INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research

International Journal of Cloud Computing

ISSN online: 2043-9997 - ISSN print: 2043-9989
https://www.inderscience.com/ijcc

Designing a hybrid heuristic-aided approach for replica placement and migration strategy for SaaS applications in edge cloud

Puneet Pahuja

DOI: 10.1504/IJCC.2025.10067418

Article History:

Received:	20 September 2023
Last revised:	17 February 2024
Accepted:	19 February 2024
Published online:	11 April 2025

Copyright © 2025 Inderscience Enterprises Ltd.

Designing a hybrid heuristic-aided approach for replica placement and migration strategy for SaaS applications in edge cloud

Puneet Pahuja

Accelitas Corporation,
Petaluma, CA 94952, USA
Email: puneet7498@gmail.com

Abstract: The 'replica placement and migration' mechanism for software-as-a-service (SaaS) developments in the edge cloud is developed. The placement of replica problem is rectified by utilising the hybrid position of wild geese and golden tortoise beetle (HPWGTB). For the similar data module, the different replicas should be placed on different data nodes. The multi-objective constraints such as network transmission cost, node load, and file unavailability are considered for an effective replica placement and migration. The developed hybrid HPWGTB is utilised to improve the load balancing of data nodes, decrease the response time, and reduce the resource utilisation of networks. The migration relationship between the target node and source node is considered for developing a migration of replica approach for accessing hotspots and minimising the migration time. The experimental outcomes are validated by comparing them with other optimisation approaches.

Keywords: SaaS applications in edge cloud; replica placement; replica migration; load balancing; multi-objective constraints; hybrid position of wild geese and golden tortoise beetle; HPWGTB.

Reference to this paper should be made as follows: Pahuja, P. (2025) 'Designing a hybrid heuristic-aided approach for replica placement and migration strategy for SaaS applications in edge cloud', *Int. J. Cloud Computing*, Vol. 14, No. 1, pp.1–24.

Biographical notes: Puneet Pahuja is currently a Cloud Engineer working in Acceitas Corporation. He obtained his BTech in Electronics and Communication from MDU Rohtak and participated in several high profile conferences. In addition to his academic career, he held several Cloud Architect positions in the USA. He specialised in the area of cloud and cloud governance. Number of scholars has completed research and many more are pursuing the same under his supervision. He has many papers on cloud which is one of the important areas of specialisation.

1 Introduction

There are various demands for SaaS developments like security, bandwidth, and real-time. High delays created by the cloud computing sector (Li et al., 2020a) are hard to face the service demands or result in a bad candidate experience. Edge computing is established to solve the issues in centralised cloud computing. Edge computing (Aral and

Copyright © 2025 Inderscience Enterprises Ltd.

Ovatman, 2018) intersects the storage, computing, and network assets to the network edge that is nearer to the data source or the candidate terminal. In addition, edge activities can be offered to face the requirements of speed connectivity, privacy protection, and latency-sensitive services (Li et al., 2021). In the field of cloud sector, a high number of data and variations among network and storage devices can result in errors and data loss. The development of the replica technology (Li et al., 2019a) can efficiently handle the issue. A better replica methodology can enhance the speed of transmission, reduce bandwidth utilisation, and decrease the delay of transmission when enhancing the reliability and handling the system load balance (Hao et al., 2019). In the sector of edge cloud modules, a high amount of information access and the network systems and the heterogeneity of storage systems can create issues like data failure and service interruption. And the multi-replica system helps to resolve these issues effectively (Liu et al., 2023). But, the multi-replica approach meets the issue of replica number adjustment. An unacceptable amount of replicas creates the issue of poor data service standards or large replica consistency maintenance prices. If the amount of copies requires to be enhanced, the presently joined replicas are required to exist on the information nodes (Xu et al., 2017).

The positioning of the replicas should select an appropriate region to manage the load balancing of the device and enhance functionality. But, because of the high amount of heterogeneous nodes in distinct regions of the edge cloud environment, an insufficient replica positioning approach causes the model load balancing (Du et al., 2011). Thus, how to locate the copies in the edge cloud sector is an issue worth experimenting with. In modern days, there have various experiments on replica positioning to reduce the storage device overhead and decrease the latency of access. But, the migration of creation of large replicas is created by the unacceptable replica positioning methodology that creates a high amount of data traffic. This needs to ensure that the replicas must be developed in the recent region when ignoring the network congestion of long delays (Guo et al., 2020). When the code's access load is higher than the threshold, several copies on the node are required to the moved to distinct nodes to minimise the load of the node and manage the load balancing in the overall model. Thus, the important migration of replica strategy is how to select the appropriate replica and the right migration route (Mansouri et al., 2021). 'Edge computing' is a configuration of partitioned computing that positions the important execution and information of the edge servers. 'Edge computing' offers products that are nearer to the network edge (Xiao et al., 2011).

Meeting with the relationship of large smart systems, the enormous development of data traffic, the high amount of advanced developments, and the enhancing requirement for the service quality for the candidates, several issues require to be taken into consideration (Khan, 2011). Meanwhile, the distance between the network among the users and the cloud computing sectors is large, which creates a delay when the candidates request validation and storage assets from the cloud computing platform. Moreover, the processing and transmission of a high number of data in the cloud sector result in an enhanced load on the main network and minimisation in the processing and rate of data transmission (Mansouri and Javidi, 2018). Also, the distributed storage developments commonly utilise static replica generation and arbitrary replica movement methodology for the distributed files (Hirsch and Madria, 2013), and the files are considered as the unit of replication. But, based on the tendency of the candidate's access and the information features, the replica generation mechanism with files as the section does not assure the present criteria. Initially, the candidates are only required to focus to take several

information units in the file (John and Mirnalinee, 2020). Next, the static methodology may not vary the amount of replicas based on the real-time platform. At last, the methodology may not utilise the overall merit of the storage assets due to the high private decision demands and high replication granularity (Hosseinzadeh et al., 2021). Furthermore, the general replica placement methodology may create an unbalanced information division after the model executes for a longer amount of time. It is not considered the general features of data access and the functionality of cluster nodes in the cloud sector (Tang et al., 2019).

The major goal of the implemented 'replica placement and migration' mechanism for SaaS applications in the edge cloud is explained below.

- To develop the 'replica placement and migration' methodology for SaaS developments that reduces the network cost and load balance issues and locates the replicas accurately with the aid of multi-objective optimisation.
- To rectify the issues of replica placement and migration applied the designed HPWGTB approach that minimises the migration time and tunes the multi-objective constraints such as network transmission cost, node load, and file unavailability.
- To enhance the load-balancing tasks of data nodes adopted the developed HPWGTB mechanism that minimises the usage of resources and reduces the response time of the network.
- To implement the HPWGTB scheme utilised the conventional 'golden tortoise beetle optimiser (GTBO)' and 'wild geese migration optimisation (GMO)' algorithms that improve the network functionality.
- To evaluate the functionality of the designed 'replica placement and migration' approach adopted several optimisation algorithms with certain functionality and statistical measures.

The implemented 'replica placement and migration' mechanism for SaaS developments in the edge cloud contains the following modules. The traditional tasks of the 'replica placement and migration' are summarised in module II. The basic and proposed concepts of 'replica placement and migration' are demonstrated in module III. The HPWGTB approach for the 'replica placement and migration' is illustrated in module IV. The design of the multi-objective optimisation model and its constraint specification for SaS application in edge cloud is given in module V. Module VI provides the results and descriptions of the suggested 'replica placement and migration' task. Module VII concludes the implemented task.

2 Existing works

2.1 Related works

In 2019b, Li et al. have recommended mechanisms for 'multi-objective optimised migration and replica placement' for the developments of SaaS. The issue of the multi-objective optimised replica was resolved based on the rapid 'non-dominated sorting genetic' task. Moreover, the suggested replica approach's applicability was displayed in

divergent occurrences like the transportation field, military sector, smart agriculture, and online platforms. In 2019c, Li et al. have implemented an information block as the information unit and the 'dynamic replica creation algorithm (DRC-GM)'. The suggested DRC-GM utilised the connection between the information block's access frequency and the number of replicas to frequently modify the amount of replicas to ensure the demands of data availability. Research outcomes displayed that the RP-FNSG and DRC-GM strategies in the field of edge computing highly enhanced the functionality of the model concerning storage space usage and higher effective networks.

