INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

OPEN ACCESS
Received: 12-02-2023
Accepted: 04-04-2023
Published: 28-04-2023

Citation: Reddy BR, Indiramma M, Nagarathna N (2023) Intelligent Particle Swarm Optimization Based Resource Provisioning Technique in Cloud Computing. Indian Journal of Science and Technology 16(16): 1241-1249. https://doi.org/10.17485/IJST/v16i16.308

* **Corresponding author**.

bramanareddy_cse@cbit.ac.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Intelligent Particle Swarm Optimization Based Resource Provisioning Technique in Cloud Computing

B Ramana Reddy[1,2*], **M Indiramma**[3], **N Nagarathna**[3]

1 Research Scholar, Department of CSE, B.M.S. College of Engineering, Bangalore, India
2 Assistant Professor, Department of CSE, Chaitanya Bharathi Institute of Technology, Hyderabad, India
3 Professor, Department of CSE, B.M.S. College of Engineering, Bangalore, India

Abstract

Objectives: To find the optimal allocation of resources and minimize the overall cost while meeting the performance requirements of the applications. **Methods:** The proposed Intelligent PSO-based resource optimization in cloud computing evaluates the quality of the solutions based on their resource allocation parameters. Cloud Sim software is used as a simulation tool for testing and evaluating new solutions and strategies in the cloud. The Closest Data Center Service Broker Policy is implemented in Cloud Simulation. **Findings:** The proposed technique assigns workloads effectively on available resources with an improvement of 10.46% in electricity consumption. **Novelty:** The algorithm can further be employed for identifying the unused VM's in the Data Center to reduce the cost.

Keywords: Cloud computing; Resource Scheduling; Load balancing; Particle Swarm Optimization; Quality of Service

1 Introduction

In the field of computer science, cloud computing technology is showing phenomenal growth due to the advancement of the internet[(1)]. Cloud computing provides infrastructure, platform, and software as services. Cloud resources are providing customers with a pay-as-you-use model. To provide quality services to customers, there is a Service Level Agreement (SLA) between the customers and the cloud service providers as shown in Figure 1. Cloud service providers need to verify if a sufficient number of resources are available to customers to ensure that QoS requirements like execution time, deadline, and budget restrictions are met. However, running too many applications on a single resource can result in a drastic performance loss, which deters cloud consumers. It is challenging to match workloads to the right resources for cloud execution. The existing approaches of Resource optimization in cloud computing involves managing multiple resources and balancing different performance metrics, which can be complex and challenging. Developing an effective optimization algorithm requires significant expertise and resources. For effective resource use, three primary QoS limitations must be taken into account[(2)].

1. Must meet deadlines with minimum execution time and cost
2. Minimum electricity consumption
3. User satisfaction

A PSO-based resource scheduling algorithm, schedules workloads in a cloud environment to save execution time, cost, and energy. By figuring out how workloads and resources interact in a cloud setting was expanded. Experimental findings enhance QoS metrics like resource availability, reliability, latency, and resource consumption to the maximum extent possible[3].

The Particle Swarm Optimization and Genetic Algorithm (PSO-GA) was used to make runtime decisions for exploring the objectives of the resource allocation plan[4]. The issue of resource allocation for cloud-based services is addressed by introducing the workload time window. Further, it builds a calculation model for optimizing VM resource allocation plans[5]. The PSO-GA-based method observed the advantages of PSO and GA and improved their inadequacies in population diversity, search range, and convergence speed. The algorithm strategy objective takes the current and future workloads into the process of producing resource allocation plans.

In the proposed research work, the intelligent Particle Swarm Optimization focuses on minimizing execution time, cost and electricity consumption.

