Power System Technology
ISSN:1000-3673

Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

Gaussian Based Convergence Factor with Squirrel Search Algorithm for Optimal Resource Allocation in Cloud Computing for Small Finance Organization

Sidagouda Basagouda Patil[1], Dr. Mukund Kulkarni[2]

[1]Research Scholar, Bharati Vidyapeeth, Kolhapur.

[2]Professor Bharati Vidyapeeth, Kolhapur.

ABSTRACT

Cloud Computing (CC) have high attention because of executing the requests, accordance with user needs and gives quality services and task execution time among Virtual Machines (VM). But, the resource allocation for different applications is the problem because of dynamic workload conditions and uncertainty in cloud network. In this manuscript, the Gaussian based Convergence Factor (GCF) with Squirrel Search Algorithm (SSA) for allocating the optimal resource in cloud environment for small finance organization. The proposed GCF with SSA effectively balances the workload and allocates much appropriate resources to users' application when ensuring a deadline constraint. The Tent chaotic map and Gaussian based Convergence Factor (GCF) are incorporated in the tradition SSA which improved the search ability and convergence rate of SSA to allocate the optimal resources in cloud for small finance organization. By selecting the correct instance types which aligns with requirements of finance organization and this involves deciding between options like on-demand, reserved or spot instances. The performance of GCF with SSA is evaluated with different metrics of implementation time, makespan, vigour ingesting and reserve utilization. The GCF with SSA reached less energy of 0.505J, less execution time of 0.472s, less makespan of 0.723s and high resource utilization of 51% for 100 tasks of 30 VMs which is efficient while compared to existing methods like Moth Search Adapted Sealion Optimization (MS-SLnO).

KEYWORDS: Chaotic map, Cloud Computing, Gaussian based Convergence Factor, Squirrel Search Algorithm and Virtual Machines

1. INTRODUCTION

Recently, the data account in type of cloud that includes the processing have maximized due to availability of different combinations in Internet of Things (IoT) [1]. The growth and popular of IoT have maximized highly, that leads to data development in perception layer of IoT [2]. The cloud management is developed in way that total server resources and client answer a data

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

which requested through explicit client through storage and database in server [3-5]. In the cloud, the intensity of data seems probable data risk breaking and hacking through groups [6]. The Cloud Computing (CC) is deployed over companies for accentuating security while computing of employment costs through accepting the virtualization is minimized [7]. The process of application and multi operation system in the same server [8]. The usage of hardware and security is improved through virtualization in cloud and gives performed well in process effectiveness and resiliency [9]. Moreover, the efficient and fast cloud technique is virtualization in the diverse environments. The partial vigor ingesting of used servers in data center is the main reason for high consumption of energy. As the outcome, the service providers pay high costs without attaining the need amount of service usage [10].

The main component of the executing method is virtualization technique, that transfers the Physical Machines (PMs) to multi–Virtual Machines (VMs) than fulfilling the requirements of several customers and users [11]. As the outcomes, the VMs have become standard phase and unit to deliver programs. Every VM which is integrated with the PM in data centre have some quantity of resource ability [12]. The user requests treated as the group of tasks in CC and every VM serves every task in accordance with available resources [13]. Hence, resource allocation process in CC is the much significance, it enables the user requests for executing automatically, without requirement of direct intervention [14]. The services of CC are generally given on demand and subjected to time restraints, generally with hours and minutes [15]. Hence, scheduling is developed for ensuring which resource usage is optimized to maximum effectiveness. In small finance organization, the optimization algorithm is employed for optimizing the resource distribution like capital, employees or operation asses in the effective and adaptive way. The main contributions of the manuscript are given below.

- The Gaussian based Convergence Factor (GCF) with Squirrel Search Algorithm (SSA) is developed to allocate the optimal resources in the cloud when ensuring the deadline constraints.
- The Tent chaotic map and Gaussian based Convergence Factor (GCF) are incorporated in the tradition SSA which improved the search ability and convergence rate of SSA to allocate the optimal resources in cloud.
- The GCF with SSA effectively optimize the cloud resources for small finance organization, minimizing costs. This ensures seamless transaction process, which is essential for customer satisfaction and business continuity.

The balanced section of research is prepared as: Segment 2 reviewed existing methods. Segment 3 delivers particulars of future technique. Segment 4 discussion and outcomes of proposed method. Segment 5 accomplishes investigation.

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

2. LITERATURE REVIEW

The existing researches which have been established for supply distribution in cloud environment are analyzed and described in this section, with its advantages and drawbacks.

Sayed Danial Alizadeh Javaheri et al. [16] suggested Clipped Double Deep Q-Learning (CDDQL) method and Particle Swarm Optimization (PSO) to assign resource in cloud architecture. The PSO algorithm was utilized for prioritizing the tasks and the CDDQL was utilized to core of Auto-CDDQL for allocating the destinated VM resources for tasks. The suggested method was introduced in the Fog and that performed autonomously. The suggested method has high scalability, but the method was unable to consider the cost of data locality, energy consumption and processing cost.

Shubhuam Singh et al. [17] presented the optimization-based resource allocation method which increased with less costs. Previous to the allocation, workload clustering was performed by utilizing the advanced k-means clustering using Lion with Advanced Mating Process (LAAM) that fine-tuned centroid. The tasks clusters were processed based in QoS and time of task. At last, the hybrid method named as Moth Search Adapted Sealion Optimization (MS-SLnO) to allocate source through considering Power Usage Effectiveness (PUE), usage of CPU and computation time. The presented method efficiently solved the issue of local optima but, that took much execution time.

Ashutosh Kumar Singh et al. [18] introduced the Floret Fertilization founded Non dominated Sorting Optimization (FP-NSO) that increased the source utilization and reduced the drive ingesting that minimized the carbon release in data center. Multi source constraint metrics were integrated with introduced method which assisted to identify the much appropriate physical machines to deploy VMs in cloud environment. The introduced method has cost unpredictability and lack in execution due to difficulties in identifying necessities.

