Nikola MATIJAŠEVIĆ, PhD Candidate (corresponding author)
nikola.matijasevic@sf.bg.ac.rs
University of Belgrade, Serbia

Marko ĐOGATOVIĆ, PhD
m.djogatovic@sf.bg.ac.rs
University of Belgrade, Serbia

Mladenka BLAGOJEVIĆ, PhD
m.blagojevic@sf.bg.ac.rs
University of Belgrade, Serbia

Resource Management in Cloud Data Centers Based on Optimisation of Average Utilisation

Abstract. *Cloud computing enables the delivery of computing services through the Internet, allowing access to resources on demand from any device and location. Efficient resource management in cloud data centres, including CPU, RAM, storage and bandwidth, is crucial for maximising utilisation and reducing costs. This paper proposes a novel resource management model optimised using a genetic algorithm, focusing on average utilisation of resources. Two approaches are evaluated: one adjusts the number of physical and virtual machines, while the other varies the probability of selecting virtual machine requests. By implementing this model, the research aims to enhance operational efficiency and service quality, demonstrating practical applications of metaheuristic algorithms in real-world cloud computing environments.*

Keywords: *Cloud computing, metaheuristic, resource management, resource utilisation.*

JEL Classification: C51, C52, C61.

1. Introduction

Cloud computing greatly impacts the economy since it lowers costs, increases efficiency, and encourages innovation (Haucap et al., 2022). Using subscription-based cloud services lowers costs because companies do not have to buy, maintain, and upgrade physical assets. Because of this agility, businesses can quickly roll out new applications and pull back on needed services, giving them a competitive advantage. In addition to enabling remote work and collaboration, cloud computing boosts productivity and innovation. Also, the cloud sector boosts employment by creating job openings.

The impact on the economy is reflected in various industries as well. Education uses cloud platforms for remote learning and resource sharing (Essam et al., 2023). In healthcare, it enhances patient care through telemedicine and big data analytics (Essam et al., 2023). The banking sector benefits from improved data security,

DOI: 10.24818/18423264/58.4.24.19

automated processes with faster transactions, and fraud detection (Bodemer, 2024). Cloud computing, as a rapidly developing and expanding field in the IT industry, has contributed to the transformation of the way data is managed, stored, and accessed, offering flexibility and scalability as the main advantages (Gourisaria et al., 2020). However, with the growth in demand for cloud services and diverse user needs, resource management has become a noteworthy challenge (Abid et al., 2020). There are various algorithms used for solving optimisation problems, and in this paper, the focus will be on the genetic algorithm which is part of the metaheuristic algorithm group (Ezugwu et al., 2021). The genetic algorithm, inspired by natural selection, has shown success in solving complex optimisation problems due to its modularity and simplicity (Alhijawi et al., 2023; Katoch et al., 2021). By applying this algorithm to resource management in cloud computing, the goal is to fulfil user requests while taking into account resource utilisation. This approach leads to improved efficiency and reduced costs for cloud service providers, but also to improved quality of cloud services for users.

The motivation for this research stems from several factors. First, the increasing demand for cloud services requires efficient resource management to maintain both service quality and control operational costs. Second, effective cloud resource management can lead to significant reductions in expenses related to hardware, energy, and maintenance, thereby enhancing operational efficiency. Finally, there is a need to apply theoretical models to real-world cloud computing scenarios in order to address practical problems faced by data centers.

More specifically, the main contributions of this paper are outlined as follows. The first contribution is the implementation of a new cloud data centre model for resource management, which relies on average utilisation of resources such as CPU, RAM, storage, and bandwidth. The second contribution is the evaluation of the proposed model using a genetic algorithm, applied through two approaches with real technical specifications and data for both physical and virtual machines, ensuring its relevance to actual cloud computing challenges.

The rest of the paper is organised as follows. Section two presents a review of the related literature. In the third chapter, the concept of cloud computing is explained, where the focus is on service and deployment models. The fourth chapter goes through the formulation of the problem and the model of resource management in cloud computing. In the fifth chapter, there is an analysis of the results obtained from the use of the genetic algorithm through two approaches, as well as their comparison and explanation. Chapter six provides concluding remarks.

2. Literature review

This part will give a brief overview of several papers dealing with the same topic. The paper by Tseng et al. (2018) explores genetic algorithms for dynamic allocation of resources and energy consumption in the data centre. It considers CPU resource usage, memory of virtual and physical machines, and data center energy consumption. A genetic algorithm uses historical data from previous time intervals

