*Intelligent Automation & Soft Computing*
DOI:10.32604/iasc.2022.024052
***Article***

Tech Science Press

# Multi-Objective Optimization of Energy Aware Virtual Machine Placement in Cloud Data Center

**B. Gomathi[1], B. Saravana Balaji[2], V. Krishna Kumar[3], Mohamed Abouhawwash[4,5,*], Sultan Aljahdali[6], Mehedi Masud[6] and Nina Kuchuk[7]**

[1]Department of Information Technology, Hindusthan College of Engineering and Technology, Coimbatore, 641032, India
[2]Department of Information Technology, Lebanese French University, Erbil, 44001, Iraq
[3]Department of Computer Science and Engineering, Sri Ramakrishna Engineering College, Coimbatore, 641022, India
[4]Department of Mathematics, Faculty of Science, Mansoura University, Mansoura, 35516, Egypt
[5]Department of Computational Mathematics, Science, and Engineering (CMSE), Michigan State University, East Lansing, MI, 48824, USA
[6]Department of Computer Science, College of Computers and Information Technology, Taif University, Taif, 21944, Saudi Arabia
[7]Department of Electronic Computing, Kharkiv Polytechnic Institute, National Technical University, Kharkiv, 61002, Ukraine
*Corresponding Author: Mohamed Abouhawwash. Email: abouhaww@msu.edu

Received: 01 October 2021; Accepted: 10 December 2021

**Abstract:** Cloud computing enables cloud providers to outsource their Information Technology (IT) services from data centers in a pay-as-you-go model. However, Cloud infrastructure comprises virtualized physical resources that consume huge amount of energy and emits carbon footprints to environment. Hence, there should be focus on optimal assignment of Virtual Machines (VM) to Physical Machines (PM) to ensure the energy efficiency and service level performance. In this paper, The Pareto based Multi-Objective Particle Swarm Optimization with Composite Mutation (PSOCM) technique has been proposed to improve the energy efficiency and minimize the Service Level Agreement (SLA) violation in Cloud Environment. In this paper, idea of MOPSO is extended with three distinct features such as Largest Processing Time (LPT) rule is applied to improve load balancing across the resources which leads to energy saving in Cloud Environment; Epsilon Fuzzy Dominance technique is used to select solutions near to Pareto front which improves the diversity of Pareto optimal solutions; and Discrete PSO along with Composite Mutation strategy in the proposed algorithm help to provide better convergence than existing approaches. Hence, the proposed algorithm produced better results than other existing algorithm such as GA and heuristics-based approach.

**Keywords:** VM placement; cloud computing; SLA violation; energy consumption; particle swarm optimization; multi-objective optimization

## 1 Introduction

Cloud computing enable the cloud providers to outsource their IT services to end users in a pay-per-use model [1]. The cloud infrastructure provides significant profit to IT companies by enabling them to focus on their new innovations to promote their business without any investment in infrastructure and human operators to deploy their services. In addition, Cloud providers deliver their services to end users in a flexible manner based on their dynamic changes in requirement. As of late, high performance in cloud infrastructure is a main concern and it has been achieved without concentrating on energy consumption in cloud environment. However, data center in cloud environment needs to host the cloud application frequently. Consequently, cloud data centers utilize substantial quantity of electrical energy that needs to contribute more on operation costs and emits carbon footprints to the environment.

To provide efficient services to end users in a profitable manner, Cloud service providers need to optimize their resource provisioning method to satisfy SLAs while improving energy efficiency [2,3]. Hence, there should be focus on optimal assignment of Virtual Machine (VM) to Physical Machine (PM) while ensuring the efficiency in energy consumption and service level performance. Thus, the proposed energy conscious VM placement decision is taken by PSOCM (Particle Swarm Optimization with Composite Mutation) algorithm.

The proposed Pareto based Multi-objective PSOCM algorithm minimizes the energy consumption as well as SLA violation as distinct dimensions while providing the dynamic assignment of VMs with multiple resources to PMs in cloud infrastructure. LPT (Longest Processing Time) [4] rule in PSOCM improves the load balancing across the resources which leads to energy saving in cloud environment. Based on LPT Rule, an empty schedule is started and iteratively place a nonscheduled task with longest processing time of all left over tasks on to VM with currently minimal workload. Furthermore, fuzzy dominance method in our proposed algorithm allocates perimeter value to each solution in the whole population and assists to measure the dominance between the solutions. Subsequently, it generates best compromised solution that takes multiple objectives such as energy consumption and Service Level Agreement (SLA) violation, simultaneously. The performance of proposed algorithm is evaluated using CloudSim toolkit and the outcomes show that the proposed algorithm produced enhanced results than existing approaches with regard to improvement in energy efficiency and minimized the SLA violation in cloud environment.

The rest of the article is prepared as follows. “Related works” section provides the interconnected work on Energy aware VM placement algorithms. “Energy Efficient Cloud Computing Model” section describes the problem formulation model for the proposed work. “Energy Efficient VM placement Algorithm using PSOCM” section gives the basic information about “Classical PSO” and “Discrete PSO algorithm” for energy efficient VM placement in Cloud Environment and “Experimental evaluation” section shows the simulation approach and assesses the proposed approach, then comparing it to an existing approach. “Conclusions” section gives the conclusions of the proposed work.

## 2 Related Works

Authors in [5] proposed multi-objective particle swarm optimization based on crowding distance algorithm to allocate service among the consumer in cloud environment while maximizing revenue for users and providers by finding optimal solution at specific time period. It generated pareto optimal solution and specified best compromise solution by using fuzzy set theory. In the Pareto set, all solutions were considered equal. Hence, it was making impossible to determine how one solution overpowered another.

Shun Yao et al. [6] presented multi-objective multi-swarm optimization technique to enhance the outcome of conflicting multiple objective workflow scheduling in data center. Non-dominated scheduling solutions are searched based on infrastructure of data center. Multiple swarms were created for different objectives and particles in different swarms can communicate with each other to provision the information sharing among swarms. But our proposed Pareto-based multi-objective PSOCM algorithm requires few parameters and has an intrinsic better communication system between its particles that helps to provide faster convergence than other optimization techniques like GA (Genetic Algorithm), ACO (Ant Colony Optimization) etc.