In 2021, Huang et al. have examined an issue of generalised service replicas that have the ability to be utilised for multiple organisational criteria. Experts derived the issue into a multi-objective system with two primary scheduling ideas, containing service latency and deployment cost. These outcomes displayed that the answers achieved by the expert's methodology were qualified concerning both accuracy and diversity that were the primary estimation factors of the multi-objective model. In 2019, Yanling et al. have explored a framework for the methodology of data replica placement for the harmonised functioning of information-intensive workflows in cloud computing and collaborative edge platform. The extensive reports displayed that the suggested system outperformed these contrasted models.

In 2020b, Li et al. have suggested a resource management mechanism to fulfil the edge cloud workloads when reducing the rented node's financial prices. With the time enhancement, the suggested resource management model reduced the default rate and overall financial cost and enhanced the usage of the 'central processing unit (CPU)'. In 2022, Mohammadi and Navimipour have deployed an advanced replica placement to enhance the replica price and mean access period utilising the ACO and fuzzy logic. The measure of fuzzy membership was also employed to establish the degree of every node according to the four features. The experimental outcomes displayed that the replica cost and the access time were enhanced contrasted to other algorithms of the replica.

In 2014, Kumar et al. have constructed the 'workload-aware data placement and replication' model for decreasing the utilisation of resources. Importantly experts observed and designed the predicted task load as a hyper-graph and enhanced the separating methodologies that reduced the mean query time. Moreover, experts displayed that the recommended system enhanced the overall throughput and transaction latencies by reducing the distributed transaction amounts. In 2022, Zade et al. have employed the fuzzy system and 'ant lion optimiser (ALO)' algorithm to establish the replica numbers. The experimental outcomes showed that the implemented system resolved the optimisation issues efficiently.

2.2 Research gaps and challenges

The common problem that occurs in the load balancing of the edge cloud cluster during the replica placement. A strong consistency strategy is utilised in the distributed file system. But, it is not applicable for all edge cloud platforms. The network congestion and insufficient bandwidth cause a delay in replica management. Also, it decreases the data availability and makes the replicas not able to connect in real-time. Table 1 provides the advantages and challenges of existing 'replica placement and migration' mechanisms for SaaS developments in the edge cloud. 'Fast non-dominated sorting genetic algorithm' (Li et al., 2019b) reduces the response and migration time. It effectively balances the load of data nodes. However, the energy and cost requirement of the replica placement is high

and it does not taken the extra I/O overhead created by the process of migration. DRC-GM (Li et al., 2019c) ensures the data availability requirements by adjusting the total number of replicas and it provides better effective network with a shorter access response. But, it does not provide enormous replicas for necessary information and it does not consider the problems of data migration after the failure of the node. MRPACO (Huang et al., 2021) can reduce the service latency and it is utilised to minimise the deployment cost. Yet, it provides limited scalability. ITO (Yanling et al., 2019) performs well with less computing budget and it is used to rectify data-intensive problems. But, it does not consider the cluster resource management and data distribution issues and it does not focus on the replication dependency problems. Resource management algorithm (Li et al., 2020b) minimises the storage overheads with high user experience and it is used to ensure data consistency. However, it does not manage the resource allocation problems between cloud and edge. Mohammadi and Jafari (Mohammadi and Navimipour, 2022) use less cost for replica placement and improved the access time and it effectively places the replica based on the user requirement. Still, it suffers from overfitting problems. In SWORD (Kumar et al., 2014), the resource and cost requirement is low and it is used to reduce the mean query span. But, the implementation time is high. ALO-Tabu (Zade et al., 2022) is used to increase the balancing of load for data nodes and it highly minimises the access time. However, it suffers from a lack of diversity and it is highly sensitive. Therefore, an advanced 'replica placement and migration' mechanism for SaaS developments in the edge cloud is developed.

Table 1 Features and challenges of existing replica placement and migration strategies for SaaS applications in edge cloud

Author [citation]	*Methodology*	*Features*	*Challenges*
Li et al. (2019b)	Fast non-dominated sorting genetic algorithm	• It reduces the response and migration time.	• The energy and cost requirements of the replica placement are high.
		• It effectively balances the load of data nodes.	• It does not taken the extra I/O overhead created by the process of migration.
Li et al. (2019c)	DRC-GM	• It ensures the data availability requirements by adjusting the total number of replicas.	• It does not provide more replicas for important data.
		• It provides a better effective network with shorter access responses.	• It does not consider the data migration problems after the failure of the node.
Huang et al. (2021)	MRPACO	• It can reduce the service latency.	• It provides limited scalability.
		• It is utilised to minimise the deployment cost.	

Table 1 Features and challenges of existing replica placement and migration strategies for SaaS applications in edge cloud (continued)

Author [citation]	*Methodology*	*Features*	*Challenges*
Yanling et al. (2019)	ITO	• It performs well with less computing budget.	• It does not consider the cluster resource management and data distribution issues.
		• It is used to rectify data-intensive problems.	• It did not focus on the replication dependency problems.
Li et al. (2020b)	Resource management algorithm	• It minimises the storage overheads with a high user experience.	• It does not manage the resource allocation problems between cloud and edge.
		• It is used to ensure the data consistency.	
Mohammadi and Jafari (2022)	Fuzzy logic	• It uses less cost for replica placement and improves the access time.	• It suffers from overfitting problems.
		• It effectively places the replica based on the user's requirement.	
Kumar et al. (2014)	SWORD	• The resource and cost requirement is low.	• The implementation time is high.
		• It is used to reduce the mean query span.	
Zade et al. (2022)	ALO-Tabu	• It is used to increase the balancing of load for data nodes.	• It suffers from a lack of diversity.
		• It highly reduces the access time.	• It is highly sensitive.

3 Novel replica placement and migration strategy for Saas applications in edge cloud: hybrid heuristic algorithm

3.1 Replica replacement

The issue of replica positioning (Li et al., 2019b) is the concentration of experiments in the management of replicas. Various experts have attained overwhelming outcomes in the replica management research. The experts suggested an approach of concentrating on the 'virtualised Hadoop' to handle the file replicas and virtual machines in a coordinated way to minimise power utilisation and decrease the waste of assets. Several experts developed a replica placement approach according to the genetic algorithm to decrease the memory overhead and face the latency request. Some of them designed the methodology of 'multiple-replica-to-multiple-service' positioning by taking the availability and

reliability. Further, several scholars experimented with the issue of replica placement with the static overhead and developed an algorithm for the replica placement according to the 'write awareness' to decrease the response period and assure the quality of the service without minimising the reliability. Also, several scholars presented a modern indexing approach in the replica placement to minimise the cost of the index and enhance the system functionality by generating indices automatically with minimal performance when the tasks are processing. Further, few researchers have a replica placement based on the data dependency that focuses on ensuring the safety of the data and attaining rapid access. In addition, an advanced 'dynamic adaptive data replica placement' approach was recommended by scholars for higher fault tolerance and scalability also the capability to manage the tasks. A distributed data distribution scheme was deployed that concentrates the replica deletion, replica positioning, and creation of replica and then simultaneously observes the requests of the candidates from the edge nodes. Subsequently, a replica placement approach was adopted by the other experts that assure the reliability of the content distribution and decreases the latency of the network when assuring a better placement methodology for the server and client start-up. Then, some of them provided a 'dynamic data placement strategy (DDPS)' for establishing the optimal region according to the replica heat. The recommended approach dynamically varies the replica of data secured on every node in the 'heterogeneous Hadoop cluster' and minimises the response period of the developments of big data. Finally, several scholars presented the approach of reliable scheduling according to the replica placement taking both safety and reliability utilising the 'game theory'.