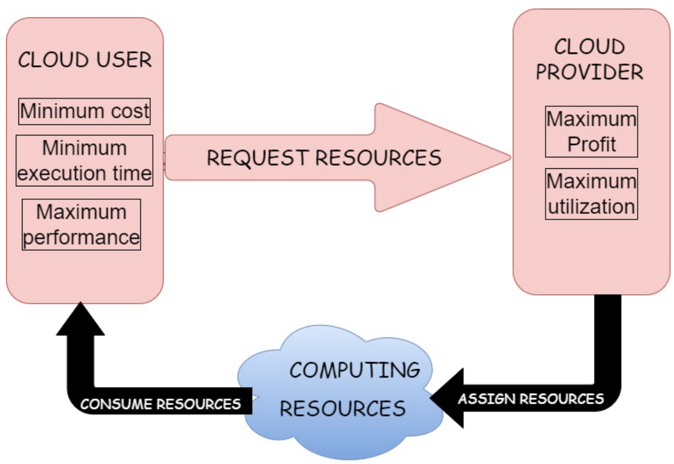

Fig 1. Resource Request in Cloud Computing

2 Methodology

2.1 Proposed Intelligent Particle Swarm Optimization-based resource

2.1.1 Scheduling Technique

In cloud computing, resource scheduling is important for resource management[6]. It directs the allocation of cloud workloads to cloud resources. After, applications are scheduled to cloud resources using an Intelligent Particle Swarm optimization-based heuristic framework, which lowers the cost of computation and data transfer. Most existing research takes into account of cloud computing's fundamental characteristics in order to execute heterogeneous cloud work loads quickly and affordably[7]. The population in the I-PSO algorithm is defined as particles, where the particles are initialized randomly. In every new generation, the fitness value of each particle is calculated and the two values of the particles need to be calculated: 1. Local best(L_{best}), 2. Global best(G_{best}). Where the L_{best} of a particle is the best result reached by the particle so far, and the G_{best} is the best result among the whole population. Intelligent PSO optimization technique works on global search[8]. Every single particle controls its own independent course according to the local and global best in every generation. Numerous NP-hard issues, including task distribution and resource scheduling, will be resolved by intelligent PSO.

Workloads : Workloads that are initiated by the user are placed in a queue for execution and processing purposes.

Resource Manager: The resource manager maintains information about resources, QoS, and SLA.

Quality of Service: Quality of Services like availability, latency, resource utilization, and reliability must be maintained.

Service Level Agreement : Provides information about suitable service level agreements between customers and cloud service providers.

Workload Analyzer: Workload Analyzer identifies different characteristics of cloud workloads.

Resource details: Maintains resource details such as availability of virtual machines, size of virtual memory, cost of cloud resources, and type of cloud resource.

Resource provisioner: It provides workload to the demanded resources for their execution in the cloud only if the resources are available in the resource pool.

Resource scheduler: It executes all the workloads on a provisioned resource efficiently.

The Proposed architecture of resource optimization is described in Figure 2.

