Multi-Objective Reinforcement Learning for Virtual Machines Placement in Cloud Computing

Chayan Bhatt, Sunita Singhal
Department of Computer Science and Engineering, Manipal University Jaipur, Jaipur, Rajasthan, India

***Abstract*—The rapid demand for cloud services has provoked cloud providers to efficiently resolve the problem of Virtual Machines Placement in the cloud. This paper presents a VM Placement using Reinforcement Learning that aims to provide optimal resource and energy management for cloud data centers. Reinforcement Learning provides better decision-making as it solves the complexity of VM Placement problem caused due to tradeoff among the objectives and hence is useful for mapping requested VM on the minimum number of Physical Machines. An enhanced Tournament-based selection strategy along with Roulette Wheel sampling has been applied to ensure that the optimization goes through balanced exploration and exploitation, thereby giving better solution quality. Two heuristics have been used for the ordering of VM, considering the impact of CPU and memory utilizations over the VM placement. Moreover, the concept of the Pareto approximate set has been considered to ensure that both objectives are prioritized according to the perspective of the users. The proposed technique has been implemented on MATLAB 2020b. Simulation analysis showed that the VMRL performed preferably well and has shown improvement of 17%, 20% and 18% in terms of energy consumption, resource utilization and fragmentation respectively in comparison to other multi-objective algorithms.**

***Keywords*—Virtual machines placement; cloud computing; reinforcement learning; energy consumption; resource utilization**

I. INTRODUCTION

In the world of digitalization, Cloud Computing has become one of the popular platforms to render an immense pool of services to its customers [1]. Cloud providers such as Amazon EC2, Google app engine, Azure, etc. provide users with different applications and platforms to run their businesses efficiently and promote Quality of Service (QoS) simultaneously [2].

Virtualization forms an essential technology of the cloud, based on which it creates Virtual Machines (VM) over the active servers to perform tasks as requested by different clients [3, 4]. The number of cloud clients has been rapidly growing to avail the wide range of day-to-day cloud services due to which there has been a drastic need for more VM to fulfill the demands of its customers. This caused the activation of more Physical Machines (PM) and spiked up the power consumption of the cloud data center. As a result, it has become a challenge for cloud providers as it causes additional operational costs that hinder the overall progress and profit of the system.

Proper placement of VM plays an important role in curbing the effect of power consumption. If VM are allocated strategically and resources of PM are optimally used, it leads to turning up less PM and minimizes overall energy consumption and carbon emissions simultaneously [5].

An efficient VM placement approach is a powerful tool for maintaining cost reduction, performance upgradation and consistent reliability of cloud data centers. However, designing an effective VM placement solution is not trivial due to large-scale cloud data centers, PM heterogeneity and use of multidimensional resources [6].

Hence, to solve the problem of VM Placement, many heuristics and meta-heuristics algorithms have been proposed to build an optimal solution for VM Placement. Differential Evolution (DE) [7], Simulated Annealing (SA) [8], Genetic Algorithms (GA) [9], Ant Colony Optimization (ACO) [10] and Particle Swarm Optimization (PSO) [11] have been continuously used to design and develop efficient solutions for placing VM on the cloud. Many hybrid algorithms have also been designed to bring an effective version placement considering different objectives.

Researchers are now focusing on multi-objective problems where different parameters such as energy usage, system performance, operational cost, completion time and QoS standards are being considered simultaneously to examine the effectiveness of VM placement solutions.

In [12], a multi-objective Mayfly Strategy has been designed for large-scale cloud data centers. It involves the collection of five dependent objective functions and converting them into minimum matrix reduction with the help of principal component analysis. This matrix serves as input to the Mayfly optimization metaheuristic and finds the optimal solution for VM placement. The comparative analysis showed that the above approach has a faster and higher convergence rate. It minimizes the power consumption, resource wastage, traffic and Service Level Agreement (SLA) violation of different configurations but involves greater computation time and cost.

Furthermore, a framework involving Deep Reinforcement Learning (RL) to solve VM placement, has been proposed by Luca et al in [13]. It works on three main objectives- minimization of software/hardware outages, co-location interference and power consumption. Six different VM placement heuristics have been used. Three of them were novel and the rest of the three were the enhancement of existing algorithms. However, the proposal required to take traffic and SLA violations into account. Similarly, Yao et al. [14] used Chebyshev scalarization in multi-objective RL to solve the problem of VM placement. The concept of the Pareto set has been used to lower energy consumption and resource wastage simultaneously. It solves the weight selection problem. However, it required enhancement in scalability and completion time.

In study [15], a multi-objective RL algorithm has been designed to minimize power consumption and SLA violations. In the proposed technique, a single objective function was used by summing up the two objectives without considering the weights. As a result, only one solution was obtained at a time by the above algorithm.

It has been observed that energy consumption has been one of the main factors that has affected the progress of the cloud as it not only increases the operational costs of the data centers but also causes high carbon emissions in the environment [16]. Similarly, many other factors such as SLA violations, resource wastage, QoS, network traffic, etc. have shown notable impact on the performance of cloud data centers and hence, became a prime focus for many researchers to work on and come up with reliable and efficient solutions [17].

Thus, to deal with the significant issue of energy consumption, this paper presents a VM Placement that minimizes the number of active servers and promotes optimal resource utilization.

