

Cost Denigration Based Data Center Allocation Policy Using Modified Parallel PSO Optimization Technique for Multi-cloud Framework

Subash Chandra Tripathy

Department of Computer Science and Engineering, Silicon University, India
E-mail: subashtripathy78@gmail.com
ORCID iD: <https://orcid.org/0009-0008-4768-3220>

Suvendu Chandan Nayak*

Department of Computer Science and Engineering, Silicon University, India
E-mail: suvendu2006@gmail.com
ORCID iD: <https://orcid.org/0000-0002-9524-9583>

*Corresponding author

Rekah Sahu

Department of Computer Science and Engineering, Silicon University, India
E-mail: sahurekha2775@gmail.com
ORCID iD: <https://orcid.org/0000-0002-9840-0357>

Received: 02 December 2025; Revised: 27 January 2026; Accepted: 10 April 2026; Published: 08 June 2026

Abstract: Most of the existing data center allocation mechanisms are either user-centric or service-provider-centric, not for both ends but in reality, both have different objectives. For example, the objective of a user is minimization of cost, response time as well as processing time whereas the objective of service provider is to maximize the profit and processing time and minimization of response time, bandwidth, energy consumption and computing overhead subject to effective resource utilization and load balancing. To address this challenge, this paper introduces a Cost Denigration-Based Data Center Allocation Policy (CD-BDAP) utilizing Particle Swarm Optimization (PSO), which simultaneously considers economic cost, response time, and energy consumption in the selection of data centers. In contrast to conventional PSO-based broker policies, CD-BDAP integrates a workload similarity-aware allocation strategy by calculating a dissimilarity index among user requests, thereby facilitating enhanced consolidation and energy efficiency. A weighted objective function is developed to balance user-centric metrics (cost and response time) with provider-centric metrics (profit and energy consumption), explicitly capturing their trade-offs. The proposed mechanism is assessed utilizing CloudAnalyst, which is constructed on CloudSim. The experimental results indicate that CD-BDAP achieves a reduction in VM cost, a decrease in response time, and an enhancement in energy efficiency, while simultaneously increasing the overall profit for service providers. The findings suggest that the integration of energy-aware cost modeling and workload similarity into PSO-based allocation can enhance both economic and performance efficiency in the selection of cloud data centers. The outcomes of CD-BDAP are compared with the existing PSO-based mechanisms and found to exhibit enhanced performance.

Index Terms: Data Center (DC), Response Time, Optimization, PSO, Energy, Cost.

1. Introduction

Due to the rapidly increasing use of cloud resources, resource management has become a critical challenge in hybrid and multi-cloud environments. Typically, resource allocation follows either a client-centric or a service provider-centric approach. An effective resource allocation and scheduling mechanism can significantly enhance resource utilization and enable efficient load balancing [1]. Each cloud service comes with its own on-demand resource

requirements. Key constraints that differentiate these services include time, cost, bandwidth, resource utilization, energy consumption, and the number of instances required. Given the variability in service size across requests, an automatic prediction and identification model is essential to select the most appropriate data center (DC) for optimized resource allocation [2]. Several service-based models have been proposed to enhance data center utilization [3, 4]. Selecting the most optimal data center (DC) has become increasingly challenging and complex due to the growing number of services and user demands [5]. The primary goal of resource allocation techniques is to assign resources on demand to meet user requirements while maximizing overall resource utilization [6].

In recent years, various heuristic approaches have been explored for resource allocation in cloud environments. Among these, swarm intelligence has emerged as a promising method, demonstrating significant results, particularly in task scheduling [7]. Among the available swarm-based methods, Particle Swarm Optimization (PSO) has shown the most substantial contributions in delivering optimized solutions. It outperforms several conventional cloud-based optimization techniques, including Simulated Annealing, Min–Min, Max–Min, and Genetic Algorithms [8, 9]. In this work, we propose a novel mechanism named CD-BDAP using PSO to identify suitable data centers (DCs) by optimizing user requests with the objective of maximizing profit for the service provider. Our approach computes an optimized list of DCs by considering multiple parameters that influence profitability. Finally, we introduce a resource allocation mechanism that ensures a balanced benefit for both cloud users and service providers. Our model is different from the conventional PSO-based models in various aspects that are listed in Table 1. Specifically, the novelty of CD-BDAP includes:

- Energy-aware VM cost optimization, where VM cost is indirectly optimized through energy consumption rather than treated as a fixed parameter.
- Integration of a similarity (dissimilarity) index among user requests into the data center allocation decision, which is not present in existing PSO-based broker policies.
- A cost denigration mechanism that prioritizes data centers with lower marginal energy-to-cost ratios, linking infrastructure efficiency to economic outcomes.

Table 1. Highlights of the Proposed CD-BDAP with conventional PSO based models

Aspect	Conventional PSO-Based DC Selection	Proposed CD-BDAP
Optimization technique	PSO	PSO
Focus of optimization	Single or limited objectives (cost <i>or</i> time)	Joint economic–energy optimization
User-centric metrics	VM cost, response time	VM cost, response time, processing time
Provider-centric metrics	Often ignored or implicit	Explicit profit and energy efficiency
Energy awareness	Indirect or absent	Explicit energy-aware VM cost optimization
Workload similarity consideration	Not considered	Similarity/dissimilarity index among user requests
Allocation decision basis	Fitness value only	Cost denigration + fitness + similarity
PSO parameter analysis	Fixed from prior work	Parameter sensitivity and convergence analysis

The rest of the paper is organized as follows: The existing related works on optimization techniques in CC are studied with the summary in Section 2. The system design model and definitions of the proposed CD-BDAP are discussed in Section 3. The detailed design mechanism of our CD-BDAP is presented with the algorithm and flowchart in Section 4. The simulation of the proposed algorithm and its performances are studied and discussed in Section 5. The comparison performance of the CD-BDAP with the benchmark techniques is analyzed and noted in Section 6. Finally, in Section 7 the conclusions with observations and the future directions are presented.

2. Related Works

Recently, various works have been proposed for DC selection for the user requirements. In our previous work, it was described that the service broker policy is responsible for finding and routing the user request to the suitable DC [10]. Peng *et al.* [11] presented a resource allocation mechanism based upon the type of user request. Syed *et al.* [12] studied a different resource allocation mechanism and observed that the existing mechanisms are not considering some important parameters, so enhancement is required. Kaur *et al.* [13] present a DQ-HEFT mechanism for workflow scheduling to optimize execution time and makespan. The work proposes the deep learning technique for optimal scheduling of the workflow in CC. Kim *et al.* [14] suggest a VM placement mechanism in CC and discuss that energy consumption is more important than other parameters in allocation. Recently, another cost-based VM allocation is suggested with respect to energy consumption. The work presents the mechanism to migrate the VM with minimum energy consumption and reduces the cost [15].

In [16], a nature-inspired ACO-Fuzzy-based resource allocation mechanism is proposed. The work contributes to minimize the end-user cost for allocating computing and network resources in the cloud. Chen *et al.* [17] addressed the workflow scheduling mechanism considering deadline as a constraint and minimizing the total renting cost of the

resource.

The authors [18] proposed a hybrid discrete multi-verse optimization algorithm to allocate VMs in DC by improving power consumption and reducing resource wastage. Different attempts are made to design hybrid mechanisms with an amalgamation of existing techniques. Xuezhai *et al.* [19] introduced a greedy-based scheduling algorithm to reduce renting cost of resources. It meets service level agreement along with other constraints. In [20], modified PSO mechanism is proposed for task scheduling considering the QoS for better RM. Zhou *et al.* [5] also suggested the modification of PSO to optimize energy consumption in task scheduling. The work has not discussed other parameters of the task and resources in CC. Jafar *et al.* [21] proposed two optimization techniques for VM placement. The work suggests the PSO and Ant colony optimization (ACO) techniques with optimum energy consumption for VM placement. Divya *et al.* [8] discussed a new load balancing technique to stabilize the total cost and the data transfer cost. The work achieves the goal by suggesting the balanced PSO (BPSO). Suchintan *et al.* [22] proposed an ACO technique for cost optimization. The work optimizes the computing cost, bandwidth cost and I/O cost for resource allocation in enterprise cloud. The work does not consider different time parameters associated with resource allocation. Moreover, a binary PSO is discussed by Mapetu *et al.* [23] for load balancing in task scheduling and claims low cost and low time. To schedule the tasks efficiently, a modified Henry gas solubility optimization is implemented which is based on whale optimization [24]. The authors observed better performance when tested on a synthetic dataset.