Threshold Based Multi-Objective Memetic Optimized Round Robin Scheduling (T-MMORRS) Technique was proposed [7] to improve the task scheduling performance by balancing both bursty and non-bursty workloads in the cloud environment. Burst detector was used to determine the workload conditions of virtual machine as normal or bursty. Based on the condition of workload, T-MMORRS Technique selected Threshold Multi-Objective Memetic Optimization (TMMO) algorithm for normal workload situation and Weighted Multi-Objective Memetic Optimized Round Robin Scheduling (WMMORRS) for burstiness workload state. The proposed Technique was improved the effectiveness of the scheduling and minimized energy usage in cloud environment.

For lowering energy consumption and maintenance costs in a cloud - based environment, a hybrid PSO (Particle Swarm Optimization) and Tabu Search (TS) algorithm [8] was used to address the Cloud Data Replication problem. The global search capability of PSO and local search capability of TS have been coupled to enhance the system's performance. In this regard, the number of data replications and their locations have been identified in order to increase data replication efficiency. However, the traditional method was used to combine many objectives into a specific objective. It must run numerous times, consuming more time each time, and providing a different optimal answer for each run.

In [9–11] they proposed multi-objective hybrid bacterial foraging approach to schedule tasks in cloud environment. It generates Pareto optimal solution to diminish the makespan while lessening the energy utilization in data center. In this approach, crossover and mutation approaches incurred extra timing to reach the optimal solution. In addition, the levels of non-domination between solutions were not measured in this work. Therefore, solutions were not closer to Pareto front. In our proposed approach, perimeter operator and epsilon fuzzy dominance technique are used to select solutions near to Pareto front which improves the diversity of Pareto optimal solutions and Composite mutation strategy in the proposed algorithm helps to provide better convergence than existing approaches.

## 3 Energy Efficient Cloud Computing Model

The cloud environment has collection of data centers that is composed by heterogeneous resources interconnected to provide set of services to end users. The next generation of cloud moves towards Green Cloud infrastructure to support energy efficient service allocation. The service requests arrived in our model is based on Poisson distribution and execution time of each request is assumed to be exponential distribution. The CPU utilization $U_i$ of host machine $H_i$ in the data centre is measured using Eq. (1):

$$U_i = \sum_{j \in \{1..n\}} u_{i,j} \tag{1}$$

where, n is the count of tasks running on Host Hi at present time and ui,j is the utilization of Hi by task tj. The power consumption of host [12] is measured using the following power model in Eq. (2).

$$P_i = K * P_{max} + (1 - K) * P_{max} * U_i \tag{2}$$

where $P_i$ is the power consumption of host i, K is the fraction of power used by idle server, Pmax is the extreme power consumption by host at fully utilized stage.

The utilization of host i is function of time, $u_i(t)$ since it may vary by its running application over the time because of different workload assignment. Hence the total energy consumption E to host i can be represented as an integral of power consumption function over the time period $[t_0, t_1]$ as shown in Eq. (3) below:

$$E = \int_{t_0}^{t_1} P(\mathrm{u_i(t)})dt \tag{3}$$

In the proposed approach, SLA violation [13] is also measured in the data centre using the following two metrics with the help of Eqs. (4) and (5):

- % of time (SLAT) in which host is over utilized (100% utilization of host).
- Performance Degradation due to Migration (PDM) [14].

SLAT is considered when VMs in host cannot achieve performance required by users due to 100% of utilization of hosts, which leads to SLA violation. PDM metric is used to measure performance degradation in VM due to VM migration as shown below.

$$SLAT = \frac{1}{NH}\sum_{i=1}^{NH}\frac{T_{FUi}}{T_{Ai}} \tag{4}$$

$$PDM = \frac{1}{NVM}\sum_{j=1}^{NVM}\frac{C_{PDj}}{C_{Ci}} \tag{5}$$

where NH is the count of host machines; $T_{FU_i}$ is the total time interval in that time SLA violation takes place due to 100% resource utilization of host i; $T_{A_i}$ is the time period during this period host i is in active condition; NVM is the count of Virtual Machines; $C_{PD_j}$ is the measurement of compromised in performance of $\mathrm{VM_j}$ as a consequence of VM migration; $C_{C_i}$ is the required host capacity of $\mathrm{VM_j}$ in its lifespan. Hence total SLA violation is calculated using Eq. (6) as follows:

$$\mathrm{T_{SLA} = SLAT * PDM} \tag{6}$$

Epsilon-fuzzy dominance sort approach [15] is used along with PSOCM to find non-dominated solutions for multi-objective task scheduling problem which improve one objective without reducing another objective. Hence, this proposed Pareto based multi-objective approach helps to minimize both energy consumption and SLA violation in order to satisfy both cloud service provider and consumer while reducing the count of VM migrations and PM shutdowns as shown in Fig. 1.

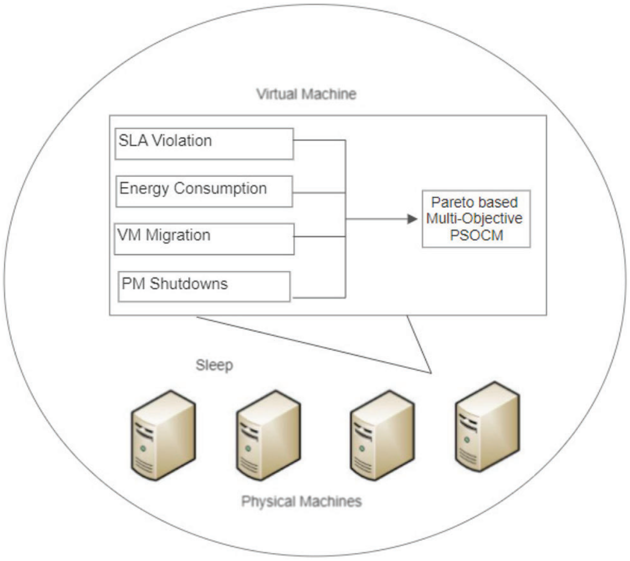


**Figure 1:** Overview of proposed algorithm

## 4 Energy Efficient VM Placement Algorithm Using PSOCM

In this section, the preliminary explanation of classical PSO is described first and then PSOCM algorithm, for energy aware VM allotment in cloud data center is described.