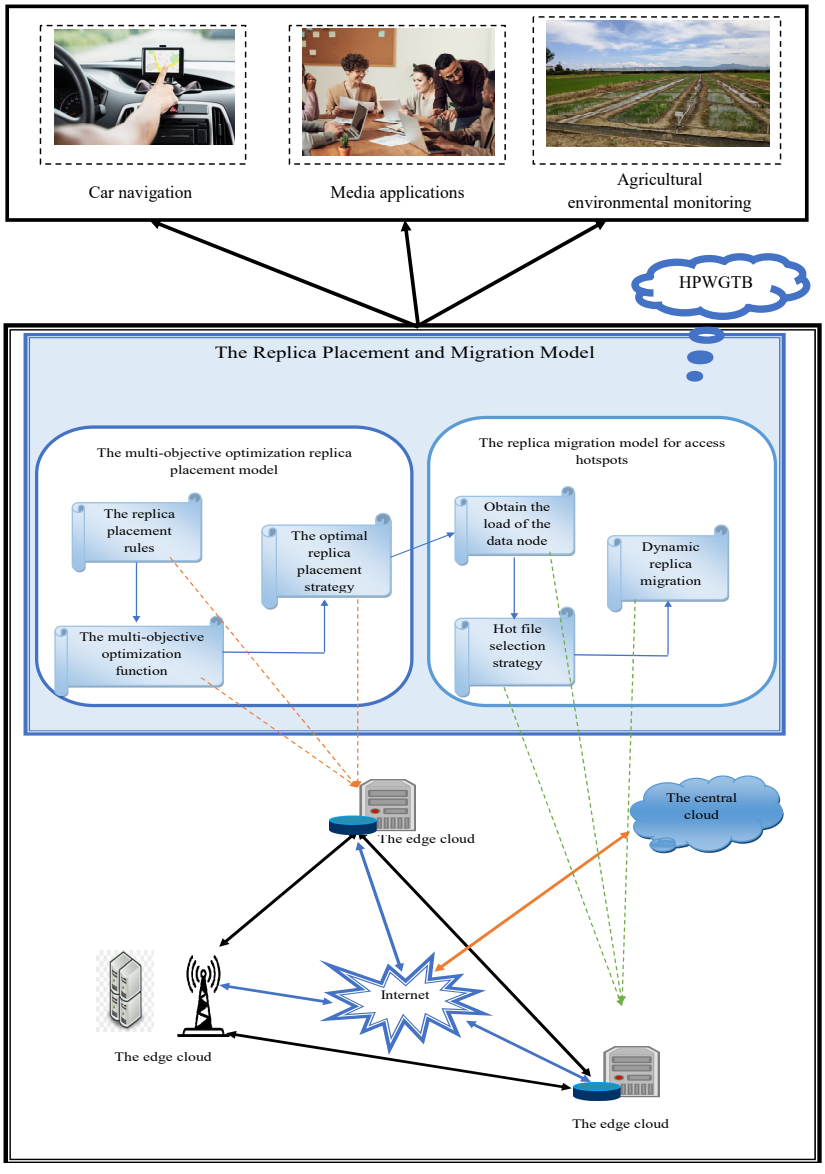
The explained works of literature have conducted the experiments based on the replica placement strategy. But, lots of similar works only focus on the section of influencing attributes like candidate access cost, distance from candidate nodes and bandwidth utilisation of network, etc. Here, no entire consideration and understanding of the replica placement. Moreover, the methodologies of 'replica placement' cloud sectors or storage systems are mostly taken instead of edge cloud sectors. So, the 'multi-objective optimised replica placement' approach in the edge cloud sector is recommended to concentrate on divergent attributes to create the placement of the replicas.

3.2 Replica migration

Since there are a lot of hurdles in the migration of replicas in the cloud sector, scholars have performed various experiments in industry and academia. Several experts provided an approach of migration to do the replication process energy efficiently and reliably and attain the fault-tolerance estimations. Some of the experts suggested an effective replica migration process for minimising the data availability and reducing the migration period and preventing the congestion of the network. A 'multi-attribute-aided computational migration' mechanism resolution was recommended by the researchers for handling where and when to migrate to assure a user experience quality and reduce the response time of the candidate. Moreover, the 'dynamic adaptive data replication migration' mechanism was deployed by some of the scholars to realise higher reliability and scalability when enhancing the capacity to manage workload variations. In addition, some of the experts recommended a methodology of migration that helps to perform seamless migration among the remote and local replicas to improve the functionality and

balancing of load while the migration is forced by the failover. The scholars suggested a 'co-migration algorithm' in that every server provides a service replica to reduce the service provider's network cost and the service latency. For data migration, a 'deep reinforcement learning' approach was designed in the edge cloud platforms with the aim of minimising the latency and enhancing the usage of resources. Furthermore, a replica migration approach was suggested by utilising the 'non-dominated sorting genetic algorithm-II' to decrease the migration of replica overhead when taking the locality and availability of the files.

Figure 1 The framework of the implemented replica placement and migration for SaaS application in edge cloud (see online version for colours)



The approaches of replica migration mentioned in earlier experiments attained higher availability, and scalability and improved the fault tolerance. Very few of the experiments are only concerning the balancing of load. Hence, the migration of a replica approach for hotspot access according to the load balancing is suggested.

3.3 Proposed optimisation model for SaaS application in edge cloud

Edge cloud computing is a sector according to the basis of cloud computing mechanism, and the ability of edge environment is constructed on the edge framework. It can minimise the bandwidth cost and pressure by observing the storage, data evaluation, and validation on the network. Moreover, it can offer cloud services like power distribution and network scheduling. Edge cloud computing can offer assets that are nearer to the candidates are minimise the period of response. However, it still meets several hurdles in solving the overload issues because of its restricted ability. The multi-replica methodology can make various data copies for the edge cloud device and secure them in distinct information nodes that enhance the data standard and availability. A better replica methodology can enhance the data communication speed, decrease the utilisation of bandwidth, and decreases the data transmission delay. It also enhances reliability and manages the load balance. However, due to the high amount of heterogeneous nodes in divergent regions of the edge cloud network, the insufficient approach of replica positioning troubles the network's load balancing. The designed 'replica placement and migration' for SaaS applications in the edge cloud are illustrated in Figure 1.

After the generation of replicas, the replicas are required to be positioned correctly to assure the functionality of the data access. Hence, a 'multi-objective optimised replica placement' approach is suggested to achieve the mechanism of better replica placement. Generally, the copies are positioned geographically partitioned to enhance the availability of the data and reduce the delay of access. But, the approach of replica positioning may create more replicas to be developed, leading high amount of data traffic. Without creating congestion in the network, it is essential to establish a better basic replica of the copied data and the correct routes to deliver the information. Thus, the migration of the replica mechanism for hotspots access according to the load balancing is suggested to attain the optimal migration of replica methodology. The issue of replica placement is handled based on the designed HPWGTB algorithm. Based on the explained replica placement condition, various attributes such as network transmission cost, node load, and file availability are taken. In the end, the replicas management consistency is carried out. In addition, the replica migration mechanism for hotspot access according to the load balancing initially generates the sector of node load determination. The HPWGTB task is adopted to enhance the data node's load balancing, minimise the usage of resources, and decrease the response period of the network. The migration connection between the source and target node is taken for enhancing the migration of replica system for accessing hotspots and to reduce the migration period. The research solutions are evaluated by contrasted with other optimisation mechanisms.

4 Hybrid position of wild geese and golden tortoise beetle for optimisation in replica placement and migration strategy

4.1 Traditional GMO

The classical GMO (Wu et al., 2022) task is adopted from the environmental features of the wild geese. The wild geese perform a unique formation for the small group migration process over long distances for reproduction and survival. In order to assure air flight continuity, every migration group of wild geese requires modification of their formation. The mathematical model and the algorithmic principles of the existing GMO are provided below.

The population of the conventional GMO is created randomly in the answer space and particular amounts of wild geese are elected as the initial head geese. The population dimension of the wild geese is pointed as A and the amount of head geese is given as B. The starting radius dimension of the migration set is referred to as $R\left(R=\frac{ud-ld}{A}\right)$. The formulation of the migration set formation is given in equation (1).

$$\begin{cases} z_j^u - z_k^u, & \text{if } j - c*(k-1)+1, \\ z_j^u - z_k^u - R + 2R*rd(1,\dim), & \text{else}, \end{cases} \tag{1}$$

Here, the attribute z_j^u refers to the place of the j^{th} separate one at the u^{th} execution (j = 1, 2, …, A). The highest amount of iteration is denoted as U (u = 1, 2, …, U). The term z_k^u indicates the k^{th} place of the separate head goose at the u^{th} execution (k = 1, 2, …, B). The amount of migration sets is given as $b\left(b=\frac{A}{B}\right)$. The mathematical model of the flight approach of the migration set is expressed in equation (2).

$$z_j^{u+1} = z_j^u + f_1\left(z_{best}^u - z_k^u\right) + f_2\left(z_l^u - z_k^u\right) \tag{2}$$

Here, the initial flight step size is denoted as f_1. Further, the term z_{best}^u points to the individual's global optimal measure and the unspecified individual head goose are indicated as z_l^u. The variables z_k^u and z_j^u refer to the migration group's head goose and the member correspondingly. The second flight step size is termed as f_2 and it is estimated in equation (3).

$$\begin{cases} f_2 = \exp\dfrac{fit(k) - fit_{ave}}{fit_{wrst} - fit_{bst}}, & fit(k) \le fit_{ave}, \\ f_2 = \exp\dfrac{fit(k) + fit_{ave} - 2fit_{bst}}{fit_{wrst} - fit_{bst}}, & fit(k) > fit_{ave}, \end{cases} \tag{3}$$

The fitness measure of the head goose is given as $fit(k)$. The variables fit_{bst}, fit_{wrst} and fit_{ave} point to the 'best, average, and worst fitness' measures of the head geese correspondingly. The variable f_2 is often utilised to handle the knowledge data's proportion of the other head geese.

