International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
***Article Received:** 25 November 2023 **Revised:** 12 December 2023 **Accepted:** 30 January 2024*

Improved Environmental Adaptation Method for Scheduling Workflows in Cloud

Ravi Prakash*
CSED, ITER, Siksha O Anusandhan University
raviprakash@soa.ac.in

Ranvijay
CSED, MNNIT Allahabad
ranvijay@mnnit.ac.in

Abstract—Cloud users are expanding at rapid rate which forces the cloud data centres execute billions of commands each second. A random user request must be planned and processed on the workflow without knowing the sequence of future requests. This makes workflow scheduling on distributed environment as NP-hard problem. In this work we present an optimization-based scheduling approach that responds to cloud's dynamic nature. The suggested technique derives from the Environmental Adaptation Method (EAM), an evolutionary algorithm established to handle optimization problems. After EAM's original proposal, multiple better versions were made to fix inherent issues. Most of the revised algorithms performed well in lower dimensions but degraded performance is seen in higher ones. Most of these methods were binary encoded, which poses issues for real-valued parameters owing to conversion cost. Improved Environmental Adaptation Method with Real parameters (IEAM-R) was presented to deal with real valued problems to increase IEAM's convergence rate. IEAM-R performs effectively on lower-dimensional benchmark functions, but not on larger dimensions. We changed IEAM-R and created a new algorithm to increase the diversity of solutions in higher dimensions. Exploration and exploitation must be redesigned to im-prove convergence rate. On all 24 benchmark functions, the proposed modified optimization algorithm with fine-tuned operators, outperforms its predecessors and other state-of-the-art algorithms. The technique is then used to the workflow scheduling issue in cloud computing, where it reduces the overall cost of cloud operation as compared to other heuristic and metaheuristic approaches.

Keywords- Optimization; Workflow Scheduling; Cloud Computing;

I. Introduction

The goal of cloud computing is to make it possible for users in different locations to access a variety of computer resources and services through the internet on an as-needed, pay-per-use basis [1] [2]. In reality, cloud service providers are vying for customers' business by reducing prices to attract more users [3]. Cloud service providers need to lower cloud-operational costs without compromising quality of service if they want to survive the competing market.

Recalling "how the cost is being calculated, how the services are being given, and how the massive demands from the cloud customers are being managed" will be necessary if we are to achieve our goal of reducing computational cost. A user either makes a specific machine request to the cloud service provider or chooses from a pool of preconfigured virtual machines (VMs) [4]. In the former scenario, the cloud service provider spins up a new Virtual Machine (VM) based on the user's requirements and then hands over the reins [5]. As the number of virtual machines (VMs) increases, it becomes more difficult to find an optimal allocation of resources using simple algorithms like FCFS (First Come First Serve), Round Robin Scheduling, Shortest Remaining Time, SJF (Shortest Job First), etc.

Scheduling objectives, such as minimizing total execution time, minimizing total cost, and balancing load among cloud resources, must be taken into account in order to schedule tasks efficiently and affordably [6] [7]. In this article, we aim to find the optimal schedule for a company's operations, one that would reduce operational expenses without sacrificing productivity [8] [9].

A. *The evolution of the proposed algorithm:*

The EAM algorithm takes into account a species' adaptability to its changing environment. According to James Mark Baldwin [10], a population's adaptability emerges through a cycle of recurrent learning. EAM outperformed the evolutionary strategies of crossover and mutation because it allowed for a more rapid adaptation of the population's fitness to its new surroundings.

The EAM method offers three operators for its algorithm. The first modifies the solution's observable properties in response to a fitness comparison between a particle and its surroundings. The Alteration operator takes into account subtle changes in the environment and incorporates their impact into the present population, and the Selection operator chooses the optimum option after Adaptation and Alteration operators have been applied. Since the utilization of the search space was low in EAM due to the alterations operator, IEAM was developed to enhance search space exploitation by selecting the particle with the highest fitness. This procedure is called exploration. The remaining members of the present generation are exploited afterwards. Exploiting in this way is most effective for unimodal issues, whereas population exploration aids in breaking out of the grip of local optimum solutions in multimodal situations. Neither of the two algorithms have offered a balanced approach between exploration and exploitation. The best solution can only be reached in a finite number of iterations if both operators are in perfect sync with one another throughout. Furthermore, IEAM regarded each particle the same except for the best one. However, if the situation were handled differently, such as giving greater weight to particles with correspondingly higher fitness values in the following generation of the population, a better solution may emerge. For a middle ground between exploration and exploitation, the Improved Environmental Adaptation Method with Real Parameters (IEAM-R) was presented. Unlike its forerunners EAM and IEAM, which used a binary encoding approach, it was designed to operate with real-valued parameters. The additional work involved in converting from real to binary and vice versa was avoided by using real parameters rather than binary encoding. The pace of convergence of IEAM-R sped up as a result. Additionally, it conducts exploration and exploitation in the same section of the population, eliminating the rest of the population, and picks a range of the population whose fitness value is discovered to be greater than a threshold value. Our methodology originates from the desire to repurpose the eliminated population. We rethought its operators to strike a better balance between exploration and exploitation.