### *4.1 Classical PSO*

PSO [16] is a meta-heuristics algorithm that conducts search in solution space using swarm of individuals. The particle is an individual in the swarm. One among the individual will become leader, i.e., individual with best performance, which leads the motion of the whole swarm. The motion of every particle depends on the leader particle and its individual knowledge. Based on the fitness function, particles are evaluated. Each particle in swarm is rendered with vector in D-dimensional space. The velocity of each particle directs movement of flying particle in search space. Here, $\mathrm{V}_{\mathrm{k}}^{\mathrm{i}}$ and $\mathrm{V}_{\mathrm{k+1}}^{\mathrm{i}}$ are the velocity of particle i at iteration k and k+1 respectively. Particles are flying through hyper dimensional search space by changing its position. All the particles are familiar with its position and value of the fitness function in that particular position. The position of particle depends on its best position $\mathrm{P}_{\mathrm{k}}^{\mathrm{i}}$ and position of the best particle $\mathrm{P}_{\mathrm{k}}^{\mathrm{g}}$ in an entire population where $\mathrm{X}_{\mathrm{k}}^{\mathrm{i}}$ and $\mathrm{X}_{\mathrm{k+1}}^{\mathrm{i}}$ represent the current position of particle i at iteration k and k+1 respectively. In each generation, velocity and position of particle is updated as follows:

$$V_{k+1}^{i} = \omega_k V_k^i + c_1 r_1 \left(P_k^i - X_k^i\right) + c_2 r_2 \left(P_k^g - X_k^i\right) \tag{7}$$

$$X_{k+1}^{i} = X_k^i + V_{k+1}^{i} \tag{8}$$

where, the variables r1 and r2 are random numbers between the range 0 and 1. c1 and c2 are the acceleration co-efficient where c1 helps particle to move towards its success and c2 helps to move particle towards success of entire population. $\mathrm{W}_{\mathrm{k}}$ is the inertia weight that balances between the global and local exploration capabilities of the swarm. The whole population moves towards the near optimal point in the solution space. The classical PSO algorithm can only be applied for continuous optimization functions. In order to solve VM-to-PM allocation problem in cloud environment, classical PSO algorithm needs to modify as discrete PSO which introduces the analogous structure to the classical PSO equations to solve Virtual Machine to Physical Machine allocation problem in cloud environment.

### *4.2 Discrete PSO for Energy Efficient VM Placement*

This section gives the detail about the discrete PSO [17] and its adaption to solve VM-to-PM allocation problem in data center. One of the critical problems in PSO algorithm is to design solution representation where the particles have to carry the essential information to solve the problem in particular domain. To make the particles to support VM placement, solution for assigning VM to PM is represented by the collection of length 'L' where 'L' is the count of VMs. The value of the array in d^th^ dimension renders PM number to which d^th^ VM is allocated. Fig. 2 shows the particle representation for VM-to-PM assignment with 5 VMs and 2 PMs.

| $\mathrm{VM}_1$ | $\mathrm{VM}_2$ | $\mathrm{VM}_3$ | $\mathrm{VM}_4$ | $\mathrm{VM}_5$ |
|---|---|---|---|---|
| 1 | 2 | 2 | 1 | 1 |

**Figure 2:** Solution representation VM-to-PM allocation problems

The initial position of particles in the swarm is generated by the permutations of the current VM-to-PM assignment solution which is also part of the initial swarm. The velocity of particles is initialized randomly. The particles are evaluated based on the objectives such as minimization of energy utilization and violation of the SLA in data centers, defined in Eqs. (1)–(6). The initial Pareto optimal solutions are selected based on the fuzzy dominance and permutation methods [18] and stored in the external Pareto archive set.

- Updating Position and Velocity of particles

To upgrade the position of particles in the swarm, following Eqs. (9) and (10) are used:

$$V_p^{i+1} = V_p^i \overset{\times}{O} \left( \left( C_1 \overset{*}{O} \left( P_p^i \overset{-}{O} X_p^i \right) \right) \overset{\times}{O} \left( C_2 \overset{*}{O} \left( P_g^i \overset{-}{O} X_p^i \right) \right) \right) \tag{9}$$

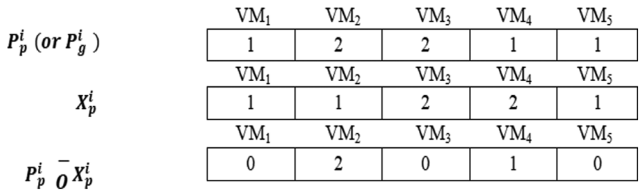
$$X_p^{i+1} = X_p^i \overset{\times}{O} V_p^{i+1} \tag{10}$$

where $C_1$ and $C_2$ are the one-dimensional random arrays, whose length is n. The values of these random arrays are either 0 or 1 and the probability of getting 1 as a value is 0.3. $V_p^i$ and $X_p^i$ are the velocity and position of particle p at iteration i, respectively. $P_p^i$ is the personal best solution of particle p at iteration i and $P_g^i$ is the global best position for the whole swarm at iteration i. The operators used in the Eqs. (9) and (10) are defined as follows:

- Difference Operator $(\bar{O})$

This operator is used to check whether there is any difference in value between current position ($\mathbf{X_p^i}$) of particle i and global ($\mathbf{P_g^i}$) or personal best position ($\mathbf{P_p^i}$) of particle. If yes, corresponding element in the result will be the value of $\mathbf{P_p^i}$(**or** $\mathbf{P_g^i}$). Otherwise, elements in result are filled by PM number using LPT rule which helps to assign heavy loaded VMs onto the PM which has minimal load.

Fig. 3 illustrates the working principle of $\bar{O}$ operator. Here, PM assignment for VMs 1, 3 and 5 is based on LPT rule. PM numbers generated by LPT rule are filled in resultant array. If $\mathbf{P_p^i}$(**or** $\mathbf{P_g^i}$) and $\mathbf{X_p^i}$ are equal, then corresponding term in the velocity equation is omitted.

