Journal of Computer Science

Original Research Paper

Multivalent Optimizer-Based Hybrid Genetic Algorithm for Task Scheduling in Cloud Applications

[1]Meena Malik, [2]Bhavna Gupta, [3]Chander Prabha, [4]Dimple Tiwari, [5]Tuğşad Tülbentçi, [6]Şahin Akdağ, [7]Fadi Al-Turjman and [8]Nitesh Singh Bhati

[1]*Department of Computer Science and Engineering, Chandigarh University, Mohali, Punjab, India*
[2]*Department of Computer Science and Engineering, SISTEC, Bhopal, India*
[3]*Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India*
[4]*School of Engineering & Technology, Vivekananda Institute of Professional Studies-Technical Campus, Delhi, India*
[5]*Faculty of Architecture, Near East University, Nicosia, Northern Cyprus*
[6]*Department of Computer and Instructional Technologies, Faculty of Education, Near East University, Nicosia, Northern Cyprus*
[7]*Departments of Artificial Intelligence, Software, and Information Systems Engineering, Research Center for AI and IoT, AI and Robotics Institute, Near East University, Nicosia, Mersin10, Turkey*
[8]*Department of Computer Science & Engineering, School of ICT, Gautam Buddha University, Greater Noida, UP, India*

Article history
Received:02-06-2024
Revised:08-08-2024
Accepted:29-08-2024

Corresponding Author:
Nitesh Singh Bhati
Department of Computer Science & Engineering, School of ICT, Gautam Buddha University, Greater Noida, UP, India
Email: niteshbhati07@gmail.com

Abstract: Cloud computing platforms provide on-demand online services without the need for direct user management. Generally, big clouds distribute functions over multiple data centers at distant locations. The major facilities offered by clouds are based upon virtual machines which provide benefits in terms of low scheduling cost, improved accessibility and availability of cloud services. While transferring the tasks for effective scheduling, the main issue arises due to the domain and characteristics difference of the source machine and the target machine. During network traffic, the challenges are more complex thereby resulting in slow data transfer which leads further issues such as delayed delivery of critical tasks. In order to address the problem of heterogeneity in cloud task, there is a strong requirement of optimal scheme for task scheduling. This research work implements an optimization scheme for scheduling tasks in cloud domains. The offered scheme uses a multivalent optimizer using genetic algorithm termed as Multivalent Optimizer based Genetic Algorithm (MO-GA). It attempts to enhance system performance by transferring the tasks through cloud networks on the basis of resources workload. Therefore, it's very important to apply proper transfer mechanisms for efficient task scheduling in cloud applications. The suggested scheme (MO-GA) ponders various parameters such as system throughput, amount of virtual machines, total number of tasks, speed and capacity. From analytical results, it can be easily identified that our scheme optimizes task scheduling even for large number of tasks efficiently. MO-GA succeeds to achieve optimized tasks' transfer time and get promising results. The scheme is investigated using MATLAB distrusted system for the simulation of the cloud environment. The proposed scheme manage enhancement and optimization of almost 15% over the existing schemes for task transfer.

Keywords: Cloud Applications, Task Scheduling, Genetic Algorithm Multivalent Optimizer

Introduction

Cloud computing helps to manage services as well as information for various organizations. This management implements delivering data center facilities such as storage capability, CPU usage and networking abilities (Hayes, 2008; Kumar and Sharma, 2018; Wickremasinghe *et al.*, 2010). The offered benefits include low data redundancy, lower corporal cost, fast information processing. Generally, the architecture follows three levels for providing different type of services such as software as a service (SaaS), platform as a service (PaaS), infrastructure as a service (IaaS) (Bokhari *et al*., 2018) represented in Fig. (1). SaaS offers services to various users by using system interface. PaaS provides operating system facilities of the cloud. IaaS represents all the hardware facilities by offering storage capability, network service and many more. Paas is primarily accountable for collecting that data by using various network facilities (Li *et al*., 2021).

Mainly three approaches are used for the deployment of cloud services. In first, services are delivered in local surroundings through a private cloud particularly for a city or may be an organization (Linthicum, 2016; Manickam and Raja gopalan, 2019). The next approach implements deployment of a public cloud to fulfill a global network which may cover across multiple countries as well. In third approach, a hybrid deployment scheme combines the private and public methods as per user domains requirements.

In order to deliver the services using private cloud, there is a requirement to connect private and public cloud together. The resulting hybrid cloud provides lots of advantages but a main challenge arise for scheduling the tasks (Abualigah and Diabat, 2021a). It may be difficult to transfer task scheduling due to difference of specifications and facilities offered by respective clouds (Yuan *et al*., 2020; Abualigah *et al*., 2020c). Here, transfer rate, processing speed, throughput and storage capacity may vary and this variation may effect delivery time, transfer workload and effective resource usage (Mansouri *et al*., 2021). The data and resource centralization approach in clouds makes it more suitable for business firms due to reduction in cost, faster, low data redundancy and better data using central operations (Alguliyev*et al*., 2019; Sreenu and Sreelatha, 2019).

An extensive range of applications use cloud structure for data collection from different banks, institutes, schools, hospitals and universities. Still various institutes try to cut down operational cost of the cloud services by using public cloud services. One of the key parameters for cloud applications is make span and to achieve minimum make span, task scheduling becomes really hard. An efficient scheduler is required to do effective task handling suitable for variety of task and changing environment. Task scheduling approach and its efficiency plays critical role for resolving the primary issue of task transfer (Nadjaran Toosi *et al*., 2018). An effective scheduling algorithm ensure minimum make span without disturbing all the necessary precedence constraints. It also ensures maximum resource utilization for improving task transfer performance. The complexity of scheduling possibilities tends to increase for ensuring task transfer efficiency and load balancing with lower energy consumption (Mapetu *et al*., 2021.) Here we can easily conclude that the complexity of scheduling scheme for task transfer turn out to be complex for ensuring transfer efficiency while lowering energy consumption. Therefore, there is a strong requirement of optimization method here in order to address the main issue of transfer efficiency. This study implements an optimization method for improving task transfer performance in the domain of cloud applications. The performance of the system for transferring tasks demands efficient scheduler in order to achieve effectiveness to meet dynamic change in active tasks in cloud scenarios.

Literature

A wide range of research works have been suggested to resolve the issue of workload transfer among heterogonous cloud centers. For task scheduling the important issue to focus is resulting transfer efficiency. To achieve virtuous efficiency there is a strong need of transferring task while maintain low response time using existing cloud resources. A well-organized cloud deployment would be able to handle transferring faster tasks on priority basis (Ashouraie and Jafari Navimipour, 2015). A wide range of computational features for transfer efficiency need scheduling the transfer priority on the basis of its corresponding delivery time. We can use features from already existing nature inspired schemes mentioned in Fig. (2), for scheduling the tasks in order to optimize the transfer of different type of tasks.