to predict resource requirements for the next time interval. A new hybrid algorithm for load balancing is presented by Mousavi et al. (2017). The goal is to improve the bandwidth of the cloud provider's network, and the proposed algorithm combines the TLBO (teaching-learning-based optimisation) algorithm and the GV (grey wolf optimisation) algorithm. The hybrid algorithm shows better performance compared to the individual use of these algorithms. The paper by Chen et al. (2021) presents an algorithm for urgent resource needs, establishing a priority scheme for the goal of the model to minimise the number of physical machines used for resource management. RAA-PI-NSGAII algorithm improves the quality of the solution set and accelerates the speed of problem-solving. The experimental results show that the proposed algorithm can efficiently allocate resources for urgent needs while ensuring balanced use of all resources. The results of the random forest and the genetic algorithm were compared in terms of resource management (Ganapathy et al., 2023). Components such as energy consumption, execution time and CPU utilisation were analysed. The genetic algorithm performed better in terms of energy consumption and execution time, but the random forest gave better results for CPU utilisation. The paper by Manavi et al. (2023) proposes a hybrid algorithm that combines a genetic algorithm and a neural network to improve task scheduling. The scheduler is composed of Neural Network Task Classification (N2TC), which is used for the classification of input tasks and Genetic Algorithm Task Assignment (GATA), which is used for the selection of classified tasks. This algorithm takes into account execution and response time, cost, and system efficiency and it outperforms the state-of-the-art algorithm. The S-MOAL (Spacy Multi-Objective Antlion) algorithm was proposed to minimise the makespan and cost of using virtual machines (Belgacem et al., 2020). The simulation results proved that this algorithm performed better than some relevant algorithms. Also, improvements in the fault tolerance and energy efficiency of virtual machines were noted. The paper by Gao et al. (2021) proposed a Hierarchical Multi-Agent Optimisation (HMAO) algorithm, which represents the combination of genetic and multi-agent optimisation algorithms, used to maximise resource utilisation and minimise bandwidth cost. In comparison to existing solutions, simulation results demonstrate that the HMAO algorithm is more effective in solving the problem of resource allocation with a large number of requested tasks. To optimise resource utilisation and to support efficient and on-demand computing resource management, a software-defined networking (SDN) based architecture for edge and cloud computing services in 5G heterogeneous networks was established (Du et al., 2022). The evolutionary game-based service selection and Stackelberg differential game-based resource-sharing mechanisms are designed. The simulation results confirm the effectiveness and stability of the proposed mechanisms. The WR-LA algorithm, a hybrid of the Whale Optimisation Algorithm (WOA) and Lion Algorithm (LA), provides an optimal solution for container resource allocation in cloud computing (Vhatkar et al., 2022). Comparative analysis and simulation results show that the WR-LA algorithm outperforms traditional models and significantly improves resource allocation efficiency and system reliability.

According to our knowledge, there is no evidence in the literature about the possible application of metaheuristics to optimise the resource management model in cloud computing based on average utilisation of data centers with a combination of resources such as CPU, RAM, storage and bandwidth.

3. Cloud computing

Cloud computing offers widely adopted, diverse Internet-based services. This includes servers, storage, databases, and software, accessible on demand from any device. Understanding its structure, service models, and applications is the key for optimal use. Cloud computing is constantly evolving, requiring staying updated for informed decisions. It is best understood through its two main models: service and deployment (Sunyaev, 2020).

3.1 Service models

As per Sunyaev (2020), cloud services are categorised into three models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The mentioned models are listed in descending order, from the highest to the lowest degree of user control. This hierarchy is determined by the following factors (Ruparelia, 2016):

1. Applications – various software solutions that are available for use by users;
2. Data – handling and processing of data while securing them in terms of privacy and immutability;
3. Runtime – an environment used to develop, test and execute applications and it behaves like a smaller operating system (container);
4. Middleware – software used to communicate between different applications, services, devices, databases, etc.;
5. Operating system - represents the most important software that can be used on a computer. It serves to manage computer memory and processes, but also to manage computer hardware and software;
6. Virtualisation – enables the creation of multiple virtual machines within each physical machine. They act as separate computers that users can access and use;
7. Servers represent the building block of every large cloud centre, as well as any other centre that processes large amounts of data. It is another name for physical machines (computers) with above-average performance and computing resources;
8. Storage – provision of the necessary storage space capacity and its management to meet the needs of users in terms of their requirements (amount of data);
9. Network – represents connecting and managing cloud resources such as servers, applications, data, services, etc. which are potentially at different locations.

The listed items can be grouped into layers as shown by (Parast et al., 2022). The infrastructure layer includes items 6, 7, 8 and 9, the platform layer includes infrastructure items and items 3, 4 and 5, and finally, the software layer includes

infrastructure and platform items and items 1 and 2. The infrastructure layer represents the basis on which cloud services are built. The platform layer refers to the environment for developing, testing, and executing applications. The software layer delivers applications and services to end users.

While users can opt for local implementations, cloud computing's emergence and benefits have pushed this option aside. Cloud providers offer cheaper computer resources without the need for server space, maintenance, data security, or IT expertise. This reduces or eliminates the user's resource management tasks compared to local options (Table 1).

Table 1. Available resources managed by the user locally and in different service models

	Local	IaaS	PaaS	SaaS
Applications	✓	✓	✓	×
Data	✓	✓	✓	×
Runtime	✓	✓	×	×
Middleware	✓	✓	×	×
Operating system	✓	✓	×	×
Virtualisation	✓	×	×	×
Server	✓	×	×	×
Storage	✓	×	×	×
Networking	✓	×	×	×

Source: Authors' own creation.