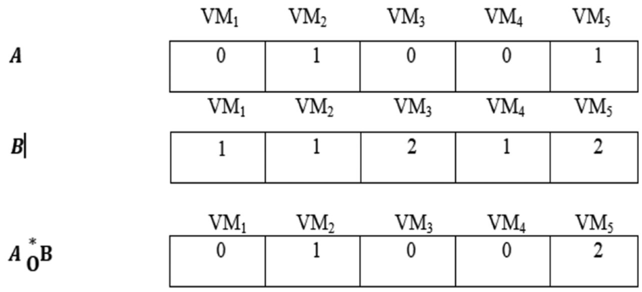


| | $VM_1$ | $VM_2$ | $VM_3$ | $VM_4$ | $VM_5$ |
|---|---|---|---|---|---|
| $P_p^i$ (or $P_g^i$ ) | 1 | 2 | 2 | 1 | 1 |
| $X_p^i$ | 1 | 1 | 2 | 2 | 1 |
| $P_p^i \overset{-}{O} X_p^i$ | 0 | 2 | 0 | 1 | 0 |

**Figure 3:** Working principle of $\bar{O}$ operator

- Multiplication Operator $(\overset{*}{O})$:

Multiplication operator is used to perform multiplication process between array A and B. Fig. 4 illustrates the working principle of $\overset{*}{O}$ operator. Here, PM assignment for VMs 1, 3 and 4 is performed based on LPT rule and PM numbers found through LPT rule are filled in the resultant array.

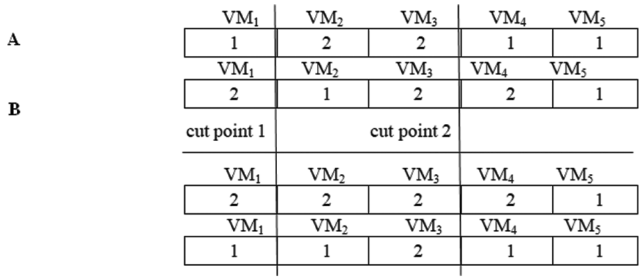


| | $VM_1$ | $VM_2$ | $VM_3$ | $VM_4$ | $VM_5$ |
|---|---|---|---|---|---|
| $A$ | 0 | 1 | 0 | 0 | 1 |
| $B$ | 1 | 1 | 2 | 1 | 2 |
| $A \overset{*}{O} B$ | 0 | 1 | 0 | 0 | 2 |

**Figure 4:** Working principle of $\overset{*}{O}$ operator

- Crossover operator ($\overset{\times}{O}$):

In this crossover operator, two cut points are selected from the particle randomly and exchange the values between particles A and B whose values are left side of first cut point and right side of second cut point as shown in Fig. 5. This operator produces two temporary particles, and one will be selected as the result of crossover operator randomly. To enhance the solution of discrete PSO, composite mutation strategy [19] is applied on the result of discrete PSO. Then, the particles are evaluated and combined the solutions from current iteration and solution from archive of previous generation. From these 2*NS solutions where NS is the dimension of the swarm, best NS solutions are selected to update the archive set using fuzzy dominance and perimeter methods [20]. The above-mentioned processes are repeated until maximum iteration is reached. Based on the optimal mapping, VMs are migrated, and then unused PMs are turned off to minimize the energy utilization while less violation in SLA.



| | $VM_1$ | $VM_2$ | $VM_3$ | $VM_4$ | $VM_5$ |
|---|---|---|---|---|---|
| A | 1 | 2 | 2 | 1 | 1 |
| B | 2 | 1 | 2 | 2 | 1 |

| $VM_1$ | $VM_2$ | $VM_3$ | $VM_4$ | $VM_5$ |
|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 1 |
| 1 | 1 | 2 | 1 | 1 |

**Figure 5:** Working principle of $\overset{\times}{O}$ operator

The algorithm for Discrete PSO for VM-to-PM mapping is listed below.

Step 1: Initialize size of the population as NS and dimension of particle as number of VMs and initialize empty schedule for LPT rule.

Step 2: Set the position and velocity of particles in swarm by random permutation of VM-to-PM assignment.

Step 3: Apply LPT rule on VM-to-PM assignment and evaluate fitness of each particles using Eqs. (1)–(6).

Step 4: The Pareto optimal solutions are selected based on fuzzy dominance and permutation methods and stored in external achieve set.

Step 5: Upgrade the velocity and position of each particle using Eqs. (9) and (10).

Step 6: Composite mutation strategy is applied to enhance solution produced by discrete PSO.

Step 7: Particles are evaluated and combined solutions from current iteration along with solution from previous generation (totally 2*NS solution). The best NS solutions are selected using on fuzzy dominance approach.

Step 8: If maximum iteration is not reached, then go to step 3. Otherwise, optimal VM-to-PM mapping is produced.

## 5 Experimental Evaluation

The CloudSim [21] simulation toolkit is utilised to estimate the performance of the proposed algorithm. One data center is simulated in CloudSim with dual type of PMs such as *HP Proliant ML 110 G4* (Intel Xeon 3040, 2 cores × 1860 MHz, 4096 MB RAM and 1 Gbps BW) and *HP ProLiant ML110 G5* (Intel Xeon 3075, 2 cores × 2660 MHz, 4096 MB RAM and 1 Gbps BW). The real power consumption characteristics are listed in Tab. 1 and used to calculate power consumption using SPEC power benchmark [9].

**Table 1:** Power consumption for HP ProLiant G4 and G5 provided by SPEC power benchmark

| Server | 0% | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|---|---|---|
| HP ProLiant G4 | 86 | 89.4 | 92.6 | 96 | 99.5 | 102 | 106 | 108 | 112 | 114 | 117 |
| HP ProLiant G5 | 93.797 | 97 | 101 | 105 | 110 | 116 | 121 | 125 | 129 | 133 | 135 |

The VM types used in this simulation environment correlate to Amazon EC2 instance type. The features of each VM instance types with single core are extra-large (2500 MIPS, 870 MB RAM), large (2000 MIPS, 3750 MB RAM), medium (1000 MIPS, 1740 MB RAM) and small (500 MIPS, 613 MB RAM). The utilization details of different workloads run by VMs are generated randomly for every 5 m. The following are the parameters used in meta-heuristics algorithm:

- GA: crossover ratio = 0.8 and mutation probability = 0.2
- GA and PSOCM: Size of the Population = 20 and Number of Iterations = 100

The number of PMs used in this experiment ranges from 50 to 150 and number of VMs ranges from 50 to 200. Three different test scenarios are used to evaluate the proposed approach with two different kinds of load in data centers, light load data center and heavy load data center. Using these specifications, the experiments are conducted using proposed approach to minimize the energy utilization and violation of SLA in data centers while reducing VM migrations and PM shutdown. The performance of the proposed approach is analysed with other existing algorithm such as GA [22–25] and heuristics-based approach that uses LRR (Robust Local Regression) technique for the detection of overloaded host and MMT (Minimum Migration Time) policy for VM selection.