Kapil N. Vhatkar and Girish P. Bhole [19] implemented the Whale Random update-based Lion Algorithm (WR-LA) that was integration of Lion Algorithm (LA) and Whale Optimization Algorithm (WOA) for optimal resource allocation. The solution of optimal source provision was developed through consider the objectives such as Balanced Cluster Use, Entire Network Distance, Scheme Letdown and Edge Reserve. The implemented method has high complexity, resource overhead and less scalability.

Avnish Thakur and Major Singh Goraya [20] developed the Phasor Particle Swarm Optimization and Dragonfly Algorithm (PPSO-DA) was utilized for generating the optimum resource allocation plan to balance load in cloud. The developed method utilized the DA features for distracting the search agent from region of worst solution through novel algorithm. The distraction magnitude experienced through search agent was contrariwise relative to their distance from position of nastiest answer. The balance among examination and mistreatment

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

stages in various stages in search process was developed through coefficient of dynamic distraction. But the developed method has high execution time.

3. PROPOSED METHODOLOGY

The aim of proposed GCF with SSA method is to assign resources of users and unceasingly monitored rank of whole VM instances for avoiding probability of over or under-resources provisioning. Additionally, proposed GCF with SSA have capability to take decisions at processing time and select good resources for processing of end-users and enhances performance of indicator parameters. This method has different components like cloud layer, source supervisor, source provisioning and de-provisioning, allocation strategy and mapping Physical Machine (PM) with Virtual Machine (VM). By assessing the workloads of organization, includes transaction volume, peak utilization times and performance requirements. This analysis supports to determine the amount and type of cloud resources required. Based on workload analysis, develop the cloud infrastructure. Select the correct instance types which aligns with requirements of organization. This involves deciding between options like on-demand, reserved or spot instances. The resources like capital, employes are represented as the variables for allocation across various financial products, branches or operational units. The few supportive civilizations listed in India under 1904 Act in initial 5 to 6 years are described: Rajahauli Village Bank, Jorhat, Jorhat Cooperative Town Bank and Charigaon Village Bank, Jorhat, Assam (1904), Tirur Primary Agricultural Cooperative Bank Ltd., Tamil Nadu (1904), Agricultural Service Cooperative Society Ltd., Devgaon, Piparia MP (1905), Bains Cooperative Thrift and Credit Society Ltd., Punjab (1905) Bilipada Service Cooperative Society Ltd., Orissa (1905), Government of India, Sectt. Cooperative Thrift & Credit Society (1905), Kanginhal Vyvasaya Seva Sahakari Bank Ltd., Karnataka (1905), Kasabe Tadvale Cooperative Multi-Purpose Society, Maharashtra (1905), Premier Urban Credit Society of Calcutta, West Bengal (1905), Chittoor Cooperative Town Bank, Andhra Pradesh (1907), Rohika Union of Cooperative Credit Societies Ltd., Bihar (1909). Under this Act, numerous non-credit initiatives come like Triplicane society in Madras that rub consumer store, weaver credit cooperatives in Dharwar and Hubli, that gave credit in the form of yarn etc. The below figure 1 represents the architecture resource allocation in cloud environment.

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

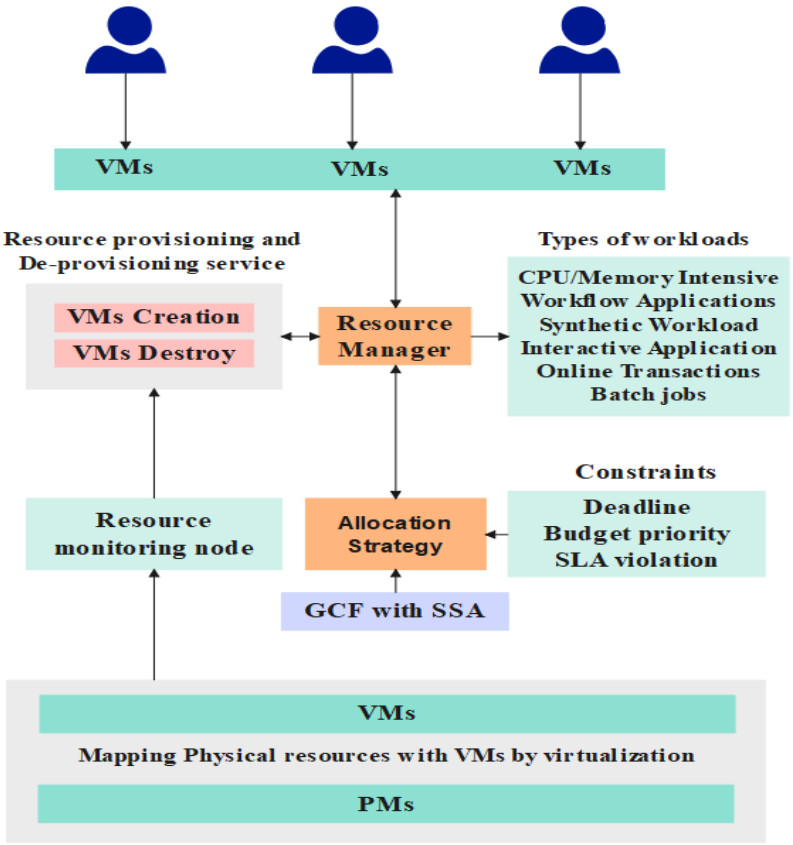

Figure 1 Resource allocation in cloud environment

3.1 Cloud layer

In cloud layer, the user submits n applications $(\zeta_1, \zeta_2, \zeta_3, \ldots, \zeta_n)$ for executing in server of Data Center (DC) which includes Physical Machine (PM) which is represented as $PM = \{PM_1, PM_2, PM_3, \ldots, PM_N\}$. The PM includes multiple heterogeneous Virtual Machines (VMs) which is represented as $(\rho_1, \rho_2, \rho_3, \ldots, \rho_m)$ for accomplishing the end-users demands. The amount of VMs deployed in workstation based on rest resource capacity of individual workstation and also resource needed through applications. Every task ζ_1 has length of $l(\zeta_1)$ and integrated with deadline $\delta(\zeta_1)$. All upcoming applications are scheduled over heterogenous resources such as memory (MM_{mem}), processing speed (ρ_s), storage (s_c), number of Central Processing Unit (CPU) depend on demands. The end-user expects to execute each task with high throughput, less time and also financial cost, when service provider try to get high gain from resources of datacenters. The whole upcoming applications are analyzed and allocated to good resource to process or else the request is rejected. At last, the proposed GCF with SSA is used to allocate the optimal requested tasks to available VMs. The acknowledgement is sent once the process is completed.