The main contributions of this paper include:

- An RL-based VM placement has been proposed that deals with the minimization of the number of active servers to host the requested VM and maintain optimal resource utilization.
- An enhanced selection policy has been applied that selects actions of choosing an appropriate PM for a respective VM. It ensures that the algorithm goes through proper exploration and exploitation.
- Two heuristics have been used for the ordering of VM. The sequence of the VM in which it is processed has been manipulated to bring a desirable and efficient VM solution.
- Pareto approximate set has been used to obtain solutions according to the objectives and have weightage as per the perspective of the users.

The rest of the paper is organized as follows: Section II demonstrates the system model and problem formulation of VM placement. It also covers a detailed description of VMRL. Section III depicts the simulation analysis of the proposed work and its comparison with other existing techniques. The conclusion of the paper has been presented in Section IV.

II. Methodology

A. Reinforcement Learning

Reinforcement learning is a decision-making approach that explores the environment and takes suitable action to gain maximum reward for a specific situation. It is a self-teaching process that uses a trial-and-error method to discover the best solution for a non-deterministic problem. It has been implemented in many automated systems that perform a lot of small decisions without human guidance. It consists of an agent that perceives the unknown environment and performs actions to reach its goal. The agent starts from the initial state and by applying actions, it moves towards its final goal by traversing through different states [18]. Based on applied action, a reward is given to the agent for that state. The reward expresses the goodness of the state and is stored in the form of a Q value along with the next state, in a Q-table. The Q - table gets updated iteratively. The Q-values of the current state can be updated using Eq. (1).

$$Q(s,a) = Q(s,a) + \alpha * (r + \gamma * Max\, Q(s',a') - Q(s,a) \quad (1)$$

Q (s, a) denotes action-value estimates of the current state, Q (s', a') denotes action-value estimates of the next state, r refers to the achieved reward, alpha is the learning rate and gamma is the discount factor. After traversing all the states, the path that accommodates maximum reward is the optimal solution.

B. Pareto Approximate Set

Most of the researchers have followed the concept of Pareto dominance [19] to solve the problem of multi-objective. By using the Pareto set, multi-objective algorithms solve large-scale complex problems and rather than giving a single best solution, it provides a set of same quality solutions based on different objectives.

C. Energy Consumption

Energy consumption has become a critical issue for cloud providers as they need to invest the maximum in curbing its overall effect on cloud data centers. It has been observed that PM consume the maximum amount of power and causes high carbon emissions. Past studies have shown that energy consumption has a linear relationship with the CPU utilization of PM [20]. Moreover, it has been proven that idle PM are primarily responsible for wasting energy and are required to be shut down when not in use. The energy model has been formulated as per Eq. (2).

$$Ei = (Emax - Eidle) * UiCPU + Eidle \quad (2)$$

Emax denotes energy consumption when PM is fully utilized and Eidle denotes energy consumption of an idle PM. UiCPU represents the CPU utilization of the particular PM. In the proposed work, the energy consumption of fully loaded PM is fixed at 185 w and for idle PM, it is 120 w.

D. Resource Utilization

Proper Resource utilization brings a positive outcome for a better VM placement approach. Optimal use of resources leads to less resource wastage and activation of servers. If the resources are not used wisely, it may cause resource fragmentation and degradation in system performance [21]. In this paper, we have considered only two resources: CPU and memory. Resource wastage of PM can be evaluated using the following Eq. (3).

$$Rwastage = \frac{|Lcpu - Lram| + e}{Ucpu + Uram} \quad (3)$$

Lcpu and Lram denote normalized residual resources of CPU and memory. Ucpu and Uram denote normalized utilization of CPU and memory.

E. Problem Statement

Cloud computing consists of different configured data centers that hold numerous PM, having different attributes and configurations [22]. It involves memory, processor, and network bandwidth. To perform the execution of tasks requested by the customers, VM are created and deployed on these PM under specific constraints as mentioned below [23].

$$i \sum mi\ (Vid)\ <=\ Pid \quad (4)$$

$$i\sum m(Vm) <\ Pm\ \ (5)\ i \sum mi\ (Vcpu)\ <=\ Pcpu \quad (6)$$

$$i \sum mi\ (Vbw)\ <=\ Pbw \quad (7)$$

Constraint in Eq. (4) denotes that each VM should be deployed on only a single PM. Constraints in Eq. (5), (6) and (7) verify that the required resources of a VM should not exceed the resource capacities of the PM. Like PM, VM also have specific configurations and attributes, concerning memory, bandwidth, and processing power.

1) VM Ordering: The proposed approach selects the best method to map the requested VM to the available PM. The order in which VM are processed has been changed to achieve better solutions for VM placement problems. We have considered two simple heuristics for the deployment of VM.

a) Heuristic I-VMRL I: In this approach, the dot product of requested resources of VM and the sum of utilizations of PM are evaluated using Eq. (8) and based on results, VM are arranged in non-ascending order. RL is applied to the new ordering of VM.

$$(VMcpu\ .Nj\ PMcpu) +\ (VMmem.Nj\ PMmem) \quad (8)$$

Here, VMcpu and VMmem are the requested resources of the VM. $\sum_{N_j=1} PMcpu$ is the sum of CPU utilization and $\sum_{N_j=1} PMmem$ is the sum of memory utilization of all available PM.

b) Heuristic II- VMRL II: This approach obtains the difference between the resource utilization of all PM and the resources required by the current VM as mentioned in Eq. (9) based on the results, VM are sorted in non-descending order.

$$(VMcpu - Nj\ PMcpu)2\ +\ (VMmem - Nj\ PMmem)2 \quad (9)$$

2) VM placement based on Reinforcement Learning (VMRL): A VM placement based on Reinforcement Learning (VMRL) has been presented in this paper. RL provides better decision-making and hence is adopted to design an effective VM Placement solution [24]. A learning agent is established to perceive the resource requirements and capacities of the VM and PM and take suitable actions to accomplish the goal of reducing the number of active PM by properly conserving the resources.