This study targets to increase efficiency of task transfer scheduler in cloud scenarios for heterogonous tasks by offering multivalent optimizer over existing genetic algorithm. Our proposed scheme combines the multivalent optimizer with Genetic Algorithm (GA) to improve its search strategy by minimizing the deficiencies in the most promising local issues. The chief motivations involve two parts: In first, the cost of heterogeneous tasks' transfer may be reduced by allocating task to all cloud resources available at the required moment. The efficiency will be computed on the basis of transfer time and other important system parameters such as transfer rate, throughput, capacity of cloud machines and processor speed.

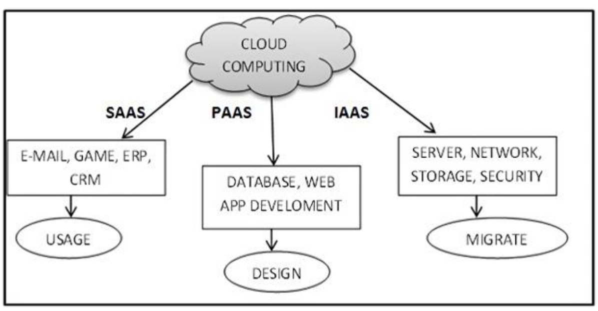

Fig. 1: Type of cloud services

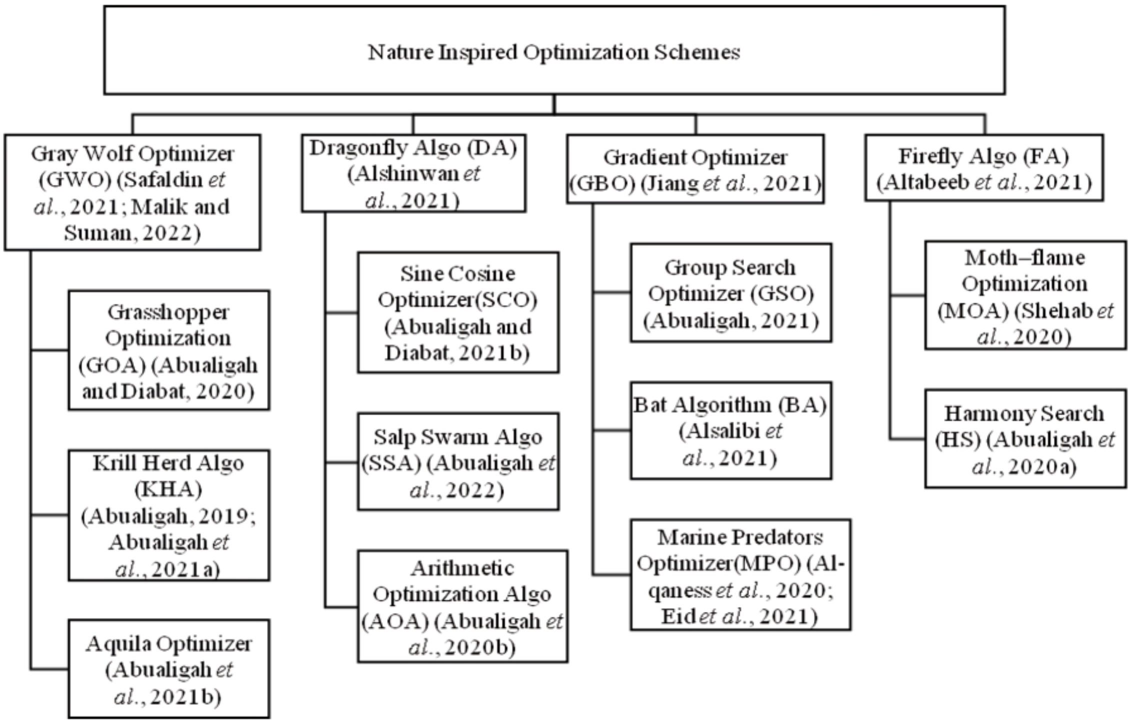

Fig. 2: Type of optimization schemes

In (Safaldin *et al.*, 2021), the author proposed Grey wolf optimizer using support vector machine (GWOSVM) that work basically for intrusion detection system (IDS), to improve the accuracy of intrusion detection thereby decreasing the processing time in Wireless Sensor Networks. By the test results it can be easily revealed that the GWOSVM with 7 wolves is proved more improved than existing algorithms. The study proposed in Malik and Suman (2022), implements a hybrid Lateral Wolf and Particle Swarm Optimization (LW-PSO) for addressing various task scheduling problems and offers an optimal load balance scheme for Cloud services. It focused for the parameters mainly average load, processor utilization and average turnaround time, average response time for performance analysis. The Dragonfly algorithm (Alshinwan *et al.*, 2021), proved to be an competent and efficient scheme for swarm optimization that may be applied in various fields like engineering, medical, energy systems etc. Here, several characteristics of dragonfly like binary, discrete, modify and hybridization of DA has been discussed and compared with existing schemes.

The study (Jiang *et al.*, 2021) discussed Gradient Based Optimizer (GBO) that is an efficient approach and use the mechanism of Feature Selection (FS). It practice Local Escape Operator (LEO) as well as Gradient Search Rule (GSR) for solving various issues. This study considered, 8 GBO versions and 8 transfer functions are encouraged for mapping the search space to a particular discrete space. The investigational results have recommended that Binary GBO schemes performs better than metaheuristic schemes.

The study in Altabeeb *et al.* (2021), implements a cooperative and hybrid firefly algorithm (CVRP-CHFA) to bargain the new vehicle routes with minimizing traveling distance. It is mainly using the mechanism of Capacitated vehicle routing problem (CVRP). The results revealed promising results using CVRP-CHFA along with outperforming the existing firefly algorithms. The study(Abualigah and Diabat, 2020), offers grasshopper algorithm which tends to be one the mosteffective and competentrecent meta-heuristic optimization algorithms. It is applicable to several types ofOptimization problems such as engineering design, wireless networking, machine learning, image processing, control of power systems.