The foraging and resting operations are unavoidable for the wild geese groups while long-distance migration. In the free foraging task, the wild geese group candidates explore arbitrarily based on the head goose data and continue a particular relation in a minimal region. After processing the foraging task, the wild geese reform and migrate. Equation (4) provides the foraging approach mathematically.

$$p_1 = z_j^u + f_3(z_k^u - z_j^u + R) + f_4\left(z_{bst}^u - z_k^u\right) \tag{4}$$

The arbitrary integers f_3 and f_4 are referred to as and they lie from 0 to 1 accordingly. The updated position is indicated as p_1. These integers are utilised to manage the step size movement of individuals while performing the foraging task. The group range's radius R is utilised to manage the length among the migration set candidates and the head goose.

While performing the long-distance migration, it is necessary to change the head goose often to attain high flight durability. Hence, a better individual goose is chosen from every migration group and then chosen as the heed geese of the next generation. This approach permits to achieve the better region and also assures the separation of the head geese's places. It helps to trade off the exploration and exploitation of GMOs.

After the exchange of the head geese, the migration group radius R is reduced and formulated in equation (5). It is aimed to raise the candidate's density in the set and develop the correctness of the exploration phase.

$$R = R * \left(1 - 0.1\left(\frac{u}{U_{\max}}\right)\right) \tag{5}$$

Here, the present iteration and the maximum execution are referred to u and $U_{\max}$ accordingly. Algorithm 1 depicts the pseudo-code for conventional GMOs.

Algorithm 1 Traditional GMO

Consider the iteration counts and population variables.
Fitness function validation
For u = 1 to $U_{\max}$
  For j = 1 to N_p
    Generate the migration group utilising equation (1)
    Estimate the flight mechanism applying equation (2)
    Update the better position p_1.
  End
End
Iterate the steps till obtain its optimum
Provide the optimal solutions.

4.2 Traditional GTBO

GTBO (Tarkhaneh et al., 2021): the classical GTBO is the bio-inspired meta-heuristic model. It has employed the functionalities of the golden tortoise beetle. The golden tortoise has the behaviour of changing its colour. The GTBO's functionality is presented as follows. Assume two parameters such as d and e and these are presented at the upper

part of the each other in the format of the stack. The optical thickness of each layer is considered as a 1/4 of wavelength. The expression is $b_d \cdot c_e = b_e \cdot c_e$, here the thicknesses of the layers are termed as b_d and b_e correspondingly. And then the variables c_d and c_e point to the reflecting indices accordingly. Equation (6) derives the higher reflective index.

$$y \cdot \alpha = \left(c_d \cdot c_d \cdot \cos\left(\theta_d\right) + b_e \cdot c_e \cdot \cos\left(\theta_e\right)\right) \tag{6}$$

The reflected light's wavelength is denoted as α and the term y refers to the constant measure. The terms θ_d and θ_e represent the common angles. 'Vigneron's' scheme is utilised in order to establish the dominating wavelength α. This task is derived in equation (7).

$$\alpha = \frac{2\gamma\sqrt{\sigma^2 - \sin^2\left(\theta_f\right)}}{j} \tag{7}$$

Here, the variable σ terms the reflective index, and the variable θ_f denotes the general angle. The layer's thickness and the constant integer are indicated as γ and j correspondingly. The issue factors are generated in the matrix format in a dimension of $o \times c$. This is given in equation (8).

$$R_{beetle} = \begin{pmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,c} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,c} \\ \vdots & \vdots & \ddots & \vdots \\ z_{o,1} & z_{o,2} & \cdots & z_{o,c} \end{pmatrix} \tag{8}$$

The factors c and o denote the variable amount and the beetle amount accordingly. The profit function of the GTBO is formulated in equation (9).

$$H_{beetle} = \begin{pmatrix} h\left[\left(x_{1,1} & g_{1,2} & \cdots & g_{1,c}\right)\right] \\ h\left[\left(x_{2,1} & g_{2,2} & \cdots & g_{2,c}\right)\right] \\ \vdots & \vdots & \ddots & \vdots \\ h\left[\left(x_{o,1} & g_{o,2} & \cdots & g_{o,c}\right)\right] \end{pmatrix} \tag{9}$$

The fitness variable is denoted as g and every beetle fitness measure is given as H_{beetle}.

The solution creation is derived in equation (10) for the mature beetle numbers.

$$W_j^H = D_j^I + U_{color} \cdot \left(D_{t1}^I - D_{bst}^I\right) \tag{10}$$

The creation of the current female beetle I in the place is indicated as D_j^I. It is subject to the creation of the male golden beetle I in the place D_{t1}. The colour-changing operator is denoted as U_{color} and the beetle's best fitness is denoted as D_{bst}^I. The derivation of the colour-changing operator U_{color} is given in equations (11) and (12).

$$U_{color} = \left(b_d \cdot c_d \cdot \cos\left(\theta_d\right) + b_e \cdot c_e \cdot \cos\left(\theta_e\right)\right) + (y \cdot \alpha) \tag{11}$$

Here,

$$\begin{cases} b_d, c_d = Rn() \\ b_e, c_e = Rn() \cdot \beta \\ \sigma = cauchy(\mu, \rho) \\ \theta_d, \theta_d = \beta \\ \eta, j, y = rd() \\ \theta_e = 2 \cdot \pi \cdot rd() \end{cases} \tag{12}$$

The general arbitrary measure is denoted as *Rn*() and that generates the numbers with the range of [1, *c*]. Further, the variable *rd*() indicated as the uniform arbitrary measure that lies in the limit of [0.1, 0.9]. The factor *Cauchy*() refers to the 'Cauchy distribution function'.

The validation of the survival operator is formulated in equation (13).

$$\begin{cases} beetle_1 = \gamma \cdot d_{t1} + (1-\gamma) \cdot (d_{t2} - \eta_1) \\ beetle_2 = \gamma \cdot d_{t2} (1-\gamma) \cdot (d_{t1} - \eta_2) \end{cases} \tag{13}$$

The generally chosen measures are pointed as d_{t1} and d_{t2} and equation (13) shows the evaluation of the measures η_1 and η_2.

The basically chosen attributes are pointed as c_{s1} and c_{s2}. The validation of the attributes v_1 and v_2 is expressed in equation (14).

$$\begin{cases} \eta_1 = (1-p) \cdot (d_{bst} - d_{t1}) \\ \eta_2 = (1-p) \cdot (d_{bst} - d_{t2}) \\ p = \dfrac{\gamma \cdot J}{|r|^{\beta}} \end{cases} \tag{14}$$

The general numbers associated with the solution dimension are ϑ and r.

The factors μ and q are the general integers included with the dimension of the solution. In the end, the position of the GTBO is updated and referred to as p_2.

Algorithm 2 Traditional GTBO

Consider the iteration counts and population variables.
Fitness function validation
For u = 1 to $U_{\max}$
  For j = 1 to N_p
    Construct the issue matrices utilising equations (8) and (9).
    Create the solution applying equation (9)
    Estimate the colour-changing operator employing equation (11).
    Update the better position p_2.
  End
End
Iterate the steps till obtain its optimum
Provide the optimal solutions.

