$x_{19,1} = 1$	1	0	0	$x_{19,1} = 1$
$j = 20$	1	0	0	$x_{20,1} = 1$	1	0	0	$x_{20,1} = 1$	1	0	0	$x_{20,1} = 1$
$j = 21$	0	0	1	$x_{21,3} = 1$	0	0	1	$x_{21,3} = 1$	0	0	1	$x_{21,3} = 1$
$j = 22$	1	0	0	$x_{22,1} = 1$	1	0	0	$x_{22,1} = 1$	1	0	0	$x_{22,1} = 1$
$j = 23$	1	0	0	$x_{23,1} = 1$	1	0	0	$x_{23,1} = 1$	1	0	0	$x_{23,1} = 1$
$j = 24$	1	0	0	$x_{24,1} = 1$	1	0	0	$x_{24,1} = 1$	1	0	0	$x_{24,1} = 1$
$j = 25$	0	1	0	$x_{25,2} = 1$	1	0	0	$x_{25,1} = 1$	1	0	0	$x_{25,1} = 1$
$j = 26$	0	0	1	$x_{26,3} = 1$	0	0	1	$x_{26,3} = 1$	0	0	1	$x_{26,3} = 1$
$j = 27$	1	0	0	$x_{27,1} = 1$	0	0	1	$x_{27,3} = 1$	0	0	1	$x_{27,3} = 1$
$j = 28$	1	0	0	$x_{28,1} = 1$	1	0	0	$x_{28,1} = 1$	1	0	0	$x_{28,1} = 1$
$j = 29$	0	1	0	$x_{29,2} = 1$	1	0	0	$x_{29,1} = 1$	1	0	0	$x_{29,1} = 1$
$j = 30$	0	1	0	$x_{30,2} = 1$	1	0	0	$x_{30,1} = 1$	1	0	0	$x_{30,1} = 1$
	$y_{ik}^{0.5}$				$y_{ik}^{0.6}$				$y_{ik}^{0.4}$			
	$k = 1$	$k = 2$	$k = 3$		$k = 1$	$k = 2$	$k = 3$		$k = 1$	$k = 2$	$k = 3$	
$i = 1$	1	0	0	$y_{11} = 1$	1	0	0	$y_{11} = 1$	1	0	0	$y_{11} = 1$
$i = 2$	1	0	0	$y_{21} = 1$	1	0	0	$y_{21} = 1$	1	0	0	$y_{21} = 1$
$i = 3$	1	0	0	$y_{31} = 1$	1	0	0	$y_{31} = 1$	1	0	0	$y_{31} = 1$
$i = 4$	1	0	0	$y_{41} = 1$	1	0	0	$y_{41} = 1$	1	0	0	$y_{41} = 1$
$i = 5$	1	0	0	$y_{51} = 1$	1	0	0	$y_{51} = 1$	0	0	1	$y_{53} = 1$
$i = 6$	0	1	0	$y_{62} = 1$	0	1	0	$y_{62} = 1$	0	1	0	$y_{62} = 1$
$i = 7$	0	0	1	$y_{73} = 1$	0	0	1	$y_{73} = 1$	1	0	0	$y_{71} = 1$

Table 7. Solution of the model (19)-(24)