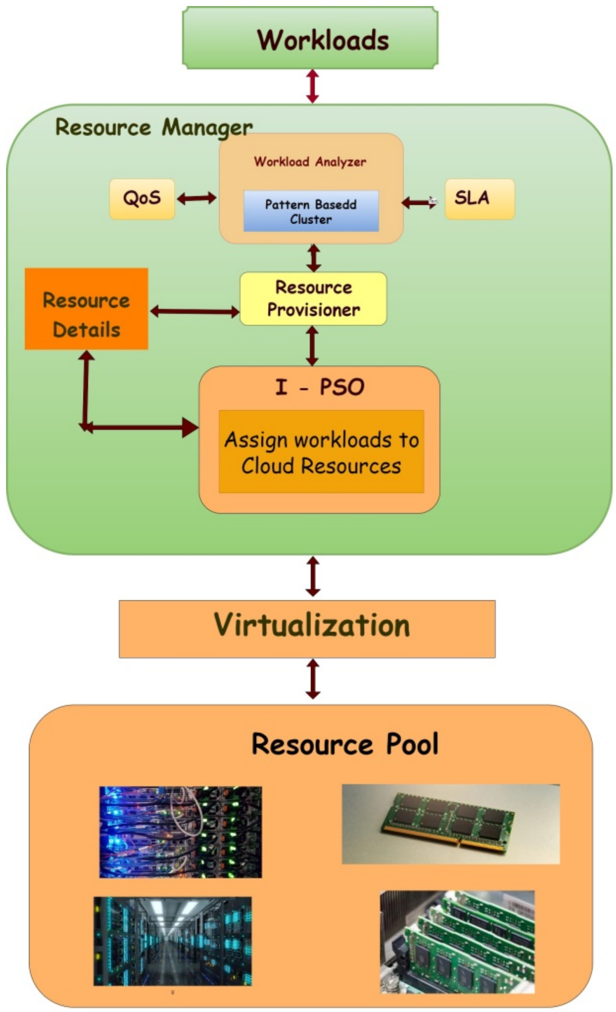

Fig 2. Architecture of proposed Resource optimization

2.1.2 Objective Function

The fitness value is calculated based on execution time, cost, and electricity consumption as shown in equation 1.

$$\text{fitness } = c1 * \text{ execution time } + c2 * \text{ execution cost } + c3 * \text{ electricitycon} \tag{1}$$

$0 \leq \text{c1} \leq 1, 0 \leq \text{c2} \leq 1$ and $0 \leq \text{c3} \leq 1$

where c1, c2, and c3 are weights to prioritize the components of the fitness function.

Execution time is the time taken to execute the workload on the allocated resources as follows:

$$\text{Execution}_{\text{ time}} = \min\left(t\left(w_i, r_i\right)\right) \text{ for } 1 \leq i \leq n \tag{2}$$

where $t(w_i, r_i)$ is the time taken by workload w_i executed by resource r_i.

Execution cost is the cost of the workloads executed by the assigned resources in the following equation.

$$\text{Execution}_{\text{cost}} = \min\left(c\left(w_i, r_i\right)\right) \text{ for } 1 \leq i \leq n \tag{3}$$

where $c(w_i, r_i)$ is the cost of workload w_i executed by resource r_i. electricity consumption is the electricity consumed in resource utilization calculated by

$$\text{electricity}_{co\,n} = \text{electricity}_{vm} + \text{electricity}_{\text{memory}} + \text{electricity}_{\text{misc}} \tag{4}$$

Where electricity_{vm} is the virtual machine electricity consumption, $\text{electricity}_{memory}$ is the electricity consumption for memory operations and $\text{electricity}_{misc}$ is the electricity consumption for fans and other miscellaneous parts.

$$\text{electricity}_{vm} = vm_{\text{idle}} + \left(vm_{\text{running}} - vm_{\text{idle}}\right) \times vm_{\text{processor}} \tag{5}$$

where vm_{idle} denotes the idle state of the virtual machine, $vm_{running}$ denotes the virtual machine running time power consumption, and $vm_{processor}$ denotes the virtual machine processor capacity.

2.2 Proposed Algorithm

In this section, the description regarding I-PSO based algorithm for resource provisioning technique in the cloud environment is shown in Algorithm 1.

Algorithm 1: Proposed Intelligent PSO-based resource scheduling Algorithm

Result: intelligent mapping of workloads to the VMs.

```
initialize VMs
initialize workloads
initialize a random feasible solution
for i=1 to PS
do
Pv ← RV
Pp ← RP
if fitness(G_best) > f itness(P )
then
G_best ← P
end
end
while iterations are not reached max
do
for P ∈Ppop
do
Pv ← update velocity
Pp ← update position
if f itness(P ) < f itness(L_best)
then
L_best ← P
end
if f itness(Lbest) < f itness(Gbest)
then
Gbest ← Lbest
end
end
return(Gbest)
end
if any unassigned VMs then
VMs moved to sleep mode.
```

32 end
Intelligent PSO Terminology:
The notations used in an I-PSO algorithm are presented in Table 1.