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

3.2 Source Supervisor

It is main module of cloud source distribution method whose impartial is interacts with various components to process tasks of end users in efficient way without violating constraints such as user budgets and deadline. The resource manager gathers diverse users requests and saves them in queue of workload to process. That analyzes each request depended on its constraints and demand, next sent that to provisioning and de-provisioning mechanisms for developing or deploying VM examples so that choose optimal resources to satisfy requests demand. The list of resources is fed to allocation strategy that schedule and allocates resources to diverse workloads utilizing GCF with SSA and returns to source supervisor.

3.3 Source a provisioning and de-provisioning

The VM examples developed and demolished through this constituent based on users request and rank of cloud incomes. That gets cloud resources rank from node of resource monitoring and end-users demands from source supervisor. The node intensive care theatres essential part in future GCF with SSA, that continuously monitor cloud resources and send data to source provisioning and de-provisioning module for additional significant act. Whether users request is extended in processing time, next many VM instances are deployed in PM for avoiding the under-provisioning conditions. Whether end-user demand is shrunk in processing time, next amount of VM instances is destroyed for avoiding the over-provisioning condition in cloud datacenter. The provision resources which are “right-sized” for present workloads. Avoid the over-provisioning for minimizing the costs. The right-sizing includes choosing optimum instance size and types, storage options and configurations of network. Introduce the policies of auto-scaling which routinely regulate the amount of active examples founded on real-time request. This safeguards that resource is scaled up in high request and scaled down in low demand, optimize the performance and costs.

3.4 Mapping the PMs with VMs

The effectiveness of future GCF with SSA is based on mapping among PM and VM instances in datacentre of cloud. This is essential problems because of heterogeneity in both PMs and VMs and unpredictable user demands. The Virtual Machine Monitor (VMM) is the essential for deploying multi VM instances over PM by virtualization. The goal of VMM is to position VM examples over much appropriate PM and enhance the resource usage.

3.5 Squirrel Search Algorithm

This algorithm is swarm intelligence-based algorithm which mimics the scavenging conduct of collectors and its effective technique of movement called as the sliding. In small finance organization, the maximizing the profits and customer satisfaction and minimizing the risk through optimizing the loan portfolio are the main objective functions. In SSA, there have four essential phase which is described below.

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

- In deciduous forest, there has n squirrels and n trees, 1 squirrel in 1 tree.
- These n tress have hickory tree and acorn trees $N_{fs}(1 < N_{fs} < n)$ and the remaining trees are normal.
- There have three types of trees in forest. The hickory tree contains good food source, acorn trees contain normal food source and normal tress contains no food.
- Every squirrel search diet separately and utilized accessible food sources finished its dynamic scavenging behavior. The below figure 2 represents the process of GCF with SSA.

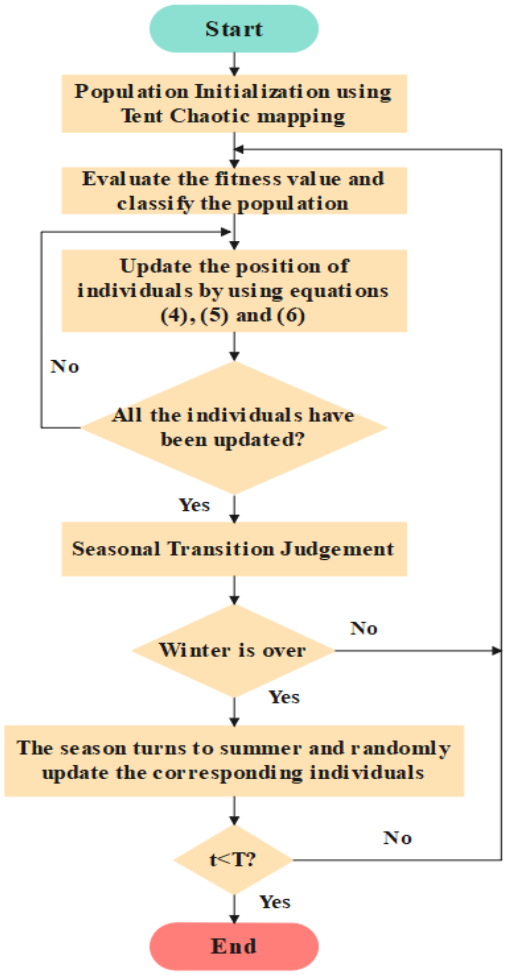

Figure 2 Process of GCF with SSA

3.5.1 Population initialization

The initialization of squirrels is performed by using the tent chaos of population (30), maximum iteration (100). The search space has possible configurations of resource allocations.