In the IaaS model, providers offer users the necessary cloud infrastructure like computing, storage, and network resources, following a "pay-as-you-go" approach in a virtualised environment. The user in this model refers to the IT administrators. A virtualised environment refers to the division of one large physical machine into multiple smaller virtual machines. Users benefit from this model as they don't need to buy or manage physical servers. Instead, they rent direct access to the infrastructure from the provider. IaaS is chosen when users prefer not to handle infrastructure maintenance and scalability, offering flexibility with easily adjustable resources.

The PaaS model includes everything that IaaS has, but with certain additions. From the user's point of view, the PaaS model introduces additional limitations regarding the degree of control, capabilities, and flexibility, while the opposite can be said from the cloud provider's point of view. The user in this model refers to the developers. Providers using the PaaS model manage the platform part. Users are left with the responsibility to manage data and to develop, test, and execute applications. This model is most often used for creating frameworks and applications based on the use of artificial intelligence, machine learning, business intelligence, etc.

The SaaS model is the best-known service model and represents an extension of the PaaS model because it also includes software (data and applications). The

provider manages the infrastructure, platform, and software, while the end-user just uses various desired applications. Therefore, this model is used only when there is a user need for ready-made applications in the cloud. The applications can be easily accessed through an Internet browser. Only the user interface through which the application is used is available to the end user, but it does not need to be downloaded, installed, or updated on the user's computer.

The choice of one of the three previously mentioned and described service models depends solely on the needs of the user. Therefore, it is necessary to delineate whether it is enough to rent only the infrastructure (IaaS model) or whether the platform must be included (PaaS model) or whether there is a desire to use ready-made applications (SaaS model). These three models are not the only ones; others are generally assigned to the Everything as a Service (XaaS) model (Sunyaev, 2020).

3.2 Deployment models

Deployment models describe how cloud services are provided and accessed, indicating who owns and manages the environment. The main model types are: public, private, community, and hybrid (Sunyaev, 2020).

The public model is the most used model in this group. All applications and services are hosted in the cloud, owned by the provider, and accessible through the Internet. Therefore, users do not need their hardware and rely on the provider's infrastructure, paying only for the resources they use. This model offers scalability, allowing users to adjust resources as needed, leading to cost savings. However, data security and privacy are major concerns, as the data is stored with the provider and might be accessed by others. Additionally, only basic services are typically offered.

The private cloud model is used exclusively within an organisation, requiring all necessary infrastructure on-site. Maintenance and upgrades are handled by IT experts, with access to data strictly controlled to minimise loss or misuse. While customisable, this cloud model requires significant initial investment and training.

The community cloud model is used by organisations with similar security, data protection, scalability, and other needs. Similar to the private model, only specific organisations have access. They share infrastructure, costs, and management, making it cost-effective. This model facilitates collaboration and joint projects through easy data exchange within the same cloud environment. However, it is less common due to the challenge of finding and grouping similar organisations.

The hybrid cloud model combines elements of the public, private, and community cloud models to provide greater flexibility. It enables the sharing of data and applications between these models. This model is ideal for organisations seeking the scalability of the public cloud and the privacy of the private cloud, allowing them to separate important and less important data. However, implementing a hybrid cloud can be complex and may require the hiring of additional experienced IT experts.

4. Problem and model formulation

In this paper, public IaaS is considered, and it represents one of the possible combinations of service and deployment models of cloud computing. Data center (DC) consists of *n* physical machines (PMs) of the same technical specifications. It is necessary to distribute *m* virtual machines (VMs) on *n* physical machines and maximise resource utilisation of the DC (bandwidth) and PMs (CPU, RAM, and storage). In the DC, the VMs handle user requests with bandwidth distributed to PMs as needed. PMs have processor resources (the number of virtual CPUs), RAM, and storage. The parameters and their definitions are given in Table 2.

Table 2. Definition of parameters

Parameter	Definition
$V = \{v_1, v_2, \dots, v_m\}$	A set of virtual machines
$P = \{p_1, p_2, \dots, p_n\}$	A set of physical machines
s_j	PM status
$\mathbb{V}_{ij}$	VM allocation
v_{ij}^{cpu}	CPU utilisation v_i in p_j
v_{ij}^{ram}	RAM utilisation v_i in p_j
v_{ij}^{str}	Storage utilisation v_i in p_j
$v_{ij}^{used\ vcpu}$	Number of used vCPUs v_i in p_j
$p_j^{total\ vcpu}$	Number of total vCPUs in p_j
$v_{ij}^{used\ rgb}$	Number of used GBs of RAM v_i in p_j
$p_j^{total\ rgb}$	Number of total GBs of RAM in p_j
$v_{ij}^{used\ sgb}$	Number of used GBs of storage v_i in p_j
$p_j^{total\ sgb}$	Number of total GBs of storage in p_j
$v_{ij}^{used\ tb}$	Number of used Tbs of bandwidth v_i in p_j
$DC^{total\ tb}$	Number of total TBs of DC bandwidth
p_j^{cpu}	CPU utilisation in p_j
p_j^{ram}	RAM utilisation in p_j
p_j^{str}	Storage utilisation in p_j
p_j^{bw}	Bandwidth utilisation in p_j
C^{avg}	Average CPU utilisation of *n* PMs
R^{avg}	Average RAM utilisation of *n* PMs
S^{avg}	Average storage utilisation of *n* PMs
B^{avg}	Average bandwidth utilisation of *n* PMs