- **Particle:** A particle is similar to a flock of birds searching for food. Every particle has velocity. Fitness values calculate a particle's performance. For the proposed I-PSO algorithm, cloud workloads are considered particles.
- **Population size:** In the proposed I-PSO, available resources in the cloud are considered as population size.
- **Random velocity:** Every particle's velocity is updated with L_{best} and G_{best} values.
- **Particle velocity:**The particle's velocity is calculated based on particle position.
- **Particle position:** positions depend on the submission status, waiting state, ready state, execution state, and completion state.
- **Global best (G_{best}):** Best position of a particle among the whole group of particles.
- **Local best (L_{best}):** Best position reached by a particle

Table 1. Notations and its description

Notation	Description
S	Number of particles
PS	Population size
RV	Random velocity
RP	Random position
Pv	Particle velocity
Pp	Particle position
Ppop	Particle population
G_{best}	Global best position
L_{best}	Local best position

In Figure 3, all the algorithm steps are described in the form of a flow chart.

3 Results and Discussion

The experiment has been conducted in a simulator with size of 700MB image conversion cloud workload. Once the simulation starts a sequence of steps will be execute as per the mentioned order.

1. Cloud user submits workload details like name and type of workload. Workload analyzer will submit these details to Resource Manager.
2. Resource manager processes workload details and then asks the user for budget restrictions and deadline details.
3. Once the cloud user submits budget and deadline restrictions the Resource manager will generate tentative schedule and cost.
4. Cloud user sends a confirmation of the details, SLA agreements to the Resource Manager[9].
5. Cloud user pays requested amount and executes the workloads on allocated resources

The performance of the proposed I-PSO resource provisioning technique is compared with the existing scheduling algorithms[10,11].

Results – Execution time: In Table 2 results compared at 45 workloads, execution time in I-PSO is 2.74% lesser than ACO, 4.31% lesser than GA. At 90 workloads, execution time in I-PSO is 3.53% lesser than ACO, 5% lesser than GA. Figure 4 shows that execution time of I-PSO is better than ACO and GA.

Results – Execution cost: In Table 3 results compared at 45 workloads, the cost incurred in I-PSO is $260, ACO is $266 and GA is $292. At 90 workloads the cost incurred in I-PSO is $412, ACO is $422 and GA is $442. Figure 5 shows that the execution cost of I-PSO is minimum compared to the ACO and GA.

Results – Electricity consumption: In Table 4 results compared at 45 workloads the I-PSO consumes 8.97% lesser than ACO, 14.10% lesser than GA. At 90 workloads the I-PSO consumes 10.46% lesser than ACO, 16.27% lesser than GA. Figure 6 shows that the electricity consumption is minimum compared to the ACO and GA.

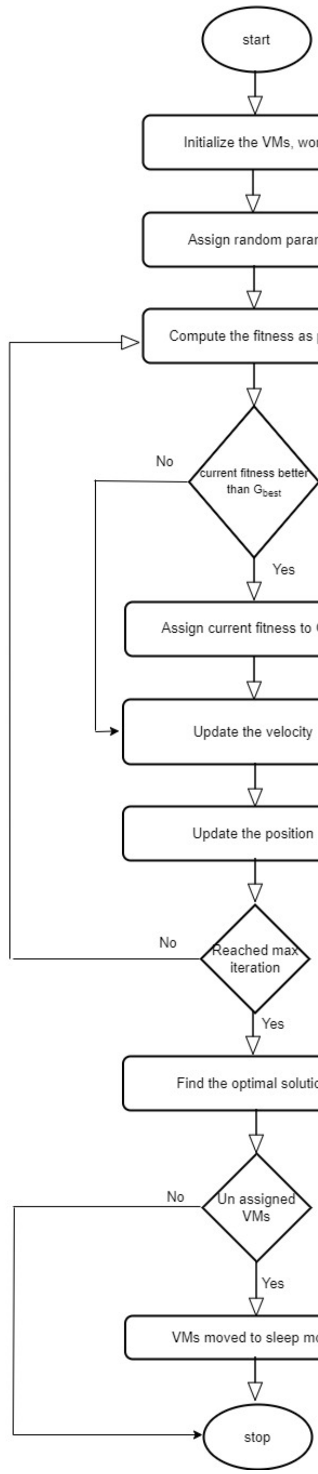

Fig 3. Flowchart of the proposed system

Table 2. Comparison of I-PSO Execution time (ms) with GA and ACO works

No. of work loads	Execution Time (ms)		
	GA	ACO	I-PSO
15	225	221	198
30	410	402	380
45	532	524	510
60	698	788	765
75	950	948	918
90	1218	1201	1160