B. Resolution of complexity of implementation to any cloud simulator:

Workflows are used to describe collections of tasks that are scheduled to run on a group of virtual machines (VMs) [11]. It is common practice in many branches of engineering to create and use workflows. Scientific workflows are defined as those developed for and utilized with cloud/grid computing [12]. First, computing resources are chosen during resource provisioning; then, in the second step, each workflow job is mapped onto the best-suited resource in accordance with the established schedule [13]. Workflow scheduling is one area where optimization techniques are being used, and they are proving effective in reducing costs. Most standard optimization techniques have previously undergone extensive testing in simulators, with each iteration demonstrating considerable improvements in performance over their predecessors. Optimizing algorithms can usually be made more effective if additional work is put into refining the algorithm [6]. The proposed technique makes use of some of the benefits of optimization algorithms as well [14]. The scheduling issue is NP-Hard, however it may be mapped by thinking of the cost of a task's execution on a group of VMs as a particle [15]. A population is a collection of all such particles. Our method uses two operators—adaptation and selection—to handle a population that is produced at random. When applied to a population, both of these operators generate a new population. The particles with the highest fitness levels are selected, while the others are abandoned, according to a predetermined criterion. This procedure is repeated until the best possible outcome is achieved.

II. BACKGROUND AND RELATED WORK

Scheduling workflows in a decentralized setting has received a lot of attention in the academic community [16] [17]. The creation of an optimum solution is of primary importance since this is an NP-hard issue and cannot be solved in polynomial time. Over the last decade, several optimization methods have been suggested for use in cloud/grid computing [18] [19]. Since cloud service providers, delivers a limited set of virtualized resources on demand, prop-erly scheduling these resources for user activity or application has become highly important, and numerous intelligent evolutionary computing methods are rapidly evolving to meet this need [20] [21].

Thus, several algorithms, such as classic task scheduling, adaptive dynamic scheduling, real-time scheduling, multi-objective scheduling citeduan2014multi, distributed and parallel scheduling, and others, are studied and encouraged as possible future solutions [22]. The goal of optimal scheduling for activities or resources in the cloud is to maximize profit for consumers and service providers [23]. SaaS, PaaS, and IaaS [24] make up the cloud computing service paradigm. As a result, these various service models provide a useful framework for classifying cloud-scheduling issues.

- "Scheduling in application layer (SaaS)"
- "Scheduling in virtualization layer (PaaS)"
- "Scheduling in infrastructure layer (IaaS)"

In scheduling at any level, problems might arise not only due to the cloud customers' limited resources and spending limits, but also due to the service providers' desire to maximize resource use. Consequently, there are many sub-sets of application layer services, including as:

- "Scheduling for provider efficiency"
- "Scheduling for user quality of services"
- "Scheduling for negotiations"

This study focuses on "scheduling for user quality of services," which primarily addresses the monetary and temporal aspects of their workflow software. Workflow scheduling aims to minimize both time and money spent. As can be seen in figure [1-5], Montage, Cybershake, epigenomics, Sipht, and Inspiral are some of the most popular processes in scientific computing.

A. Environmental Adaptation Method [EAM]

The core of the EAM has been built on three operators named adaptation, alteration and selection.

- *Adaptation*
  A particle or solution may adapt to its surroundings by taking on the characteristics of the group as a whole or, alternatively, the conditions of the moment. The average fitness of the particles in the current generation determines the final condition that the following generation will reach. The calculation for the adaptation operator is shown in equation 1.
  $$P_{i+1} = (\sigma * (P_i)^{ei} + \varphi)\%2^b \quad (1)$$
  Where,
  P_{i+1}= Adapted population obtained for P_i solution
  P_i= Decimal representation of binary version of the P_i solution
  σ and φ are random numbers.
  b represents the total number of bits used
  e_i is the current environment which equates to $f(Pi)/favg$, where f is the fitness.
- *Alteration*
  The future generation's results might be affected by noise in the present created environment. In order to mitigate the impact of background noise, the modification operator flips a small number of bits in the particle's representation in the population generation we get after applying the adaptation operator.
- *Selection*
  After integrating the original people with the ones gained via adaptation and modification, the best solutions are chosen. Each generation of the population is subjected to these operators in turn until either the maximum number of generations is reached or the target value is achieved.