The article (Abualigah and Diabat, 2021b) discusses a Sine Cosine Algorithm (SCA), which works for population optimization algorithm and it was inspired by the scheme offered by Mirjalili in 2016, motivated by the trigonometric sine and cosine functions.From the results and comparison it can be easily stated that Sine Cosine algorithm can successfully solving problems in domain ofmachine learning, image processing, software engineering, networking etc. In (Abualigah, 2021), authors suggested, Group Search Optimization which is a nature inspired scheme for resolving optimization problems on the basis of animal searching behavior in real life. It exhibits quite dominant due todecision making problems. The performance analysis represents promising results as compared to various popular existing optimization algorithms. The Moth-Flame Optimization algorithm (Shehab *et al*., 2020), is used in wide range of optimization problems. The study is proved to be beneficial for different researchers and practitioners in many fields, including optimization, medical, engineering, clustering and data mining

The Krill Herd approach in article (Abualigah, 2019; Abualigah *et al*., 2021a), represents the amount of text information on the internet and in modern applications has increased, leading to a growing interest in the area of text analysis. This interest is driven by the need to process a large amount of unorganized text information. Here the author discussed the impact of COVID-19 on health, economy and society and how researchers from different fields have been working to find solutions using AI and other intelligent data analysis concept. He also highlighted the most influential authors and journals on this topic and demonstrated the growing value of open access publications. Similarly, the author in (Abualigah *et al*., 2022), implements the Salp Swarm algorithm, where the hill climbing technique is used for solving engineering design problems. The scheme is composed of two stages, at first hybridization of the SSA is done by means of the HC local search in order to improve its expected capabilities. At the second stage, the scheme applies a selection scheme to enhance the exploration capabilities of the hybrid SSA. In Alsalibi *et al*. (2021), Dynamic Membrane-driven Bat Algorithm (DBMA) is implemented to improve optimization efficiency in various applications. The scheme targets to enhance population diversity by balancing the exploration-exploitation tradeoff. The proposed algorithm is tested on a set of benchmark functions and compared to other variants of the bat algorithm.

The article (Abualigah *et al*., 2020a), proposed the Harmony search algorithm (HSA) which is basically a swarm intelligence optimization scheme that may be applied to various clustering applications, such as data clustering, text clustering, image processing and wireless sensor networks. The author identifies the limitations of HSA and suggests potential research directions to enhance its performance and applicability to solve complex optimization problems. The Aquila Optimizer (AD) (Abualigah *et al*., 2021b) is a new population-based optimization method, which is inspired by the hunting behavior of Aquila birds. The optimization mechanisms of the proposed algorithm are represented in four methods. The experimental results show that the developed QO algorithm outperforms well-known metaheuristic methods in finding the optimal solution.

Arithmetic Optimization algorithm (Abualigah *et al*., 2020b) is a new meta-heuristic method which utilizes the distribution behavior of the main arithmetic operators including Addition, subtraction, multiplication and division. Experiments show excellent results by AOA as compared to otherwell-known optimization techniques. The rise of COVID-19 pandemic became the biggest challenge for mankind. Adaptive neuro-fuzzy Inference System (ANFIS) (Al-qaness *et al*., 2020) is modelled to forecast the number of covid infected people in four countries, Italy, Iran, Korea and the USA. It is based on Marine predators algorithm, that optimizes ANFIS parameters. The results reflects MPA-ANFIS outperformed all the other techniques in almost all measuring areas. In (Eid *et al*., 2021) the author have proposed an improved Marine Predators Algorithm, that lead to rapid convergence and avoid local minima stagnation for the original MPA. There are two standard test systems, which are thought to test the efficiency and performance of the algorithm, 69-Bus and 118-bus.

Cloud computing has been started globally to provide central support for storing and processing various tasks hereby avoiding various organizational costs. It need to transfer high number of tasks thereby maintaining low response time (Chen *et al*., 2017). The weak scheduling will cause multiple issues for delivering optimal cloud services. Therefore, it is very important to implement optimal task scheduling schemes for efficient use of available cloud resources. This study implements MO-GA scheme for scheduling and optimization of different type of cloud tasks performance.

The Second important phase involves the integration of MO and GA to attain effective processes scheduling for cloud tasks transfer by optimizing the transfer time. The newly designed MO-GA is capable to plan the transfer tasks scheduling on the basis of existing workload of the cloud resources. In contrast, the GA is capable to enhance the traditional MVO approach with the help of mutation and crossover procedure in order to improve the already started schedule using MVO. The experimental research is presented by two different scheduling set-ups for the validation of efficiency of the proposed scheme. The experimental results demonstrate that the offered scheme provide improved results as compared to existing methods. The work contribution is outlined below:

- A novel inventive scheduling approach for cloud application to attain minimum transfer time using available resources
- Combining the features of existing genetic algorithm with multivalent optimization to enhance the searching ability
- Experimenting different assessment criteria along with changing scenarios to examine the proposed method

The Proposed Algorithm

This segment explains the research methodology comprising the full research design of the proposed scheme, suggested parameters and process, modeling environment and evaluation practices. The methodology is generated following the various related works already present for transferring the heterogeneous cloud tasks. The system suggests a complete set of associated procedures and settings to accomplish and address our decided objectives for our suggested work. On the basis of objectives and requirements of the scenario, the methodology has been planned to design the research.

Materials and Methods

This study emphasize, effective scheduling of dissimilar kinds of tasks in the cloud environment. Here, the fundamental challenge identified is transferring the task-centered effectiveness parameters. Multiple already existing studies are examined and reviewed for regulating competency features which may be implemented for scheduling the priority basis of heterogeneous tasks. The important parameters considered include processing speed, storage ability, distance, throughput and transfer rate. The task scheduling considering system efficiency requires maximum utilization of available cloud resources thereby maintaining effective task transfer. To fulfill the requirements, this study implements Multivalent optimization scheme combined with well-known genetic algorithm resulting into a novel scheme named as MO-GA.

Task Scheduling

The task scheduling may be described as allocation of different tasks to different Virtual Machines (VMs) and ensuring completion of all in minimum possible period. Suppose a system '*S*' in cloud having '*K*' physical machines represented as K_{PM} and every physical machine contains '*K*' virtual machines represented as K_{VM} (Abualigah and Diabat, 2021a). So all the physical and virtual machine in the system may be represented as:

$$S = \{PM1\,, PM2\,, PM3 \ldots\ldots\ldots\ldots\ldots\ldots\ldots PMK-1, PMK\} \quad (1)$$

$$PMK = \{\, VM1\,, VM2\,, VM3 \ldots\ldots\ldots\ldots\ldots\ldots\ldots VMK-1, VMK\} \quad (2)$$