Table 2. Symbol and notation

Symbol / Notation	Description
DC_i	i^{th} data center
U	Set of user requests
u_i	i^{th} user request
VM	Virtual Machine
VM_i	Virtual machine allocated to user u_i
VM_{total}	Total VM cost paid by the user
$request_{cost}$	Request cost
VM_{cost}	VM usage cost
DT_{cost}	Data transfer cost
VMP_{cost}	VM processing (CPU) cost
M_{cost}	Main memory cost
ST_{cost}	Storage cost
(x)	Number of memory units (MB)
(y)	Number of storage units
$Revenue$	Total income earned by the service provider
$profit$	Net profit of the service provider
$energy$	Energy consumption cost
Net_r	Network resource cost
Pow_i	Power consumption of the i^{th} physical server
$Energy_{total}$	Total energy consumed by a data center
$util_i$	Utilization of the i^{th} server
EPT	Expected processing time
$ResTime$	Response time
$ProTime$	Processing time
DIB	Degree of load imbalance
I	Index of dissimilarity
w	Inertia weight in PSO
$(c1, c2)$	Acceleration coefficients in PSO
P_{best}	Personal best solution
G_{best}	Global best solution
$(w1, w2, w3, w4)$	Weights in multi-objective optimization
N	Number of particles / candidate data centers
k	Iteration index

3. System Definitions and Metrics

This section highlights various parameters that are considered to outline the allocation model. We claim that there are different parameters and constraints of the user request, network and data centre, and formulate most of them for designing our proposed algorithm. Table 2 presents the various symbols and notations used in this work.

Definition 1: (request Cost) – Concerned with the VM cost per user request based upon the resource type and duration of resource utilization time. Assume the VM_i is configured for user request U_i with resource type r_i for duration t and is defined as:

$$request_{cost} = \sum_{r_i} C(VM_i) * t \quad (1)$$

Definition 2: (Data Center) – A data center (DC) can be represented as a tuple $DC = \{R, AR, OS, VMM, VMP_{cost}, M_{cost}, ST_{cost}, DT_{cost}, PU\}$ where R = the region for locating the DC, AR = architecture of the DC, OS = operating system to compute the user application on demand, VMM = VM manager, VMP_{cost} = the cost per VM, M_{cost} = memory cost, ST_{cost} = storage cost, DT_{cost} = data transfer cost and PU = the physical hardware units.

Definition 3: (Total VM cost) - The actual monetary cost incurred by a user for executing a request in a cloud data center is comprised of two primary components: the virtual machine (VM) usage cost and the data transfer cost. Both elements directly contribute to the total amount charged to the user. The data transfer cost encompasses both inter-region and intra-region communication overhead, which significantly impacts user expenditure and the operational efficiency of the service provider. We formulate it through Eq. (2).

$$C(VM_{total}) = C(VM_{cost}) + C(DT_{cost}) \quad (2)$$

The formulation of VM usage cost in Eq. (3) further disaggregates the cost into VM processing cost, main memory cost, and storage cost. This decomposition is warranted by the fact that cloud providers independently price computing, memory, and storage resources, thereby enabling flexible provisioning and detailed billing. The VM processing cost accounts for CPU execution charges, while memory and storage costs are represented as linear functions of allocated memory units (x) and storage units (y), respectively.

The VMcost is derived in Eq. (3) using VMP_{cost} , M_{cost} and ST_{cost} as:

$$C(VM_{cost}) = C(VMP_{cost}) + \sum_{i=1}^x C(M_{cost}) + \sum_{i=1}^y C(ST_{cost}) \quad (3)$$

Where x = number of required memory units in MB of main memory

y = storage units in X-MB [25, 26] and the said values are addressed by the service provider during the DC configuration using a different pricing model.

Definition 4: (Expected Profit of Service Provider): It is the deduced cost for utilizing the resource r for a user request u that is produced from the revenue. The revenue is the collection of charges for the service while allocating the resources according to the user request [27]. Here, we compute the revenue as the collection of VM_{total} , Power consumption cost $C(Pow)$ and Network consumption resource cost $C(Net_r)$ and it is derived in Eq. (4).

$$profit_{Expected} = C(VM_{total}) + C(Pow) + C(Net_r) \quad (4)$$

Definition 5: (Estimated Processing Time): The cost is directly influenced by the estimated processing time which involves the resource utilization. Assume request u demanding n units of resource type r with processing speed γ , and it is defined in Eq. (5).

$$EPT_u = n * r / \gamma \quad (5)$$

Definition 6: (Power Consumption): The power consumption Pow at time t for the data center containing N_{phy} number of physical servers, the power consumption for the i^{th} physical server Pow_i of the data center is derived in Eq. (6).

$$Pow(t) = \sum_{i=1}^{N_{phy}} Pow_i(t) \quad (6)$$

CPU is the major power consumption component of the physical server in the data center. Every CPU has constant power consumption Pow_{idle} when it is in an idle state and can reach a maximum of Pow_{max} at full virtualization with N_{vm} number of virtual machines [28]. Therefore, we derive the $Pow_i(t)$ for the i^{th} server through the Eq. (7).

$$Pow_i(t) = Pow_{idle_i} + (Pow_{max_i} - Pow_{idle_i}) * util \quad (7)$$

Thus, the total energy consumed by the server is defined in Eq. (8).

$$Energy = \int_0^t Pow(utilization(t)) \quad (8)$$

Definition 7: (Utilization): We estimate the utilization ($util$) for the server i at time t hosting N_{vm} number of virtual machines with expected processing time EPT and is defined in Eq. (9).

$$util_{i,t} = \sum_{j=1}^{N_{vm}} EPT_j * util_j(t) \quad (9)$$

Definition 8: (Network resource cost): It defines the multiplication of the amount of data transferred, the unit price of network (P_{net}) and the number of network units (r). The network device cost is based upon the in-bound P_{net}^{in} and out-bound P_{net}^{out} cost. We define the cost function for network resource cost $C(Net_r)$ in Eq. (10).

$$C(Net_r) = (\int_t^{t+} P_{net}^{in} + P_{net}^{out}) * r \quad (10)$$

4. Proposed CD-BDAP Model

In this section, the proposed model suggests the concept of PSO with the formulation of new fitness functions incorporated into DC allocation. CD-BDAP defines various objective functions using different influencing parameters related to the data center and user request.

4.1. User–Provider Trade-off

The allocation of cloud data centers inherently involves conflicting objectives between cloud users and service providers. Users primarily aim to minimize service costs and response times, whereas service providers seek to maximize profit by enhancing resource utilization efficiency and reducing operational costs, such as energy consumption and network overhead [29]. Optimizing one objective in isolation often compromises the other, rendering it impractical to achieve simultaneous global optimality for both parties. In CD-BDAP, this conflict is addressed by formulating the data center selection problem as a weighted multi-objective optimization rather than a single-objective optimization. User-centric objectives, including VM usage cost and response time, and provider-centric objectives, including profit and energy consumption, are integrated into a unified fitness function using configurable weights. These weights enable the allocation policy to explicitly control the relative importance of user satisfaction versus provider profitability, thereby transforming the problem into a controllable trade-off rather than an implicit assumption.