B. Improved Environmental Adaptation Method [IEAM]

The impact of the EAM's modification operator to eliminate background noise does not allow for effective exploration of the search space. IEAM incorporates a parameter based on the current global best solution to efficiently use the search space. The IEAM selection operator is made in a manner that the best solution of the current generation may guide the remainder of the particles in the next population generation. The IEAM selection operator is presented in equation 2.

$$P_{i+1} = (\sigma * (P_i)^{ei} + \varphi(G_i - P_i))\%2^b \quad (2)$$

Where, G_i = Decimal equivalent of the binary version of the best solution and rest of the computations are same as EAM.

C. Improved Environmental Adaptation Method with Real parameter [IEAM-R]

Many of the issues we face have to do with genuine principles [25] [26]. Real-valued calculations on huge datasets need a lot of space for storing intermediate and final results. Increases in storage technology (primary/secondary) have allowed us to bypass the formerly insurmountable issue of storing space and make direct use of actual numbers. Thus, the idea of employing binary values merely to accommodate storage limits is unnecessary [27]. Working with real parameters, like in the case of IEAM-R, also avoids the need for the extra computing work involved in translating between the real and binary representations. Furthermore, IEAM-R determines the population average fitness and eliminates any solutions whose fitness is higher than the average fitness threshold.

Many methods for scheduling workflows have been developed; they typically include heuristics, meta-heuristics, and hybrid algorithms that combine the best features of each.

D. Existing Scheduling Algorithms in Cloud Computing:

- *An efficient Multi Queue Job Scheduling for Cloud Computing:*
  The fragmentation of FCFS and Round-robin causes unnecessary space consumption and a rise in the price of the user's application. This issue might be avoided by using the MQS that the author suggested (Multi Queue Scheduling). First, it sorts the tasks in ascending order, next it divides them into queues based on their size (medium, small, and big), and last it assigns them to a virtual machine. By taking this course of action, you may save money and avoid wasting precious storage spaces.
- *Improved Max-Min Scheduling Model for Task Scheduling in Cloud*
  The Min-Min algorithm causes an issue with resource distribution. Compared to using either Min-Min or Max-Min alone, RASAs that use both have a longer makespan. While RASA is useful, the makespan that can be achieved with Improved Max-Min Scheduling is greater, and it results in reduced waiting time for scheduled operations.
- *HEFT (Heterogeneous Earliest Finish Time)*
  It is the first cost-optimized, on time, and hybrid cloud-based workflow scheduling algorithm. The author focuses primarily on the factors of time, cost, and resources [28].

Additionally, numerous meta-heuristic algorithms have previously been implemented and evaluated on cloud simulators, including Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Artificial Bee Colony (ABC), Cat Swarm Optimization (CSO), and Ant Colony Optimization (ACO). In comparison to heuristic algorithms, meta-heuristic ones go much closer to the global optimum solution. If you compare PSO to other meta-heuristic algorithms, you will find that its makespan is shorter.