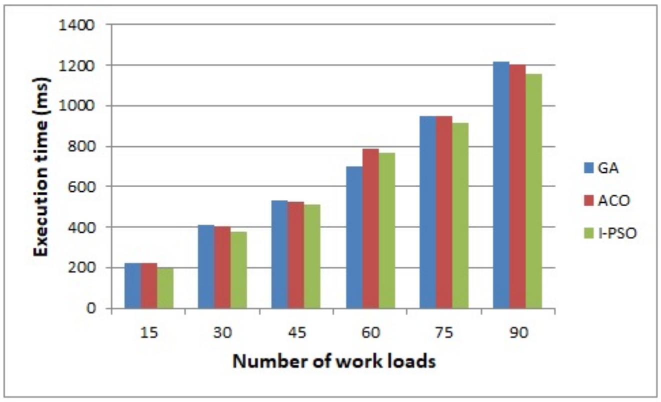

Fig 4. Execution time vs no.of work loads

Table 3. Comparison of I-PSO Execution cost($) with GA and ACO works

No. of work loads	Execution cost ($)		
	GA	ACO	I-PSO
15	196	176	169
30	230	204	198
45	292	266	260
60	332	307	297
75	368	346	336
90	442	422	412

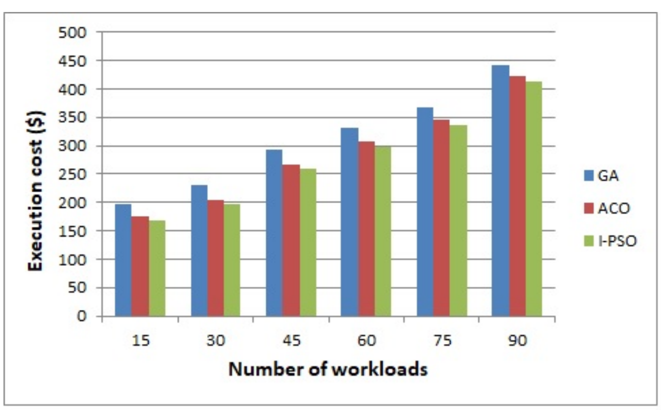

Fig 5. Execution cost vs no.of work loads

Table 4. Comparison of I-PSO electricity consumption (kwh) with GA and ACO works

No. of work loads	Electricity Consumption (Kwh)		
	GA	ACO	I-PSO
15	82	76	70
30	88	82	76
45	89	85	78
60	92	87	81
75	96	92	84
90	100	95	86

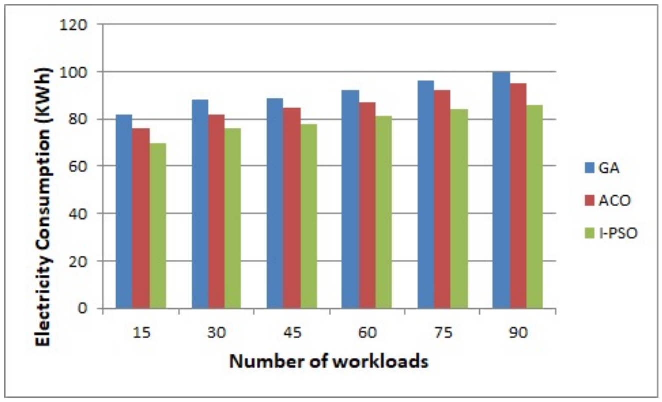

Fig 6. Electricity consumption vs no.of workloads

4 Conclusion

In this research work, different techniques such as load balancing and resource allocation methods are provided which ensures that resources are used optimally. Furthermore, the Proposed Intelligent PSO algorithm can help predict idle virtual machines, enabling providers to proactively allocate resources. However, the I-PSO algorithm is unable to predict the upcoming workloads due to ongoing changes of user requirements.