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

Every squirrel describes various resource allocation to different task in cloud. In this research, utilized the Tent chaotic map for generating the diverse set of initial solutions and mathematical expression is given in equation (1),

$$x_{n+1} = \begin{cases} \frac{x_n}{\mu}, & if\ x_n < \mu \\ \frac{1 - x_n}{1 - \mu}, & if\ x_n \geq \mu \end{cases} \quad (1)$$

Tis process supports in spreading the initial resource allocation strategy in search space, covers wide range of possibilities. After including the tent chaotic map, the mathematical expression for initial population is given in equation (2),

$$FS_i = FS_L + x_{n+1}(1, d) \times (FS_U - FS_L), \qquad (i = 1,2, \dots, n) \quad (2)$$

3.5.2 Population classification

The squirrels' locations are estimated through fitness function f. The variable for decision (squirrels' position vector) is fed as input to fitness function and respective result $f\ (FS_i) = f(FS_{i1}, FS_{i2}, \dots, FS_{id}),\ (i = 1,2, \dots, n)$ represents fitness value of ith squirrel. The fitness values describe the food source quality fought through ith squirrel. The squirrels in hickory tree (FS_h) is the one which has minimum fitness values. The squirrels in acorn tree (FS_a) describes individuals which has fitness value range from $2 - N_{fs} + 1$. The balanced individual squirrels are describes as normal tress (FS_n).

3.5.3 Position update

The squirrels' position is updated through gliding for trees of various class. The n_1 squirrels which are in usual tress move towards acorn tress, n_2 squirrels move towards hickory tree. The n_3 squirrels in acorn tress move to hickory tree. The Gaussian based convergence factor a is incorporated in the traditional position update formula of SSA which improves the search ability and convergence rate of SSA in resource allocation. The mathematical expression for Gaussian based convergence factor is given in equation (3),

$$a_{GCF} = \begin{cases} \phi \cdot \frac{1}{\sqrt{2\pi}(T_{max}/3)} e^{\frac{t^2}{2(T_{max}/3)^2}}, & t \leq \delta T_{max} \\ \phi \cdot \frac{1}{\sqrt{2\pi}(T_{max}/3)} e^{\frac{t^2}{2(T_{max}/3)^2}}, & \delta T_{max} \leq t < T_{max} \end{cases} \quad (3)$$

In the above equation (3), the ϕ represents the decreasing function which changes with number of iterations, the T_{max} signifies the highest amount of iterations and the t signifies the present iteration number. The convergence factor a depends on Gaussian distribution that initially allowed much exploration and gradually minimizes, that allows the smooth transition

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

from examination to mistreatment of algorithm. The mathematical expression of location update after incorporating the Gaussian based convergence factor is given from equation (4) – (6),

$$FS_i^{t+1} = \begin{cases} FS_i^t + a_{GCF} \times d_g \times G_c \times (FS_h^t - FS_i^t), & R \geq P_{dp} \\ random\ location, & or\ else \end{cases} \quad (4)$$

$$FS_i^{t+1} = \begin{cases} FS_i^t + a_{GCF} \times d_g \times G_c \times (FS_a^t - FS_i^t), & R \geq P_{dp} \\ random\ location, & or\ else \end{cases} \quad (5)$$

$$FS_i^{t+1} = \begin{cases} FS_a^t + a_{GCF} \times d_g \times G_c \times (FS_h^t - FS_a^t), & R \geq P_{dp} \\ random\ location, & or\ else \end{cases} \quad (6)$$

In the above equations, the t represents the present iteration, the R represents the random number in [0,1] range and P_{dp} represents probability of predator presence. If the $R \geq P_{dp}$ represents that squirrels are harmless and slither by forestry in food search. If the $R < P_{dp}$ represents that squirrels are in risk of predation and forced to utilize the random walk for identifying the closest hiding location. The G_c represents the gliding constant of value 1.9, d_g represents the random gliding distance and its mathematical expression is given in equation (7),

$$d_g = \frac{h_g}{\tan(\varphi) \times sf} \quad (7)$$

In the above equation (7), the h_g and sf represents the constant value of 8 and 18 respectively and the $\tan(\varphi)$ is the gliding angle and the mathematical formula for measuring the gliding angle is given in equation (8),

$$\tan(\varphi) = \frac{D}{L} \quad (8)$$

In the above equation (8), the D represents drag force and the L represents lift force and the mathematical appearance is given in reckoning (9) and (10),

$$D = \frac{1}{2\rho V^2 S C_D} \quad (9)$$

$$L = \frac{1}{2\rho V^2 S C_L} \quad (10)$$

In the above equations, the ρ represents the air density, the V represents the squirrel's gliding speed, the S represents the surface area, the C_D represents the coefficient of frictional drag and the C_L represents the random number.

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

3.5.4 Seasonal Transition Judgement

The condition for seasonal monitoring is used in SSA supports a method to jump out of local optima. In initial phase of every iteration, the SSA needed whole populace to be in season state, that represents whole persons are updated. While whole individuals are updated, mathematical formula for change in season is given in equation (11) and (12),

$$S_c^t = \sqrt{\sum_{k=1}^{d} \left(FS_{h,k}^t - FS_{aj,k}^t\right)^2} \quad j = 1, 2, \ldots, N_{fs} \tag{11}$$

$$S_{min} = \frac{10e^{-6}}{(365)^{t/(T/2.5)}} \tag{12}$$

In the above equations, the T represents the highest number of iterations and the t represents present number of repetitions. While cyclical continuous is less than minimum $(S_c^t < S_{min})$, season variations from season to seasonal. While period changes, whole individuals gliding to F_h stayed in updated locations and whole individuals to F_a without encounter the predator relocating its location and the mathematical formula is given in equation (13),

$$FS_i^{new} = FS_L + Levy(n) \times (FS_U - FS_L) \tag{13}$$

The mathematical formula for Levy distribution is measured by using the equation (14),

$$Levy(x) = 0.01 \times \frac{r_a \times \xi}{|r_b|^{\frac{1}{\beta}}} \tag{14}$$

In the above equation (15), the r_a and r_b represents the usually dispersed chance number [0,1], the β signifies continuous worth and the exact formulation for measuring the ξ is given in equation (16),

$$\xi = \frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)^{\frac{1}{\beta}}}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}} \tag{15}$$

In the above equation (15), the $\Gamma(x) = (x-1)$.