Parameter	Definition
C^{max}	Maximum CPU utilisation of PM
R^{max}	Maximum RAM utilisation of PM
S^{max}	Maximum storage utilisation of PM
B^{max}	Maximum bandwidth utilisation of DC

Source: Parameters used by authors.

In this model, there are two binary parameters, namely the PM status parameter s_j and the VM allocation parameter $\mathbb{V}_{ij}$. The first parameter tells whether PM p_j is active (0 if it is not or 1 if it is), while the second one gives information about whether VM v_i is allocated (assigned) to PM p_j (0 if it is not or 1 if it is). Based on $\mathbb{V}_{ij}$, the value of the parameter s_j can be obtained as shown by equation (1).

$$s_j = \max_{\mathrm{i}} \mathbb{V}_{ij}. \tag{1}$$

The focus is on the average utilisation of the four resources that make up the components in the objective function *F* that is maximised as shown by equation (2).

$$F = C^{avg} + R^{avg} + S^{avg} + B^{avg} \longrightarrow max. \tag{2}$$

The first component of the objective function is related to the calculation of the average CPU utilisation by *n* PMs (C^{avg}). Its formulation is given by equation (3).

$$C^{avg} = \frac{\sum_{j=1}^{n} p_j^{cpu}}{\sum_{j=1}^{n} s_j}. \tag{3}$$

The parameter p_j^{cpu} represents the CPU utilisation of PM p_j and ranges from 0 to 1. It is calculated according to equation (4).

$$p_j^{cpu} = s_j \sum_{i=1}^{m} v_{ij}^{cpu}. \tag{4}$$

The parameter v_{ij}^{cpu} refers to the CPU utilisation of VM v_i within PM p_j and ranges from 0 to 1. Its calculation is given by equation (5).

$$v_{ij}^{cpu} = \mathbb{V}_{ij} \frac{v_{ij}^{used\ vcpu}}{p_j^{total\ vcpu}}. \tag{5}$$

The parameter $v_{ij}^{used\ vcpu}$ corresponds to the number of used (requested) vCPUs by VM v_i within PM p_j. The parameter $p_j^{total\ vcpu}$ corresponds to the number of total vCPUs available to PM p_j and is calculated according to equation (6). The parameter $p_j^{cpu\ number}$ is the number of CPUs used by PM p_j, p_j^{cores} is the number of CPU cores, and $p_j^{threads}$ is the number of core threads.

$$p_j^{total\ vcpu} = p_j^{cpu\ number} \cdot p_j^{cores} \cdot p_j^{threads}. \tag{6}$$

The next component of the objective function is related to the calculation of the average RAM utilisation by *n* PMs (R^{avg}). Its formulation is given by equation (7).

$$R^{avg} = \frac{\sum_{j=1}^{n} p_j^{ram}}{\sum_{j=1}^{n} s_j}. \quad (7)$$

The parameter p_j^{ram} represents the RAM utilisation of PM p_j and ranges from 0 to 1. It is calculated according to equation (8).

$$p_j^{ram} = s_j \sum_{i=1}^{m} v_{ij}^{ram}. \quad (8)$$

The parameter v_{ij}^{ram} refers to the RAM utilisation of VM v_i within PM p_j and ranges from 0 to 1 as given by equation (9). The parameter $v_{ij}^{used\ rgb}$ corresponds to the GB of RAM used (requested) by VM p_i within PM p_j. The parameter $p_j^{total\ rgb}$ corresponds to the total GB of RAM available to PM p_j.

$$v_{ij}^{ram} = \mathbb{V}_{ij} \frac{v_{ij}^{used\ rgb}}{p_j^{total\ rgb}}. \quad (9)$$

The third component of the objective function is related to the calculation of the average storage utilisation by *n* PMs (S^{avg}) as given by equation (10).

$$S^{avg} = \frac{\sum_{j=1}^{n} p_j^{str}}{\sum_{j=1}^{n} s_j}. \quad (10)$$

The parameter p_j^{str} represents the storage utilisation of PM p_j and ranges from 0 to 1. It is calculated according to equation (11).

$$p_j^{str} = s_j \sum_{i=1}^{m} v_{ij}^{str}. \quad (11)$$

The parameter v_{ij}^{str} refers to the storage utilisation of VM v_i within PM p_j and ranges from 0 to 1 as given by equation (12). The parameter $v_{ij}^{used\ sgb}$ corresponds to GB of storage used (requested) by VM v_i within PM p_j. The parameter $p_j^{total\ sgb}$ corresponds to the total GB of storage available to PM p_j.