4.3 Adaptive concept of HPWGTB

The features of traditional GMO and GTBO are utilised to construct the HPWGTB. The conventional GMO is motivated by the character of the wild geese swarming. The wild geese utilise a unique manner for the migration in long-distance for the reproduction and the survival of the small groups. On the other hand, the classical GTBO is motivated by the golden tortoise functionalities. This approach is designed according to the survival mechanism and the 'beetle's dual attractiveness' to create new answers for the optimisation issues. The traditional GMO has better computational functionality contrasted to the other existing models. Also, it has better optimisation outcomes, better competitiveness in difficult problems, and strong applicability. Also, the conventional GTBO is highly effective and resolves the engineering complexities. However, the existing GMO utilises more attributes that enhance the computational burden also its initial flight step size utilises the arbitrary integer, so that the effectiveness of the model is decreased. When considering the GTBO, it troubles solving the complex optimisation issues. So, to solve the conventional model's issue, it develops the approach named HPWGTB. In this approach, according to the optimal position, the process performs. The introduced derivation of the designed HPWGTB is expressed in equation (15).

$$p = \frac{p_1 + p_2}{100} \tag{15}$$

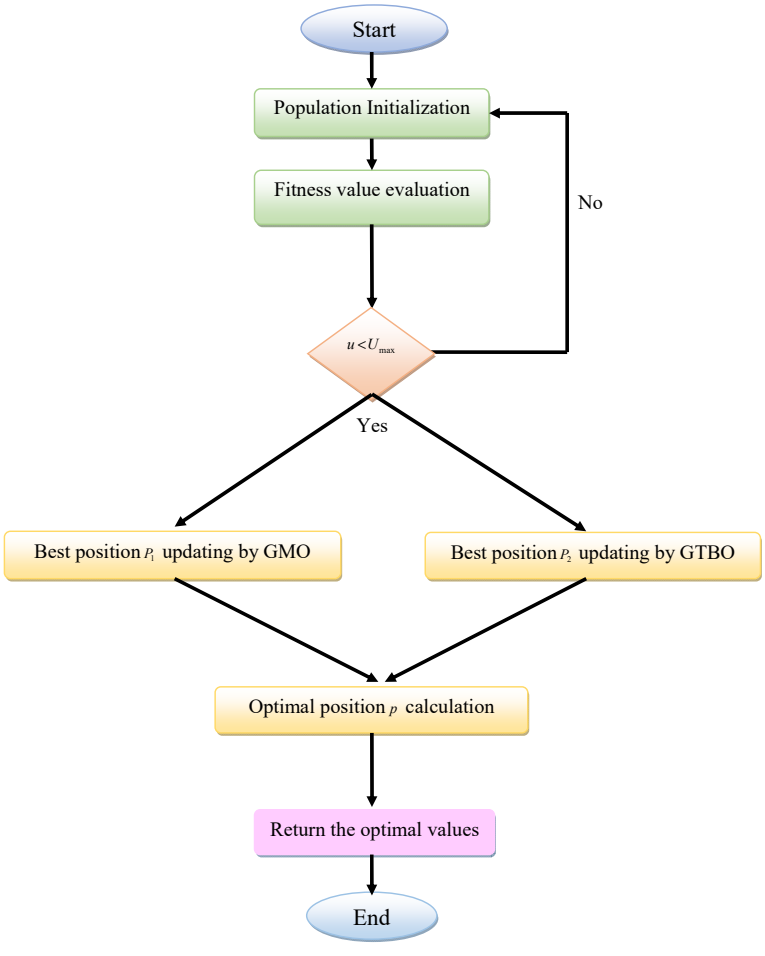
Algorithm 3 Recommended HPWGTB

Consider the iteration counts and population variables.
Fitness function validation
For $u = 1$ to U_{max}
  For $j = 1$ to N_p
    Initialise the attribute p.
    GMO approach
    Generate the migration group utilising equation (1)
    Estimate the flight mechanism applying equation (2)
    Update the better position p_1.
    GTBO approach
    Construct the issue matrices utilising equations (8) and (9).
    Create the solution applying equation (9)
    Estimate the colour-changing operator applying equation (11).
    Update the better position p_2.
    Calculate the optimal position utilising equation (15).
  End
End
Iterate the steps till obtain its optimum
Provide the optimal solutions.

The updated optimal position is referred to as p. Further, the variables p_1 and p_2 refer to the good positions of the conventional GMO and GTBO correspondingly. The flowchart

of the implemented HPWGTB is shown in Figure 2 and then the pseudo-code is depicted in Algorithm 3.

Figure 2 The flowchart of the developed HPWGTB (see online version for colours)



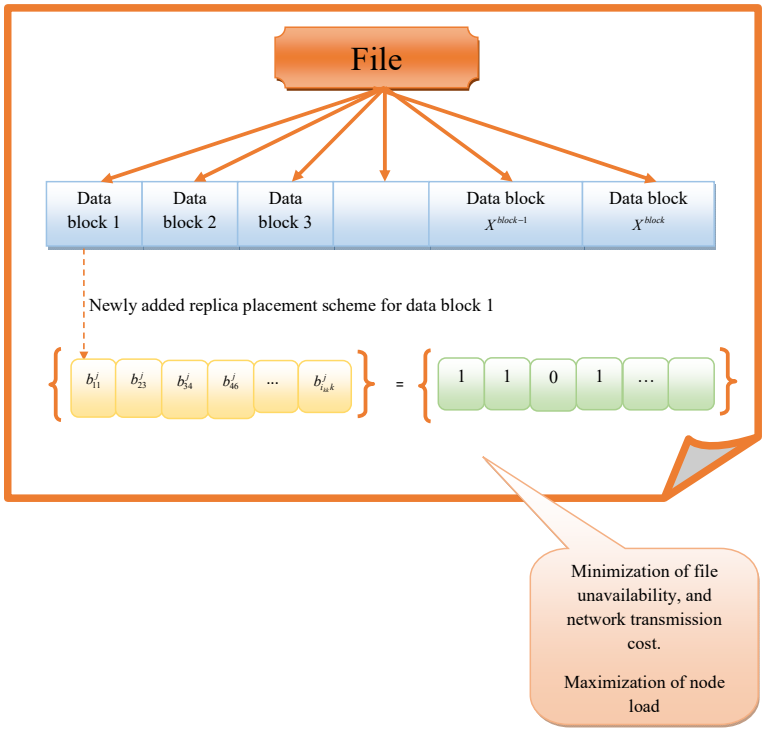
5 Design of multi-objective optimisation model and its constraint specification for SaaS application in edge cloud

5.1 Optimisation for replica placement and migration

It is well known that the issue of replica placement according to the 'multi-objective optimisation' is 'NP-hard'. In the sector of economics, there is one solution named 'Pareto solution' but that is a 'non-inferior' answer set. In addition, the issue of replica positioning is a problem of combinatorial optimisation denoted by the 'travelling salesman problem (TSP)'. Moreover, the issue of replica migration such as file access hotspots created by the burst requests is emerged. Hence, these issues are resolved by the

designed HPWGTB approach. This is performed based on the position updating that improves the convergence and network functionalities. Figure 3 displays the 'chromosome coding' mechanism of the suggested 'replica placement and migration' strategy for SaaS developments in the edge cloud.

Figure 3 The illustration of the chromosome coding task in designed replica placement and migration mechanism for SaaS applications in edge cloud (see online version for colours)



5.2 Objective function

In the placement of the replica approach for the information blocks, various attributes are required to be taken that involve the network transmission cost, node load, and file unavailability. The derivation of the multi-objection optimisation for the 'replica placement and migration' is formulated in equation (16).

$$\begin{cases} \min F(y) = [f_1(y), f_{21}(y), \ldots, f_m(y)] \\ \text{s.t. } y \in Y \\ Y \subseteq R^n \end{cases} \tag{16}$$

The objective function is pointed here as $F(y)$ that is the combined function that takes some of the sub-objective functions. The function integrated factor is denoted as R'' and the variable Y indicates the collection of solutions that fulfil the attributes. The feasible

solution is represented as *y* and that fulfils the attributes. Hence, with the fundamental concept of replica placement and migration issues can be expressed in equation (17).

$$F(y) = \arg\min_{\{Dn^x\}} \left[fu + ntc + \frac{1}{nl} \right] \tag{17}$$

Here, the attributes *fu* and *ntc* denote the 'file unavailability and network transmission cost'. Then, the 'node load' is pointed as *nl*. The term Dn^x refers the replicas which are assigned to which Data nodes.