The agent works in the state space S_t = {s1, s2……….sn} where n is equal to the number of requested VM. St represents the set of normalized resource utilization in two – dimensions i.e., for CPU and memory utilization. A graphical representation of VMRL is depicted in Fig. 1.

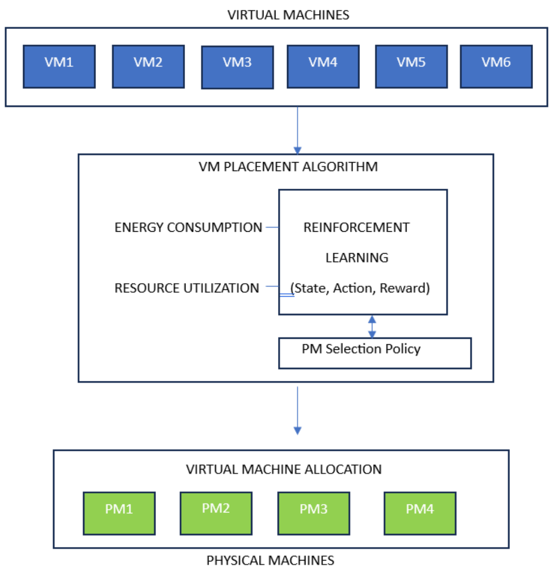


Fig. 1. Graphical representation of working of VMRL algorithm.

a) Selection: The agent performs actions to select a particular PM for the corresponding VM and ensures that the PM satisfies all the constraints defined in the VM policy [25]. The action space denotes all the PM that can accommodate a particular VM at time step t. The period between two iterations is considered as time step t. The selection of action is to be performed in two phases. In phase I, selection probabilities are assigned to the favorable PM by applying an enhanced Tournament-based selection method. All the feasible PM are divided into sub-groups and are provided with respective selection probabilities based on their Q-value by using Eq. (10).

$$Pi = \frac{2(i-1)}{K(K-1)}\ q\ +\ \frac{2(k-i)}{K(K-1)}(1-q);\ \text{i} \in \{1,2,3\ldots.\ \text{K}\} \quad (10)$$

Here Pi is the selection probability of a favorable PM that can host a specific VM and K is the population size. The PM with a higher Q-value is assigned a higher selection probability whereas the PM with a low Q-value is assigned a smaller selection probability.

b) Sampling: In Phase II, a sampling algorithm called the Roulette Wheel is applied to generate the fittest PM. In this approach, each possible action is appointed a portion on roulette according to selection probabilities assigned in phase I and the roulette wheel is spun K times to select suitable PM successively. This procedure helps to maintain a balance between exploration and exploitation processes and ensures that past experiences lead to fast convergence toward optimal VM placement solutions.

c) Rewards: Rewards are awarded as per the action performed by the agent in every state and are stored in the form of Q- value in the Q- table. They are calculated based on the two objectives. Let r = {r1, r2} where r1 denotes the reward for the first objective and r2 denotes the reward for the second objective as defined in Eq. (10) and Eq. (11). Better rewards are given for the actions that give favorable outcomes as per the defined objectives.

$$r1 = \frac{E}{E' t+1+ \eta} \tag{11}$$

Et depicts the total energy consumed by all PM at time step t, and E t+1 depicts the total energy consumed by all PM after the current VM gets allocated to PM.

$$r2 = \frac{R}{R' t+1+ \eta} \tag{12}$$

Rt depicts the resources wasted by all PM at time step t, and Rt+1 depicts the resource wasted by all PM after the current VM gets allocated to PM. By traversing from one state to another and by collecting the rewards, the agent achieves its goal in the form of an optimal VM solution. The path that obtains maximum cumulative rewards is the best VM solution.

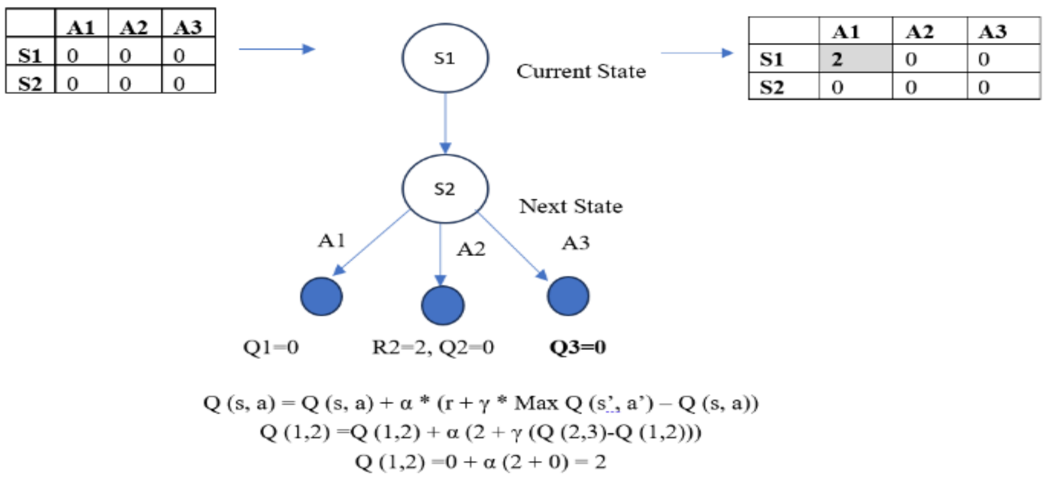
The algorithm begins with the initialization of parameters and Q-table. Let us assume that there are three PM as {PM1, PM2, PM3} and six VM as {VM1, VM2, VM3, VM4, VM5, VM6}. Hence, there will be a Q-table of 3x6 columns as depicted in Fig. 2. Initially, all the values of the Q-table are initialized with zero.