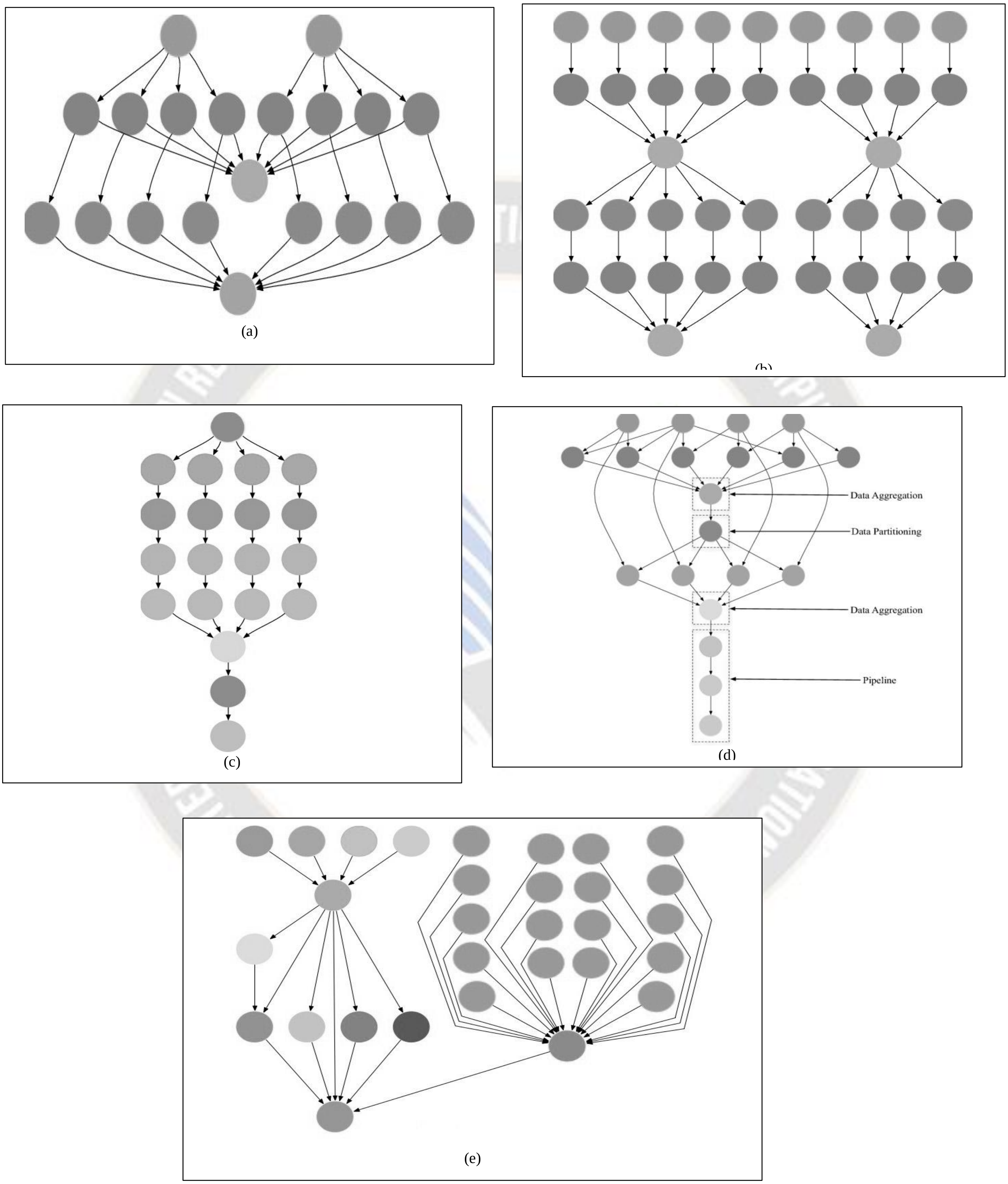

Figure 1. Some well-known Scientific Workflows (a) CyberShake Workflow (b) Inspiral Workflow (c) Epigenomics Workflow (d) Montage Workflow (e) Sipht Workflow

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
***Article Received:** 25 November 2023 **Revised:** 12 December 2023 **Accepted:** 30 January 2024*

There is no single algorithm that guarantees optimum task-resource scheduling to reduce workflow costs and times, therefore there is always room for a better algorithm to be developed. This is especially true in a cloud-computing environment, where there is a constant need for new and improved algorithms.

III. Workflow scheduling via Optimization Algorithms

Figure [1a-1e] displays the most popular cloud/grid-optimized scientific workflows. Scheduling workflows may be thought of as a Directed Acyclic Graph (DAG) with the notation G = (V, E), where V is the collection of tasks and E is the set of data dependencies between these tasks. Because of interdependencies between workflow activities, we encounter two kinds of cost:
Execution Cost (C_{ex}) — It is the cost involved in executing the task submitted to VM which depends on number of instructions required to complete that task.
Data Transfer Cost (C_{tr}) — It is the cost involved in data transfer (related to the submitted task) between any two VMs in the workflow. Thus the total cost of mapping a task M, can be defined as in equation 3:

$$C_{total}(M) = C_{ex}(M) + C_{tr}(M) \quad (3)$$

IV. Proposed Approach

The suggested optimization method is based on a modification of the Improved Environmental Adaptation Method (IEAM) developed by Mishra et al. [29]. The IEAM algorithm, which uses binary encoded computations, has a very high convergence rate but exhibits discrepancies under the conditions stated: when our initial population is small, there may be a significant discrepancy be-tween obtained solutions and desired solutions, and there is also the additional overhead of converting from decimal to binary and vice-versa. The suggested method, denoted as "IEAM-RP," avoids these issues by dealing with real parameters directly rather than their binary equivalent. We also modified the algorithm's adaptation operator, which is represented as equation 4 and 5. In the modified version of the adaptation operator, we have given special attention to the best particle of the generation, because it was quite easy to guide a single particle at a time rather than the whole population. Guiding a single particle, which is also the current best one relives the computational burden involved in guiding the whole population. The rest of the particles can now follow the best particle's path with less computational complexity. Therefore, the convergence of this algorithm is extremely rapid in comparison to its rival algorithms. This is the most desirable quality to possess when working with the cloud, where users anticipate a real-time response to the work that they have submitted.
Adaption method followed by best particle:

$$P_{i+1} = \frac{P_i * F(X_i)}{F_{avg}} + \beta \quad (4)$$

Adaption method followed by rest of the particles:

$$P_{i+1} = P_i + \beta * (B_P - W_P) \quad (5)$$

Where, B_P and W_P are best position and worst position of a particle respectively.
P_i is the position value of a particle.
β is a tuning parameter whose value lie between 0 and 1.
$F(X_i)$ is fitness of i^{th} particle.
F_{avg} is current environmental fitness.