5.3 Definition of constraints

The factors which are employed in the designed replica placement and migration approach are described as follows.

5.3.1 File unavailability fu

Assume the *n* data blocks presented in the file *s* indicated by $\{c_1, c_2, \ldots, c_n\}$. For the c_i data block, there are j_r replicas. If entire nodes for the data block c_i are not available then the data block c_i is also not available. Consider that every node is independent of everyone. The factor $P(\bar{U}A_i)$ denotes the data node's A_i unavailability likelihood. Thus, the likelihood of the unavailable data block c_i is expressed in equation (18).

$$\begin{aligned} P(\bar{C}A_i) &= P(\bar{U}A_1 \times \bar{U}A_2 \times \ldots \times \bar{U}A_{jr}) \\ &= P(\bar{U}A_1) \times P(\bar{U}A_2) \times \ldots \times P(\bar{U}A_{jr}) \end{aligned} \tag{18}$$

Based on the classical theory, it is considered that the data node reliability enables the 'exponential distribution'. Various copies are in distinct information nodes. Thus, with the connection between the data node reliability and the various replicas, the unavailable data block c_i likelihood is derived in equation (19).

$$P(\bar{U}A_i) = \prod_{i=1}^{j_r} (1 - e^{-\sigma_i Uj_r}) \tag{19}$$

Here, the j_r replica lifetime is indicated as Uj_r for the predicted measure of the data node failure in the time *U*. And the variable σ indicates the rate of node failure. To ensure the availability of files, it is essential that the presence of data blocks. The data block unavailability makes the overall file unavailable. So, equation (20) formulates the probability of file unavailability.

$$\begin{aligned} P(\bar{S}A_i) &= P(\bar{C}A_1 \cup \bar{C}A_2 \cup \ldots \cup \bar{C}A_n) \\ &= \sum_{r=1}^{n} P(\bar{C}A_r) - \sum_{1 \le r < k \le n} P(\bar{C}A_r \cap \bar{C}A_k) \\ &\quad + (-1)^{n-1} P(\bar{C}A_1 \cap \bar{C}A_2 \cap \ldots \cap \bar{C}A_n) \end{aligned} \tag{20}$$

Consider that every data section is independent of everyone. The data block c_i unavailability does not trouble the data block

$c_k \cdot P(\overline{C}A_r \cap \overline{C}A_k) = P(\overline{C}A_r) \times P(\overline{C}A_k), r \neq k.$ Based on equations (19) and (20), the file S unavailability probability is given in equation (21).

$$P(\overline{S}A) = \sum_{r=1}^{n} (-1)^{r+1} D_n^r \left(\prod_{i=1}^{j_r} 1 - e^{-\sigma_i U j_r} \right)^r \tag{21}$$

The expectation of the file s availability is represented as $1 - P(\overline{S}A) \geq A_{except}$ and A_{expect}.

5.3.2 Node loads nl

To estimate the node load, disk space usage S_{disk}^i, I/O access rate $S_{I/O}^i$, bandwidth utilisation S_{bw}^i, memory utilisation S_{mu}^i, and CPU utilisation S_{CPU}^i are employed. The node's i load $K(i)$ is derived in equation (22).

$$K(i) = w_1 S_{CPU}^i + w_2 S_{mu}^i + w_3 S_{bw}^i + w_4 S_{I/O}^i + w_5 S_{disk}^i \tag{22}$$

Here, the factors w_1, w_2, w_3, w_4 and w_5 and $K(i) \in [0, 1]$ denote the five metric weight coefficients. These coefficients of weights denote the necessity of every data node load indicator and fulfil $\sum w_i = 1$. To attain the weight coefficients, the analytic hierarchy approach is utilised.

- For the data block, the minimum the CPU utilisation is, the lesser the load of CPU, and the load ability of the CPU is stronger.
- For the data block, the minimum occupancy of storage is, the lesser the storage load, and the storage load ability is stronger.
- The bandwidth of the network troubles the cluster's load management. The minimum the bandwidth utilisation is, the more private the cluster is.
- For the placement of the replica, the ability of the disk to write and read is necessary.
- For the data block, the minimum the disk space utilisation is, the lesser the node load and the ability of disk space load is stronger.

5.3.3 Network transmission cost ntc

The transmission price of the network among the copies requires to be taken in the positioning of the replica. The factor R_0 denotes the replica file size that is required to be positioned. The variable $B(i, j)$ points the bandwidth of the network between the two nodes. The cost of transmission $TC(i, j)$ is formulated in equation (23).

$$TC(i, j) = \frac{R}{B(i, j)} \tag{23}$$

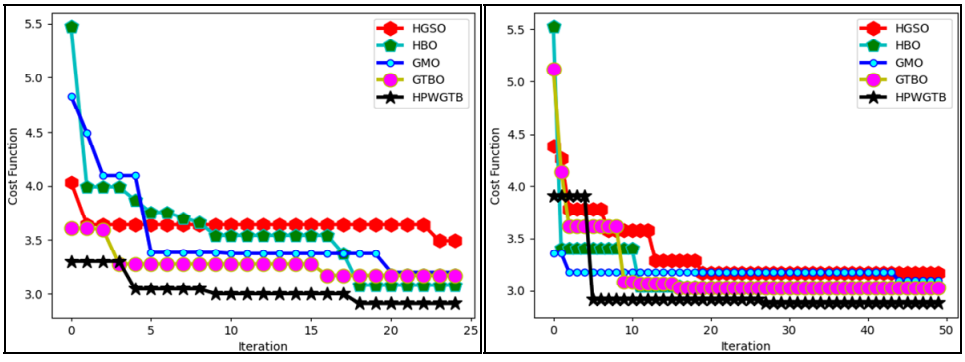
These three constraints helped to achieve a better replica migration and placement mechanism for SaaS developments in the edge cloud.

6 Results and discussion

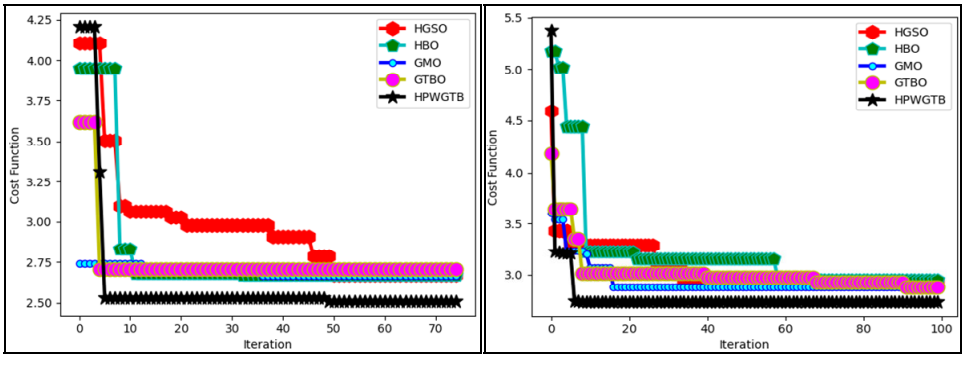
6.1 Simulation setup

The recommended 'replica placement and migration' strategy for SaaS applications in the edge cloud was implemented in MATLAB 2020a and the promising solutions were achieved. The chromosome length was determined according to the amount of replicas. The maximum iteration and the population were 205 and 10 appropriately. Numerous optimisation approaches such as Henry gas solubility optimisation (HGSO) (Hashim et al., 2019), heap-based optimiser (HBO) (Elsayed et al., 2021), GMO (Zade et al., 2022), and GTBO (Tarkhaneh et al., 2021) were taken into consideration for the performance validation of the suggested replica placement and migration strategy.

Figure 4 The cost function validation of the suggested HPWGTB approach over divergent optimisation algorithms concerning, (a) Convergence-25 (b) Convergence-50 (c) Convergence-75 (d) Convergence-100 (see online version for colours)



(a) (b)



(c) (d)

6.2 Cost function analysis of the designed HPWGTB algorithm over other conventional algorithms

The designed HPWGTB is involved in the cost function estimation against various conventional algorithms and is presented in Figure 4. The convergence values such as 25,

50, 75, and 100 are utilised to validate the cost measure of the suggested HPWGTB algorithm. The cost function of the HPWGTB is estimated based on the iteration values. The cost function of the suggested HPWGTB is minimised by 83.5% of HGSO, 84% of HBO, 84.5% of GMO, and 85% of GTBO appropriately when the iteration count is 20 for the convergence 100 in Figure 4(d). From this evaluation, it is confirmed that the designed HPWGTB has better convergence rates than the other classical approaches.