$$v_{ij}^{str} = \mathbb{V}_{ij} \frac{v_{ij}^{used\ sgb}}{p_j^{total\ sgb}}. \quad (12)$$

The last component of the objective function is related to the calculation of the average bandwidth utilisation by *n* PMs (B^{avg}) as given by equation (13).

$$B^{avg} = \frac{\sum_{j=1}^{n} p_j^{bw}}{\sum_{j=1}^{n} s_j}. \quad (13)$$

The parameter p_j^{bw} represents the bandwidth utilisation of PM p_j and ranges from 0 to 1 according to equation (14). The parameter $v_{ij}^{used\ tb}$ corresponds to Tb of

bandwidth used (requested) by VM v_i within PM p_j. The parameter $DC^{total\ tb}$ corresponds to the total Tb of bandwidth available to the DC.

$$p_j^{bw} = s_j \frac{\sum_{i=1}^{m} \mathbb{V}_{ij} v_{ij}^{used\ tb}}{DC^{total\ tb}}. \tag{14}$$

After the objective function components, it is necessary to introduce and explain the constraints of the model. They are given by equations (15) to (23).

$$\forall i, j, m, n \in \mathbb{N}. \tag{15}$$
$$0 \leq p_j^{cpu} < C^{max},\ \forall j. \tag{16}$$
$$0 \leq p_j^{ram} < R^{max},\ \forall j. \tag{17}$$
$$0 \leq p_j^{str} < S^{max},\ \forall j. \tag{18}$$
$$0 \leq \sum_{j=1}^{n} p_j^{bw} < B^{max}. \tag{19}$$
$$0 \leq v_{ij}^{cpu}, v_{ij}^{ram}, v_{ij}^{str} \leq 1,\ \forall i, j. \tag{20}$$
$$\sum_{j=1}^{n} \mathbb{V}_{ij} = 1,\ \forall i. \tag{21}$$
$$\mathbb{V}_{ij} \in \{0,1\},\ \forall i, j. \tag{22}$$
$$s_j \in \{0,1\},\ \forall j. \tag{23}$$

The first constraint (equation (15)) says that the parameters *i*, *j*, *m*, and *n* belong to the set of natural numbers. The next three constraints (equations (16), (17) and (18)) refer to the utilisation of PM p_j in terms of all three resources (CPU, RAM and storage) and they range from 0 to some value which must be less than 1 (C^{max}, R^{max} and S^{max}). The constraint in equation (19) ensures that the utilisation of the entire DC, in terms of bandwidth, ranges from 0 to some value that is less than 1 (B^{max}). Equation (20) is a constraint that prevents VM v_i from occupying completely any of the three resources (CPU, RAM and storage) within PM p_j. The constraint given by equation (21) implies that VM v_i can be allocated only to PM p_j (one VM cannot be allocated to two or more PMs). The last two constraints (equations (22) and (23)) indicate that $\mathbb{V}_{ij}$ and s_j are binary parameters (values 0 or 1).

5. Simulation results

The results were obtained based on the equations and constraints of the previous chapter. The genetic algorithm was implemented using the Python programming language (McKinney, 2022). It is necessary to define constants for the maximum utilisation of resources as $C^{max} = 0.85$, $R^{max} = 0.95$, $S^{max} = 0.85$ and $B^{max} = 0.90$. The population (*M*) ranges from 1000 to 3500 (step 500) and the iterations (*N*) range from 50 to 350 (step 50). For each run, the time required to reach the solution was noted. The probability of mutation is defined as $p^{mut} = 0.01$. Technical specifications for VMs, PMs, and DC are given in Table 3.

Table 3. Technical specifications of physical and virtual machines and data center

		vCPU (number)	RAM (GB)	Storage (GB)	Bandwidth (Tb)
VM requests	1	1	0.5	10	0.5
	2	1	1	25	1
	3	1	2	55	2
	4	2	2	65	3
	5	2	4	80	3
	6	4	8	160	4
	7	6	16	320	5
	8	8	32	640	6
	9	16	64	1280	10
	10	24	96	1600	15
PM		128	128	3840	/
DC		/	/	/	$10 \cdot n$

Source: Calculation made by authors.

The first approach is based on fixating the probabilities of selecting VM requests, but also on changing the number of PMs and VMs. The probabilities of selection for each of the 10 potential VM requests are 5%, 5%, 5%, 10%, 10%, 25%, 30%, 5%, 2%, and 3%, respectively. Changing the number of PMs is done by changing the parameter *n* (from 10 to 20, step 1), and changing the number of VMs through *m* (from 20 to 80, step 10). The obtained results are shown in Table 4.