After application of adaption method on particles, selection operator issued to select the best solutions, which is left, unchanged and explained in algorithm 1. Number of best solutions is equal to the initial population size, which is selected from parent population, and offspring has already generated using adaption method. This process is iterated until stopping criteria is met as explained in algorithm 2.

Algorithm 1: Selection Operator of IEAM-RP

```
T_POP = merge(P_i, P_{i+1})
S_POP = sort(T_POP)
P_i = select fittest individual from S_POP
return P_i
```

Algorithm 2: Proposed Algorithm— IEAM-RP

```
Initialize a population of particles POP_i with random values.
repeat
    for i=1 to MaxIteration do
        Evaluate Fitness of each particle
        P_{i+1} = Adaption(POP_i, Fitness_i)
        POP_{i+1} = Selection(POP_i, POP_{i+1})
    end for
Until stopping criteria is met or optimal, solution is found.
```

Table 1: Terminology Used in the Algorithm 1 and 2.

T_{POP}	Temporary Population
S_{POP}	Population after Sorting
P_i	Current Population
P_{i+1}	Adapted Population

POP_i	Population at i[th] generation

A. Application of IEAM-RP in cloud

The proposed algorithm is applied on widely accepted simulator known as cloudsim. The pre-requisites of the scheduling strategy can be listed as:

- Set execution cost of all the VMs.
- Set data transfer cost for each pair of VMs.
- Set number of instructions (in millions) for all VMs.
- Set workflow in millions of instructions (MI) for all the tasks.
- Set workflowDataTransfer for all the tasks.

Execution cost represents the cost involved in 1 sec of task execution in i_{th} VM say exeCost(V_i). Data transfer cost matrix transferCost[V_i][V_j] is set which represents total data transfer cost per MB from i_{th} virtual machine to j_{th} virtual machine. Another matrix is used which defines the processing capacity of i_{th} virtual machine in mips[V_i]. Other steps include setting up matrices containing information about workflow. First, we have to provide number of instructions for each task in a workflow to be processed by the machine. This information is stored in workflowMI matrix where workflowMI[t_i] is the number of instructions in millions. A matrix workflowDataTransfer[t_i][t_j] is also used which contains the amount of data to be transferred in MB from task t_i to t_j.

After setting the above mentioned parameters, we can compute IEAM-RP[t_i] for all the task in a workflow. IEAM-RP[t_i] will give the task to resource mapping according to which tasks can be assigned to the respective virtual machines. The fitness function for the problem of cost minimization can be shown by equation 6 where C_{ex} can be calculated as:

$$C_{ex}\left(t_i, V_j\right) = \frac{MI(t_i)}{mips(v_j)} * exeCost[v_j] \qquad (6)$$

Where, $C_{ex}\left(t_i, V_j\right)$ is the execution cost when task ti is mapped to virtual machine V_j

The transfer cost of equation 7 can be calculated as:

$$C_{tr}(t_i, t_j, v_k, v_l) = workflowDataTransfer[i][j] * vmTransferCost[k][l] \qquad (7)$$

Where,

$C_{tr}(t_i, t_j, v_k, v_l)$ is the data transfer cost from task i to task j where tasks i and j are mapped on virtual machines k and l respectively. workflowDataTransfer[i][j] is the amount of data to be transferred in MB from task i to task j. vmTransferCost[k][l] is the cost of transferring the data from virtual machine k to virtual machine l.

B. Data Structures with standard data used in cloudsim

In cloudsim we have scheduled 10 tasks on 8 virtual machines by assuming a population size of 25 which iterate at most 10 times.