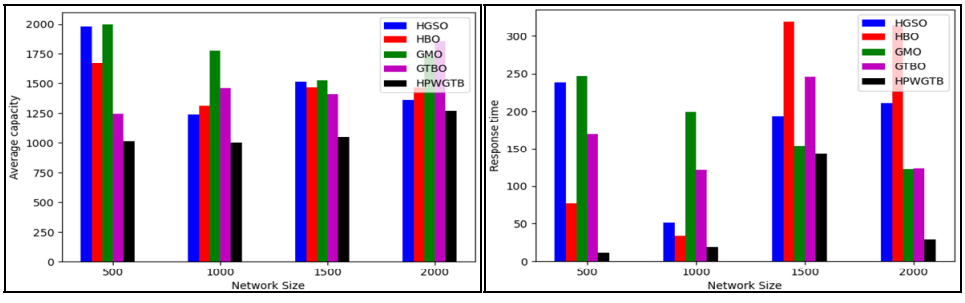
6.3 Statistical investigation of the recommended HPWGTB algorithm over distinct algorithms

Table 2 depicts the statistical validation of the suggested HPWGTB task against divergent traditional algorithms for the various convergences. When taking the best measure for the convergence value 50, the recommended HPWGTB is advanced by 10.7% of HGSO, 8.2% of HBO, 10.3% of GMO, and 8.2% of GTBO correspondingly. Hence, it is confirmed that the developed HPWGTB attained better performance rates.

Table 2 The statistical examination of the suggested HPWGTB over multiple conventional algorithms for the various convergences

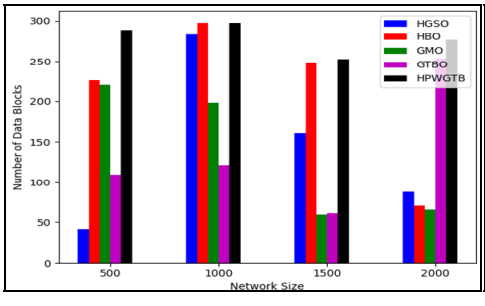
Terms	*HGSO (Hashim et al., 2019)*	*HBO (Elsayed et al., 2021)*	*GMO (Wu et al., 2022)*	*GTBO (Tarkhaneh et al., 2021)*	*HPWGTB*
			Convergence-25		
Worst	4.035179945	5.472284886	4.83087632	3.607960778	3.300696005
Best	3.492798469	3.078559337	3.199259919	3.171212293	2.913743107
Mean	3.645661225	3.578274884	3.53106538	3.27802456	3.03302979
Median	3.641852442	3.542541362	3.373347843	3.276724252	3.000615132
Std	0.089169059	0.492610787	0.424144612	0.129885974	0.125900158
			Convergence-50		
Worst	4.385703824	5.524099469	3.357980249	5.122461726	3.911210509
Best	3.172198224	3.032519203	3.095371866	3.030874389	2.883284149
Mean	3.341964032	3.157769073	3.175172503	3.182464752	3.00276892
Median	3.172198224	3.032519203	3.177867463	3.030874389	2.921220821
Std	0.288694195	0.368752125	0.045902338	0.370932588	0.30334781
			Convergence-75		
Worst	4.108193393	3.947143687	2.738975182	3.613508829	4.205429819
Best	2.672222655	2.669319733	2.663541531	2.707143924	2.506140603
Mean	2.969514711	2.814516614	2.684369056	2.755483385	2.621065505
Median	2.980760519	2.669319733	2.663541531	2.707143924	2.529190688
Std	0.362159726	0.39261264	0.027358672	0.203657772	0.387057416
			Convergence-100		
Worst	4.594984571	5.174528487	3.604871614	4.186917493	5.375248473
Best	2.936136791	2.950959778	2.879873186	2.884726645	2.737224895
Mean	3.065123372	3.223617677	2.939518997	3.022152645	2.788054558
Median	2.941787419	3.158479528	2.879873186	2.977003134	2.737224895
Std	0.224798384	0.497383321	0.15642374	0.19893914	0.28054199

Figure 5 The functionality investigation of the suggested replica placement and migration strategy over various traditional algorithms regarding, (a) average capacity (b) average response time (c) number of data blocks (d) file unavailability (e) migration time (f) network transmission cost (g) node load (h) resource utilisation (i) transferred data size (see online version for colours)

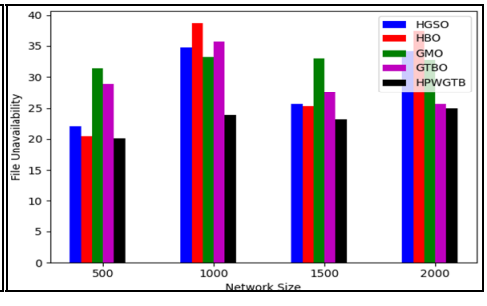


(a)

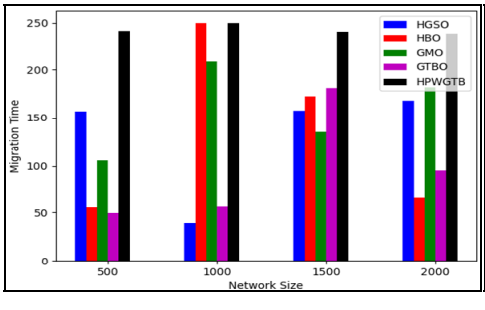
(b)



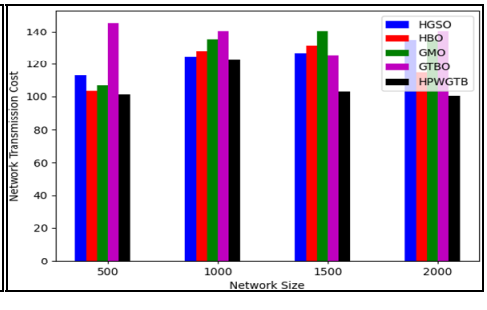
(c)



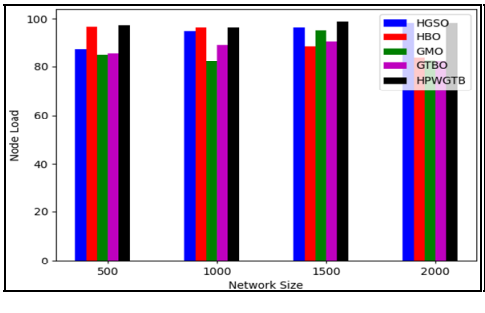
(d)



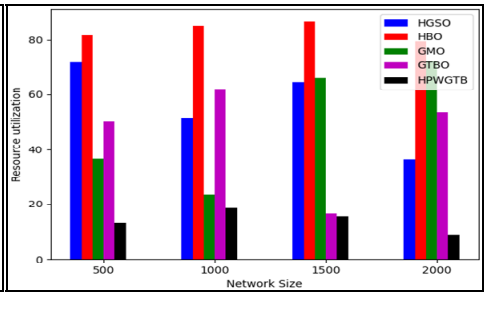
(e)



(f)

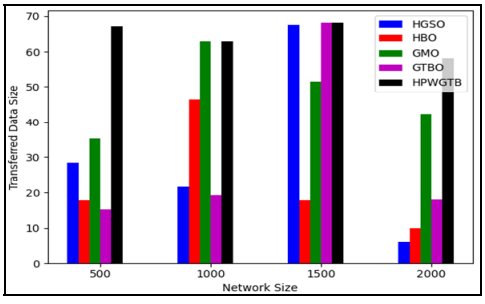


(g)



(h)

Figure 5 The functionality investigation of the suggested replica placement and migration strategy over various traditional algorithms regarding, (a) average capacity (b) average response time (c) number of data blocks (d) file unavailability (e) migration time (f) network transmission cost (g) node load (h) resource utilisation (i) transferred data size (continued) (see online version for colours)



(i)

6.4 Performance validation of the suggested replica placement and migration task over divergent optimisation approaches

Figure 5 illustrates the performance estimation of the suggested replica placement and migration task over distinct algorithms. According to the size of the network, several performance metrics estimate the functionality of the approach. The recommended replica placement and migration task's network transmission cost is reduced by 87.5% of HGSO, 87% of HBO, 86.3% of GMO, and 86.1% of GTBO accordingly when the size of the network is 1,000 in Figure 5(f). From this, it is guaranteed that the designed replica placement and migration task has higher functionality rates than the older mechanisms.