	$\mathcal{F}_2 = 7.7874$			
	$\boldsymbol{x_{jk}}$			
	$k = 1$	$k = 2$	$k = 3$	
$j = 1$	1	0	0	$x_{11} = 1$
$j = 2$	1	0	0	$x_{21} = 1$
$j = 3$	1	0	0	$x_{31} = 1$
$j = 4$	1	0	0	$x_{41} = 1$
$j = 5$	1	0	0	$x_{51} = 1$
$j = 6$	1	0	0	$x_{61} = 1$
$j = 7$	1	0	0	$x_{71} = 1$
$j = 8$	1	0	0	$x_{81} = 1$
$j = 9$	1	0	0	$x_{91} = 1$
$j = 10$	1	0	0	$x_{10,1} = 1$
$j = 11$	1	0	0	$x_{11,1} = 1$
$j = 12$	1	0	0	$x_{12,1} = 1$
$j = 13$	0	1	0	$x_{13,2} = 1$
$j = 14$	1	0	0	$x_{14,1} = 1$
$j = 15$	1	0	0	$x_{15,1} = 1$
$j = 16$	1	0	0	$x_{16,1} = 1$
$j = 17$	1	0	0	$x_{17,1} = 1$
$j = 18$	1	0	0	$x_{18,1} = 1$
$j = 19$	1	0	0	$x_{19,1} = 1$
$j = 20$	1	0	0	$x_{20,1} = 1$
$j = 21$	0	0	1	$x_{21,3} = 1$
$j = 22$	1	0	0	$x_{22,1} = 1$
$j = 23$	1	0	0	$x_{23,1} = 1$
$j = 24$	0	0	1	$x_{24,3} = 1$
$j = 25$	1	0	0	$x_{25,1} = 1$
$j = 26$	0	0	1	$x_{26,3} = 1$
$j = 27$	0	0	1	$x_{27,3} = 1$
$j = 28$	1	0	0	$x_{28,1} = 1$
$j = 29$	0	1	0	$x_{29,2} = 1$
$j = 30$	0	1	0	$x_{30,2} = 1$
	$\boldsymbol{y_{ik}}$			
	$k = 1$	$k = 2$	$k = 3$	
$i = 1$	1	0	0	$y_{11} = 1$
$i = 2$	1	0	0	$y_{21} = 1$
$i = 3$	1	0	0	$y_{31} = 1$
$i = 4$	1	0	0	$y_{41} = 1$
$i = 5$	1	0	0	$y_{51} = 1$
$i = 6$	0	0	1	$y_{63} = 1$
$i = 7$	0	1	0	$y_{72} = 1$

Table 8. Comparison of solutions and models

	$(x^{0.5}, y^{0.5})$	$(x^{0.6}, y^{0.6})$	$(x^{0.4}, y^{0.4})$	(x, y)	Relative improvement of the solution (x, y), %
$\mathcal{F}_1^{0.5} = \mathcal{F}_1(\gamma = 0.5)$	124.0661	–	–	**125.5718**	**1.21**
$\mathcal{F}_1^{0.6} = \mathcal{F}_1(\gamma = 0.6)$	–	100.3885	–	**103.4330**	**3.03**
$\mathcal{F}_1^{0.4} = \mathcal{F}_1(\gamma = 0.4)$	–	–	141.0612	**147.7106**	**4.71**
$\mathcal{F}_2$	7.6605	7.5080	7.2543	**7.7874**	**1.66; 3.72; 7.35**

Analysis of the results shows that the solution (x, y) provided by the second model $(\mathcal{F}_2)$ is the best in all cases: $\mathcal{F}_1^{0.5}(x^{0.5}, y^{0.5}) < \mathcal{F}_1^{0.5}(x, y)$ ($124.0661 < 125.5718$), $\mathcal{F}_1^{0.6}(x^{0.6}, y^{0.6}) < \mathcal{F}_1^{0.6}(x, y)$ ($100.3885 < 103.4330$) and $\mathcal{F}_1^{0.4}(x^{0.4}, y^{0.4}) < \mathcal{F}_1^{0.4}(x, y)$ ($141.0612 < 147.7106$). The comparison is graphically shown in Figure 3. This result can also be concluded from the relation $\mathcal{F}_2(x, y) > \mathcal{F}_2(x^{0.5}, y^{0.5}) > \mathcal{F}_2(x^{0.6}, y^{0.6}) > \mathcal{F}_2(x^{0.4}, y^{0.4})$. The comparison is shown in Figure 4. Another important result is that, in the first model, the best solution is obtained when $\gamma = 0.5$, and the worst solution is obtained when $\gamma = 0.4$. This is confirmed by the relative improvements given in the last column of Table 8. Thus, solution corresponding to $\gamma = 0.5$ is out performed by the best solution by 1.21% and 1.66% based on $\mathcal{F}_1$ and $\mathcal{F}_2$ values, respectively. Analogously, on $\mathcal{F}_1$ and $\mathcal{F}_2$ values, the solution corresponding to $\gamma = 0.6$ is outperformed by the best solution by 3.03% and 3.72%, respectively. And the solution corresponding to $\gamma = 0.4$ is outperformed by the best solution by 4.71% and 7.35%, respectively. In Figures 3 and 4, the best results are colored in bold red.