Our execution cost matrix (exeCost[][]) is:

1.21	1.2	1.24	1.18	1.12	1.27	1.25	1.14

Our transfer cost matrix (trCost[][]) is:

0	0.17	0.2	0.2	0.21	0.21	0.18	0.18
0.17	0	0.2	0.2	0.21	0.21	0.18	0.18
0.2	0.2	0	0.17	0.22	0.22	0.19	0.19
0.2	0.2	0.17	0	0.22	0.22	0.19	0.19
0.21	0.21	0.22	0.22	0	0.17	0.2	0.2
0.21	0.21	0.22	0.22	0.17	0	0.2	0.2
0.18	0.18	0.19	0.19	0.2	0.2	0	0.17
0.18	0.18	0.19	0.19	0.2	0.2	0.17	0

Our mips matrix (mips[]) is:

1.011	1.004	1.013	1	0.91	1.043	1.023	0.998

Our workflow in million instruction matrix (millionInstructions[]) is:
(From column 1 to 5)

8000	6000	7000	9000	10000

(from column 6 to 10)

9000	6000	7000	9000	8000

Our workflow data transfer matrix (workflowDataT ransfer[][]) is:

0	80	90	100	0	0	0	0	0	0
0	0	0	0	60	0	0	0	0	0
0	0	0	0	50	70	80	0	0	0
0	0	0	0	0	50	0	60	0	0
0	0	0	0	0	0	60	0	0	0
0	0	0	0	0	0	0	80	90	0
0	0	0	0	0	0	0	0	100	0
0	0	0	0	0	0	0	0	0	50
0	0	0	0	0	0	0	0	0	90
0	0	0	0	0	0	0	0	0	0

Our Directed Acyclic Graph (DAG) representation for the Task Dependency as shown in figure 2.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
***Article Received:** 25 November 2023 **Revised:** 12 December 2023 **Accepted:** 30 January 2024*

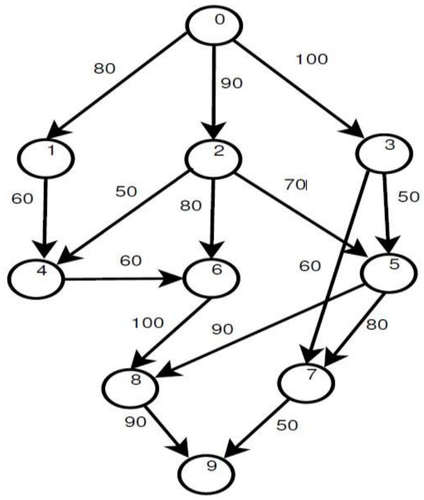

Figure 2. DAG representation of our task dependent workflow in cloud.

V. Results and Comparisons

We have used cloudsim as a simulation tool and implemented the proposed algorithm "IEAM-RP" to minimize the execution cost of tasks submitted on the suggested workflow. Same data set under identical environments were used to get results from other variants of Particle Swarm Optimization (PSO) algorithms like PSO-RAND, PSO-TVIW, PSO-TVAC, and base PSO along with other applied metaheuristic algorithms named GA, GA-PSO, and TLBO. A comparative table has also been attached as table 2. The results were also obtained from FCFS scheduling strategy (The default scheduling strategy used in cloudsim). A quick analysis of result as in figure 3 shows that IEAM-RP gives better results as compared to other applied heuristic approaches in workflow scheduling of the cloud.

Table 2. Total cost involved while scheduling different algorithms on the same workflow having identical simulation criteria for every algorithm

PSO-TVAC	**PSO-TVIW**	**PSO-RAND**	**FCFS**	**IEAM-RP**	**PSO**	**GA**	**GA-PSO**	**TLBO**
157 000	146 000	159 500	162 500	122 500	127 196	125 266	124 157	126 057
145 000	157 000	154 000	162 500	141 000	147 572	147 237	141 550	148 267
153 000	157 500	142 500	162 500	136 000	142 426	139 577	137 212	140 609
148 000	159 000	156 000	162 500	134 500	135 575	141 148	134 555	140 789
149 000	145 500	143 000	162 500	125 120	130 577	129 271	126 888	129 101
157 500	150 100	145 100	162 500	135 250	140 152	136 266	135 736	140 049
150 000	148 100	158 000	162 500	134 120	141 734	138 777	135 321	141 279
148 000	166 000	157 000	162 500	147 500	153 118	148 502	149 586	150 405
147 000	149 000	133 000	162 500	129 000	129 112	129 723	130 141	132 942
156 000	153 000	156 000	162 500	135 990	136 550	139 527	137 469	138 004

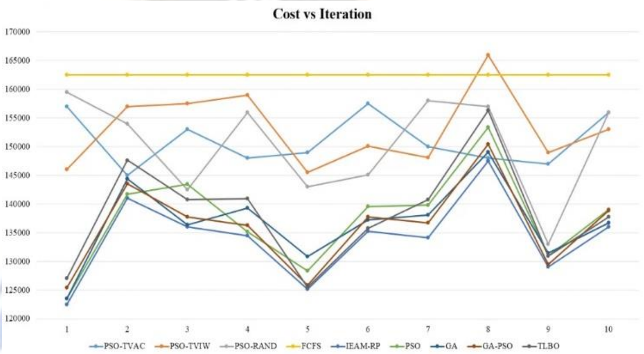

Figure 3. Cost Vs Iteration graph of various optimization technique when applied to the suggested workflow