To represents tasks submitted by different users and a virtual machine's serial number along with its processing speed measurement unit in terms of MIPS, we can denote is as follow:

$$TK = \{\, Task1, Task2, Task3 \ldots\ldots\ldots\ldots TaskK-1, TaskK\} \quad (3)$$

$$VMK = \{VMK, SID, VMK, MIPS\} \quad (4)$$

To represent N^{th} task into the sequence it can denoted as T_N and the features of tasks may be defined as:

$$TN = \{TaskSID, TasklenN, ECTN\,, PIN\} \quad (5)$$

Here *TaskSID* represents serial number and *TasklenN* denotes length of the instruction of the task (unit: million instructions). The estimated completion timeis denoted as ECTK and task priority is referred as PIK. The matrix representation for estimated Completion Time (ECT) in matrix form of the size *Ktsk*×*Kvm Indicates* the required execution time to execute the task on virtual machine.

$$ECT = TasklenN/\,MIPSK \quad (6)$$

k = 1,2,3……no of VMs. *TasklenN* refers task length *N*. The object function aims to decrease the make span by identifying the most suitable set of tasks for executing on VMs where ECT_{NK} represents the execution time for Nth task using K^{th} *VM* and VM_K Virtual machine number, *Ntsk* is the number of tasks. *TasklenN* is the task–length. The fitness value is defined as:

$$F = max\{ECT\}, \forall\, [l, Ntsk]\ belongingtokthvirtualmachine \quad (7)$$

The Proposed Approach-Multivalent Optimizer

MO designed for an organized population segment, where computing the fitness value is involved for effective task scheduling. Let's consider, {VM_1,VM_2, VM_3, VM_4} are virtual machines and transfer tasks currently active are {T_1, T_2, T_3 and T_4}. In array [i][j], the population for MO (Grey hole) is symbolized for currently active tasks, idenotes total tasks and j is total virtual machines. Here, the fitness calculation for each column of array and identification of optimal machine for all the task will be calculated on the basis of corresponding transfer speed. For instance, lets assume T1/V4, T2/V3, T3/V4 and T4/V2 denotes the optimum solutions for all tasks given in the array (i.e., black hole). On the basis of population in black hole, the GA will be applied to optimize the task scheduling based on the available cloud resources according to the intersection between the transfer tasks. The MO solutions get updated by Eq. (9):

$$xij\,.\,d1 >= Ni(Di)\,, xkj\,.\,d1 < Ni\,(Di) \quad (9)$$

$$xij = \{xijford2 \geq WEPandxij = \{\, Xj + TDRford2 < WEP\&d3 < 0.5 \quad (10)$$

$$TDR = T * ((UB - LB) * d + LB \quad (11)$$

$$T = (\,1 - (l1/p)/\,(L1/p)) \quad (12)$$

x_{ij} is representing i^{th} parameter of the j^{th} division , Di is the universe division, d1,d2,d3,d4 is the random value in the [0,1] interval.

Genetic Algorithm

To attain a correct schedule for task processing in the cloud environments genetic algorithm is used for better results. Basically, the main objective of using genetic algorithm is to moderate the population boundaries by controlling the MO black hole. By reducing the size of task population lead to optimal solution by Genetic Algorithm.

The primary working mechanisms for the genetic algorithm are as follow:

- Selection mechanism: This method helps to discover random pair solutions to apply the operations such as crossover or mutation on the basis of its probability
- Crossover mechanism: This method selects chromosomes' pair in order to produce next generation chromosomes with the help of two-point manner. Several features from the first parent get transmitted to the new generation chromosome and rest of the features are transmitted by remaining parents. The probability of crossover used here is 0.8
- Mutation mechanism: It is used in a genetic algorithm in order to preserve the variation of the population by means of varying the chromosome using a minor probability from [0, 1]. The mutation probability denoted by (Pm) 0.2 is used here. Moreover, Mp1 and Mp2 which are simply random numbers chosen from [0,1], are defined two mutation points for performing mutation
- Termination Criteria: If Fitness value is not improving further then termination condition is applied. Using termination criteria, the best or fittest chromosome get identified from the population

Hybrid Multi-Verse Optimizer with Genetic Algorithm (MO-GA)

The genetic algorithm is one of most popular optimization method which focus to solve constrained as well as unconstrained problems. It follows adaptive heuristic style for searching and belongs to evolutionary algorithms. It follows the natural selection style, which suggests to select the most fittest value for generating the successive generation. In order to compute and find the fittest score there is a requirement to get individual's competency level. To get best offspring, individual fitness score must be optimal or close to optimal.

So, MO can be collaborated with GA to solve cloud optimization issues. The MO implements decrease in size of the task's boundaries. The illustration possibly will be as Task Boundaries (TB) = (V, E), V can denote number of tasks need to be executed and E denote the transfer paths through the cloud's VMs. Alternatively, the GA is recommended to be applied for optimization of the task scheduling considering all the necessary efficiency properties. In the Fig. (3), the overall design of the proposed methodology has been represented.

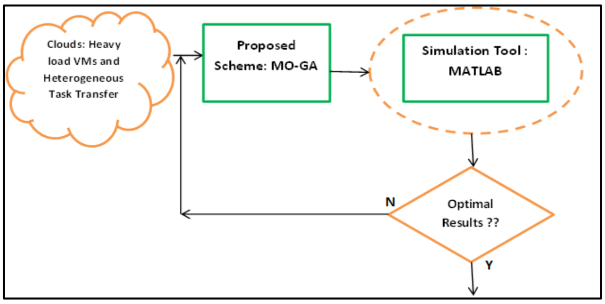

Fig. 3: illustrates the methodology design of this study

Multivalent Optimizer (MO) is our offered scheme which will follow evolutionary features to optimize the possible solutions on the basis of universe theory. According to the theory, every object present in universe (i.e., cloud tasks) is part of the initial hole that is generally referred as a white hole. Then on the basis of some specific circumstances, several objects can switch to some other smaller hole which is referred as a black hole. This black hole possibly will accept some other objects which may be part of a different universe which can be specified as a wormhole.

Now, there are high chances that features of white hole objects will be different from the features acquired by objects of the worm universe. Here, the black hole is playing responsibility for scheduling of every object on the basis of different characteristics. Here key factor to consider is that every object of the universe need to fulfill certain parameters then only it is allowed to switch to a black hole.

Results

This section indicates a brief discussion associated to experimental backgrounds for task transfer in cloud scenarios. On the basis of various experiments, the results have been displayed along with the comparison to the existing schemes for scheduling the task transfer. The proposed scheme is a hybrid version of multivalent approach and genetic algorithm to achieve optimization while transferring different type of tasks from one VM to another that is clearly represented in Fig. (4). Therefore, it mainly focuses for decreasing the transfer time.