Table 4. Objective function for different numbers of physical and virtual machines

		Number of VMs						
		20	30	40	50	60	70	80
Number of PMs	10	1.04673	1.30101	1.34384	1.52242	1.62057	1.82633	**2.01891**
	11	0.96891	1.23995	1.15131	1.41344	1.52415	1.59124	1.75120
	12	0.97072	1.06681	1.13398	1.30298	1.39609	1.54169	1.61487
	13	0.90688	0.99932	1.02557	1.25779	1.30478	1.40504	1.52442
	14	0.93930	1.02005	0.96989	1.14036	1.25235	1.35641	1.41472
	15	0.78616	0.88909	0.92558	1.03085	1.13501	1.26724	1.30822
	16	0.85948	0.87513	0.84906	1.02062	1.08251	1.24542	1.25593
	17	0.74374	0.75714	0.83733	0.99154	1.00806	1.23107	1.16980
	18	0.87851	0.84666	0.76032	0.97024	0.97744	1.04367	1.08387
	19	0.72122	0.74061	0.74804	0.83885	0.92298	0.97727	1.06640
	20	0.71436	0.74789	0.74335	0.82168	0.90501	0.96596	1.00470

Source: Calculation made by authors.

The best value of the objective function is 2.01891 and it was achieved by combining the values of the parameters $n = 10$ and $m = 80$. However, this best result needs to be confirmed and presented differently by calculating the DC utilisation percentage. This represents the ratio between the obtained values of the objective function and the maximum possible value of the objective function. The

values calculated in this way are shown in Table 5. The maximum possible value of the objective function is calculated as $C^{max} + R^{max} + S^{max} + \frac{B^{max}}{\sum_{j=1}^{n} s_j}$.

Table 5. Data center utilisation based on the objective function for different numbers of physical and virtual machines expressed as a percentage (%)

		Number of VMs						
		20	30	40	50	60	70	80
Number of PMs	10	36.99	46.47	48.36	55.11	58.93	66.41	**73.42**
	11	34.24	44.28	41.68	51.17	55.42	58.08	63.91
	12	34.30	38.39	41.05	47.38	50.95	56.27	59.11
	13	32.05	35.97	37.13	45.74	47.62	51.43	55.80
	14	33.19	36.71	35.27	41.62	45.84	49.65	51.92
	15	28.08	32.18	33.66	37.74	41.65	46.50	48.11
	16	30.37	31.68	30.99	37.36	39.73	45.70	46.19
	17	26.56	27.53	30.56	36.30	37.07	45.18	43.10
	18	31.04	30.65	27.83	35.52	35.95	38.45	40.00
	19	25.76	26.93	27.38	30.85	34.00	36.06	39.35
	20	25.51	27.20	27.21	30.22	33.34	35.64	37.13

Source: Calculation made by authors.

For a clearer presentation and view, the obtained results of the objective function values and results of DC utilisation (expressed in percentages) are shown through the heat map in Figure 1 and Figure 2, respectively.

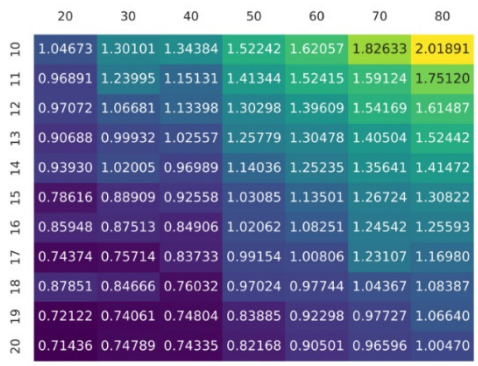

Figure 1. Heat map of the results of the objective function
Source: Calculation made by authors.

Figure 2. Heat map of data center utilisation results (%)
Source: Calculation made by authors.

The best DC utilisation is 73.42% and it also corresponds to the best value of the objective function. Accordingly, this result was achieved with the same values of the parameters, i.e., combination of values $n = 10$ and $m = 80$. The execution time is 2.69192 s. Genetic algorithm parameters are $M = 1500$ and $N = 350$.

Another approach is based on fixating the number of PMs and VMs, but changing the probabilities of selecting VM requests. Arbitrarily chosen and fixed values of the parameters are $n = 15$ and $m = 60$. For this example, 10 different selection variants are introduced for each of the 10 VM requests. They are shown in Table 6 in a way where the probability of choosing a VM request is expressed in percentages for each variant.

Table 6. Different variants of choice of virtual machine requests expressed as a percentage (%)

		VM requests									
		1	2	3	4	5	6	7	8	9	10
Different variants of choice	I	1%	3%	4%	7%	9%	11%	12%	15%	18%	20%
	II	3%	5%	6%	7%	9%	11%	12%	14%	16%	17%
	III	5%	7%	8%	8%	9%	11%	11%	13%	14%	14%
	IV	7%	9%	9%	8%	10%	10%	11%	11%	12%	13%
	V	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
	VI	12%	12%	11%	10%	10%	10%	9%	9%	9%	8%
	VII	15%	12%	11%	11%	9%	9%	8%	8%	9%	8%
	VIII	18%	14%	12%	11%	9%	9%	7%	7%	7%	6%
	IX	21%	15%	13%	12%	10%	8%	6%	6%	5%	4%
	X	23%	17%	14%	12%	10%	8%	6%	5%	3%	2%

Source: Calculation made by authors.