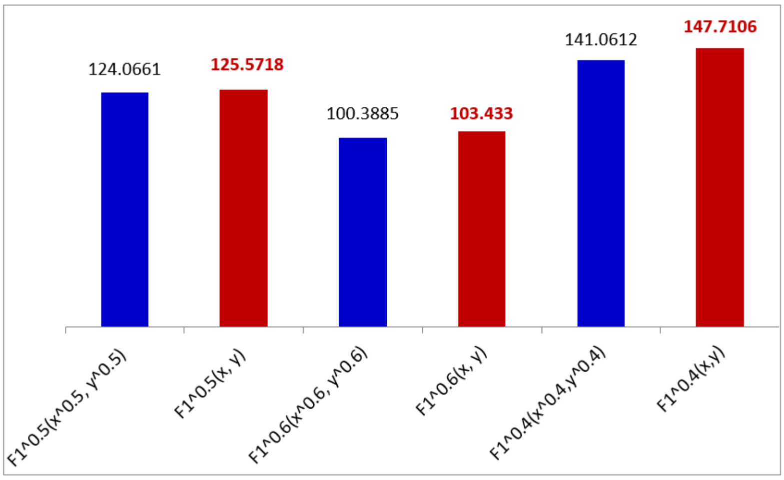

Fig.3. Comparison of solutions based on $\mathcal{F}_1$-values for different values of the parameter γ

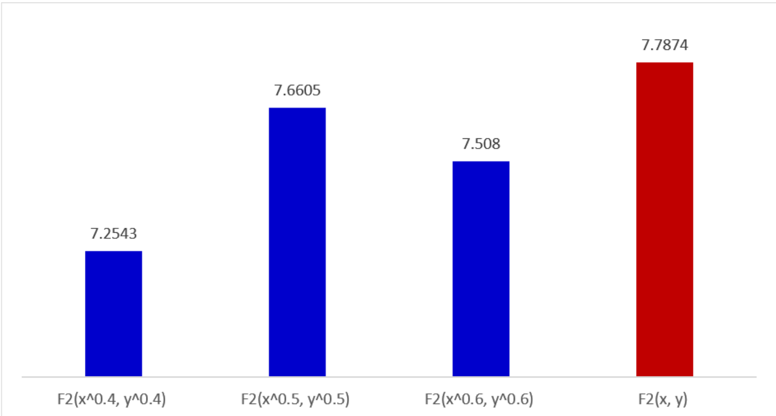

Fig.4. Comparison of solutions based on $\mathcal{F}_2$-values

Thus, referring to the best solution (x, y) in the example under consideration (see Table 7), cloud resources should be distributed to the servers as follows: cloud resources 1-12, 14-20, 22, 23, 25, and 28 should be distributed on the server 1, cloud resources 13, 29 and 30 should be distributed on the server 2, and finally, cloud resources 21, 24, 26 and 27 should be distributed on the server 3. Users should be assigned to servers as follows: users 1-5 should be assigned to server 1, user 6 should be assigned to server 3, and user 7 should be assigned to server 2.

5. Conclusions

The article offered a strategy for the optimal placement of tasks in cloud computing environment to minimize the processing time and network delays in cloud computing environment. In traditional mobile cloud computing, there are delays in delivering the results obtained to the user due to the loading of the Internet when solving the problems on cloud servers. Moreover, when the tasks are solved on a cloud server, the power consumption of mobile devices increases and its energy reserves quickly deplete. The proposed scheduling model reduced the mobile power consumption and network latency by uploading applications into appropriate cloudlets, taking into account the frequency of use of applications. Consequently, the proposed model enabled the users to reduce additional (extra) costs and processing time. The article proposed a method for reducing delays by significantly reducing the network infrastructure overload by arranging the optimal distribution of the most frequently used resources by certain user groups, as well as more remote resources on the cloud network servers serving those groups.

References

[1] R.S. Somula and S. Ra, "A survey on mobile cloud computing: Mobile Computing + Cloud Computing (MCC=MC+CC)," *Scalable Computing: Practice and Experience*, vol.19, no.4, pp. 309-337, 2018.

[2] M. Ala'anzy, M. Othman, Z.M. Hanapi and M.A. Alrshah, "Locust Inspired Algorithm for Cloudlet Scheduling in Cloud Computing Environments," *Sensors*, 21, 7308, pp.1-19, 2021.

