3D Res (2019) 10:12
https://doi.org/10.1007/s13319-019-0222-2

3DR EXPRESS

# LGSA: Hybrid Task Scheduling in Multi Objective Functionality in Cloud Computing Environment

**N. Manikandan · A. Pravin**

Received: 14 December 2018 / Revised: 15 February 2019 / Accepted: 24 February 2019 / Published online: 15 March 2019


**Abstract** Cloud computing turns to be a big shift from the conventional perception of the IT resources. It is a transpiring computing technology that is increasingly stabling itself as the promising future of distributed on-demand computing. The processes comprised in it are the ones that act as a vital backbone and which strengthen the entire stream of cloud computing as a whole. In specific, Task scheduling is the one such phenomena that enhances the cloud computing in terms of performance. Hence task scheduling that is considered as a predominant one amidst others is what this paper comprises all about. Maximizing the profit via assigning the whole task to the virtual machine is what the problem of scheduling deals with. Although there prevails many more ways to resolve this problem, this paper explores one such solution that consumes lesser number of resources, having lower cost and much importantly consuming lesser energy. By making a profound research regarding this approach of scheduling so as to represent the multi-objective function, both lion optimization algorithm and gravitational search algorithm are hybridized. In spite of having certain drawbacks which could be avoided although, the brighter side relies the merits of making use of both lion search and gravitational search algorithm. There could be many means of measurement for computing the performance of the algorithm. The different algorithms that aid to depict the comparable study encompasses gravitational search algorithm, genetic algorithm and lion, particle swarm optimization. The experimental results serve as the evident for depicting the bitterness of our proposed algorithm compared to the prevailing approaches. As an unexplored path may seem trivial but is effective so does the betterment of our lion approach.

**Keywords** Task scheduling · GSA · Multi objective function · LOA · Hybrid LGSA

## 1 Introduction

Cloud computing phenomena, amalgamated with exploring internet remote servers is considered to be flourishing paradigm. [1]. Cloud computing is phenomena developed to offer consumers with resources for exclusive application. As the advancements in cloud computing has seen great heights consumers can now get access to resources which will help them to access their personal files stored on any system with the aid of a network connection. The most common among the famous cloud is being the Google Drive [2].

N. Manikandan (✉) · A. Pravin
School of Computing, Sathyabama Institute of Science and Technology, Chennai, Tamilnadu, India
e-mail: macs2005ciet@gmail.com

N. Manikandan
St. Joseph's College of Engineering, Chennai, Tamilnadu, India

The applications of the cloud is on the increasing side and it has many optional features and enabled services such as virtualization, low cost, elasticity and pay per usage [3]. The customers have to pay based on their individual requirements. Cloud computing comes in with the capacity to offer services based on the ever changing user needs. In general clouds hold diverse resources that are distributed and heterogeneous. Clouds can be used both privately as well as publicly. Their utilisation in small and large enterprises depends upon their needs accordingly. Cloud computing is an advantage for the customers as they pay for the services they use. The services required by the user will alone be charged by the service providers [4]. The three different kinds of services provided by the cloud is as given below (1) Software as a service (SaaS), (2) Platform as a service (PaaS), (3) Infrastructure as a service (IaaS) [5, 6]. Reliability, high flexibility, scalability, cost efficiency, location and independence of software makes cloud computing to be accepted widely [7, 8].

The resources in cloud computing are managed with the aid of virtualization technology. This mechanism enables several virtual machines to run on a single physical machine (PM), which is the resource server [9]. Virtualization helps in distinguishing the resources related to hardware in a physical machine into several execution environments called virtual machines (VM). The benefits of virtualisation being the ease of administration, cost efficiency in maintenance, conservation in energy consumption which in turn leads to green computing. The ever increasing demand for virtualisation has resulted in the decline of quality of service (QoS). Thus VM migration has now become the feasible solution for all such problems. The accuracy in VM migration can result in proper handling of overload balancing and issues related to performance. VM migration has now become inexorable due its practical infeasibility in load condition of physical machine [10–12]. To ensure quality in thin cloud the proper trade-off between service-level agreement (SLA) and quality of servicing has to be enhanced. SLA is used as the medium for VM migration in cloud to make use of the available resources according to the needs of the user in general. It offers the owners of the cloud to option to combine host and switch off the host which is underutilized for the purpose of consolidation.

The efficiency of the cloud is decided by VM migration as it facilitates the user to access computing environment in a single underlying host PM and also enhances the accuracy by way of proper resource sharing. Several research studies have been undertaken in the field of virtual machine migration and host consolidation [6]. The problems that arise as a result of changing consumer needs is yet to be addressed by the advancements in the same field. The optimization algorithm is seen as a viable solution to remove the problems related to VM placement., the optimization algorithm is employed to track a moving optimum rather than just simply looking for the optimal solution [11]. The central issue related to VM migration based on optimization being controlling the resources, consumption of energy, cost of migration etc. To arrive at an optimal solution the local heuristics may not be enough and meta-heuristics has to be applied to effectively tackle such issues. Meta-heuristic is a frequent main process to display and change the functions of secondary-heuristics to provide better results. This page focuses on the methods deployed in the design of VM migration approach based on the resources that are used, cost of migration and the energy consumption. The primary objective being the establishment of a VM in order to enable the resources accordance needed to meet the requests made by the loaded PM to the unloaded PM. The VM placement is carried out based on the parameters of resource usage, energy consumption and migration sot to ensure optimal performance of the cloud [13].