4.2. Weighted Multi-Objective Optimization Formulation

In cloud environments, the allocation of data centers entails addressing multiple conflicting objectives that cannot be optimized simultaneously to their individual optima. Specifically, the minimization of user-centric metrics, such as service cost and response time, often conflicts with the maximization of service-provider-centric objectives, such as profit and energy efficiency. To address this issue, the proposed CD-BDAP mechanism conceptualizes data center selection as a weighted multi-objective optimization problem, wherein competing objectives are explicitly integrated into a single fitness function through the use of adjustable weights. The optimization goals can be expressed as: $\min VM_{cost}$, $\min EPT_u$, $\min Energy$ and $\max profit_{Expected}$. These objectives are unified into a scalar fitness function as:

$$Fitness = w_1 \left(-\frac{VM_{cost}}{\max(VM_{cost})} \right) + w_2 \left(\frac{profit_{Expected}}{\max(profit_{Expected})} \right) - w_3 \left(\frac{EPT_u}{\max(EPT_u)} \right) - w_4 \left(\frac{Energy}{\max(Energy)} \right) \quad (11)$$

where $w_1, w_2, w_3, w_4 \geq 0$ and $w_1 + w_2 + w_3 + w_4 = 1$.

The normalization terms ensure that objectives with different units and scales contribute proportionally to the fitness evaluation. By adjusting the weight parameters, CD-BDAP facilitates explicit control over the balance between user satisfaction and provider profitability. Assigning higher weights to user-centric objectives results in lower costs and reduced response times, whereas prioritizing provider-centric objectives with higher weights enhances energy efficiency and profit maximization. This formulation permits the optimization process to explore a range of feasible trade-off solutions, rather than imposing a singular, inflexible allocation policy.

4.3. Optimized Profit Based DC Selection

We discuss how to calculate the optimized DC list to gain maximum profit based upon the computed user request list to prepare the optimized DC list. The position and the vector are two major computing parameters in PSO to compute the optimized solution. The following steps of the designed model are discussed.

Step 1: In our proposed system we formulate the $Profit_{expected}$ as position and VM_{cost} as speed. So, the computed fitness function $f(DC)$ is formulated in Eq. (11).

Step 2: The DC with minimum pow and maximum $util$ is set as the position (P_{best}) treated as $DC_{initial}$ using Eq. (12).

$$DC_{initial} = \max f (util) \text{ subject to } \min f(pow) \quad (12)$$

Step 3: Compute the global best (G_{best}) using the fitness function formulated in Eq. (9) as the preferred DC (DC_{prefer}) in the region using Eq. (13).

$$DC_{prefer} = \max(f(DC)) \quad (13)$$

The work aims to allocate a suitable DC for each user request. Here, every computed possible solution is treated as the particle. Let us assume the representation of each particle as DC_i that is i^{th} particle as DC for k^{th} generation in the population space and can be defined in Eq. (14).

$$DC_i^k = \{DC_{i1}^k, DC_{i2}^k, \dots, DC_{iD}^k\} \quad (14)$$

Where, N = number of candidate solutions for DC.

$$DC^k = \{DC_1^k, DC_2^k, \dots, DC_N^k\} \text{ and here } 1 \leq i \leq N$$

Step 4: Set the position and velocity for the DC as DC_{ij}^k and V_{ij}^k , subject to $1 \leq i \leq N$, $1 \leq j \leq D$ and $1 \leq k \leq Max_{itr}$. Where, Max_{itr} = maximum number of iterations that is considered as a termination criterion.

Step 5: Compute Pbest and gbest DC for k^{th} iteration. So, after the k^{th} iteration, the particle DC_j^k iteration has a personal best as:

$$DC_{pbest_i}^k = \{DC_{pbest_{i1}}^k, DC_{pbest_{i2}}^k, \dots, DC_{pbest_{iD}}^k\}$$

And the global best within the population as:

$$DC_{gbest}^k = \{DC_{gbest_1}^k, DC_{gbest_2}^k, \dots, DC_{gbest_D}^k\}$$

Where DC_{gbest}^k = best computed DC according to our formulated fitness function.

Step 6: The particle position needs to be changed to a new position concerning velocities in the PSO mechanism after each iteration. So, we compute updated velocity using Eq. (16).

$$V_{ij}^k = \alpha * V_{ij}^{k-1} + c_1 * r_1 [DC_{pbest_{ij}}^k - DC_{ij}^{k-1}] + c_2 * r_2 [DC_{gbest_j}^{k-1} - DC_{ij}^{k-1}] \quad (16)$$

Where, α = inertia weight which puts a constraint on uncontrolled velocity.

c_1, c_2 = are acceleration coefficients

r_1, r_2 = random variables ranging from 0 to 1.

Step 7: The newly updated position is derived as:

$$DC_{ij}^k = DC_{ij}^{k-1} + V_{ij}^k \quad (17)$$

For each update, the current fitness value of the particle is calculated based on the new fitness value of the personal best modification.

$$DC_{pbest_i}^k = \begin{cases} DC_i^k, & \text{if } f(DC_i^k) < f(DC_{pbest_i}^{k-1}) \\ DC_{pbest_i}^{k-1}, & \text{otherwise} \end{cases} \quad (18)$$

Then the global best is also modified as follows.

$$DC_{gbest}^k = \begin{cases} DC_{pbest_i}^k, & \text{if } f(DC_{pbest_i}^k) < f(DC_{gbest}^k) \\ DC_{gbest}^k, & \text{otherwise} \end{cases} \quad (19)$$

Hence, each particle dimension is similar to the DC specifications and the positions of DCs are initialized randomly from the DC list. The proposed Optimize_DC algorithm gives the mapped index of the DC in the list which is optimized.

Algorithm: Optimize_DC()

Inputs: Data Center List D , Max_{itr} (*terminating criteria*)

Outputs: Optimized DC List D

```

1.  Begin
2.  Initialize the particles data centers with random positions and velocities
3.  Set  $DC = \{DC_1, DC_2, \dots, DC_N\}$ 
4.  Set the Data Center Matrix  $DC[i][j]$  where  $1 \leq j \leq D$ 
5.  Set  $C(VM_{cost})$  and  $C(DT_{cost})$ 
6.  Set  $DC_{initial} = \min(C(VM_{cost}), C(DT_{cost}))$ 
7.  Assign  $DC_{gbest}^k = DC_{pbest}^k = DC_{initial}$ 
8.  Calculate initial  $f(DC_{pbest})$  and  $f(DC_{gbest})$ 
9.  while ( $k < Max_{itr}$ ) do
10.   for  $j=1$  to  $D$  do
11.     Compute inertia parameter  $\alpha$ 
12.     for  $i=0$  to  $N$  do
13.      Update Particle velocity
14.       $V_{ij}^k = \alpha * V_{ij}^{k-1} + c_1 * r_1 [DC_{pbest_{ij}}^k - DC_{ij}^{k-1}] + c_2 * r_2 [DC_{gbest_j}^{k-1} - DC_{ij}^{k-1}]$ 
15.      Update particle position using i.e  $DC_{ij}^k = DC_{ij}^{k-1} + V_{ij}^k$ 
16.      if  $f(DC_i^k) < f(DC_{pbest_i}^{k-1})$  then
17.        $DC_{pbest_i}^k = DC_i^k$ 
18.       if ( $f(DC_{pbest_i}^k) < f(DC_{gbest}^k)$ ) then
19.         $DC_{gbest}^k = DC_{pbest_i}^k$ 
20.       end if
21.     end if
22.   Increment  $k++$ 
23. end for
24. end while
25. Stop

```

4.4. DC Allocation Mechanism

Let us consider $C = \{c_1, c_2, \dots, c_N\}$ as the set of DC clusters and $D \subset C$ where D is the optimized DC list computed using PSO optimization technique. The proposed allocation mechanism computes degree of load imbalance (DIB) and index of dissimilarity (I) [30] which presents the unevenness in the allocation among the user requests.