In this paper, idea of MOPSO is extended with three distinct features in the proposed algorithm such as Largest Processing Time (LPT) rule is applied to improve load balancing across the resources which leads to energy saving in Cloud Environment; Epsilon Fuzzy Dominance technique is used to select solutions near to Pareto front which improves the diversity of Pareto optimal solutions; and Discrete PSO along with Composite Mutation strategy in the proposed algorithm help to provide better convergence than existing approaches. Hence, the proposed algorithm produced better results than other existing algorithm such as GA and heuristics-based approach.

### *5.1 Data Center with Light Load*

The following three different scenarios have been created in simulation environment to determine the accomplishment of the proposed approach in lightly loaded data centers.

1. Scenario 1: 50 PMs are available for 50 VMs
2. Scenario 2: 100 PMs are available for 100 VMs
3. Scenario 3: 150 PMs are available for 150 VMs

Figs. 5–7 present the Pareto-based optimal solutions composed by PSOCM, GA and heuristics approaches using above mentioned three different scenarios. The performance of the proposed approach is compared with GA and heuristics approach with the help of best compromised solutions using energy consumption and SLA violation metrics while assigning VMs to suitable PMs. Fig. 6 presents the proposed PSOCM approach which produced better results by reducing the energy consumption 6% lesser than GA and 8% lesser than heuristics approach. At the same time, the proposed approach reduced SLA violation 13% lesser than GA and 58% lesser than heuristics approach.

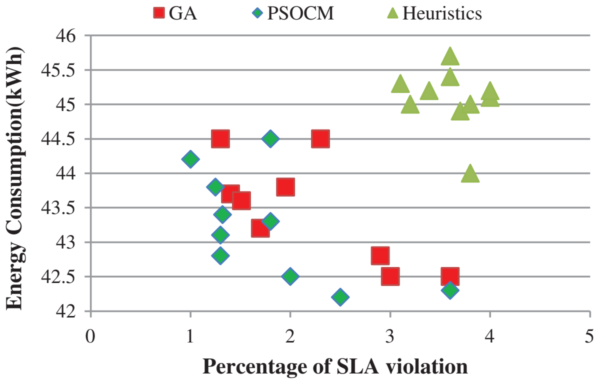


**Figure 6:** Pareto front of energy consumption and percentage of SLA violation for 50 VMs*50 PMs

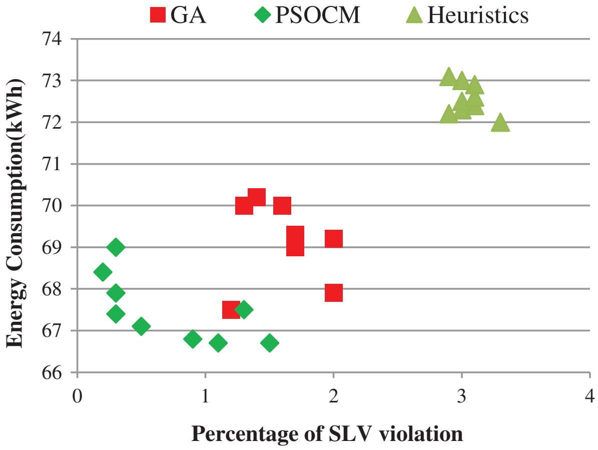


**Figure 7:** Pareto front of energy consumption and percentage of SLA violation for 100 VMs*100 PMs

Fig. 7 shows the outcome of the second scenario. The suggested PSOCM approach minimized the energy consumption by 8% than GA and 9% than heuristics approach. At the same time, PSOCM reduced the SLA violation by 8% lesser than GA and 55% lesser than heuristics approach. Fig. 8 exhibits the result of the last scenario. Here, PSOCM is 3% more energy efficient than both GA and heuristics approach. Concurrently, PSOCM reduced the SLA violation by 48% lesser than GA and 64% lesser than heuristics approach.

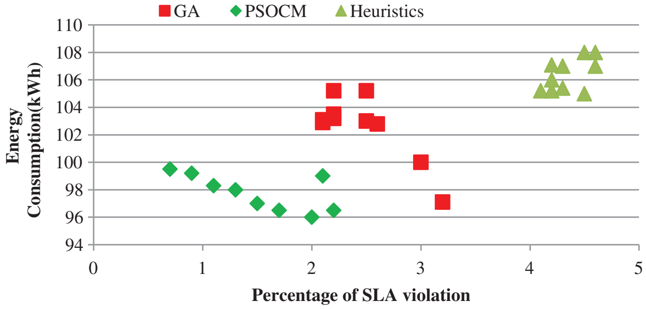


**Figure 8:** Pareto front of energy consumption and percentage of SLA violation for 150 VMs*150 PMs

In the entire lightly loaded scenarios, the proposed approach gave better result than GA and heuristics approach with regard to energy saving and SLA violation as shown in the summarized Figs. 9 and 10.