The proposed scheme distributes task by following the load balancing criteria. Thus, it need to consider various properties of VMs such as storage capacity, transfer speed, CPU utilization and throughput. Then genetic algorithm will be applied to enhance the scheduling on the basis of mutation and crossover procedures. The outcome received by hybridization of MO-GA will be fed as its initial population for further enhancement. Hence, it can be stated that VMs workload is assured using MO-GA and further enhancement for workload is handled by original genetic algorithm.

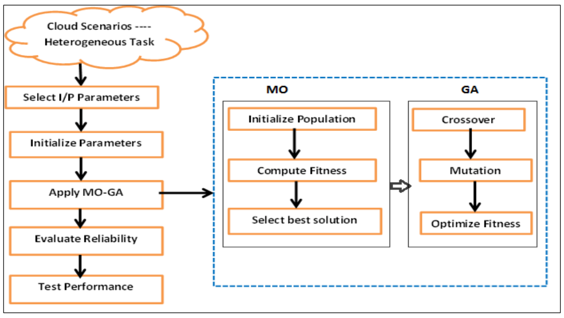

Fig. 4: Represents hybrid version of multivalent approach and genetic algorithm to achieve optimization

For experimental evaluation, MATLAB has been used and to meet the requirements of cloud environment for heterogeneous task transfer distributed toolbox has been utilized. The simulation parameters used are presented below in Table (1). All the tests are conducted using 5 different rounds having 500 iterations each.

The work tries two different ways for scheduling task transfers having 500, 1000tasks respectively. This variation in number of tasks is mainly focused to simulate the testing of all issues which may arise due to scheduling and these issues may increase with increase in number of tasks. The task identification is done by using four features: Unique ID of task, size of the task, estimated task transfer time and priority of the task. The size of task indicate computational estimate for transfer time. Additionally, to analyze expected and actual transfer time expected ECT has been used.

For cloud environment, mainly two approaches are used for the placement of dataset. In first approach, virtual machines directly transfer the datasets to other machines available on cloud. In second method, the datasets are transmitted through data centers which efficiently manage large sets in short span only. In our work we have implemented data centers approach for MO processes. For experimentation two different virtual machines have been used and parameters undertaken are stated in the Table (1).

The experimentation is done using two different modes considering 500 tasks and 1000 tasks in the dataset. Two different data centers have been created to process MO approach. MO is responsible for task transfer scheduling on the basis of expected ECT for assuring the workload balance in all VMs. In order to optimize the performance of GA, crossover process as well as mutation process will be applied on the considered dataset for optimal transfer time.

In Fig. (5), ECT is displayed for optimizing the schedule of 500 tasks on the basis of elementary algorithms only i.e. GA and MVO. Initially, the MO has been used to recognize the efficiency of the MO-GA scheme. By following the MO procedures, the ECT was calculated for both virtual machines' complete executions. By observing the results, we can easily conclude the optimal ECT is 5.6 s approx. for best case and in worst case it is 654 s, alternately, the MO also decreases the average ECT to 649 s for task transfer. Furthermore, the optimal ECT used for the two VMs in GA 2.96 s and 593 in best case and worst case respectively. The selected two algorithms found practical enough for optimizing the average ECT to get minimum solution for tasks transfer. Moreover, the GA decreases the ECT time by almost 590 s. In short, the task transfer schedule for 500 tasks, GA and MO both are suitable for optimizing the average ECT. Conversely, the GA found to be more efficient over MO in the worst as well as best cases for optimal ECT.

The Fig. (6), represents the average ECT of 1000 tasks for task transfer on the basis of MO and GA. The MO optimal results represents that the total ECT of 1000 transfer tasks is about 5.27 s compared with 199 as initial ECT. Hence, the MVO reduces the transfer ECT by about 194 s. On the other hand, the optimized ECT based on GA for1000 tasks records about 4.7 s comparing with 190 s as initial ECT. Thus, the GA reduces the ECT by about 186 s for the dataset of 1000 tasks. In conclusion, the GA is more effective than the MVO in optimizing the ECT of the transfer 1000 tasks using the two virtual machines. Although the optimized ECT is useful for MVO and GA, it can be noticed that the optimization of each algorithm is less effective when handling a larger dataset. The optimization of MVO and GA is better for the 500 tasks than the optimization for 1000 tasks.

Table 1: Simulation parameters

Parameter	Value	Parameter	Value
Number of cloudlets	100–2000	No of hosts	2
Length	1000–2000	Host's RAM	2048 MB
Size	10,000	Host's Storage	100 GB
VM's RAM	512 MB	Host's Bandwidth	1 Gb/s
VM's Bandwidth	1 Gb/s	DC	2
No of CPU	1	Policy type for host	Time shared
VM's MIPS	100–1000		

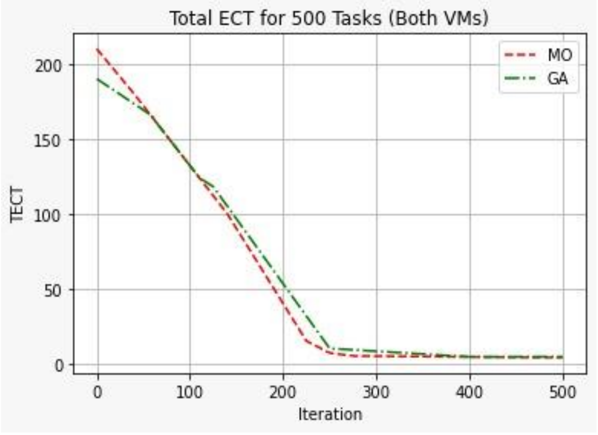

Fig. 5: Average ECT of 500 tasks for task transfer for MO and GA

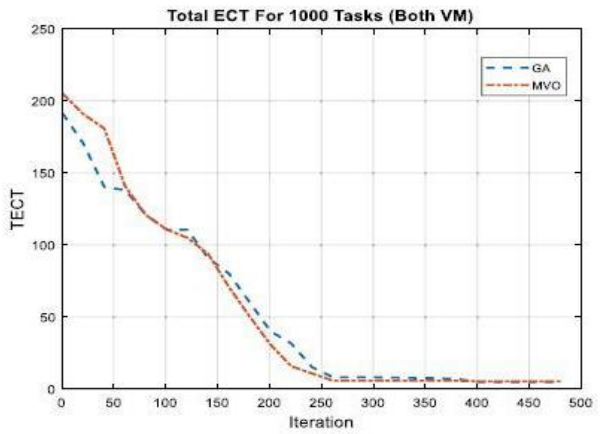

Fig. 6: Average ECT of 1000 tasks for task transfer for MO and GA

Figures (5-6), clearly indicates that with growing size of dataset, it becomes difficult to optimize the virtual machines. In the Table (2) a brief summary for optimal ECT on the basis of several experiments and scenarios has been included for MO as well as GA by considering separate set of500, 1000 tasks. By observing and comparing entries in Table (2) it can be stated easily that, MO-GA manage to provide improved ECT to transfer 1000 tasks for both type of machines.