3.5.5 Objective function

In small finance organization, need to optimize the loan portfolio through allocating funds across various loan types when maximizing returns and minimizing the risk. The mathematical formula for objective is given as equation (16),

$$Maximize \sum (Loan\ Amount_i \times Interest\ Rate_i) - \sum (Default\ Risk_i) \tag{16}$$

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

In the above equation (16), i represents the various loan products. The developed GCF with SSA allocates the optimal resources in the cloud when ensuring the deadline constraints. The GCF with SSA effectively optimize the cloud resources for small finance organization, minimizing costs. This ensures seamless transaction process, which is essential for customer satisfaction and business continuity. The GCF with SSA handles the multiple types of objective functions like maximizing the profit, minimizing the cost.

4. Experimental Analysis

The performance of the developed GCF with SSA is simulated with python environment and configurations used are i5 processor, 8 GB RAM and windows 10 OS. The performance metrics used to evaluate the developed GCF with SSA are energy consumption, Makespan, execution time and resource utilization.

4.1 Analysis of energy consumption

In the below figure 3 and 4, the energy consumption of GCF with SSA is evaluated based on number of tasks. In scenario 1, the performance of GCF with SSA is analysed with 30 VMs and in scenario 2, the performance of GCF with SSA is analysed with 60 VMs. The energy (in Joule) is needed to execute the task with minimum, that resulted high network lifetime. The existing algorithms used to evaluate the GCF with SSA are Tunicate Swarm Optimization (TSO), Reptile Search Algorithm (RSA), Whale Optimization Algorithm (WOA) and traditional SSA. In scenario 1, the GCF with SSA consumed minimal energy of 0.505 J for 100 tasks, 0.532 J for 150 tasks, 0.579 J for 200 tasks, 0.602 J for 250 tasks and 0.632 J for 300 tasks. In scenario 2, the GCF with SSA consumed minimal energy of 0.541 J for 100 tasks, 0.584 J for 150 tasks, 0.605 J for 200 tasks, 0.628 J for 250 tasks and 0.664 J for 300 tasks. While compared with existing algorithms, the proposed GCF with SSA consumed minimal energy and showed the effective performance.

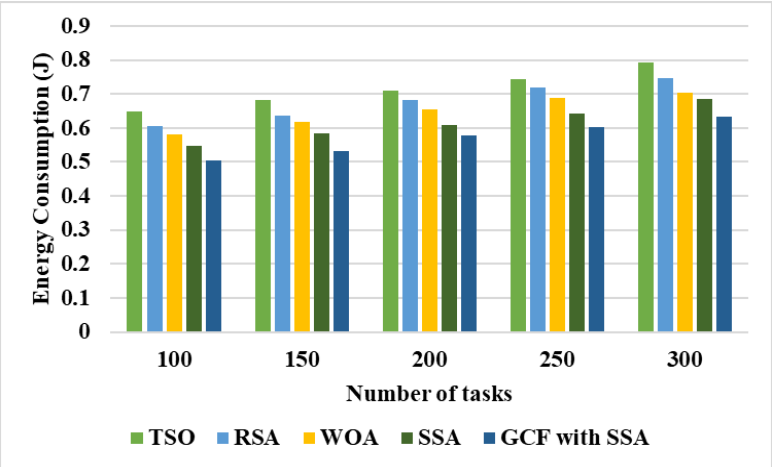

Figure 3 Number of tasks vs Energy consumption (VM=30)

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

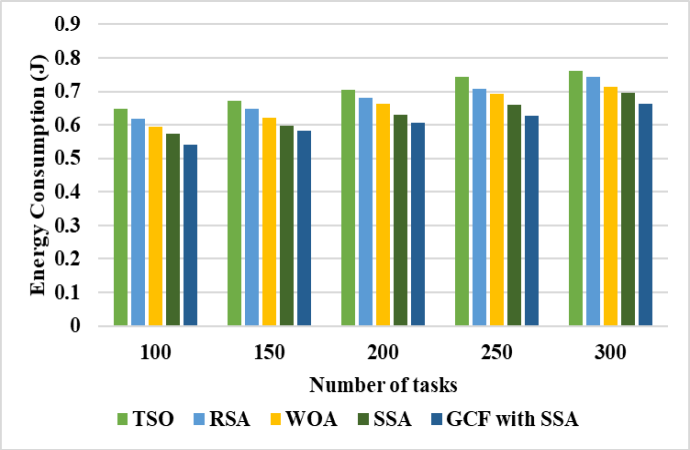

Figure 4 Number of tasks vs Energy consumption (VM=60)

4.2 Analysis of Execution time

In the below figure 5 and 6, the execution time of GCF with SSA is evaluated based on number of tasks. In scenario 1, the performance of GCF with SSA is analysed with 30 VMs and in scenario 2, the performance of GCF with SSA is analysed with 60 VMs. Every task takes various execution time (in seconds), that based on application. Though, to enhance the accuracy of PM, VMs requires to process the tasks with less execution time. The existing algorithms used to evaluate the GCF with SSA are TSO, RSA, WOA and traditional SSA. In scenario 1, the GCF with SSA consumed minimal time for completion of 0.472s for 100 tasks, 0.432s for 150 tasks, 0.392s for 200 tasks, 0.365s for 250 tasks and 0.304s for 300 tasks. In scenario 2, the GCF with SSA consumed minimal completion time of 0.491s for 100 tasks, 0.446s for 150 tasks, 0.414s for 200 tasks, 0.383s for 250 tasks and 0.343s for 300 tasks. While compared with existing algorithms, the proposed GCF with SSA consumed minimal completion time and showed the effective performance.