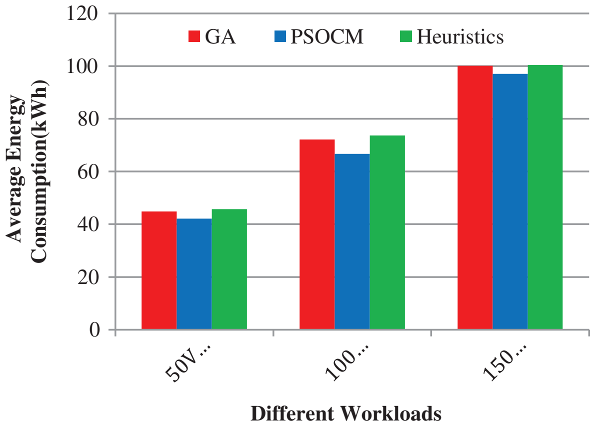


**Figure 9:** Average energy consumption of different workloads

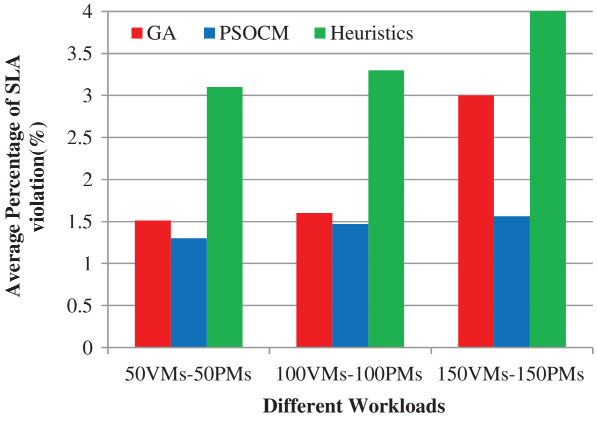


**Figure 10:** Average SLA violations of different workloads

### *5.2 Data Center with Heavy Load*

To check the performance of the proposed approach in heavy loaded data center, the following three different scenarios have been created.

1. Scenario 1: 100 PMs are available for 100 VMs
2. Scenario 2: 100 PMs are available for 150 VMs
3. Scenario 3: 100 PMs are available for 200 VMs

The first scenario in heavy loaded type was the same as the second scenario in lightly loaded center which was already explained. Fig. 11 shows the Pareto front for energy consumption and SLA violation for the second scenario. In heavy loaded type, the PSOCM approach reduced energy consumption by 4% lesser than GA and 8% lesser than heuristic. Simultaneously, it reduced SLA violation by 35% lesser than GA and 63% lesser than heuristic approach.

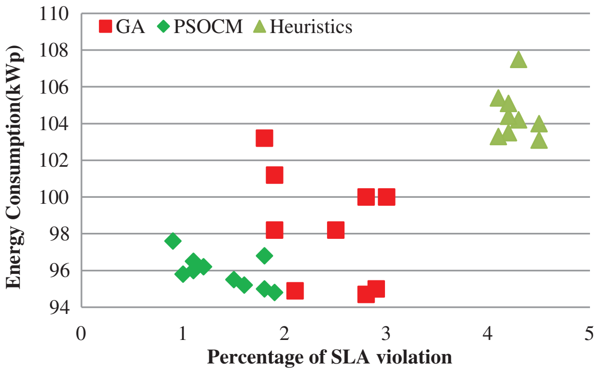


**Figure 11:** Pareto front of energy consumption and percentage of SLA violation for 150 VMs*100 PMs

In the same way in Fig. 12, PSOCM approach reduced the energy consumption by 3% lesser than GA and 7% lesser than heuristics approach. At the same time, proposed approach reduced SLA violation by 36% lesser than GA and 60% lesser than heuristics approach. Figs. 13–16 showed that the proposed approach produced better results than both GA and heuristics approach in heavy loaded data center scenario. It implicitly says that PSOCM algorithm is more compatible with regard to SLA violation and energy consumption in heavy loaded data centers. But slight degradations in performances were happened in heuristics-based approach and GA due to PM shut down and VM migration.