From 3rd row of Table (2), it can be observed MO-GA enhance the average ECT using two machines for transferring 1000 tasks. The outcome clearly specifies that the MO-GA attain around 1.5 s for transferring1000 tasks by two machines. Therefore, the MO-GA would be able to get improved ECT as compared to working with MO or GA as a separate mode. The optimal ECT is about 5.1 s when simulating with MO and4.7 s using GA.

From Figs. (7-8), it can be easily identified that the crossover procedure is very important for the enhancement of optimal ECT. Alternatively, the incorporation with MO offers efficient ECT for the bulky data as compared to applying both schemes separately. As compared to other existing approaches, such as Gray Wolf Optimizer GWO and Arithmetic Optimization-AO, the suggested MO-GA approach manage improved results. From Fig. (8), it can be clearly stated that the offered scheme outdone all the existing approaches when used 1000 tasks.

The results in Table 3 clearly indicate that the small size dataset for example transfer tasks of 500, easily can be handled with the help of single optimization scheme for instance either GA or MVO. The actual challenge for scheduling of task transfer get clear when there is need to handle a bulky dataset, for example over and above 1000 tasks. With the help of a particular approach either MVO or GA may possibly increase the ECT. But, the additional improvement is required for optimizing the ECT.

Table 2: A brief summary for optimal ECT on the basis of different scenarios

No of tasks	GA	MO
Crossover (2 VMs)	AvgBest	AvgBest
500	108.55 2.78	117.7 5.53
1000	54.9 4.9	53.9 5.3
Individual VMs 1000 tasks	AvgBest	AvgBest
VM1	68.9 14.5	86.11 43.08
VM2	67.6 15.01	88.94 32

Table 3: Indicates the average and best case scenario for all the considered schemes by considering task size 500 and 1000 respectively

Approach	Task size	Avg	Best
MO-GA	500	23	1.7422
	1000	60	2.156
MO + only Crossover	500	74.32	2.224
	1000	76.373	2.976
MO + only Mutation	500	30.426	2.654
	1000	106.242	5.012
Only MO	500	71.564	12.564
	1000	84.546	99.643
GWO	500	1.243	40.223
	1000	4.332	60.342
AO	500	1.323	39.786
	1000	5.676	62.346

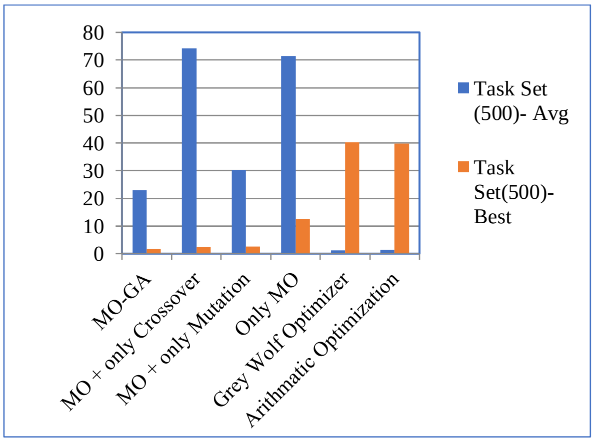

Fig. 7: Average ECT of 1000 tasks for task transfer forMO and GA

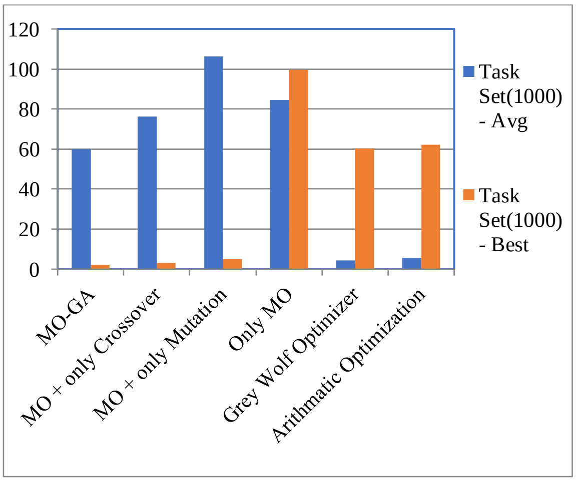

Fig. 8: Average ECT of 1000 tasks

Therefore, the MO-GA is suggested for the improvement of ECT in order to optimize the bulky datasets. The offered collaborative scheme implements the scheduling faster as compared to a single approach. As example, by MO-GA the ECT for 1000 transfer tasks is calculated 3 s but, the optimal ECT by using MO is 27 s, while the best score for GA is recorded 23 s. Therefore, it found to be very effective for optimization of ECT with the collaboration of two stage optimization. Initially, the MO phase tries to schedule the tasks for balancing the workload on available machines. Then, the GA stage operates in two modes crossover and mutation to find better re solutions in order to improve the ECT. It focus replacing the different tasks on all the available resources so as to increase the total ECT.

Discussion

This section presents discussion regarding the analytical results obtained by the offered scheme i.e. MO-GA for the optimization of cloud tasks. It also signifies the applied datasets along with all the experimental set ups to implement various results on the basis of different scenarios. In short, the ECT optimization for transfer tasks have faced a lot of issues while transferring the bulky data sets. While employing MO-GA, it has been observed that it offers an acceptable time as compared to using only MVO or GA.

In brief, this study is important in the way of efficient and effective task scheduling in cloud application for transferring tasks to improve data transfer rate by using unlike capacity of cloud resources. The leading contributions of this study are the following:

1. Hypothetically, this study identifies the core affecting factors for the effective time of cloud tasks transfer. It basically depends upon bandwidth, length of tasks, speed and capacity. The output of important characteristics should be carefully examined for addressing time optimization in the cloud transfer schedule
2. The work mainly contribute by describing the efficiency of MO-GA approach for scheduling task transfers. The gaps are identified for task transfer techniques while implementing several optimization algorithms
3. The proposed scheme i.e. MO-GA resulted an efficient algorithm to ensure the workload and capability of all the existing resources in cloud environments. The scheme can efficiently manage the cloud tasks distribution by analyzing several cloud resources capabilities, for example the capacity and transfer speed of the offered virtual machines
4. The considered scheme GA is suitable for improving the schedule of task transfers rather than scheduling the tasks itself. As per the availability of resources over the domain, the crossover and mutation processes in GA accounts for designing better opportunities for the allocation of tasks' transfer. Therefore, the GA plays decisive role in assistance rather than controlling the entire operation of scheduling
5. The MO-GA offers a possibility for efficient transfer time for the schedule of big set of cloud tasks. The characteristics of two algorithms can be considered on the basis of two steps: One is to organize the initial schedule of task transfer by MO scheme for addressing the load of resources and second is to improve the scheduling by applying GA to find improved opportunities for the optimization of transfer time having several different capabilities of cloud resources