Workload and load imbalance

$$DIB = \frac{\max(ExeTime) - \min(ExeTime)}{Avg(ExeTime)} \quad (20)$$

Index of Dissimilarity: For DC allocation, we compute the index of dissimilarity to compute the overall extent of uneven distribution of two lists (DCs and user requests). The index score is interpreted as the percentage of one of the two groups and finds the matches of the user request and DC. It can be applied to the distribution of more than two groups and it is necessary to compare each group with every other.

Assuming N as the number of DC clusters in region R , U_{i1} is the number of users of group $G1$ in the i th cluster, U_{i2} as the number of users of group $G2$ in the i th cluster and U_1 and U_2 : the total number of users in each group. We can define index of dissimilarity (I) as:

$$I = \frac{1}{2} \sum_{i=1}^N \left| \frac{U_{i1}}{U_1} - \frac{U_{i2}}{U_2} \right| \quad (21)$$

The above computation of I provides ease of calculation and interpretation, but the redistribution is not possible at all. So, it can be redefined as:

$$I = \frac{1}{2} \frac{\sum N_i |U_{ij} - U_j|}{N U_j (1 - U_j)} \quad (22)$$

where $U_{ij} = \frac{N_{ij}}{N_i}$ and $U_j = \frac{N_j}{N}$

The denominator is the maximum value for the numerator term, which gives a condition of maximum segregation in each area. The areas with excess in one group can be combined, so the numerator can be rewritten as: $\sum |N_{ij} - \frac{N_i N_j}{N}|$

Let's assume $E_{ij} = \frac{N_i N_j}{N}$.

The excess number of users can be moved from one group to another group with the expected frequencies in each group when maximum segregation exists. The index is the ratio of the number that moves to achieve even distribution to the maximum possible. The index of dissimilarity for each group can be calculated in Eq. (23).

$$I = \frac{1}{2} \frac{\sum_i |N_{ij} - E_{ij}|}{N U_j (1 - U_j)} \quad (23)$$

We can compute generalized index for all the groups by substituting Eq. (23) as in Eq. (24).

$$I_G = \frac{1}{2} \frac{\sum_j \sum_i |N_{ij} - E_{ij}|}{\sum_j N U_j (1 - U_j)} \quad (24)$$

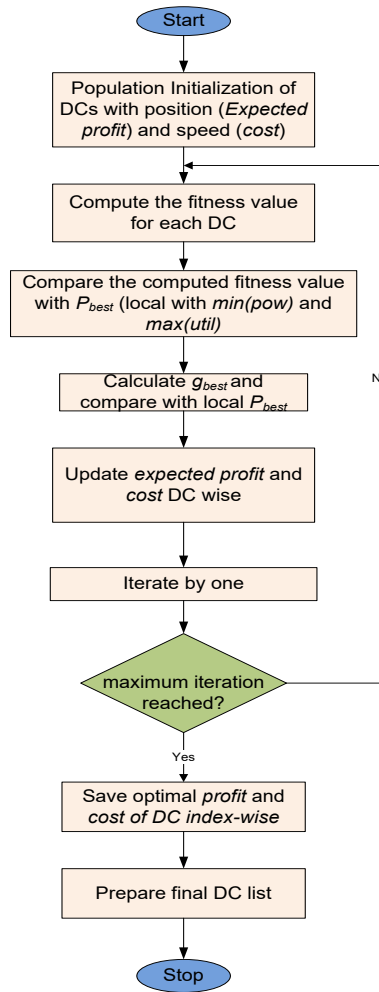


Fig. 1. Flow of the proposed DC optimization model

Algorithm: DC Allocation ()

1. **Begin**
2. Process Request Q , where $Q \neq \{\Phi\}$
3. **if** ($L = \{\Phi\}$ and $D = \{\Phi\}$)
4. Call Optimize_DC(), generate Optimized Data Center List D
5. **for** $i = 1$ to n
6. Select user L_i and D_i
7. Compute index of dissimilarity I_G in D_i
8. **if** ($I_G < I$)

```

9.      Evaluate  $Profit_{expected}$ 
10.     Assign  $L_i$  to  $D_i$ 
11.     Compute  $DIB$ 
12.     if( $DIB < 1$ )
13.         Select next user from  $L$ 
14.         Compute  $Cost_{Total}$ ,  $ResTime$ , and  $ProTime$ 
15.         Update  $L$ 
16.     Else
17.         Update  $L$  and  $D$ 
18.     end if else
19. Else
20.     Select next DC from  $D$ 
21. end of if else
22. end for
23. end if
24. Repeat until  $L$  is empty
25. Stop

```

5. Simulation and Performance Evaluation

We develop two Java-based applications: one for processing the user request and another to process the DCs. These two applications are running in *GuiMain.java* class. The proposed CD-BDAP is studied for three different sets of tasks 100, 200 and 300 with various requirements. During simulation, we configure the network bandwidth and delay. The bandwidth varies from 500 to 2000 Mbps and the network delay varies from 25 to 250 milliseconds. We have configured 3 sets of data centers during simulation; the tasks and DCs are shown in Table 3. To implement the proposed Optimize_DC (), we set the variable c_1 and c_2 as 0.2 and w as 0.6 [31] for better performance.

Table 3. Tasks and DCs used in simulation

Set	Tasks	No. of VMs	No. of DCs
1	100	32	5
2	200	54	7
3	300	80	10

Cost Optimization: We compute the cost as per our discussion, the optimized cost for 100, 200 and 300 tasks is shown in Figures 2 to 4. In Figure 2, the initial cost is initiated from 0.823\$ which is high in the 1st round of iteration. Progressively the cost is updated and the new cost is computed in each iteration as per our proposed member function. For a few of the initial tasks there is a deviation in cost. After the number of iterations increases, the cost deviation is marginal and reaches a point where the cost is constant and does not change even if increasing the number of iterations. For 100 tasks, it is optimized with a minimum cost of 0.3931\$ as shown in Figure 2. Similarly for 200 tasks the cost is optimized from 0.8797\$ which is more than the initial cost of our 1st experiment for 100 tasks but the optimized cost is 0.375\$, which is less than 0.3931\$, as shown in Figure 3. In our 3rd experiment we examine 300 tasks and find that the initial cost is the highest among all our experiments, which is 0.9006\$ but the optimized cost is the best among all, which is 0.3615\$.

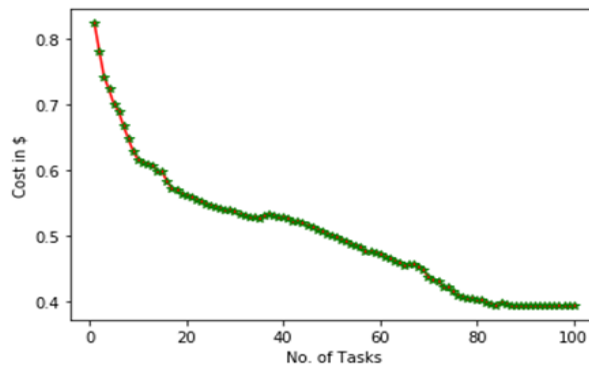


Fig. 2. Computing optimized cost for 100 tasks

It is observed that the initial cost is high when there are more tasks. As the iteration progresses, the initial cost is updated and reaches a level where it is not minimized even after increasing the number of iterations. When the proposed

system computes a greater number of tasks, it requires some computing cost, for which the initial cost is high and drops slowly. The comparison of the computed cost in our experiments is shown in Figure 5.