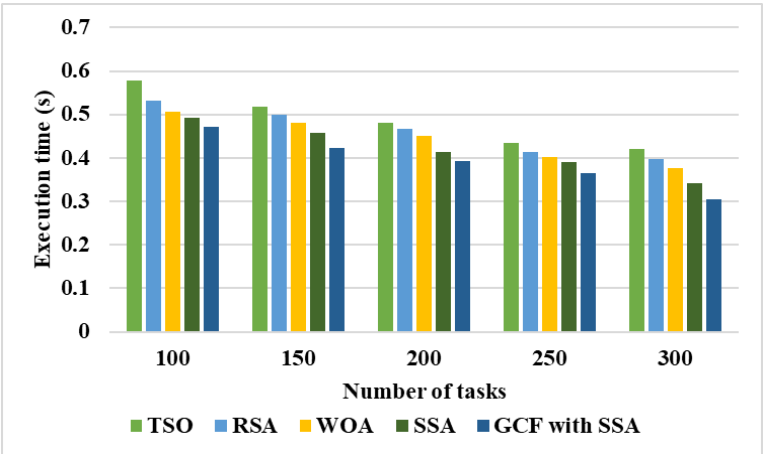

Figure 5 Number of tasks vs Execution time (VM=30)

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

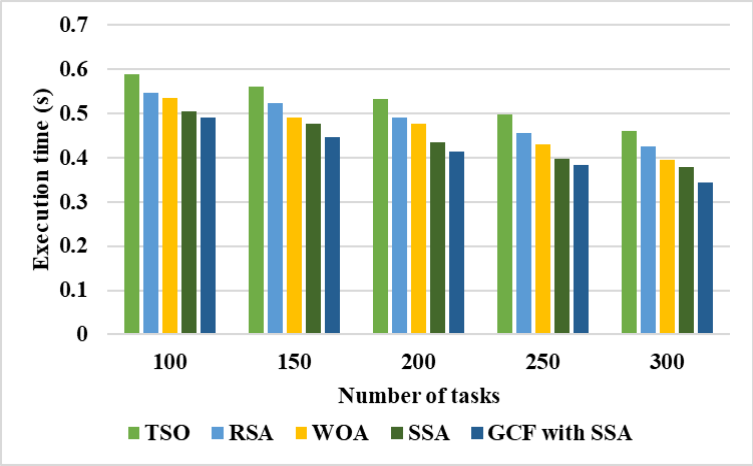

Figure 6 Number of tasks vs Execution time (VM=60)

4.3 Analysis of Makespan

In the below figure 7 and 8, the makespan of GCF with SSA is evaluated based on number of tasks. In scenario 1, the performance of GCF with SSA is analyzed with 30 VMs and in scenario 2, the performance of GCF with SSA is analyzed with 60 VMs. Makespan represents the job completion time, the positive outcome is that the minimal makespan. The existing algorithms used to evaluate the GCF with SSA are TSO, RSA, WOA and traditional SSA. In scenario 1, the GCF with SSA consumed minimal energy of 0.732s for 100 tasks, 0.692s for 150 tasks, 0.652s for 200 tasks, 0.613s for 250 tasks and 0.654s for 300 tasks. In scenario 2, the GCF with SSA consumed minimal energy of 0.808s for 100 tasks, 0.793s for 150 tasks, 0.757s for 200 tasks, 0.718s for 250 tasks and 0.693s for 300 tasks. While compared with existing algorithms, the proposed GCF with SSA consumed minimal makespan and showed the effective performance.

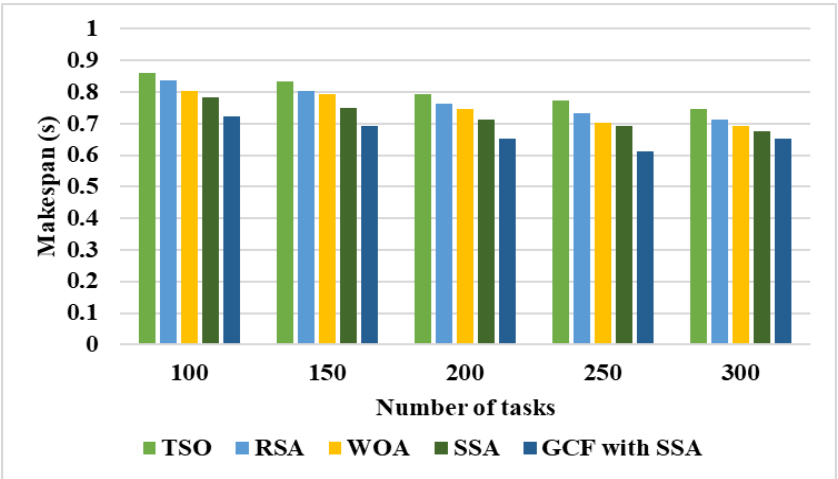

Figure 7 Number of tasks vs Makespan (VM=30)

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

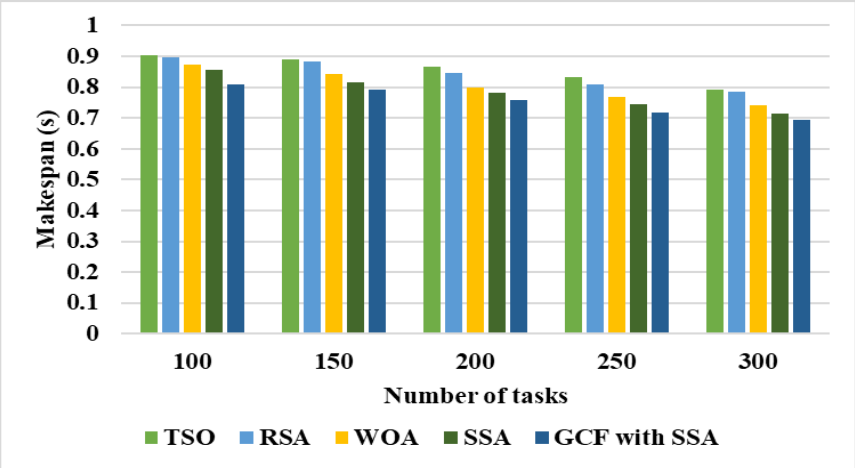

Figure 8 Number of tasks vs Makespan (VM=60)

4.4 Analysis of Resource Utilization (%)

In the below figure 9 and 10, the resource utilization of GCF with SSA is evaluated based on number of tasks. In scenario 1, the performance of GCF with SSA is analyzed with 30 VMs and in scenario 2, the performance of GCF with SSA is analyzed with 60 VMs. The existing algorithms used to evaluate the GCF with SSA are TSO, RSA, WOA and traditional SSA. In scenario 1, the GCF with SSA reached resource utilization of 51% for 100 tasks, 57% for 150 tasks, 70% for 200 tasks, 79% for 250 tasks and 85% for 300 tasks. In scenario 2, the GCF with SSA reached resource utilization of 55% for 100 tasks, 61% for 150 tasks, 76% for 200 tasks, 83% for 250 tasks and 91% for 300 tasks. While compared with existing algorithms, the proposed GCF with SSA reached high resource utilization and showed the effective performance.