Now, the VM will be arranged as per the heuristic I and II respectively. After the VM ordering, for state S1, the VM will be selected from the VM list and the PM that can accommodate the current VM will be searched. Action will be performed using an enhanced Tournament based selection policy. The respective reward will be calculated using Eq. (11) and Eq. (12). After the evaluation of the reward, the Q-table will be updated based on the reward and next state using Eq. (1) as depicted in Fig. 3.

Similarly, all the states will be accessed and their Q-values will be updated. After several iterations, the Q-table will get populated with the values. Let us assume that the Q-table has been updated with the values as depicted in Fig. 4.

	A1	**A2**	**A3**
S1	0	0	0
S2	0	0	0
S3	0	0	0
S4	0	0	0
S5	0	0	0
S6	0	0	0

Fig. 2. Q-Table for storing Q values of states for different actions.

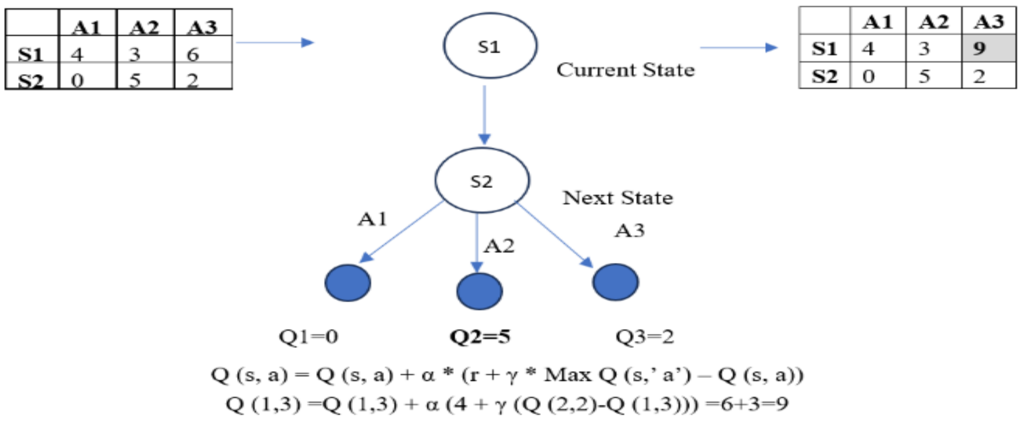


	A1	**A2**	**A3**
S1	0	0	0
S2	0	0	0

	A1	**A2**	**A3**
S1	**2**	0	0
S2	0	0	0

Q (s, a) = Q (s, a) + α * (r + γ * Max Q (s', a') – Q (s, a))
Q (1,2) =Q (1,2) + α (2 + γ (Q (2,3)-Q (1,2)))
Q (1,2) =0 + α (2 + 0) = 2

Fig. 3. Updated Q-Table.



	A1	**A2**	**A3**
S1	4	3	6
S2	0	5	2

	A1	**A2**	**A3**
S1	4	3	**9**
S2	0	5	2

Q (s, a) = Q (s, a) + α * (r + γ * Max Q (s,' a') – Q (s, a))
Q (1,3) =Q (1,3) + α (4 + γ (Q (2,2)-Q (1,3))) =6+3=9

Fig. 4. Updated Q-Table after few iterations.

In this time step, exploitation is performed and the state-action pair having the highest q-value is selected. This procedure continues to execute till the algorithm reaches maximum iteration and provides optimal VM placement solution.

The flowchart and the pseudocode for the proposed algorithm have been shown below in Fig. 5 and Algorithm 1.