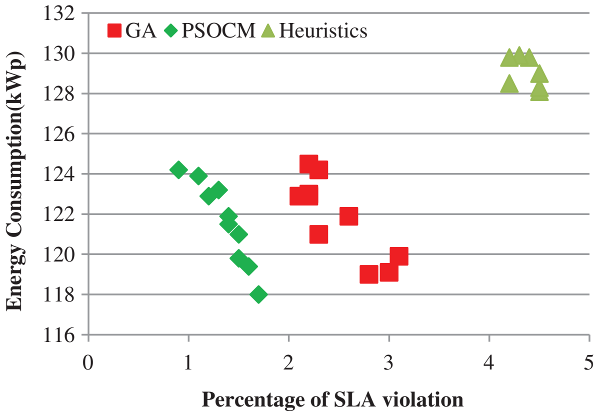


**Figure 12:** Pareto front of energy consumption and percentage of SLA violation for 200 VMs*100 PMs

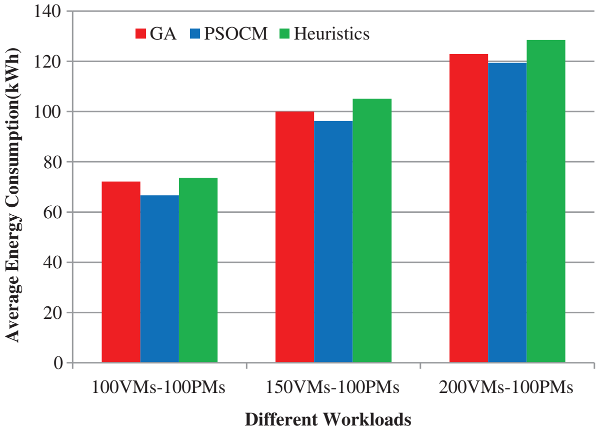


**Figure 13:** Average energy consumption of different workloads

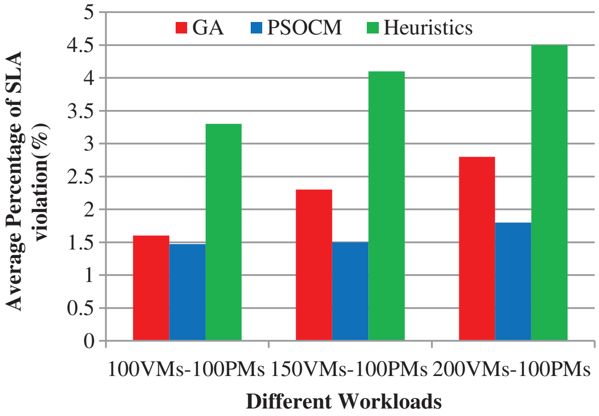


**Figure 14:** Average SLA violations of different workloads

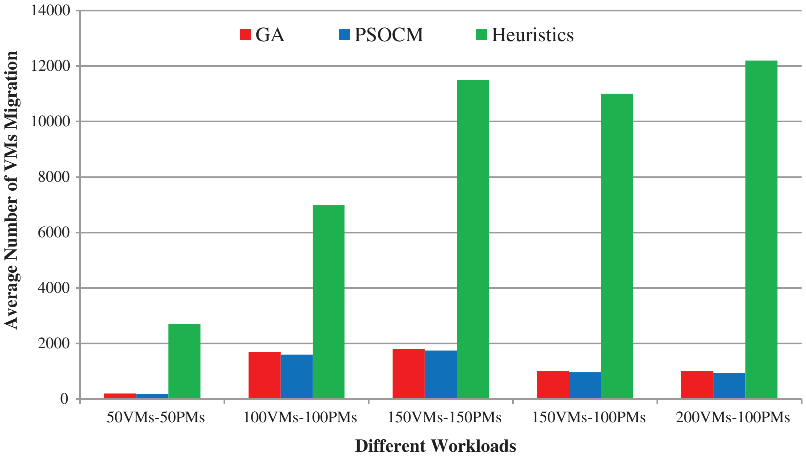


**Figure 15:** Average number of VM migrations

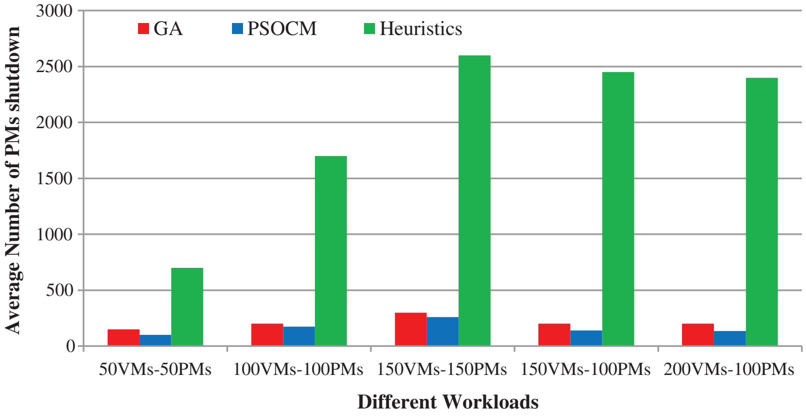


**Figure 16:** Average number of PM shutdowns

Moreover, the number of VM migrations in different workloads using proposed approach was reduced from 3% to 6% lesser than GA as well as from 77% to 92% lesser than heuristics approach. At the same time, the number of PM shutdowns in different workloads using proposed approach was reduced from 13% to 33% lesser than GA as well as from 85% to 94% lesser than heuristics approach.

In Figs. 15 and 16, VM migration and PM shutdown take place in heuristics-based approach due to overloading in PM or under loading in PM. Before VM migration and PM shutdown take place, heuristics approach does not check adaption for new VM which leads to over adaption in destination system. However, the proposed approach and GA algorithms allowed changing in solution only when there is improvement in the current VM-to-PM allocation. At the same time, proposed approach produced better results than GA as a consequence of large number of non-dominated solutions in archive set. Hence, it assisted to select best solution from the Pareto archive set.

The comparison results of the PSOCM, GA and heuristics approaches are showed in Tab. 2. This table concludes that proposed PSOCM approach produced better results in energy utilization, violation of SLA, number of VM migration and number of PM shutdowns than GA and heuristic approach in different types of loads. Results concluded that the proposed PSOCM approach reduced energy utilization and SLA violations more effectively than GA and heuristics-based approaches both in heavily and lightly loaded data center settings. Moreover, ε-fuzzy dominance sorting technique in the proposed approach helped to produce better and faster convergence than GA by selecting solutions closer to Pareto front.

**Table 2:** Best compromised solutions for different work loads

| Objectives | Algorithms | 50 * 50 | 100 * 100 | 150 * 150 | 150 * 100 | 200 * 100 |
|---|---|---|---|---|---|---|
| Energy consumption | GA | 44.8 | 72.7 | 100.1 | 100 | 122.9 |
| | Heuristics | 45.7 | 73.7 | 100.4 | 105.1 | 128.5 |
| | PSOCM | 42.1 | 66.7 | 97 | 96.2 | 119.4 |
| SLA violation | GA | 1.51 | 1.6 | 3 | 2.3 | 2.8 |
| | Heuristics | 3.1 | 3.3 | 4.3 | 4.1 | 4.5 |
| | PSOCM | 1.3 | 1.47 | 1.56 | 1.5 | 1.8 |
| No. of VM migration | GA | 200 | 1700 | 1800 | 1000 | 1000 |
| | Heuristics | 2700 | 7000 | 11500 | 11000 | 12200 |
| | PSOCM | 190 | 1600 | 1750 | 960 | 940 |
| No. of PM shutdowns | GA | 150 | 200 | 300 | 200 | 200 |
| | Heuristics | 700 | 1700 | 2600 | 2450 | 2400 |
| | PSOCM | 100 | 175 | 260 | 140 | 135 |

Apart from that, the proposed approach enhances the searching ability in solution space to improve the optimal solution using perimeter assignment operator. In addition, composite mutation strategies and fast local search algorithm [23] in the proposed approach were used to enhance the Pareto solution further produced by proposed approach. In contrast, VM migration and PM shutdown take place in heuristics-based approach due to overloading in PM or under loading in PM. Before VM migration and PM shutdown take place, heuristics approach does not check adaption for new VM which leads to over adaption in destination system. However, the proposed approach and GA algorithms allowed changing in solution only when there is improvement in the current VM-to-PM allocation.