Figure 9 Number of tasks vs Resource utilization (VM=30)

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

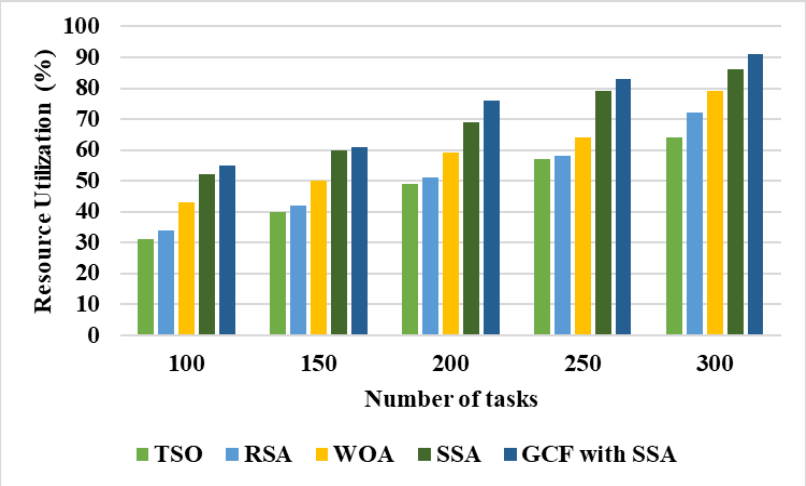

Figure 10 Number of tasks vs Resource Utilization (VM=60)

4.5 Comparative Analysis

In this section, the performance of GCF with SSA is compared to existing techniques like AutoCDDQL [16], MS-SLnO [17] and FP-NSO [18] based on task size and number of tasks. In the below table 1, the performance of GCF with SSA is compared with AutoCDDQL [16] with metrics of makespan, response time, resource utilization and energy consumption. In the below table 2, the performance of GCF with SSA is compared with MS-SLnO [17] with metrics of energy consumption, execution time, makespan, and resource utilization. In the below table 3, the performance of GCF with SSA is compared with FP-NSO [18] with metrics of resource utilization and execution time. While comparing with these 3 existing methods, the proposed GCF with SSA consumed less time and energy and reached the maximum of makespan, response time and resource utilization.

Table 1 Comparison of AutoCDDQL [16]

Methods	Task size (kb)	Makespan (s)	Response time (s)	Resource Utilization (%)	Energy consumption (kWh)
AutoCDDQL [16]	20	250	2.8	30	75
	40	520	7	65	99
	60	825	15	90	295
	80	1150	19	95	390
Proposed GCF with SSA	20	267	3.5	45	70
	40	545	9	78	84
	60	871	17	95	271
	80	1189	22	97	375

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

Table 2 Comparison of MS-SLnO [17]

Methods	Number of VMs	Number of tasks	Energy consumption (J)	Execution time (s)	Makespan (s)	Resource utilization
MS-SLnO [17]	30	100	0.740	0.278	0.760	0.740
		200	0.530	0.167	0.123	0.530
		300	0.669	0.275	0.145	0.669
		400	0.605	0.377	0.214	0.605
	60	100	0.868	0.397	0.919	0.868
		200	0.839	0.265	0.123	0.839
		300	0.774	0.209	0.155	0.774
		400	0.782	0.286	0.217	0.782
Proposed GCF with SSA	30	100	0.732	0.256	0.742	0.726
		200	0.521	0.143	0.114	0.515
		300	0.634	0.258	0.126	0.648
		400	0.543	0.361	0.201	0.587
	60	100	0.852	0.372	0.900	0.852
		200	0.817	0.256	0.103	0.814
		300	0.758	0.197	0.137	0.742
		400	0.769	0.263	0.203	0.757

Table 3 Comparison of FP-NSO [18]

Methods	Number of VMs	Resource Utilization (%)	Execution time (s)
FP-NSO [18]	100	42.59	3.95
	400	42.47	15.75
	800	42.87	85.57
	1200	42.99	148
Proposed GCF with SSA	100	43.67	3.46
	400	43.51	15.16
	800	43.98	85.23
	1200	44.26	133

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

4.6 DISCUSSION

The performance and results of GCF with SSA are evaluated with various number of tasks and by varying the VMs. The results of the GCF with SSA is compared with TSO, RSA, WOA and traditional SSA with metrics of execution time, energy consumption, Makespan and resource utilization. Additionally, the it is compared with existing methods like AutoCDDQL [16], MS-SLnO [17] and FP-NSO [18] with different metrics. The proposed GCF with SSA effectively balances the workload and allocates much appropriate resources to users' application when ensuring deadline constraints. The Tent chaotic map and GCF are incorporated in the tradition SSA which improved the search ability and convergence rate of SSA to allocate the optimal resources in cloud. The performance of GCF with SSA is evaluated with different metrics of execution time, makespan, energy consumption and resource utilization. The GCF with SSA handles the multiple types of objective functions like maximizing the profit, minimizing the cost.