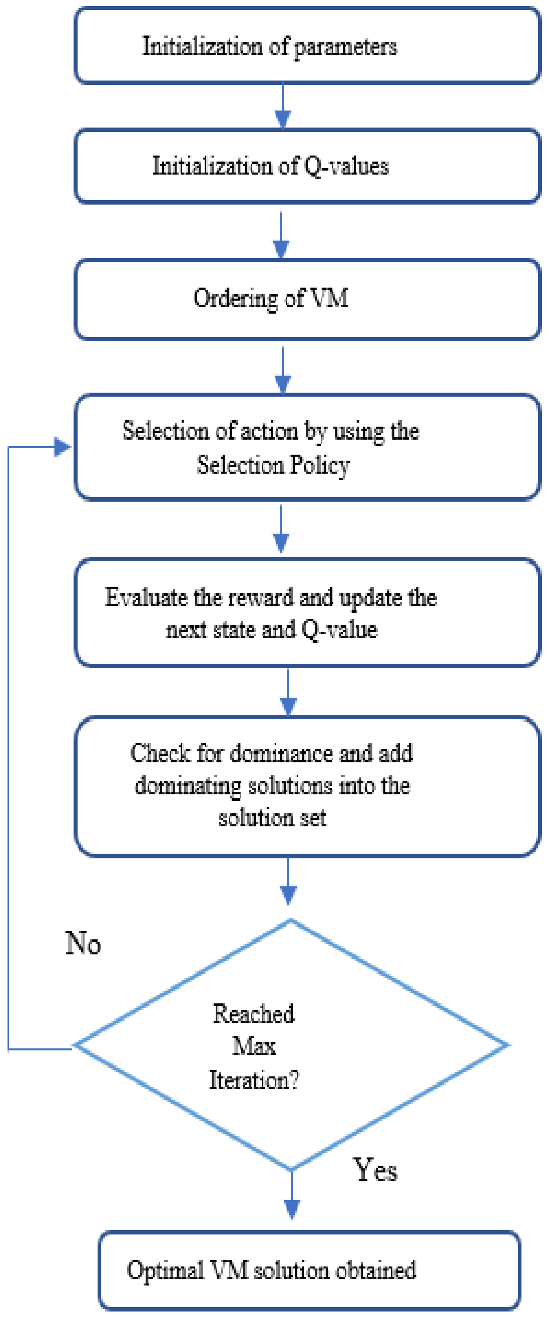

Initialization of parameters
Initialization of Q-values
Ordering of VM
Selection of action by using the Selection Policy
Evaluate the reward and update the next state and Q-value
Check for dominance and add dominating solutions into the solution set
No
Reached Max Iteration?
Yes
Optimal VM solution obtained

Fig. 5. Flowchart of proposed framework: VMRL.

Algorithm I: VM Placement using Reinforcement Learning (VMRL)

```
1.  Create a set S and initialize it to null.
2.  Create a Q-table and initialize all its values with zero
3.  For i = 1 to M (no. of VM)
4.  Initialize the current state.
5.  Generate the ordering of VM. Select a VM from the VM set V
6.  Select the action based on the selection policy. The action provides the mapping of the selected VM to an available PM that satisfies all the constraints.
7.  Generate the reward for the action.
8.  Update the q-value and the next state in the Q-table.
9.  Check for the Pareto dominance. If the solution is not dominated by the solutions in set S then add it to set S otherwise discard the solution.
10. Discard all the solutions inside set S that are dominated by the current solution.
11. Repeat steps 3 to 8 till maximum iteration. Return the optimal solution in set S.
```

III. RESULTS AND DISCUSSION

The performance metrics and experiment setup have been described to evaluate the verification and validation of VMRL. The experiments have been performed on MATLAB 2020b. The hardware configuration used for implementing the proposed work has a 3.2GHz CPU, 1 TB HDD and 8GB RAM. To find the effectiveness of VMRL, our proposed framework has been compared with three different multi-objective algorithms i.e., MOPSO (Multi-Objective Particle Swarm Optimization), MOACO (Multi-Objective Ant Colony Optimization) and VMPORL (VM Placement based on multi-objective RL) respectively, in term of resource utilization, number of active PM, energy consumption and resource fragmentation. The specifications of the servers used are mentioned in Table I.

TABLE I. SPECIFICATIONS OF PM

Server Type	MIPS	Storage (GB)	Memory (GB)
HP ProLiantG4	1860	4	1000
HP ProliantG5	2660	4	1000

A. Quality Indicators

Quality indicators are required for multi-objective algorithms to examine the quality of Pareto approximate set [26]. For the proposed VMRL, three quality indicators have been used: Overall Non-dominated Vector Generation (ONVG), Chi-Square and Hypervolume. The comparison results with other three algorithms in terms of these quality indicators have been provided in Table II. It is seen that the weight selection holds a significant role in multi-objective optimization and its quite challenging to find out an appropriate weight which will provide good Pareto Front. For proposed VMRL, 10 randomly generated weight tuples have been used and an average of 10 trials has been considered. It is observed that as per statistical analysis of all three indicator values, VMRL has performed well as compared to other multi-objective algorithms.

TABLE II. QUALITY INDICATORS FOR ALGORITHMS

Algorithm	ONVG	Chi-square	Hypervolume
MOPSO	23.54	22.87	165.59
MOACO	25.37	23.54	174.34
VMPORL	33.01	32.42	315.67
VMRL(I)	39.23	40.58	345.23
VMRL(II)	37.45	41.22	355.17

B. *Comparison with other Algorithms*

The comparison among the algorithms has been evaluated on the dataset, having VM into a batch of 200, 400, 600 and 800. The dataset is the real-time workload trace GWA-T-12 Bitbrains [27] that stores performance metrics of VM from distributed data centers of Bitbrains.

1) Scenario I: Fig. 6 depicts the experimental results of VMRL by using heuristic – I. It is observed that the performance metrics are strongly affected by the data center workload. The dot product provides control over the trade-off between the power consumed and Quality of Experience (QoE). Fig. 6(a) shows the performance of VMRL for activating the number of servers to deal with the requested VM. It is seen that the VMRL outperforms the other multi-objective algorithms i.e., MOACO, MOPSO and VMPORL. This indicates that VMRL can host a greater number of VM on lesser PM as compared to other algorithms thereby contributing to energy and resource savings. This can be seen in Fig. 6 (b), Fig. 6 (c) and Fig. 6(d) where VMRL has shown a considerable improvement of 17 % in energy consumption, 20% in resource utilization and 18% in resource fragmentation respectively in comparison to other approaches.