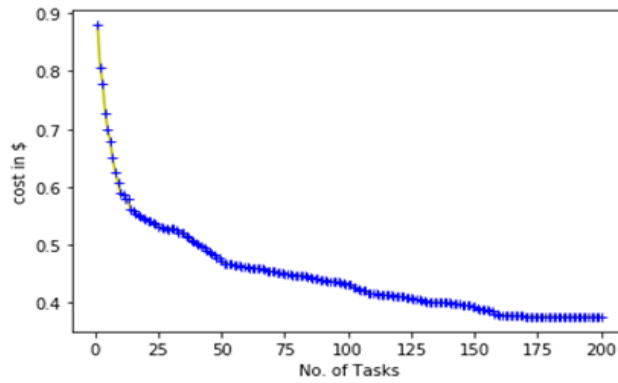


Fig. 3. Computing optimized cost for 200 tasks

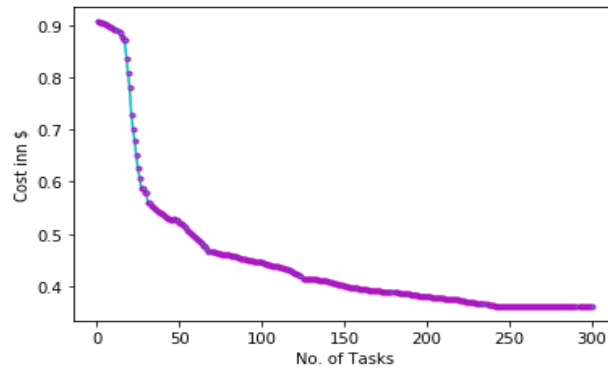


Fig. 4. Computing optimized cost for 300 tasks

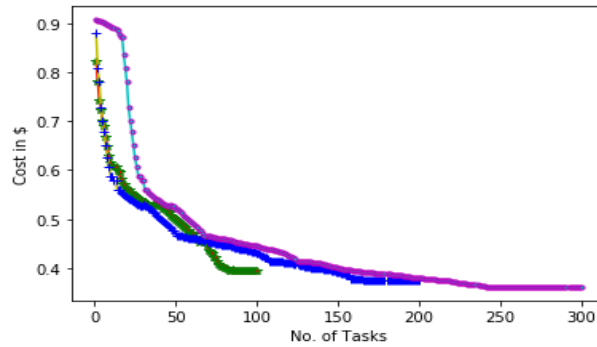


Fig. 5. Comparison of optimized cost for 100, 200 and 300 tasks

Table 4. Observation of various cost in \$

Test	Broker policy	Cost with CD-BDAP model		
		VM Cost \$	Data Transfer Cost \$	Total Cost \$
1	CDP	1.67	2.20	3.87
	ORT	1.53	2.11	3.64
	RDLB	1.86	2.32	4.18
2	CDP	1.55	1.72	3.27
	ORT	1.27	1.79	3.06
	RDLB	1.43	2.08	3.51
3	CDP	0.92	0.77	1.69
	ORT	0.83	0.60	1.43
	RDLB	1.07	0.85	1.92

Here, we find the total cost, response time and the data center processing time. These three computing parameters are evaluated using three existing broker policies such as: Optimised Response Time (ORT), Closest Data center Policy (CDP) and Dynamic Load Balancing (DLB). We notice that the computed cost of the proposed model performs better in ORT than in the other two broker policies, as depicted in Table 4. Similarly, it also computes better overall response time in our simulation as compared to others as shown in Table 5. Meanwhile the proposed CD-BDAP mechanism shows better DC processing time than the CDP broker policy, as shown in Table 6. It happens due to the optimization computing mechanism which computes the optimized DC.

Table 5. Generation of overall response time in avg, max and min in milliseconds (ms)

Test	Broker policy	Overall response time with CD-BDAP model		
		Avg	Min	Max
1	CDP	53.33	32.38	74.29
	ORT	47.22	34.14	64.30
	RDLB	63.03	39.63	86.43
2	CDP	58.48	38.75	78.22
	ORT	54.36	36.91	71.82
	RDLB	88.08	44.72	98.74
3	CDP	84.04	49.35	118.73
	ORT	77.87	46.29	109.46
	RDLB	100.06	62.83	137.29

Table 6. Evaluation of required dc processing time in milliseconds (ms)

Test	Broker policy	DC processing time with CD-BDAP model		
		Avg	Min	Max
3	CDP	1.95	0.74	3.17
	ORT	2.63	0.81	4.46
	RDLB	17.44	7.28	27.61
4	CDP	2.72	0.76	4.68
	ORT	3.11	0.84	5.39
	RDLB	25.85	12.68	39.03
5	CDP	5.35	1.21	9.49
	ORT	6.06	1.86	10.27
	RDLB	36.88	17.38	56.39

Power Consumption: We compute the power consumption cost using Eq. 6 through the simulation for the above task sets. To evaluate the power consumption, we refer to the existing model of power consumption and utilization model for VM [32] as shown in Table 7. We consider three levels of the VM except the idle level. In the idle level, the power consumption is 78.5 watts and the utilization is from 1 to 20%, and similarly for other levels.

Table 7. Utilization and power consumption [32]

Level	Utilization	Power Consumption (in watts (38 approx.))(pi)
Idle	1~20	78.5
1	21~30	P30=83
	31~40	P40=85
	41~50	P50=88
2	51~60	P60=93
	61~70	P70=102
3	71~80	P80=109
	81~90	P90=122
	91~100	P100=136

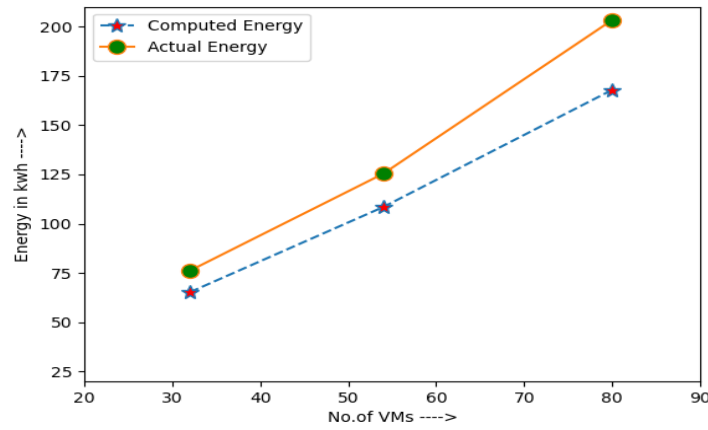


Fig. 6. Computation of energy consumption

We compute the actual power consumption for our set of VMs with reference to Table 7 without the optimization and also compute the power consumption using our proposed mechanism with the same reference. It is observed that the actual power consumption is higher and increases as the number of VMs increases. The computed energy consumption in the proposed mechanism is also increased but the deviation is smooth as shown in Figure 6. The proposed mechanism does not allow drastic changes in energy consumption with the increase of the number of VMs. We evaluate the mean values of both the energy and study that the proposed mechanism consumes 15.196% less energy as compared to the actual energy consumption referenced in the above table.