## 6 Conclusion

In this paper, PSOCM approach was proposed to dynamically allocate VMs-to-PMs depends on the present utilization of resources in cloud environment. This Pareto-based VM placement approach helped to reduce the energy utilization and violation of SLA in data centers. The performance of the proposed approach was evaluated using CloudSim toolkit and compared the efficiency of the proposed approach with GA and heuristics approaches. The simulation results showed that proposed approach gave better results than GA and heuristics approaches regarding energy consumption and SLA violation in both heavily loaded and lightly loaded data centers. Moreover, PSOCM and GA approaches accept the changes only if there is improvement in the current VM-to-PM assignment, but heuristics approach made changes in the assignment whenever there is overloaded or under loaded problem without considering the improvement. The proposed approach provides better VM placement in Cloud Environment, but Pareto approach takes time to provide optimal result.

**Acknowledgement:** We would like to give special thanks to Taif University Research supporting project number (TURSP-2020/73), Taif University, Taif, Saudi Arabia.

**Funding Statement:** This work was supported by Taif University Researchers Supporting Project (TURSP) under Number (TURSP-2020/73), Taif University, Taif, Saudi Arabia.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud computing and emerging IT platforms: Vision hype and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[2] F. Moges and S. Lemma Abebe, "Energy-aware VM placement algorithms for the openstack neat consolidation framework," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 8, no. 1, pp. 1–14, 2019.

[3] M. H. Sayadnavard, A. T. Haghighat and A. M. Rahmani, "A reliable energy-aware approach for dynamic virtual machine consolidation in cloud data centers," *The Journal of Supercomputing*, vol. 75, no. 4, pp. 2126–2147, 2019.

[4] R. L. Graham, "Bounds on multiprocessing timing anomalies," *SIAM Journal on Applied Mathematics*, vol. 17, no. 2, pp. 416–429, 1969.

[5] F. Sheikholeslami and N. J. Navimipour, "Service allocation in the cloud environments using multi-objective particle swarm optimization algorithm based on crowding distance," *Swarm and Evolutionary Computation*, vol. 35, pp. 53–64, 2017.

[6] G. Shun Yao, Y. Sheng Ding and K. Rong Hao, "Multi objective workflow scheduling in cloud system based on cooperative multi-swarm optimization algorithm," *Journal of Central South University*, vol. 24, no. 5, pp. 1050–1062, 2017.

[7] J. Prassanna and N. Venkataraman, "Threshold based multi objective memetic optimized round robin scheduling for resource efficient load balancing in cloud," *Mobile Networks and Applications*, vol. 24, no. 4, pp. 1214–1225, 2019.

[8] A. S. Abohamama and E. Hamouda, "A hybrid energy aware virtual machine placement algorithm for cloud environments," *Expert Systems with Applications*, vol. 150, no. 4, pp. 1–21, 2020.

[9] S. Srichandan, T. Ashok Kumar and S. Bibhudatta, "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 210–230, 2018.

[10] M. Abdel Basset, R. Mohamed, M. Abouhawwash, R. K. Chakrabortty and M. J. Ryan, "EA-MSCA: An effective energy-aware multi-objective modified sine-cosine algorithm for real-time task scheduling in multiprocessor systems: Methods and analysis," *Expert Systems with Applications*, vol. 173, no. 3, pp. 114699, 2021.

[11] M. Abdel Basset, D. E. Shahat, K. Deb and M. Abouhawwash, “Energy-aware whale optimization algorithm for real-time task scheduling in multiprocessor systems,” *Applied Soft Computing*, vol. 93, no. 2, pp. 106349, 2020.

[12] A. Beloglazov, J. Abawajy and R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing,” *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

[13] A. Beloglazov and R. Buyya, “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers,” *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

[14] K. Zhu, H. Song, L. Liu, J. Gao and G. Cheng, “Hybrid genetic algorithm for cloud computing applications,” in *Proc. 2011 IEEE Asia-Pacific Services Computing Conf. (APSCC)*, Fuzhou, China, pp. 182–187, 2011.

[15] B. Gomathi, K. Krishnasamy and B. Saravana Balaji, “Epsilon fuzzy dominance sort-based composite discrete artificial bee colony optimisation for multi-objective cloud task scheduling problem,” *International Journal of Business Intelligence and Data Mining*, vol. 13, no. 3, pp. 247–266, 2018.

[16] R. Kennedy and J. Eberhart, “Particle swarm optimization,” in *Proc. IEEE Int. Conf. on Neural Networks*, Glasgow, UK, pp. 1942–1948, 1995.

[17] A. H. Kashan and B. Karimi, “A discrete particle swarm optimization algorithm for scheduling parallel machines,” *Computers and Industrial Engineering*, vol. 56, no. 1, pp. 216–223, 2009.

[18] P. Koduru, Z. Dong, S. Das, J. L. Roe and E. Charbit, “A multiobjective evolutionary-simplex hybrid approach for the optimization of differential equation models of gene networks,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 5, pp. 572–590, 2008.

[19] E. Nowicki and C. Smutnicki, “A fast tabu search algorithm for the permutation flow-shop problem,” *European Journal of Operational Research*, vol. 91, no. 1, pp. 160–175, 1996.

[20] C. A. Coello Coello, G. T. Pulido and M. S. Lechuga, “Handling multiple objectives with particle swarm optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

[21] R. N. Calheiros, R. Ranjan, C. A. F. De Rose and R. Buyya, “CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services,” *Technical Report, GRIDS-TR-2009-1*, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, 2009. arXiv:0903.2525v1 [cs.DC].

[22] A. Mosa and N. W. Paton, “Optimizing virtual machine placement for energy and SLA in clouds using utility functions,” *Journal of Cloud Computing*, vol. 5, no. 1, pp. 1–17, 2016.

[23] X. Li and M. Yin, “A hybrid cuckoo search via Lévy flights for the permutation flow shop scheduling problem,” *International Journal of Production Research*, vol. 51, no. 16, pp. 4732–4754, 2013.

[24] M. Masud, G. S. Gaba, K. Choudhary, M. Hossain, M. F. Alhamid *et al.,* “Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare,” *IEEE Internet of Things Journal*, vol. 2, no. 4, pp. 1–14, 2021.

[25] S. Ibrahim, H. Alhumyani, M. Masud, S. S. Alshamrani, O. Cheikhrouhou *et al.,* “Framework for efficient medical image encryption using dynamic S-boxes and chaotic maps,” *IEEE Access*, vol. 8, no. 13, pp. 160433–160449, 2020.