To further improve this research work, the computation of the fitness value can be more accurate by considering the priority of virtual machines. Thus, the highest priority virtual machine will be assigned first and this result in a lower execution time. The future scope will require a continuous improvement and innovation to address emerging challenges and technologies. One key area of focus will be the development of several sophisticated machine learning algorithms that can accurately predict resource demands and dynamically allocate resources in real-time. This will enable cloud providers to achieve higher levels of efficiency, reduce costs, and improve overall performance.

References

1) Mishra SK, Sahoo B, Parida PP. Load balancing in cloud computing: A big picture. *Journal of King Saud University - Computer and Information Sciences*. 2020;32(2):149–158. Available from: https://doi.org/10.1016/j.jksuci.2018.01.003.
2) Singh A, Kumar R. Performance Evaluation of Load Balancing Algorithms Using Cloud Analyst. *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. 2020;p. 156–162. Available from: https://doi.org/10.1109/Confluence47617.2020.9058017.
3) Kruekaew B, Kimpan W. Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm With Reinforcement Learning. *IEEE Access*. 2022;10:17803–17818. Available from: https://doi.org/10.1109/ACCESS.2022.3149955.
4) Chen Z, Yang L, Huang Y, Chen X, Zheng X, Rong C. PSO-GA-Based Resource Allocation Strategy for Cloud-Based Software Services With Workload-Time Windows. *IEEE Access*. 2020;8:151500–151510. Available from: https://doi.org/10.1109/ACCESS.2020.3017643.
5) Reddy RB, Indiramma M. Efficient Throttled load balancing algorithm to improve the response time and processing time in data center. *Concurrency And Computation-Practice & Experience*. 2022;34(23). Available from: https://doi.org/10.1002/cpe.7208.
6) Malik N, Sardaraz M, Tahir M, Shah B, Ali G, Moreira F. Energy-Efficient Load Balancing Algorithm for Workflow Scheduling in Cloud Data Centers Using Queuing and Thresholds. *Applied Sciences*. 2021;11(13):5849. Available from: https://doi.org/10.3390/app11135849.
7) Sweekriti M, Shetty S, Shetty. Analysis of load balancing in cloud data centers. *Journal of Ambient Intelligence and Humanized Computing*. 2019;p. 1–9. Available from: https://doi.org/10.1007/s12652-018-1106-7.

8) Kothilaxman RR, Lathigara A, Aluvalu R, Viswanadhula UM. PGWO-AVS-RDA: An intelligent optimization and clustering based load balancing model in cloud. 2022. Available from: https://doi.org/10.1002/cpe.7136.
9) Samriya JK, Patel SC, Khurana M, Tiwari PK, Cheikhrouhou OK. Intelligent SLA-Aware VM Allocation and Energy Minimization Approach with EPO Algorithm for Cloud Computing Environment. *Mathematical Problems in Engineering*. 2021;2021:1–13. Available from: https://doi.org/10.1155/2021/9949995.
10) Singh H, Tyagi S, Kumar P, Gill SS, Buyya R. Metaheuristics for scheduling of heterogeneous tasks in cloud computing environments: Analysis, performance evaluation, and future directions. *Simulation Modelling Practice and Theory*. 2021;111:102353. Available from: https://doi.org/10.1016/j.simpat.2021.102353.
11) Ragmani A, Elomri A, Abghour N, Moussaid K, Rida M. An improved Hybrid Fuzzy-Ant Colony Algorithm Applied to Load Balancing in Cloud Computing Environment. *Procedia Computer Science*. 2019;151:519–526. Available from: https://doi.org/10.1016/j.procs.2019.04.070.