2) Scenario 2: Fig. 7 depicts the simulation analysis of VMRL by using heuristic II. Fig. 7(a) depicts that the VMRL can give optimal solutions by hosting different batches of VM on lesser PM. In Fig. 7(b), it is seen that the proposed technique was successful in bringing down the overall energy consumption without having outcomes of past experiences. Fig. 7(c) and Fig. 7(d) represent the resource utilization and fragmentation of VMRL in comparison to other algorithms. It is seen that VMRL gives the best results in all instances, proving it to be efficient and robust. VMPORL also came up with good outcomes on the performance metrics. The enhanced selection policy used for the selection of appropriate PM has brought a desirable change in the simulation results and upgraded the performance of VMRL in comparison to other existing techniques.

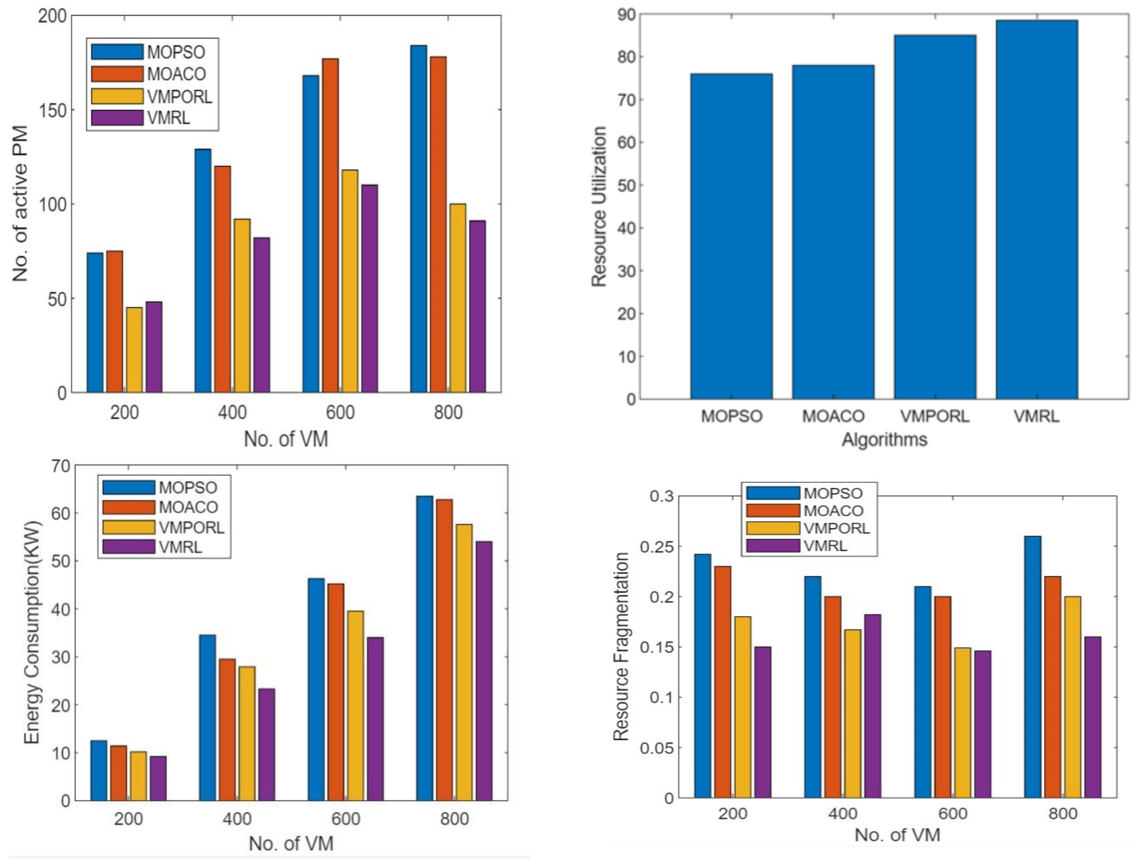


Fig. 6. Comparisons of algorithms using Heuristic I in terms of (a) Number of active servers (b) Resource Utilization of PM (c) Energy Consumption (d) Resource Fragmentation of PM.