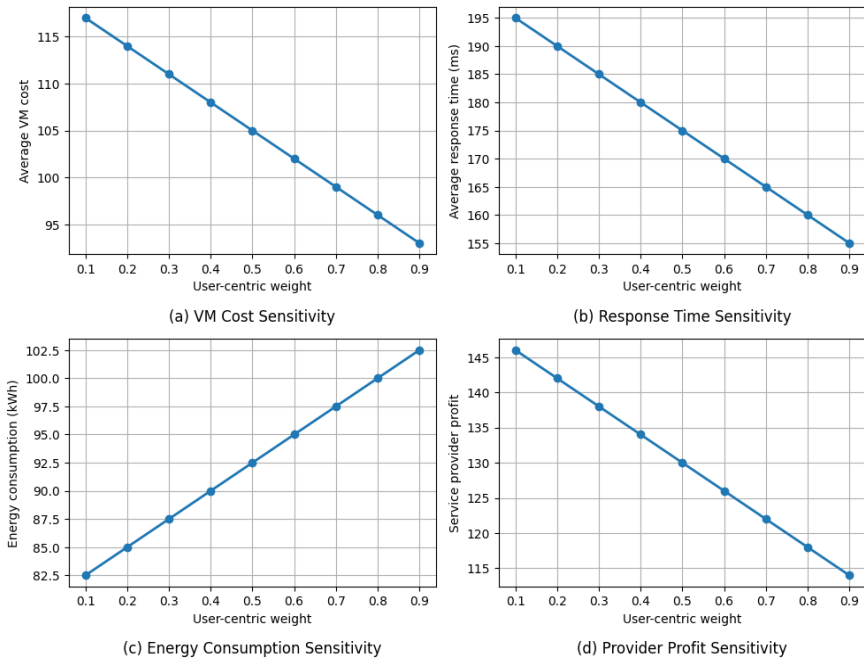


Fig. 7. Sensitivity analysis of CD-BDAP

Figure 7 illustrates the sensitivity of CD-BDAP to variations in the objective weights of the fitness function. As the weight assigned to user-centric objectives increases, there is a consistent reduction in both the average VM cost and response time, indicating an enhancement in user-level performance. Conversely, energy consumption and service provider profit exhibit a gradual increase and decrease, respectively, due to the reduced emphasis on provider-centric efficiency [33]. Notably, the observed variations across all performance metrics are smooth and monotonic, suggesting that CD-BDAP does not rely on fragile parameter tuning. Instead, the allocation decisions adapt predictably to changes in optimization priorities. This confirms that the weighted fitness formulation provides stable and interpretable trade-offs between user satisfaction and provider profitability, thereby validating the heuristic design of the proposed objective function. The sensitivity plots are based on controlled simulation experiments using CloudAnalyst under identical workload and configuration settings.

We perform parameter tuning experiments by examining the convergence behavior under varying inertia weights and acceleration coefficients. Figure 8 demonstrates the effect of altering the inertia weight (w) on the convergence of the fitness function. Among the tested values, ($w = 0.5$) achieved more rapid convergence and a lower final fitness

value. Similarly, the impact of acceleration coefficients on convergence behaviour is illustrated. The configuration ($c1 = c2 = 2.2$) consistently outperformed other settings by achieving stable and faster convergence.

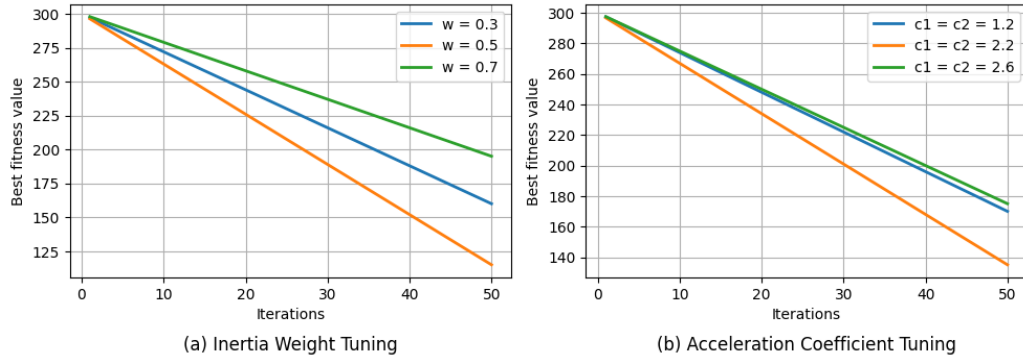


Fig. 8. Parameter tuning analysis

6. Performance Analysis with PSO-based Techniques

This section compares the performance of the proposed CD-BDAP with the benchmark PSO based mechanisms. Researchers have proposed various mechanisms with different objectives. In this paper, we compare the existing works related that aim to optimize the cost and processing time. We have created new Java classes in the existing *cloudsim.ext* package of Cloud Analyst. These packages are imported to the existing package *cloudsim.ext.gui*.

In Section 3 we have defined the VM_{cost} using Eq. (3) which computed the total VM cost VM_{total} defined in Eq. (2). We compared the computed VM_{total} of the proposed mechanism with the existing cost-aware PSO (CA-PSO) [34], Modified PSO (MPSO) [35] and hybrid PSO (HPSO) [36] for 100, 200 and 300 tasks, as shown in Figure 9. It is noted that the proposed mechanism provides better outcomes as compared with others in terms of cost. In the existing mechanisms it is observed that as the number of tasks increases, the total costs also increase in the same ratio, whereas in CD-BDAP the total cost increases slightly.

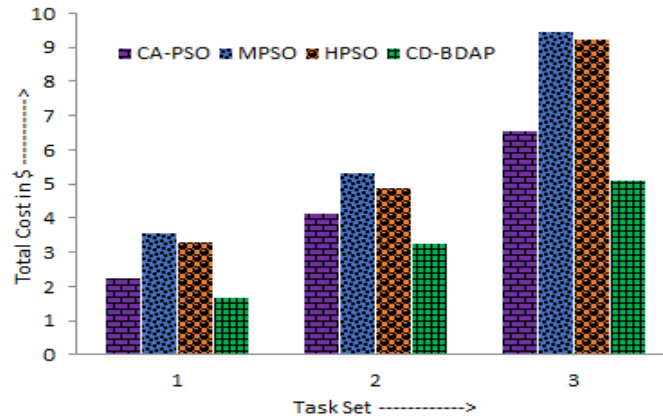


Fig. 9. Total cost compared with other PSO based techniques

Processing time is one of the major parameters in task allocation which specifies the efficiency of the allocation technique. Researchers have proposed various mechanisms to minimize processing time in cloud computing. The idea is to accept a number of tasks and provide better resource utilization. Most of the existing works implement the PSO mechanism to optimize the processing time. In this paper, we consider the existing works: Improved PSO (IPSO) [37], hybrid improved PSO (H-IPSO) [38], PSO-ACO [39] and PSO-DS [40]. We implement and compare these mechanisms with the performance of our proposed mechanism CD-BDAP for processing time. The comparative performance is studied for 100, 200 and 300 tasks, as shown in Figure 10.

Finally, we observe that the total cost is optimized as compared to the existing mechanisms. It is slightly increased with an increase in tasks for which the processing time is also optimized. The proposed mechanism requires less processing time to complete the tasks we have considered in our simulation. As our mechanism requires less processing time, it also reduces the total cost, which means that the proposed CD-BDAP mechanism provides better resource allocation and achieves effective resource utilization as compared to other existing techniques.

To ascertain the statistical significance of the performance enhancements achieved by the proposed CD-BDAP mechanism, a paired t-test is conducted in comparison to the baseline PSO-based mechanism. Each experiment is executed in 10 independent runs under identical CloudAnalyst configurations, encompassing the same workload size,

data center configuration, network delay, and bandwidth parameters. For each run, performance metrics are documented for both CD-BDAP and the corresponding baseline policy. The paired t-test is deemed appropriate in this context as the same experimental instances are used, resulting in dependent observations. A significance level of $\alpha = 0.05$ is employed. The hypotheses are stated as follows:

- Null hypothesis (H_0): There is no statistically significant difference between CD-BDAP and the baseline PSO-based method.
- Alternative hypothesis (H_1): CD-BDAP performs significantly better than the baseline method.