VI. Conclusion & Future Work

Many metaheuristic algorithms have been proposed for scheduling workload on cloud data center because scheduling load on a pool of virtual machines is an NP-hard problem. There is always a scope of improvement in the optimization algorithms since they are of randomized nature. We were also been able to create a new optimization algorithm, which is a variant of a well-established algorithm name IEAM. We tested that algorithm on the workflow scheduling problem is cloud datacenter and found better results than the existing and applied metaheuristic algorithms in the current research. In the proposed algorithm, the adaptation operator used in IEAM-RP solely governs the particles' movement. In future we would like to change the way the particles moves in the search domain in other words we would like to fine-tune the adaptation operator in a hope to find better results than the current one.

References

[1] B. K. Ferguson, Porous pavements. CRC Press, 2005. https://www.taylorfrancis.com/books/edit/10.1201/9781420038439/porous-pavements-bruce-ferguson

[2] S. B. Shaw and A. Singh, "A survey on scheduling and load balancing techniques in cloud computing environment," in Computer and Communication Technology (ICCCT), 2014 International Conference on. IEEE, 2014, pp. 87–95. https://ieeexplore.ieee.org/abstract/document/7001474

[3] S. Patidar, D. Rane, and P. Jain, "A survey paper on cloud computing," in 2012 Second International Conference on Advanced Computing & Communication Technologies. IEEE, 2012, pp. 394–398. https://ieeexplore.ieee.org/abstract/document/6168399/

[4] P. Salot, "A survey of various scheduling algorithm in cloud computing environment," International Journal of research and engineering Technology (IJRET), ISSN, pp. 2319–1163, 2013. https://shorturl.at/ijyI1

[5] L. K. Arya and A. Verma, "Workflow scheduling algorithms in cloud environment-a survey," in Engineering and Computational Sciences (RAECS), 2014 Recent Advances in. IEEE, 2014, pp. 1–4. https://ieeexplore.ieee.org/abstract/document/6799514

[6] S. S. Manvi and G. K. Shyam, "Resource management for infrastructure as a service (iaas) in cloud computing: A survey," Journal of Network and Computer Applications, vol. 41, pp. 424–440, 2014. https://www.sciencedirect.com/science/article/abs/pii/S1084804513002099

[7] M. A. Tawfeek, A. El-Sisi, A. E. Keshk, and F. A. Torkey, "Cloud task scheduling based on ant colony optimization," in Computer Engineering & Systems (ICCES), 2013 8th International Conference on. IEEE, 2013, pp. 64–69. https://ieeexplore.ieee.org/abstract/document/6707172

[8] L. Liu, M. Zhang, Y. Lin, and L. Qin, "A survey on workflow management and scheduling in cloud computing," in Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on. IEEE, 2014, pp. 837–846. https://ieeexplore.ieee.org/abstract/document/6846537/

[9] Prakash, Ravi, and Ranvijay. "Multi-operator based improved environmental adaptation method for application in real-world optimization problems." International Journal of Information Technology (2023): 1-15. https://link.springer.com/article/10.1007/s41870-023-01505-2

[10] J. M. Broughton, "The genetic psychology of james mark baldwin." American Psychologist, vol. 36, no. 4, p. 396, 1981. https://psycnet.apa.org/doiLanding?doi=10.1037%2F0003-066X.36.4.396

[11] D. Kliazovich, P. Bouvry, and S. U. Khan, "Dens: data center energy-efficient network-aware scheduling," Cluster computing, vol. 16, no. 1, pp. 65–75, 2013. https://link.springer.com/article/10.1007/s10586-011-0177-4

[12] Sharma, Vijay Kumar, Swati Sharma, Mukesh Rawat, and Ravi Prakash. "Adaptive Particle Swarm Optimization for Energy Minimization in Cloud: A Success History Based Approach." In Towards the Integration of IoT, Cloud and Big Data: Services, Applications and Standards, pp. 115-130. Singapore: Springer Nature Singapore, 2023. https://link.springer.com/chapter/10.1007/978-981-99-6034-7_7

[13] Garg, Saurabh Kumar, Steve Versteeg, and Rajkumar Buyya. "A framework for ranking of cloud computing services." Future Generation Computer Systems 29, no. 4 (2013): 1012-1023. https://www.sciencedirect.com/science/article/abs/pii/S0167739X12001422

[14] Xu, Fei, Fangming Liu, Hai Jin, and Athanasios V. Vasilakos. "Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions." Proceedings of the IEEE 102, no. 1 (2013): 11-31. https://ieeexplore.ieee.org/abstract/document/6670704