Conclusion

Cloud computing has been started globally to provide central support for storing and processing various tasks thereby avoiding various organizational costs. It need to transfer high number of tasks thereby maintaining low response time. The weak scheduling will cause multiple issues for delivering optimal cloud services. Therefore, it is very important to implement optimal task scheduling schemes for efficient use of available cloud resources. This study implements MO-GA scheme for scheduling and optimization of different type of cloud tasks. Here, the main

Challenge arises while scheduling the transfer of heterogeneous tasks in the cloud scenarios. The variations in resource specifications is another issue in cloud domains which may follow low task transfer due to speed, processing and storing capacity variation. The proposed optimization scheme will try to solve the problems with the help of scheduling criteria which follows transfer time adeptness as well as open resources. The proposed scheme focus on task scheduling as per available machines in the cloud for a balanced task distribution and ensuring productivity of virtual machines for bandwidth, speed and capacity. Alternately, GA works for expansion of workload on various cloud resources for reducing task transfer time. Crossover and mutation process helps to improve this transfer by using multiple virtual machines. Here the outcome of MO scheme will be fed as starting population for the next stage i.e., GA scheme and here crossover and mutation procedures try to optimize task transfer time by detecting lower load machines. For the last objective, simulation of MO-GA algorithm is done using MATLAB to find and represents its efficiency. From simulation plots, it can be stated very easily that the hybridization of MO with GA manages faster task transfer as compared to their individual performance. The offered scheme manage enhancement and optimization of almost 15% over the existing schemes for task transfer. The scheme is providing excellent optimization results for large tasks as well thereby reflecting effectiveness for its performance.

On the basis of results for the proposed scheme, following options are advised as future works.

We can use the MO-GA for large data over cloud to transfer large number of tasks and check its effectiveness for real environment. The collaboration can be tested with other optimization algorithms as well such greedy, PSO for scheduling the task transfer in order to optimize the transfer time.

Acknowledgment

We welcome the suggestions from various reviewers' for comprehensive review, that can help to improve the quality of our work.

Funding Information

We declare that this manuscript received no funding from any sources.

Author’s Contributions

Meena Malik: Conceptualization, methodology, investigation, experimentation and made significant contributions to the writing of the manuscript.

Bhavna Gupta: Revised and edited the manuscript.

Chander Prabha: Data collected and analysis.

Dimple Tiwari: Validation, review and editing.

Tuğşad Tülbentçi: Data curation and formal analysis.

Şahin Akdağ: Draft preparation and visualization.

Fadi Al-Turjman: Reviewed the manuscript and provided insightful, constructive feedback.

Nitesh Singh Bhati: Research plan design and supervision.

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

Ethics

This study is author's original research and has not been submitted or published elsewhere.

Data Availability

The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no competing interests.

Reference

Abualigah, L. (2021). Group Search Optimizer: A Nature-Inspired Meta-Heuristic Optimization Algorithm with its Results, Variants and Applications. *Neural Computing and Applications*, *33*(7), 2949–2972. https://doi.org/10.1007/s00521-020-05107-y

Abualigah, L. M. Q. (2019). *Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering* (1st Ed.,). Springer Cham. https://doi.org/10.1007/978-3-030-10674-4

Abualigah, L., & Diabat, A. (2020). A Comprehensive Survey of the Grasshopper Optimization Algorithm: Results, Variants and Applications. *Neural Computing and Applications*, *32*(19), 15533–15556. https://doi.org/10.1007/s00521-020-04789-8

Abualigah, L., & Diabat, A. (2021a). A Novel hybrid Antlion Optimization Algorithm for Multi-Objective Task Scheduling Problems in Cloud Computing Environments. *Cluster Computing*, *24*(1), 205–223. https://doi.org/10.1007/s10586-020-03075-5

Abualigah, L., & Diabat, A. (2021b). Advances in Sine Cosine Algorithm: A comprehensive survey. *Artificial Intelligence Review*, *54*(4), 2567–2608. https://doi.org/10.1007/s10462-020-09909-3

Abualigah, L., Alsalibi, B., Shehab, M., Alshinwan, M., Khasawneh, A. M., & Alabool, H. (2021a). A parallel hybrid krill herd algorithm for feature selection. *International Journal of Machine Learning and Cybernetics*, *12*(3), 783–806. https://doi.org/10.1007/s13042-020-01202-7

Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-qaness, M. A. A., & Gandomi, A. H. (2021b). Aquila Optimizer: A Novel Meta-Heuristic Optimization Algorithm. *Computers & Industrial Engineering*, *157*, 107250. https://doi.org/10.1016/j.cie.2021.107250

Abualigah, L., Diabat, A., & Geem, Z. W. (2020a). A Comprehensive Survey of the Harmony Search Algorithm in Clustering Applications. *Applied Sciences*, *10*(11), 3827. https://doi.org/10.3390/app10113827

Abualigah, L., Diabat, A., Mirjalili, S., Elaziz, M. A., & Gandomi, A. H. (2020b). The Arithmetic Optimization Algorithm. *Computer Methods in Applied Mechanics and Engineering*, *376*, 113609. https://doi.org/10.1016/j.cma.2020.113609

Abualigah, L., Shehab, M., Alshinwan, M., Alabool, H., Abuaddous, H. Y., Khasawneh, A. M., & Diabat, M. A. (2020c). TS-GWO: IoT Tasks Scheduling in Cloud Computing Using Grey Wolf Optimizer. In *Swarm intelligence for cloud computing* (1st Ed., pp. 127–152). Chapman and Hall/CRC. https://doi.org/10.1201/9780429020582-5

Abualigah, L., Shehab, M., Diabat, A., & Abraham, A. (2022). Selection Scheme Sensitivity for A Hybrid Salp Swarm Algorithm: Analysis and Applications. *Engineering with Computers*, *38*(2), 1149–1175. https://doi.org/10.1007/s00366-020-01067-y

Alguliyev, R. M., Imamverdiyev, Y. N., & Abdullayeva, F. J. (2019). PSO-Based Load Balancing Method in Cloud Computing. *Automatic Control and Computer Sciences*, *53*(1), 45–55. https://doi.org/10.3103/s0146411619010024