The selection probabilities expressed as percentages are generated to cover all possible cases. So, in variant I VM requirements are chosen in ascending order, in variant V the requirements are chosen equally, and in variant X it is in descending order. Based on these variants, the results shown in Table 7 were obtained.

Table 7. Objective function and data center utilisation (%) for different choices of virtual machine requests

		Results	
		Objective function value	DC utilisation
Different variants of choice	I	/	/
	II	/	/
	III	/	/
	IV	/	/
	V	**1.56216**	**57.45**
	VI	1.48197	54.39
	VII	1.45941	53.56
	VIII	1.25263	45.97
	IX	1.09460	40.17
	X	0.82868	30.33

Source: Calculation made by authors.

The obtained results of the objective function value and DC utilisation (expressed in percentages) are shown graphically in Figure 3 and Figure 4, respectively.

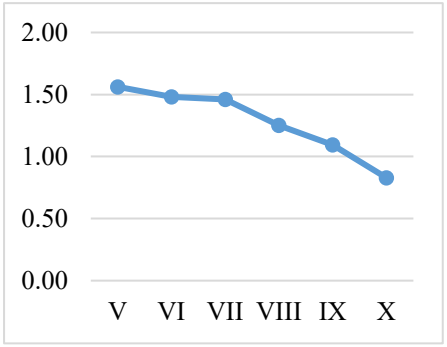

Figure 3. Graphical representation of the objective function results
Source: Calculation made by authors.

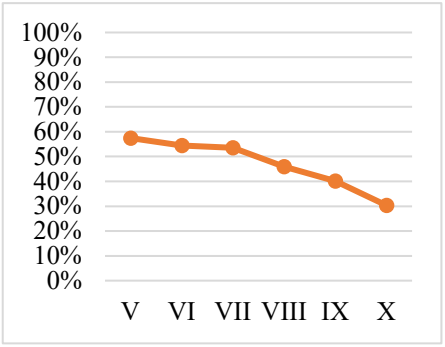

Figure 4. Graphical representation of the data center utilisation results (%)
Source: Calculation made by authors.

Results with a forward slash sign (/) indicate that the PMs and DC failed to satisfy all of the VMs requests. In other words, all VM requests claimed more resources than PMs and DC have, so these cases were dropped. This is logical because in the first variants of the selection of VM requests (I, II, III, and IV) there are the most requests that require a lot of resources. Therefore, the best result is sought from the remaining variants of the VM request selection. That result was obtained for variant V, where the value of the objective function is 1.56216, and the DC utilisation is 57.45%. The execution time is 8.09296 s. Also, the values of the genetic algorithm parameters that achieved this result are $M = 3500$ and $N = 400$. The results of the other variants (VI, VII, VIII, IX, and X) decrease compared to variant V because the number of VM requests that require a small amount of resources increases, and the number of VM requests that require a medium and a large amount of resources decreases significantly.

From the cloud provider's perspective, both approaches contribute to the same goal. Effective resource management achieves significant savings in electrical energy, preserves the longevity of computer components, and reduces the need for maintenance, all of which lead to cost savings. Additionally, optimising resource utilisation enhances overall business efficiency, enabling cloud providers to offer more reliable and scalable services. This efficiency not only improves operational performance but also allows providers to better meet the demands of their clients, fostering customer satisfaction and business growth.

6. Conclusions

For cloud computing, the use of metaheuristic algorithms can lead to more efficient use of resources, improved performance, and higher user satisfaction. This paper dealt with the use of genetic algorithm in solving the problem of resource management in cloud computing. Resource management is measured through DC utilisation. The efficiency and adaptability of the genetic algorithm make it an ideal tool for managing the complex and dynamic nature of cloud computing. Solving this problem came down to two approaches. The first approach was related to fixating

the probabilities of selecting VM requests and changing the number of PMs and VMs, while the second approach was based on fixating the number of PMs and VMs and changing the probabilities of selecting VM requests. As a result of the first approach, a DC utilisation of 73.42% was obtained with parameter values $n = 10$, $m = 80$, $M = 1500$ and $N = 350$ (execution time of 2.69192 s). The result of the second approach represents a DC utilisation of 57.45% in the variant of VM request V with parameter values $M = 3500$ and $N = 400$ (execution time of 8.09296 s).

Resource management in cloud computing significantly impacts business and the economy by optimising the use of computing resources, leading to cost savings and increased operational efficiency. Businesses benefit from reduced expenses on hardware, energy, and maintenance, allowing them to allocate resources to other strategic areas. This optimisation also enhances scalability and flexibility, enabling companies to adapt quickly to market demands. On a broader scale, these efficiencies contribute to economic growth by driving innovation, improving productivity, and creating new business opportunities across various sectors.

Future work could focus on the application of other metaheuristic algorithms or the genetic algorithm with other techniques related to selection, crossover, and mutation. On the other hand, future work could explore and solve other problems in cloud computing or deepen the problem that was the subject of this work. In addition to the above, as the field continues to develop and expand, it will be important to re-evaluate and refine the problems being solved, as well as new algorithms.