[15] S. Su, J. Li, Q. Huang, X. Huang, K. Shuang, and J. Wang, "Cost-efficient task scheduling for executing large programs in the cloud," Parallel Computing, vol. 39, no. 4, pp. 177–188, 2013. https://ieeexplore.ieee.org/abstract/document/6670704

[16] M. D. Assuncao, R. N. Calheiros, S. Bianchi, M. A. Netto, and R. Buyya, "Big data computing and clouds: challenges, solutions, and future directions," arXiv preprint arXiv:1312.4722, 2013. https://www.sciencedirect.com/science/article/abs/pii/S0743731514001452

[17] Y. Fang, F. Wang, and J. Ge, "A task scheduling algorithm based on load balancing in cloud computing," in International Conference on Web Information Systems and Mining. Springer, 2010, pp. 271–277. https://www.researchgate.net/publication/220774995_A_Task_Scheduling_Algorithm_Based_on_Load_Balancing_in_Cloud_Computing

[18] Richa Shukla, Bramah Hazela, Shashwat Shukla, Ravi Prakash, and Krishna K. Mishra "Variant of Differential Evolution Algorithm," Advances in Computer and Computational Sciences, year 2017, Springer Singapore, pages=601-608, vol. 7344, 2016. https://link.springer.com/chapter/10.1007/978-981-10-3770-2_56

[19] Z.-H. Zhan, X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H. Chung, and Y. Li, "Cloud computing resource scheduling and a survey of its evolutionary approaches," ACM Computing Surveys (CSUR), vol. 47, no. 4, p. 63, 2015. https://dl.acm.org/doi/abs/10.1145/2788397

[20] M.-A. Vasile, F. Pop, R.-I. Tutueanu, V. Cristea, and J. Ko lodziej, "Resource-aware hybrid scheduling algorithm in heterogeneous distributed computing," Future Generation Computer Systems, vol. 51, pp. 61–71, 2015. https://www.sciencedirect.com/science/article/abs/pii/S0167739X14002532

[21] R. Patel and H. Mer, "A survey of various qos-based task scheduling algorithm in cloud computing environment," International Journal of Scientific & Technology Research (IJSTR), vol. 2, no. 11, pp. 109–112, 2013. https://www.ijstr.org/final-print/nov2013/A-Survey-

Of-Various-Qos-based-Task-Scheduling-Algorithm-In-Cloud-Computing-Environment.pdf

[22] L. Diallo, A.-H. A. Hashim, R. F. Olanrewaju, S. Islam, and A. A. Zarir, “Two objectives big data task scheduling using swarm intelligence in cloud computing,” Indian Journal of Science and Technology, vol. 9, no. 28, 2016. https://ischolar.sscldl.in/index.php/indjst/article/view/132239

[23] Masdari, Mohammad, Sima ValiKardan, Zahra Shahi, and Sonay Imani Azar. "Towards workflow scheduling in cloud computing: a comprehensive analysis." Journal of Network and Computer Applications 66 (2016): 64-82. https://www.sciencedirect.com/science/article/abs/pii/S108480451600045X

[24] S. Rana, A. Choudhary, and K. Mathai, “A critical analysis of workflow scheduling algorithms in infrastructure as a serivce cloud and its research issues,” in Electrical, Electronics and Computer Science (SCEECS), 2016 IEEE Students’ Conference on. IEEE, 2016, pp. 1–6. https://ieeexplore.ieee.org/abstract/document/7509305

[25] T. Dziok, K. Figiela, and M. Malawski, “Adaptive multi-level workflow scheduling with uncertain task estimates,” in Parallel Processing and Applied Mathematics. Springer, 2016, pp. 90–100. https://link.springer.com/chapter/10.1007/978-3-319-32152-3_9

[26] P. Yue, X. Shengjun, and L. Mengying, “An improved multi-objective optimization algorithm based on npga for cloud task scheduling,” International Journal of Grid and Distributed Computing, vol. 9, no. 4, pp. 161–176, 2016. https://www.earticle.net/Article/A272916

[27] S. Xue, W. Shi, and X. Xu, “A heuristic scheduling algorithm based on pso in the cloud computing environment,” International Journal of u-and e-Service, Science and Technology, vol. 9, no. 1, pp. 349–362, 2016. https://shorturl.at/blyGT

[28] N. Chopra and S. Singh, “Heft based workflow scheduling algorithm for cost optimization within deadline in hybrid clouds,” in Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on. IEEE, 2013, pp. 1–6. https://ieeexplore.ieee.org/abstract/document/6726627

[29] K. Mishra, S. Tiwari, and A. Misra, “Improved environmental adaptation method and its application in test case generation,” Journal of Intelligent and Fuzzy Systems, vol. 27, pp. 2305–2317, 01 2014. https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs1195