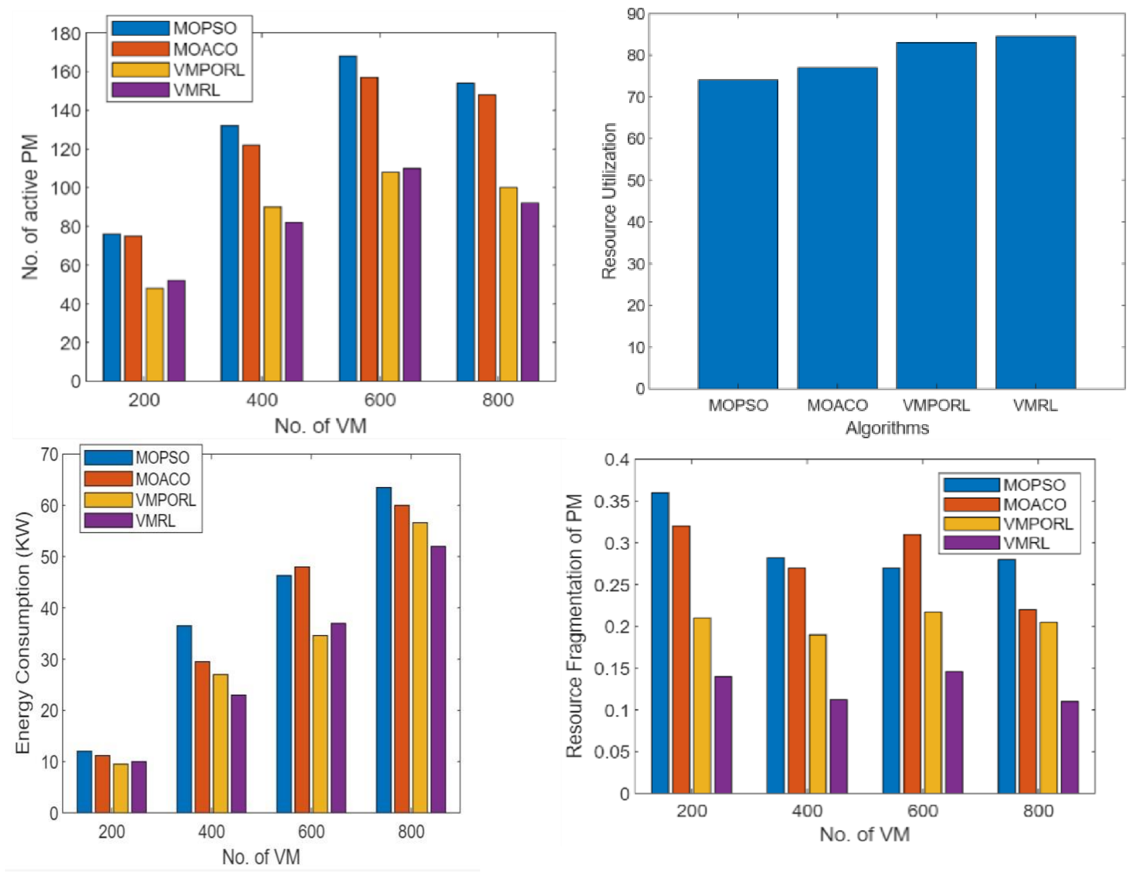


Fig. 7. Comparisons of algorithms using Heuristic II in terms of (a) Number of active servers (b) Resource Utilization of PM (c) Energy Consumption (d) Resource Fragmentation of PM.

IV. Conclusion

VM placement holds a significant role in cloud computing. An effective placement of VM on an appropriate PM can lead to minimum resource wastage and energy consumption [28]. In this paper, a multi-objective VM Placement approach has been presented that works on the principle of Reinforcement Learning. An enhanced selection strategy has been applied to select the actions suitable for mapping VM with PM. Two resource-managing heuristics have been used for ordering VM, considering the target objectives. Pareto approximate set has been built to provide optimal VM solution. The proposed technique has been implemented on MATLAB 2020b. It has been compared with other multi-objective algorithms. The simulation analysis showed that the VMRL has performed considerably well in comparison to other existing algorithms. In future studies, VM migrations and cost minimization could be taken into account to deal with the wider perspective of cloud platform.

References

[1] Huanlai Xing, Jing Zhu, Rong Qu, Penglin Dai, Shouxi Luo, Muhammad Azhar Iqbal, “An ACO for energy-efficient and traffic-aware virtual machine placement in cloud computing,” *Swarm and Evolutionary Computation*, 101012, ISSN 22106502, 2022.https://doi.org/10.1016/j.swevo.2021.101012.

[2] Gharehpasha, S., Masdari, M. & Jafarian, A., “Virtual machine placement in cloud data centers using a hybrid multi-verse optimization algorithm”, *Artificial Intelligent Rev* 54,2221–2257, 2021. https://doi.org/10.1007/s10462-020-09903-9.

[3] Farzaneh, S. M., & Fatemi, O., “A novel virtual machine placement algorithm using RF element in cloud infrastructure”, *The Journal of Supercomputing*, Vol 78, pp.1288-1329, 2021.doi:10.1007/s11227021-03863-9.

[4] Azizi, S., Zandsalimi, M. & Li, D., “An energy-efficient algorithm for virtual machine placement optimization in cloud data centers”, *Cluster Computing* 23, 3421–3434, 2020.https://doi.org/10.1007/s10586-020-03096-0.

[5] Li, Z., Lin, K., Cheng, S., “Energy-Efficient and Load-Aware VM Placement in Cloud Data Centers”, *Journal of Grid Computing* 20, 2022.https://doi.org/10.1007/s10723022-09631-0.

[6] S. Omer, S. Azizi, M. Shojafar, and R. Tafazolli, “A priority, power and traffic-aware virtual machine placement of IoT applications in cloud data centers”, Journal of Systems Architecture, vol. 115, Article ID 101996, 2021.https://doi.org/10.1016/j.sysarc.2021.101996.

[7] Dubey, K., Nasr, A.A., Sharma, S.C., El-Bahnasawy, N., Attiya, G., El-Sayed, A, “Efficient VM Placement Policy for Data Centre in Cloud Environment”, *Soft Computing: Theories and Applications*, pp. 301–309, 2020.https://doi.org/10.1007/978-981-15-07519_28.

[8] K. Karmakar, R. K. Das, and S. Khatua, “An ACO-based multi-objective optimization for cooperating VM placement in the cloud data center", *Journal of Supercomputing*, vol. 78, no.3, pp. 3093-3121, 2022.https://doi.org/10.1007/s11227-021-03978-z.

[9] Nawaf Alharbe, Abeer Aljohani and Mohamed Ali Rakrouki, “A Fuzzy Grouping Genetic Algorithm for Solving a Real-World Virtual Machine Placement Problem in a Healthcare-Cloud”, *Algorithms*, pp. 1-17, 2022.https://doi.org/10.3390/a15040128.