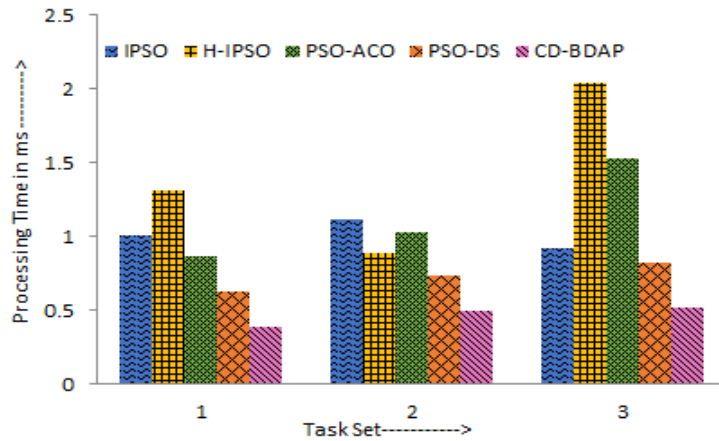


Fig. 10. Average processing time comparing with other pso based techniques

Table 8. Paired t-test results

Metric	t-Statistic	p-Value	Significance
Total VM Cost	-17.62	< 0.002	Significant
Average Response Time	-13.65	< 0.001	Significant
DC Processing Time	-20.24	< 0.002	Significant
Energy Consumption	-12.64	< 0.001	Significant

The results of the paired t-test indicate that the enhancements achieved by CD-BDAP over existing PSO-based mechanisms are statistically significant across all assessed metrics, as shown in Table 8. Specifically, CD-BDAP results in a notable reduction in VM cost, response time, processing time, and energy consumption. Given that all p-values are substantially below the 0.05 threshold, the null hypothesis is rejected for each metric. These findings confirm that the observed performance improvements are not attributable to random variation in simulation runs but are instead due to the proposed energy-aware cost denigration mechanism and workload similarity-based allocation strategy.

7. Conclusion and Future Works

Resource allocation and data center selection continue to pose significant challenges in cloud computing, primarily due to the conflicting objectives of cloud users and service providers. This study introduces CD-BDAP, a PSO-based data center allocation policy that effectively balances user-centric performance requirements with provider-centric economic objectives through an energy-aware cost optimization framework. Unlike traditional policies that consider VM cost as a static parameter, the proposed approach indirectly optimizes VM cost by minimizing energy consumption and operational overhead. Furthermore, CD-BDAP incorporates a workload similarity-aware allocation strategy utilizing a dissimilarity index, which facilitates enhanced consolidation and more efficient resource utilization. The allocation problem is formulated using a weighted objective function, allowing explicit control over the trade-offs between user cost, response time, energy consumption, and service provider profit, rather than relying on implicit assumptions. Simulation results obtained using CloudAnalyst demonstrate that CD-BDAP consistently reduces VM cost and response time while enhancing energy efficiency and overall provider profit, compared to existing broker policies and PSO-based allocation techniques under identical experimental conditions. Future research will extend CD-BDAP by incorporating real cloud workload traces, dynamic energy models, and multi-objective optimization techniques to further investigate the trade-offs between economic efficiency and quality of service in large-scale cloud environments.

All the Declarations and Statements

Author Contributions Statement

Suwendu Chandan Nayak: Conceptualization, Methodology, Formal Analysis, Review and Editing and Supervision: proposed research ideas, constructed the overall framework, and supervised model execution.

Subash Chandra Tripathy: Contributed to the literature survey, documented the technical background of the study and Software Implementation: Handled data acquisition, and implementing the research model, Performed in-depth analysis of experimental results, prepared performance charts, and ensured the statistical robustness of the evaluation.

Rekha Sahu: Model, Validation, and Performance Evaluation: Led the model design, validated results using standard metrics, benchmarked performance against existing methods and ensured the statistical robustness of the evaluation.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

None.

Data Availability Statement

None.

Ethical Declarations

None

Acknowledgments

None

Declaration of Generative AI in Scholarly Writing

None

References

- [1] J. Son, A. V. Dastjerdi, R. N. Calheiros, and R. Buyya, "SLA-Aware and energy-efficient dynamic overbooking in SDN-based cloud data centers," *IEEE Trans. Sustain. Comput.*, vol. 2, no. 2, pp. 76–89, 2017.
- [2] S. Zou, W. Ji, and J. Huang, "BTP: automatic identification and prediction of tasks in data center networks," *Journal of Cloud Computing*, vol. 11, no. 1, Art. no. 36, 2022.
- [3] A. Mulahuwaish, S. Korbil, and B. Qolomany, "Improving datacenter utilization through containerized service-based architecture," *Journal of Cloud Computing*, vol. 11, no. 1, Art. no. 44, 2022.
- [4] N. Tripathy, S. Sahoo, S. C. Nayak, and C.-C. Lee, "An efficient task offloading approach to reduce energy consumption in fog architecture," *Computers and Electrical Engineering*, vol. 128, Art. no. 110648, 2025.
- [5] Z. Zhou, J. Chang, Z. Hu, J. Yu, and F. Li, "A modified PSO algorithm for task scheduling optimization in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 24, Art. no. e4970, 2018.
- [6] Nayak, Suwendu Chandan, and Chitaranjan Tripathy. "Deadline based task scheduling using multi-criteria decision-making in cloud environment." *Ain Shams Engineering Journal* 9, no. 4, pp 3315-3324, 2018.
- [7] E. Pacini, C. Mateos, and C. García Garino, "Distributed job scheduling based on Swarm Intelligence: A survey," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 252–269, 2014.
- [8] D. Chaudhary and B. Kumar, "A new balanced particle swarm optimisation for load scheduling in cloud computing," *Journal of Information & Knowledge Management*, vol. 17, no. 1, Art. no. 1850009, 2018.
- [9] A. Ullah, N. M. Nawi, and S. Ouhamme, "Recent advancement in VM task allocation system for cloud computing: review from 2015 to 2021," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 2529–2573, 2022.
- [10] Nayak, Suwendu Chandan, and Chitaranjan Tripathy. "Deadline sensitive lease scheduling in cloud computing environment using AHP." *Journal of King Saud University-Computer and Information Sciences* 30, no. 2, pp 152-163, 2018.
- [11] Bhoi, Ashok Kumar, Manas Ranjan Kabat, Suwendu Chandan Nayak, and Gopinath Palai. "Renewable energy source based quality of service (QoS)-aware routing mechanism in cloud network." *Wireless Networks* 28, no. 4, pp. 1703-1718, 2022.
- [12] S. H. H. Madni, M. S. A. Latiff, Y. Coulilaly, and S. M. Abdulhamid, "Recent advancements in resource allocation techniques for cloud computing environment: a systematic review," *Cluster Comput.*, vol. 20, no. 3, pp. 2489–2533, 2017.
- [13] A. Kaur, P. Singh, R. S. Bathth, and C. P. Lim, "Deep-Q learning-based heterogeneous earliest finish time scheduling algorithm for scientific workflows in cloud," *Software: Practice and Experience*, vol. 52, no. 3, pp. 689–709, 2022.
- [14] M. H. Kim, J. Y. Lee, S. A. Raza Shah, T. H. Kim, and S. Y. Noh, "Min-max exclusive virtual machine placement in cloud computing for scientific data environment," *J. Cloud Comput.*, vol. 10, no. 1, 2021.
- [15] İ. Çağlar and D. T. Altılar, "Look-ahead energy efficient VM allocation approach for data centers," *J. Cloud Comput.*, vol. 11, no. 1, 2022.