Al-qaness, M. A. A., Ewees, A. A., Fan, H., Abualigah, L., & Abd Elaziz, M. (2020). Marine Predators Algorithm for Forecasting Confirmed Cases of COVID-19 in Italy, USA, Iran and Korea. *International Journal of Environmental Research and Public Health*, *17*(10), 3520. https://doi.org/10.3390/ijerph17103520

Alsalibi, B., Abualigah, L., & Khader, A. T. (2021). A Novel Bat Algorithm with Dynamic Membrane Structure for Optimization Problems. *Applied Intelligence*, *51*(4), 1992–2017. https://doi.org/10.1007/s10489-020-01898-8

Alshinwan, M., Abualigah, L., Shehab, M., Elaziz, M. A., Khasawneh, A. M., Alabool, H., & Hamad, H. A. (2021). Dragonfly Algorithm: A Comprehensive Survey of its Results, Variants and Applications. *Multimedia Tools and Applications*, *80*(10), 14979–15016. https://doi.org/10.1007/s11042-020-10255-3

Altabeeb, A. M., Mohsen, A. M., Abualigah, L., & Ghallab, A. (2021). Solving Capacitated Vehicle Routing Problem Using Cooperative Firefly Algorithm. *Applied Soft Computing*, *108*, 107403. https://doi.org/10.1016/j.asoc.2021.107403

Ashouraie, M., & Jafari Navimipour, N. (2015). Priority-Based Task Scheduling on Heterogeneous Resources in the Expert Cloud. *Kybernetes*, *44*(10), 1455–1471. https://doi.org/10.1108/k-12-2014-0293

Bokhari, M. U., Makki, Q., & Tamandani, Y. K. (2018). A Survey on Cloud Computing. In V. Aggarwal, V. Bhatnagar, & D. Mishra (Eds.), *Big Data Analytics. Advances in Intelligent Systems and Computing* (Vol. 654, pp. 149–164). Springer Singapore. https://doi.org/10.1007/978-981-10-6620-7_16

Chen, W., Xie, G., Li, R., Bai, Y., Fan, C., & Li, K. (2017). Efficient Task Scheduling for Budget Constrained Parallel Applications on Heterogeneous Cloud Computing Systems. *Future Generation Computer Systems*, *74*, 1–11. https://doi.org/10.1016/j.future.2017.03.008

Eid, A., Kamel, S., & Abualigah, L. (2021). Marine Predators Algorithm for Optimal Allocation of Active and Reactive Power Resources in Distribution Networks. *Neural Computing and Applications*, *33*(21), 14327–14355. https://doi.org/10.1007/s00521-021-06078-4

Hayes, B. (2008). Cloud computing. *Communications of the ACM*, *51*, 9–11. https://doi.org/10.1145/1364782.1364786

Jiang, Y., Luo, Q., Wei, Y., Abualigah, L., & Zhou, Y. (2021). An Efficient Binary Gradient-Based Optimizer for Feature Selection. *Mathematical Biosciences and Engineering*, *18*(4), 3813–3854.

Kumar, M., & Sharma, S. C. (2018). Deadline Constrained Based Dynamic Load Balancing Algorithm with Elasticity in Cloud Environment. *Computers & Electrical Engineering*, *69*, 395–411. https://doi.org/10.1016/j.compeleceng.2017.11.018

Li, J., Zhang, X., Han, L., Ji, Z., Dong, X., & Hu, C. (2021). OKCM: Improving Parallel Task Scheduling in High-Performance Computing Systems Using Online Learning. *The Journal of Supercomputing*, *77*(6), 5960–5983. https://doi.org/10.1007/s11227-020-03506-5

Linthicum, D. S. (2016). Emerging Hybrid Cloud Patterns. *IEEE Cloud Computing*, *3*(1), 88–91. https://doi.org/10.1109/mcc.2016.22

Malik, M., & Suman. (2022). Lateral Wolf Based Particle Swarm Optimization (LW-PSO) for Load Balancing on Cloud Computing. *Wireless Personal Communications*, *125*(2), 1125–1144. https://doi.org/10.1007/s11277-022-09592-3

Manickam, M., & Rajagopalan, S. P. (2019). Retracted Article: A Hybrid Multi-Layer Intrusion Detection System in Cloud. *Cluster Computing*, *22*(S2), 3961–3969. https://doi.org/10.1007/s10586-018-2557-5

Mansouri, N., Javidi, M. M., & Mohammad Hasani Zade, B. (2021). A CSO-based approach for secure data replication in cloud computing environment. *The Journal of Supercomputing*, *77*(6), 5882–5933. https://doi.org/10.1007/s11227-020-03497-3

Mapetu, J. P. B., Kong, L., & Chen, Z. (2021). A Dynamic VM Consolidation Approach Based on Load Balancing Using Pearson Correlation in Cloud Computing. *The Journal of Supercomputing*, *77*(6), 5840–5881. https://doi.org/10.1007/s11227-020-03494-6

Nadjaran Toosi, A., Sinnott, R. O., & Buyya, R. (2018). Resource Provisioning for Data-Intensive Applications with Deadline Constraints on Hybrid Clouds Using Aneka. *Future Generation Computer Systems*, *79*, 765–775. https://doi.org/10.1016/j.future.2017.05.042

Safaldin, M., Otair, M., & Abualigah, L. (2021). Improved Binary Gray Wolf Optimizer and SVM for Intrusion Detection System in Wireless Sensor Networks. *Journal of Ambient Intelligence and Humanized Computing*, *12*(2), 1559–1576. https://doi.org/10.1007/s12652-020-02228-z

Shehab, M., Abualigah, L., Al Hamad, H., Alabool, H., Alshinwan, M., & Khasawneh, A. M. (2020). Moth–Flame Optimization Algorithm: Variants and Applications. *Neural Computing and Applications*, *32*(14), 9859–9884. https://doi.org/10.1007/s00521-019-04570-6

Sreenu, K., & Sreelatha, M. (2019). W-Scheduler: Whale Optimization for Task Scheduling in Cloud Computing. *Cluster Computing*, *22*(S1), 1087–1098. https://doi.org/10.1007/s10586-017-1055-5

Wickremasinghe, B., Calheiros, R. N., & Buyya, R. (2010). CloudAnalyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications. *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 446–452. https://doi.org/10.1109/aina.2010.32

Yuan, H., Bi, J., & Zhou, M. (2020). Profit-Sensitive Spatial Scheduling of Multi-Application Tasks in Distributed Green Clouds. *IEEE Transactions on Automation Science and Engineering*, *17*(3), 1097–1106. https://doi.org/10.1109/tase.2019.2909866