5. CONCLUSION

The effective optimization-based Resource allocation framework is developed which protected the cloud infrastructure and allocated the optimal resource to process the demand of users by using decision-making process. Here, the GCF with SSA is utilize to search and allocate good possible resource to request and ignore probability of imbalance workload. The developed GCF with SSA have high convergence rate because it consumes less amount of time to identify optimal resource. The performance of GCF with SSA is evaluated with different metrics of execution time, makespan, energy consumption and resource utilization. The GCF with SSA reached less energy of 0.505J, less execution time of 0.472s, less makespan of 0.723s and high resource utilization of 51% for 100 tasks of 30 VMs which is efficient while compared to existing methods. The GCF with SSA handles the multiple types of objective functions like maximizing the profit, minimizing the cost. In future, the security of cloud and resource allocation in cloud can be performed to further improve the performance of VMs in cloud environment.

REFERENCES

[1]. Sangaiah, A.K., Javadpour, A., Pinto, P., Rezaei, S. and Zhang, W., 2023. Enhanced resource allocation in distributed cloud using fuzzy meta-heuristics optimization. Computer Communications, 209, pp.14-25.

[2]. Bashir, S., Mustafa, S., Ahmad, R.W., Shuja, J., Maqsood, T. and Alourani, A., 2023. Multi-factor nature inspired SLA-aware energy efficient resource management for cloud environments. Cluster Computing, 26(2), pp.1643-1658.

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

[3]. Al Reshan, M.S., Syed, D., Islam, N., Shaikh, A., Hamdi, M., Elmagzoub, M.A., Muhammad, G. and Talpur, K.H., 2023. A fast converging and globally optimized approach for load balancing in cloud computing. IEEE Access, 11, pp.11390-11404.

[4]. Baburao, D., Pavankumar, T. and Prabhu, C.S.R., 2023. Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method. Applied Nanoscience, 13(2), pp.1045-1054.

[5]. Hua, W., Liu, P. and Huang, L., 2023. Energy-efficient resource allocation for heterogeneous edge-cloud computing. IEEE Internet of Things Journal.

[6]. Jeong, B., Baek, S., Park, S., Jeon, J. and Jeong, Y.S., 2023. Stable and efficient resource management using deep neural network on cloud computing. Neurocomputing, 521, pp.99-112.

[7]. Zhou, G., Tian, W., Buyya, R. and Wu, K., 2023. Growable Genetic Algorithm with Heuristic-based Local Search for multi-dimensional resources scheduling of cloud computing. Applied Soft Computing, 136, p.110027.

[8]. Zavieh, H., Javadpour, A., Li, Y., Ja'fari, F., Nasseri, S.H. and Rostami, A.S., 2023. Task processing optimization using cuckoo particle swarm (CPS) algorithm in cloud computing infrastructure. Cluster Computing, 26(1), pp.745-769.

[9]. Nabi, S. and Ahmed, M., 2022. PSO-RDAL: Particle swarm optimization-based resource-and deadline-aware dynamic load balancer for deadline constrained cloud tasks. The Journal of Supercomputing, 78(4), pp.4624-4654.

[10]. Mansour, R.F., Alhumyani, H., Khalek, S.A., Saeed, R.A. and Gupta, D., 2023. Design of cultural emperor penguin optimizer for energy-efficient resource scheduling in green cloud computing environment. Cluster Computing, 26(1), pp.575-586.

[11]. Godhrawala, H. and Sridaran, R., 2023. A dynamic Stackelberg game based multi-objective approach for effective resource allocation in cloud computing. International Journal of Information Technology, 15(2), pp.803-818.

[12]. Subbaraj, S., Thiyagarajan, R. and Rengaraj, M., 2023. A smart fog computing based real-time secure resource allocation and scheduling strategy using multi-objective crow search algorithm. Journal of Ambient Intelligence and Humanized Computing, 14(2), pp.1003-1015.

[13]. Jena, U.K., Das, P.K. and Kabat, M.R., 2022. Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. Journal of King Saud University-Computer and Information Sciences, 34(6), pp.2332-2342.

[14]. Amer, D.A., Attiya, G., Zeidan, I. and Nasr, A.A., 2022. Elite learning Harris hawks optimizer for multi-objective task scheduling in cloud computing. The Journal of Supercomputing, 78(2), pp.2793-2818.

Power System Technology
ISSN:1000-3673
Received: 06-01-2025 Revised: 15-02-2025 Accepted: 05-03-2025

[15]. Chandrashekar, C., Krishnadoss, P., Kedalu Poornachary, V., Ananthakrishnan, B. and Rangasamy, K., 2023. HWACOA scheduler: Hybrid weighted ant colony optimization algorithm for task scheduling in cloud computing. Applied Sciences, 13(6), p.3433.
[16]. Alizadeh Javaheri, S.D., Ghaemi, R. and Monshizadeh Naeen, H., 2024. An autonomous architecture based on reinforcement deep neural network for resource allocation in cloud computing. Computing, 106(2), pp.371-403.
[17]. Singh, S., Singh, P. and Tanwar, S., 2023. Energy aware resource allocation via MS-SLnO in cloud data center. Multimedia Tools and Applications, 82(29), pp.45541-45563.
[18]. Singh, A.K., Swain, S.R., Saxena, D. and Lee, C.N., 2023. A bio-inspired virtual machine placement toward sustainable cloud resource management. IEEE Systems Journal, 17(3), pp.3894-3905.
[19]. Vhatkar, K.N. and Bhole, G.P., 2022. Optimal container resource allocation in cloud architecture: A new hybrid model. Journal of King Saud University-Computer and Information Sciences, 34(5), pp.1906-1918.
[20]. Thakur, A. and Goraya, M.S., 2022. RAFL: A hybrid metaheuristic based resource allocation framework for load balancing in cloud computing environment. Simulation Modelling Practice and Theory, 116, p.102485.
[21]. Kumar, M., Dubey, K., Singh, S., Kumar Samriya, J. and Gill, S.S., 2023. Experimental performance analysis of cloud resource allocation framework using spider monkey optimization algorithm. Concurrency and Computation: Practice and Experience, 35(2), p.e7469.
[22]. Zhang, X., Zhao, K., Wang, L., Wang, Y. and Niu, Y., 2020. An improved squirrel search algorithm with reproductive behavior. IEEE Access, 8, pp.101118-101132.