- [16] S. Mishra, A. K. Sangaiah, M. N. Sahoo, and S. Bakshi, "Pareto-optimal cost optimization for large scale cloud systems using joint allocation of resources," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 11, pp. 15375–15393, 2023, doi: 10.1007/s12652-019-01601-x.
- [17] L. Chen, Y. Guo, X. Li, and R. Ruiz, "Hybrid resource provisioning for workflow scheduling in cloud computing," in *Human Centered Computing (HCC 2016)*, Lecture Notes in Computer Science, vol. 9567, Cham: Springer, 2016, pp. 34–46, doi: 10.1007/978-3-319-31854-7_4.
- [18] S. Gharehpasha, M. Masdari, and A. Jafarian, "Virtual machine placement in cloud data centers using a hybrid multi-verse optimization algorithm," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 2221–2257, 2021.
- [19] X. Zeng, S. K. Garg, Z. Wen, P. Strazdins, A. Y. Zomaya, and R. Ranjan, "Cost efficient scheduling of MapReduce applications on public clouds," *J. Comput. Sci.*, vol. 26, pp. 375–388, 2018.
- [20] A. Xu, Y. Yang, Z. Mi, and Z. Xiong, "Task scheduling algorithm based on PSO in cloud environment," in *Proc. 2015 IEEE 12th Int. Conf. Ubiquitous Intelligence and Computing, Int. Conf. Autonomic and Trusted Computing, and 15th Int. Conf. Scalable Computing and Communications (UIC-ATC-ScalCom-CBDCOM-IOP)*, 2015, pp. 1055–1061.
- [21] J. Meshkati and F. Safi-Esfahani, "Energy-aware resource utilization based on particle swarm optimization and artificial bee colony algorithms in cloud computing," *J. Supercomput.*, vol. 75, no. 5, pp. 2455–2496, 2019.
- [22] S. Mishra, M. N. Sahoo, A. K. Sangaiah, and S. Bakshi, "Nature-inspired cost optimisation for enterprise cloud systems using joint allocation of resources," *Enterprise Information Systems*, vol. 15, no. 2, pp. 174–196, 2021, doi: 10.1080/17517575.2019.1605001.
- [23] J. P. B. Mapetu, Z. Chen, and L. Kong, "Low-time complexity and low-cost binary particle swarm optimization algorithm for task scheduling and load balancing in cloud computing," *Appl. Intell.*, vol. 49, no. 9, pp. 3308–3330, 2019.
- [24] M. Abd Elaziz and I. Attiya, "An improved Henry gas solubility optimization algorithm for task scheduling in cloud computing," *Artificial Intelligence Review*, vol. 54, no. 5, pp. 3599–3637, 2021.
- [25] J. Venkata Krishna, G. Apparao Naidu, and N. Upadhyaya, "A Lion-Whale optimization-based migration of virtual machines for data centers in cloud computing," *International Journal of Communication Systems*, vol. 31, no. 8, Art. no. e3539, 2018.
- [26] S. P. Praveen, K. T. Rao, and B. Janakiramaiah, "Effective Allocation of Resources and Task Scheduling in Cloud Environment using Social Group Optimization," *Arab. J. Sci. Eng.*, vol. 43, no. 8, pp. 4265–4272, 2018.
- [27] H. M. Alkhashai and F. A. Omara, "An enhanced task scheduling algorithm on cloud computing environment," *Int. J. Grid Distrib. Comput.*, vol. 9, no. 7, pp. 91–100, 2016.
- [28] A. Al-maamari and F. A. Omara, "Task Scheduling Using PSO Algorithm in Cloud Computing Environments," *Int. J. Grid Distrib. Comput.*, vol. 8, no. 5, pp. 245–256, 2015.
- [29] I. Casas, J. Taheri, R. Ranjan, L. Wang, and A. Y. Zomaya, "GA-ETI: An enhanced genetic algorithm for the scheduling of scientific workflows in cloud environments," *J. Comput. Sci.*, vol. 26, pp. 318–331, 2018.
- [30] A. M. Manasrah, T. Smadi, and A. Almomani, "A Variable Service Broker Routing Policy for data center selection in cloud analyst," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 29, no. 3, pp. 365–377, 2017.
- [31] A. S. A. Beegom and M. S. Rajasree, "Integer-PSO: a discrete PSO algorithm for task scheduling in cloud computing systems," *Evol. Intell.*, vol. 12, no. 2, pp. 227–239, 2019.
- [32] C.-H. Lien, M. F. Liu, Y.-W. Bai, C. H. Lin, and M.-B. Lin, "Measurement by the software design for the power consumption of streaming media servers," in *Proc. IEEE Instrumentation and Measurement Technology Conference (IMTC 2006)*, 2006, pp. 1597–1602, doi: 10.1109/IMTC.2006.328685.
- [33] Bhoi, Ashok Kumar, Manas Ranjan Kabat, Suvendu Chandan Nayak, and Gopinath Palai, "Energy efficient task allocation and consolidation in multicast cloud network," *Wireless Networks* 28, no. 8, pp. 3349–3366, 2022.
- [34] G. Zhao, "Cost-Aware Scheduling Algorithm Based on PSO in Cloud Computing Environment," *Int. J. Grid Distrib. Comput.*, vol. 7, no. 1, pp. 33–42, 2014.
- [35] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. 1998 IEEE Int. Conf. Evol. Comput. (ICEC)*, Anchorage, AK, USA, 1998, pp. 69–73. DOI: 10.1109/ICEC.1998.699146.
- [36] S. N. Sivanandam and P. Visalakshi, "Dynamic task scheduling with load balancing using parallel orthogonal particle swarm optimization," *Int. J. Bio-Inspired Comput.*, vol. 1, no. 4, pp. 276–286, 2009.
- [37] S. Zhan and H. Huo, "Improved PSO-based task scheduling algorithm in cloud computing," *J. Inf. Comput. Sci.*, vol. 9, no. 13, pp. 3821–3829, 2012.
- [38] A. June, "Research Paper on Optimized Utilization of Resources Using PSO and Improved Particle Swarm Optimization (IPSO) Algorithms in Cloud Computing," vol. 2, no. 2, pp. 499–505, 2014.
- [39] J. JU, W. BAO, Z. WANG, Y. WANG, and W. LI, "Research for the Task Scheduling Algorithm Optimization based on Hybrid PSO and ACO for Cloud Computing," *Int. J. Grid Distrib. Comput.*, vol. 7, no. 5, pp. 87–96, 2014.
- [40] I. Casas, J. Taheri, R. Ranjan, and A. Y. Zomaya, "PSO-DS: a scheduling engine for scientific workflow managers," *J. Supercomput.*, vol. 73, no. 9, pp. 3924–3947, 2017.

Authors' Profiles



Subash Chandra Tripathy is a PhD scholar in the Department of Computer Science and Engineering at Silicon University, India and also working as an assistant professor in the School of Computer Science at SoA University, India. He completed his MCA from North Odisha University and his MTech in Computer Science and Engineering from BPUT, Odisha, India. He has more than 17 years of teaching experience and 4 years of research experience in the field of cloud computing, fog computing and machine learning. He has published a number of research articles in international journals and conferences.



Dr. Suvendu Chandan Nayak completed B.Tech and M.Tech in Computer Science and Engineering Biju Patnaik University of Technology, India in the year 2006 and 2011 respectively. He received his Ph.D. degree from Veer Surendra Sai University of Technology (Formally University College of Engineering, Burla) India in Computer Science & Engineering in the year 2019. Currently, he is a Senior Assistant Professor in the Department of Computer Science and Engineering at Silicon University, Bhubaneswar, India. Five MTech, two MSc and one PhD scholars have received their degrees under his supervision. Currently, four PhD and one Msc scholar are continuing their research under him. He is the Co-PI of a funded project of Govt. of Odisha, India. His area of research is AI, ML, Biomedical, Cloud computing and IoT. He has published more than 45 research papers in different International / National journals and conferences. He is associated with different foreign universities as a supervisor and examiner.



Dr. Rekah Sahu is currently working as Assistant Professor in the Department of Computer Science and Engineering at Silicon University, Odisha, India. She received her MSc and MTech in Computer Science from Utkal University, India and PhD in Computer Science and Engineering from KIIT Deemed to be University, India in the year 2021. Her area of research is AI, ML, Biomedical and Computer vision. She has delivered a number of expert talks on AI and ML at different workshops. She has published more than 40 research articles in different international journals and conferences. Currently, 3 PhD scholars and 3 MSc scholars are working under her supervision.

How to cite this paper

Subash Chandra Tripathy, Suvendu Chandan Nayak, Rekah Sahu, "Cost Denigration Based Data Center Allocation Policy Using Modified Parallel PSO Optimization Technique for Multi-cloud Framework", International Journal of Information Technology and Computer Science(IJITCS), Vol.18, No.3, pp.28-44, 2026. DOI:10.5815/ijitcs.2026.03.03