7 Conclusions

The 'replica placement and migration' approach for SaaS developments in edge cloud have been constructed to rectify the troubles in the classical methodologies. The problem of replica placement was resolved by adopting the recommended HPWGTB scheme. For a similar block of data, various replicas might be positioned on various nodes. The multi-objective attributes such as 'file availability, node load, and the transmission cost' were taken for efficient replica migration and placement. The recommended HPWGTB scheme was utilised to improve the information node's load balancing, decrease the usage of resources, and minimise the response period of the network. The migration connection among the 'source and the target' node was considered for improving the migration of replica model for access hotspots and the decrease in the migration period. The experimental answers were estimated by contrasting them with other optimisation techniques. The designed replica placement and migration task's file unavailability was minimised by 98.3% of HGSO, 98.4% of HBO, 97.8% of GMO, and 98.2% of GTBO appropriately when the size of the network is 1,500. Therefore, the constructed replica placement and migration strategy has attained its supremacy against other traditional mechanisms. In future work, the replica and migration strategy will be extended for the car navigations and live videos for attaining high reliability and low latency. Also, the

larger-scale simulations will be performed in the future using deep learning models for providing enhanced performance. Need to investigate information based on the technologies of data acceleration and data replica self-repair in cloud computing environments.

References

Aral, A. and Ovatman, T. (2018) 'A decentralized replica placement algorithm for edge computing', *IEEE Transactions on Network and Service Management*, June, Vol. 15, pp.516–529.

Du, Z., Hu, J., Chen, Y., Cheng, Z. and Wang, X. (2011) 'Optimized QoS-aware replica placement heuristics and applications in astronomy data grid', *Journal of Systems and Software*, July, Vol. 84, pp.1224–1232.

Elsayed, S.K., Kamel, S., Selim, A. and Ahmed, M. (2021) 'An improved heap-based optimizer for optimal reactive power dispatch', *IEEE Access*, Vol. 9, pp.58319–58336.

Guo, J., Li, C. and Luo, Y. (2020) 'Fast replica recovery and adaptive consistency preservation for edge cloud system', *Soft Computing*, Vol. 24, pp.14943–14964.

Hao, P., Hu, L., Jiang, J., Che, X., Li, T. and Zhao, K. (2019) 'Framework for replica placement over cooperative edge networks', *Journal of Ambient Intelligence and Humanized Computing*, Vol. 10, pp.3011–3021.

Hashim, F.A., Houssein, E.H., Mabrouk, M.S., Al-Atabany, W. and Mirjalili, S. (2019) 'Henry gas solubility optimization: a novel physics-based algorithm', *Future Generation Computer Systems*, Vol. 101, pp.646–667.

Hirsch, D. and Madria, S. (2013) 'Data replication in cooperative mobile ad-hoc networks', *Mobile Networks and Applications*, Vol. 18, pp.237–252.

Hosseinzadeh, M., Masdari, M., Rahmani, A.M., Mohammadi, M., Aldalwie, A.H.M., Majeed, M.K. and Karim, S.H.T. (2021) 'Improved butterfly optimization algorithm for data placement and scheduling in edge computing environments', *Journal of Grid Computing*, Vol. 19.

Huang, T., Lin, W., Xiong, C., Pan, R. and Huang, J. (2021) 'An ant colony optimization-based multiobjective service replicas placement strategy for fog computing', *IEEE Transactions on Cybernetics*, November, Vol. 51, pp.5595–5608.

John, S.N. and Mirnalinee, T.T. (2020) 'A novel dynamic data replication strategy to improve access efficiency of cloud storage', *Information Systems and E-Business Management*, Vol. 18, pp.405–426.

Khan, S.U. (2011) 'Mosaic-Net: a game theoretical method for selection and allocation of replicas in ad hoc networks', *The Journal of Supercomputing*, Vol. 55, pp.321–366.

Kumar, K.A., Quamar, A., Deshpande, A. and Khuller, S. (2014) 'SWORD: workload-aware data placement and replica selection for cloud data management systems', *The VLDB Journal*, Vol. 23, pp.845–870.

Li, C., Liu, J., Lu, B. and Luo, Y. (2021) 'Cost-aware automatic scaling and workload-aware replica management for edge-cloud environment', *Journal of Network and Computer Applications*, 15 April, Vol. 180, p.103017.

Li, C., Song, M., Zhang, M. and Luo, Y. (2020a) 'Effective replica management for improving reliability and availability in edge-cloud computing environment', *Journal of Parallel and Distributed Computing*, September, Vol. 143, pp.107–128.

Li, C., Bai, J., Chen, Y. and Luo, Y. (2020b) 'Resource and replica management strategy for optimizing financial cost and user experience in edge cloud computing system', *Information Sciences*, April, Vol. 516, pp.33–55.

Li, C., Wang, Y., Chen, Y. and Luo, Y. (2019a) 'Energy-efficient fault-tolerant replica management policy with deadline and budget constraints in edge-cloud environment', *Journal of Network and Computer Applications*, October, Vol. 143, pp.152–166.

Li, C., Wang, Y., Tang, H. and Luo, Y. (2019b) 'Dynamic multi-objective optimized replica placement and migration strategies for SaaS applications in edge cloud', *Future Generation Computer Systems*, November, Vol. 100, pp.921–937.

Li, C., Wang, Y., Tang, H., Zhang, Y., Xin, Y. and Luo, Y. (2019c) 'Flexible replica placement for enhancing the availability in the edge computing environment', *Computer Communications*, 15 October, Vol. 146, pp.1–14.

Liu, J., Xie, M., Chen, S., Xu, G., Wu, T. and Li, W. (2023) 'TS-REPLICA: a novel replica placement algorithm based on the entropy weight TOPSIS method in spark for multimedia data analysis', *Information Sciences*, May, Vol. 626, pp.133–148.

Mansouri, N. and Javidi, M.M. (2018) 'A hybrid data replication strategy with fuzzy-based deletion for heterogeneous cloud data centers', *The Journal of Supercomputing*, Vol. 74, pp.5349–5372.

Mansouri, N., Javidi, M.M. and Zade, B.M.H. (2021) 'Hierarchical data replication strategy to improve performance in cloud computing', *Frontiers of Computer Science*, Vol. 15.

Mohammadi, B. and Navimipour, N.J. (2022) 'A fuzzy logic-based method for replica placement in the peer to peer cloud using an optimization algorithm', *Wireless Personal Communications*, Vol. 122, pp.981–1005.

Tang, Y., Wang, H., Guo, K., Luo, T. and Chi, T. (2019) 'A new replica placement mechanism for mobile media streaming in edge computing', July, Vol. 33, No. 7.

Tarkhaneh, O., Alipour, N., Chapnevis, A. and Shen, H. (2021) 'Golden tortoise beetle optimizer: a novel nature-inspired meta-heuristic algorithm for engineering problems', *Neural and Evolutionary Computing*, April.

Wu, H., Zhang, X., Song, L., Zhang, Y., Gu, L. and Zhao, X. (2022) 'Wild geese migration optimization algorithm: a new meta-heuristic algorithm for solving inverse kinematics of robot', *Computational Intelligence and Neuroscience*, September.

Xiao, N., Chen, T. and Liu, F. (2011) 'RSEDP: an effective hybrid data placement algorithm for large-scale storage systems', *The Journal of Supercomputing*, Vol. 55, pp.103–122.

Xu, X., Yang, C. and Shao, J. (2017) 'Data replica placement mechanism for open heterogeneous storage systems', *Procedia Computer Science*, Vol. 109, pp.18–25.

Yanling, S., Chunlin, L. and Hengliang, T. (2019) 'A data replica placement strategy for IoT workflows in collaborative edge and cloud environments', *Computer Networks*, January, Vol. 148, pp.46–5915.

Zade, B.M.H., Mansouri, N. and Javidi, M.M. (2022) 'A new hyper-heuristic based on ant lion optimizer and Tabu search algorithm for replica management in a cloud environment', *Artificial Intelligence Review*, Vol. 56, pp.9837–9947.