[10] Brad Everman, Maxim Gao, Ziliang Zong, “Evaluating and reducing cloud waste and cost—A data-driven case study from Azure workloads”, *Sustainable Computing: Informatics and Systems*, 100708, ISSN 2210-5379, 2022.https://doi.org/10.1016/j.suscom.2022.100708.

[11] Manoj Kumar and Suman, “Meta-Heuristics Techniques in Cloud Computing: Applications and Challenges”, *Indian Journal of Computer Science and Engineering (IJCSE)*, vol. 12, 2021. https://doi.org/10.21817/indjcse/2021/v12i2/211202055.

[12] Durairaj, S., Sridhar, R., “MOM-VMP: multi-objective mayfly optimization algorithm for VM placement supported by principal component analysis (PCA) in cloud data center”, *Cluster Computing*,2023.https://doi.org/10.1007/s10586-023-04040-8.

[13] Caviglione, L., Gaggero, M., Paolucci, M., “Deep reinforcement learning for multi-objective placement of virtual machines in cloud datacenters”, *Soft Computing* 25, pp. 12569–12588,2021. https://doi.org/10.1007/s00500-020-05462-x.

[14] Qin, Y., Wang, H., Yi, S., “Virtual machine placement based on multi-objective reinforcement learning”, *Appl Intell* 50, pp. 2370–2383,2020. https://doi.org/10.1007/s10489020-01633-3.

[15] S. Long, Z. Li, Y. Xing, S. Tian, D. Li and R. Yu, “A Reinforcement Learning-Based Virtual Machine Placement Strategy in Cloud Data Centers”, *In* IEEE *6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Yanuca Island, Cuvu, Fiji,* 223230, 2020. doi: 10.1109/HPCC-SmartCity-DSS50907.2020.00028.

[16] Alhammadi A.S.A., Vasanthi V., “Multi-Objective Algorithms for Virtual Machine Selection and Placement in Cloud Data Center”, In Proceedings of the 2021 International Conference of Advanced Technology and Engineering 2021. https://doi.org/10.1109/ICOTEN52080.2021.9493496.

[17] Sayyidshahab Nabavi, Linfeng Wen, Sukhpal Singh Gill, Minxian Xu, “Seagull optimization algorithm based multi-objective VM placement in edge-cloud data centers”, *Internet of Things and Cyber-Physical Systems*, Volume 3, pp. 28-36, ISSN 2667-3452, 2023.https://doi.org/10.1016/j.iotcps.2023.01.002.

[18] Hummaida, A.R., Paton, N.W. & Sakellariou, R., “Scalable Virtual Machine Migration using Reinforcement Learning.”, *Journal of Grid Computing* 20, Vol 15, pp.1-16, 2022. https://doi.org/10.1007/s10723-022-09603-4.

[19] Alahmad, Y., Agarwal, A., “Multiple objectives dynamic VM placement for application service availability in cloud networks.”, *Journal of Cloud Computing* 13, pp.1-20, 2024. https://doi.org/10.1186/s13677-024-00610-2.

[20] Salami, Hamza Onoruoiza, Abubakar Bala, Sadiq M. Sait, and Idris Ismail. "An energy-efficient cuckoo search algorithm for virtual machine placement in cloud computing data centers." *The Journal of Supercomputing* 77, pp.13330-13357,2021. https://doi.org/10.1007/s11227-021-03807-3.

[21] Bhatt, C., Singhal, S. “Anatomy of Virtual Machine Placement Techniques in Cloud” In: Sharma, D.K., Peng, SL., Sharma, R., Zaitsev, D.A. (eds) Micro-Electronics and Telecommunication Engineering. ICMETE 2021. Lecture Notes in Networks and Systems, vol 373. Springer, Singapore,2022. https://doi.org/10.1007/978-981-16-8721-1_59.

[22] Ghasemi, A., Toroghi Haghighat, A., “A multi-objective load balancing algorithm for virtual machine placement in cloud data centers based on machine learning”, *Computing* 102, pp. 2049–2072, 2020. https://doi.org/10.1007/s00607-020-00813-w. 20.

[23] Mejahed, Sara, and M. Elshrkawey. "A multi-objective algorithm for virtual machine placement in cloud environments using a hybrid of particle swarm optimization and flower pollination optimization." *PeerJ Computer Science* 8, 2022: e834.

[24] Eswaran, S., Dominic, D., Natarajan, J. and Honnavalli, P.B. “Augmented intelligent water drops optimization model for virtual machine placement in cloud environment.” *IET Network 9*: pp. 215-222,2020. https://doi.org/10.1049/iet-net.2019.0165.

[25] Sun, Wei, Yan Wang, and Shiyong Li. "An optimal resource allocation scheme for virtual machine placement of deploying enterprise applications into the cloud." *AIMS Mathematics* 5, no. 4, pp. 3966-3989,2020. doi: 10.3934/math.2020256.

[26] Azizi, Sadoon, Maz’har Zandsalimi, and Dawei Li. "An energy-efficient algorithm for virtual machine placement optimization in cloud data centers." *Cluster Computing* 23, pp. 3421-3434,2020. https://doi.org/10.1007/s10586-020-03096-0.

[27] Grid Workload Archive -T-12 Bitbrains. http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains..

[28] Sandeep Kumar Bothra, Sunita Singhal, and Hemlata Goyal, “Deadline-Constrained Cost-Effective Load-Balanced Improved Genetic Algorithm for Workflow Scheduling”, Int. J. Inf. Technol. Web Eng. 16, pp. 1–34,2021. https://doi.org/10.4018/IJITWE.2021100101.