[3] A. Nasr, N.A. El-Bahnasawy, G. Attiya and A. El-Sayed, "Cloudlet Scheduling Based Load Balancing on Virtual Machines in Cloud Computing Environment," *Journal of Internet Technology*, vol.20, no.5, pp. 1376-1378, 2019.

[4] L. Lin, P. Li, J. Xiong and M. Lin, "Distributed and Application-aware Task Scheduling in Edge-clouds," *14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN) 6-8 Dec. 2018*, pp. 165-170, 2018.

[5] M. Yuyi, C. You, J. Zhang, K. Huang and K. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp.2322-2358, 2017.

[6] A. Nasir, Y. Zhang, A. Taherkordi and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal, v*ol. 5, no. 1 pp. 450-465, 2017.

[7] T. Shi, M. Yang, X. Li, Q. Lei and Y. Jiang, "An energy-efficient scheduling scheme for time-constrained tasks in local mobile clouds," *Pervasive and Mobile Computing*, vol. 27, pp. 90 – 105, 2016.

[8] R.K. Alekberov, "Strategy for reducing delays and energy consumption in cloudlet-based mobile cloud computing," *International Journal of Wireless Networks and Broadband Technologies*, vol.10, pp. 32-44, 2021.

[9] T. Zhao, S. Zhou, X. Guo, Y. Zhao and Z. Niu, "A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing," *IEEE Globecom Workshops (GC Wkshps),* pp. 1–6, 2015.

[10] D.G. Roy, D.De, A. Mukherjee and R. Buyya, "Application-aware cloudlet selection for computation offloading in multi-cloudlet environment," *J Supercomputer,* vol. 73, pp. 1672–1690, 2017.

[11] Z. Peng, D. Cui, J. Zuo J, Q. Li, B. Xu and W. Lin, "Random task scheduling scheme based on reinforcement learning in cloud computing," *Cluster Computing,* pp.1595-1607, 2015.

[12] M. Sachula, Y. Wang, Z. Miao and K. Sun, "Joint optimization of wireless bandwidth and computing resource in cloudlet-based mobile cloud computing environment," *Peer-to-Peer Networking and Applications,* vol. 11, no. 3, pp. 462-472, 2018.

[13] M. Quwaider and Y. Jararweh, "Cloudlet-based efficient data collection in wireless body area networks," *Simul Model Pract Theory*, vol.50, pp. 57–71, 2015.

[14] F. Zhang, J. Ge, Z. Li, C. Li, C. Wong, L Kong, B. Luo and V. Chang, "A load-aware resource allocation and task scheduling for the emerging cloudlet system," *Future Generation Computer Systems*. vol. 87, pp. 438-456, 2018.

[15] A. Muneera, M. Al-Ayyoub, Y.Jararweh, L.Tawalbeh and E.Benkhelifa, "Power optimization of large-scale mobile cloud system using cooperative cloudlets," *IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW),* pp.34-38, 2016.

[16] L. Tong, Y. Li and W. Gao, "A hierarchical edge cloud architecture for mobile computing," *in: Proceedings of the 35th IEEE INFOCOM*, pp. 1–9, 2016.

[17] D.K. Sajnani, A. R. Mahesar, A. Lakhan, and A. Jamali, "Latency Aware and Service Delay with Task Scheduling in Mobile Edge Computing," *Communications and Network*. vol.10, no.04, 2018.

[18] A. Beloglazov, J. Abawajy and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Gener Comput Syst*, vol.28, pp.755–768, 2012.

[19] A. Mukherjee and D. De, "Low power offloading strategy for femto-cloud mobile network," *Eng Sci Technol Int J*, vol. 19, pp. 260–270, 2016.

[20] Y. Liu, M. J. Lee and Y. Zheng, "Adaptive Multi-Resource Allocation for Cloudlet-Based Mobile Cloud Computing System," *IEEE Transactions on Mobile Computing*, vol. 15, pp. 2398-2410, 2016.

[21] X. Sun and N. Ansari, "PRIMAL: PRofit Maximization Avatar Placement for Mobile Edge Computing," *Proceedings of 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22-27 May*, pp. 1-6, 2016.

[22] G. Shreya, A. Mukherjee, S. Ghosh and R. Buyya, "Mobi-IoST: mobility-aware cloud-fog-edge-iot collaborative framework for time-critical applications," *IEEE Transactions on Network Science and Engineering*, pp.1-15, 2019.