References

[1] Abid, A., Manzoor, M.F., Farooq, M.S., Farooq, U., Hussain, M. (2020), *Challenges and Issues of Resource Allocation Techniques in Cloud Computing*. *KSII Transactions on Internet and Information Systems*, 14(7), 2815-2839.

[2] Alhijawi, B., Awajan, A. (2023), *Genetic algorithms: theory, genetic operators, solutions, and applications. Evolutionary Intelligence*. *Springer*, 7, 1245-1256, https://doi.org/10.1007/s12065-023-00822-6.

[3] Belgacem, A., Beghdad-Bey, K., Nacer, H., Bouznad, S. (2020), *Efficient dynamic resource allocation method for cloud computing environment*. *Cluster Computing*, 23, 2871–2889.

[4] Bodemer O. (2024), *Revolutionizing Finance: The Impact of AI and Cloud Computing in the Banking Sector*. *ResearchGate*, https://researchgate.net/publication/378704632.

[5] Chen, J., Du, T., Xiao, G. (2021), *A multi-objective optimization for resource allocation of emergent demands in cloud computing*. *Journal of Cloud Computing: Advances, Systems and Applications*, 10, https://doi.org/10.1186/s13677-021-00237-7.

[6] Du, J., Jiang, C., Benslimane, A., Guo, S., Ren, Y. (2022), *SDN-Based Resource Allocation in Edge and Cloud Computing Systems: An Evolutionary Stackelberg Differential Game Approach*. *IEEE/ACM Transactions on Networking*, 30(4), 1613-1628.

[7] Essam, M., Nagy, S. (2023), *Cloud computing and industries transformation*. *DELL Technologies Proven Professional Knowledge, 1-17,* https://learning.dell.com/content/dam/dell-emc/documents/en-us/2023KS_Essam-Cloud_Computing_and_Industries_Transformation.pdf.

[8] Ezugwu, A., Shukla, A., Nath, R., Akinyelu, A., Agushaka, O., Chiroma, H., Muhuri, P. (2021), *Metaheuristics: a comprehensive overview and classification along with bibliometric analysis. Artificial Intelligence Review*, 54, 4237-4316, https://doi.org/10.1007/s10462-020-09952-0.

[9] Ganapathy, S., Tamilarasi, K., Natarajan, V., Periasamy, J.K. (2023), *Resource Allocation in Cloud Computing*. *Journal of Advances in Information Technology*, 14(5), 1062-1072.

[10] Gao, X., Liu, R., Kaushik, A. (2021), *Hierarchical Multi-Agent Optimization for Resource Allocation in Cloud Computing*. *IEEE Transactions on Parallel and Distributed Systems*, 32(3), 692-707.

[11] Gourisaria, M.K., Samanta, A., Saha, A., Patra, S.S., Khilar, P.M. (2020), *An Extensive Review on Cloud Computing*. *Data Engineering and Communication Technology*, 1079, 53-78.

[12] Haucap, J., Fritz, D., Thorwarth, S. (2022), *The economic impact of cloud computing in Europe*. *DICE Consult*, https://www.europeancloudalliance.com/wp-content/uploads/2022/11/Cloud-Computing-in-Europe-fin.pdf.

[13] Katoch, S., Chauhan, S.S., Kumar, V. (2021), *A review on genetic algorithm: past, present, and future*. *Multimedia Tools and Applications*, 80(1), 8091-8126.

[14] Manavi, M., Zhang, Y., Chen, G. (2023), *Resource Allocation in Cloud Computing Using Genetic Algorithm and Neural Network*, *ArXiv*. https://doi.org/10.48550/arXiv.2308.11782.

[15] McKinney, W. (2022), *Python for data analysis. O'Reilly Media Inc*. California, United States of America.

[16] Mousavi, S., Mosavi, A., Varkonyi-Koczy A.R., Fazekas, G. (2017), *Dynamic Resource Allocation in Cloud Computing*. *Acta Polytechnica Hungarica*, 14(4).

[17] Parast, F.K., Sindhav, C., Nikam, S., Yekta, H.I., Kent, K.B., Hakak, S. (2022), *Cloud computing security: A survey of service-based models*. *Computers & Security*, 114.

[18] Ruparelia, N.B. (2016), *Cloud Computing*. *The MIT Press*. Massachusetts, United States of America.

[19] Sunyaev, A. (2020), *Internet Computing. Springer Cham*. 2, https://doi.org/10.1007/978-3-031-61014-1.

[20] Tseng, F.H., Wang, X., Chou, L.D., Chao, H.C., Leung, V.C.M. (2018), *Dynamic Resource Prediction and Allocation for Cloud Data Center Using the Multiobjective Genetic Algorithm. IEEE Systems Journal*, 12(2), 1688-1699.

[21] Vhatkar, K., Bhole, G. (2022), *Optimal container resource allocation in cloud architecture: A new hybrid model*. *Journal of King Saud University - Computer and Information Sciences*, 34(5), 1906-1918.