[23] A. Mukherjee, D. Priti, D. De and R. Buyya, "IoTF2N: "An energy-efficient architectural model for IoT using Femtoletbased fog network," *The Journal of Supercomputing*, vol. 75, no. 11, pp. 7125-7146, 2019.

[24] M. Shiraz, S. Abolfazli, Z. Sanaei and A. Gani, "A Study on Virtual Machine Deployment for Application Outsourcing in Mobile Cloud Computing," *The Journal of Supercomputing*, vol. 63, pp. 946-964, 2013.

[25] N. Kumar and R. Kumar, "Scheduling of Tasks (Cloudlets) in Heterogeneous Processing Cloud Environment," *International*

Journal on Emerging Technologies, vol. 11, no.3, pp. 417-421, 2020.
[26] C. Gogos, C. Valouxis, P.Alefragis, G.Goulas, N.Voros and E.Housos, "Scheduling independent tasks on heterogeneous processors using heuristics and column pricing," *Future Gener Comp Syst,* vol. 60, pp. 48-66, 2016.
[27] L. Liu, M. Zhang, R. Buyya and Q. Fan, "Deadline-constrained coevolutionary genetic algorithm for scientific workflow scheduling in cloud computing," *Concur Comput: Pract Exper*. vol. 12, pp.3942-3954, 2017.
[28] P. Zhang and M. Zhou, "Dynamic cloud task scheduling based on a two-stage strategy," *IEEE Trans Autom Scie Eng,* pp.772-783, 2018.
[29] R.G. Alekberov and O.R. Alekperov, "Procedure of effective use of cloudlets in wireless metropolitan area network environment," *International Journal of Computer Networks & Communications*, vol. 11, no. 1, pp.93–107, 2019.
[30] Y. Shen, Z. Bao, X. Qin and J. Shen, "Adaptive task scheduling strategy in cloud: when energy consumption meets performance guarantee," *World Wide Web.* pp. 155-173, 2016.
[31] E. Ahmed, A. Akhunzada, M. Whaiduzzaman, A. Gani, S.H. Ab Hamid and R. Buyya, "Network-centric performance analysis of runtime application migration in mobile cloud computing," *Simul Model Pract Theory,* vol. 50, pp. 42–56, 2015.
[32] P. Azad and N.J. Navimipour, "An energy-aware task scheduling in the cloud computing using a hybrid cultural and ant colony optimization algorithm," *Int J Cloud Appl Comput.* pp.20-40, 2017.
[33] G.H. Bindu, K.Ramani, C.S. Bindu, "Energy aware multi objective genetic algorithm for task scheduling in cloud computing," *Int J Int Protocol Tech.* pp. 242-249, 2018.

Authors' Profiles

Rasim M. Alguliyev is a Director of the Institute of Information Technology of the Azerbaijan National Academy of Sciences (ANAS) and the Academician-Secretary of the ANAS. He is a full member of ANAS and a Full Professor. He received his BSc and MSc in Electronic Computing Machines from the Azerbaijan Technical University in 1979. He received his PhD and Doctor of Science (higher degree after PhD) in Computer Science in 1995 and 2003, respectively. His research interests include information security, e-government, data mining, big data, online social network analysis, cloud computing, evolutionary and swarm computation and scientific metrics. He is the author of more than 600 papers, four monographs, four patents and several books.

Rashid G. Alakbarov graduated from Automation and Computer Engineering faculty of Azerbaijan Polytechnic University named after C. Ildirim. He received his PhD degree in 2006 from Supreme Attestation Commission under the President of the Republic of Azerbaijan. His primary research interests include various areas in cloud computing, data processing, computer networks, virtual computing, particularly in the area of distributed computing. He is head of department at the Institute of Information Technology as of 2002. Since 2010, he has been leading the development of "AzScienceNet" infrastructure. In 2021, he was appointed an executive director of the institute by the decision of the Presidium of Azerbaijan National Academy of Sciences. He is the author of 95 scientific papers, including 5 inventions.

How to cite this paper: Rasim M. Alguliyev, Rashid G. Alakbarov, "Integer Programming Models for Task Scheduling and Resource Allocation in Mobile Cloud Computing", International Journal of Computer Network and Information Security(IJCNIS), Vol.15, No.5, pp.13-26, 2023. DOI:10.5815/ijcnis.2023.05.02