Cloud computing is not just another typical practice in the world of Information Technology, but it is rather an advancing revolution. Grid Computing, Utility Computing and dispersed frameworks are directly associated with Cloud computing. It can be emphasised that framework processing is the spine of Cloud computing. Virtual assets and administrations are provided by Cloud computing with the goal of diminishing infrastructure expense. Due to its capability to provide virtualization and reflection, Cloud computing is realised and is generally popular [14]. Cloud computing is a paradigm which is developing rapidly in compliance with clients demanding services that yield better results and effective performance. Hence, load balancing has developed as a sensational and noticeable research area. It requires a concerned attention to have the nodes in a cloud environment a balanced load and this task of allocating the cloud tasks is assigned to cloud computing. This strategy is called load balancing. However in cloud computing, the process of load balancing emphases to enhance resource consumption so as to get the highest throughput, to have the quick response time and also to restrict the overload of any single resource, thereby the process exhibit strong influence towards the performance. Better load balancing makes cloud computing more efficient and boosts user satisfaction. Therefore, "it is the process of confirming the even distribution of work load on the pool of system nodes or processors so that the running task is accomplished without any disturbance". The objectives of load balancing are to enhance the performance and thereby improve the stability of the system, construct a system which is fault-tolerant and provide upcoming variations in the system such as security updates, revealing the customer time and the resources which could be used for the future tasks. Cloud load balancing is one of those processes that can be executed in cloud computing which can be completed separately as well as on a group basis. There are several algorithms designed for dispensing the load to diverse tasks. After finishing the literature survey, it can be concluded that many of these algorithms recommended so far are too complex. However in case of Round robin scheduling algorithm, it considers only the existing load on each virtual machine. This is a fixed method of load balancing. [15] Static load balancing methods propose the simplest simulation and examining of environment but fail to model the heterogeneous nature of the cloud. Another algorithm known as the throttled algorithm is completely based on virtual machines. In this algorithm, the order goes as: client initially requests the load balancer to inspect whether the correct virtual machine accesses that load and implements the operations which is given by the user or the client. The Escel algorithm says that the load balancer is important for checking of jobs which request execution. The responsibility of load balancer is to line up these jobs and assign them to different virtual machines. The balancer regularly looks over the queue for new jobs and then allots those jobs to the list of free simulated servers. The list of jobs that are allotted to virtual servers is also kept track of by the balancer, which supports them to recognise which virtual machines are free and need to be allocated fresh jobs. The name Escel proposes that this concept works on equally spreading the execution load on different virtual machines.

In this paper, we suggest the combination of gravitational search algorithm and lion optimization search algorithm which serves as a multiple objective task scheduling. This model can reflect the demands of the tasks for the resources in detail. The multiple objective functions that is used in this paper is cost, energy, resource utilization, the scheduled task is also based on the multiple objective function. When using lion search and gravitational search algorithm, it's easy to fall into a local optimum therefore, this paper proposes the idea of a hybrid LGSA algorithm that can evaluate and that could adjust the quality of solution in order to avoid falling into that local optimum. Main novelty of this paper is to desire the multiple objective functions.

### Contribution of Work

1. There is an approach known as the LGSA which is done for task scheduling, which is the hybrid of the lion optimization algorithm. The LGSA algorithm was proposed to resolve the multi-objective optimization scheduling problem. Here in the hybridization optimal and sub optimal solution with maximum profit is being obtained.
2. The multi-objective optimization model proposed in this paper. This problem controls the process of overloading and under loading problems. Here multi objective optimization can be optimal cost, energy and resource utilization is achieved.

This paper is arranged into various sections. In the Sect. 2 it provides the details about the reviews taken for the existing system problems and the details are furnished as the literature survey paper given below. In the Sect. 3 it encloses the explanation about the problem statement which tells about the problem which needs to be resolved in the proposed work. Also it says about the framework that is used in the proposed model. In Sect. 4 it provides information about the discussion part and the outcome obtained in the proposed model. In Sect. 5 it gives a brief explanation about the conclusion part which says about the purpose using the various models in the proposed system.

## 2 Literature Survey

In a cloud environment, the number of job and the available sources Increases fastly, especially when virtual sources (VMs) should be considered. By increasing the number of jobs and sources, the entire favourable job are being found source mappings increases exponentially, and the selection of an optimal job is based on problem of energy, resource utilization and time consuming. To overcome this parameter we go for the literature survey papers [16–19].

The drastic growth of technology in cloud demands increased number of the services and tasks that are provided by it in the virtual machine [20]. To solve these problems many algorithms have been proposed. All these algorithms are tested on different situations, compared with various stimulation results and existing algorithms. Through this process the best of all is chosen as the algorithm. A new algorithm called as heuristic-based load balancing algorithm (HBLBA) is proposed here. This algorithm consists of two phases called as server configuration and task VM—mapping. This is considered to be a very efficient algorithm. HBLBA adopts the concept of queuing model through which tasks are scheduled on time. Performance comparison is done with various parameters such as makespan, waiting time, scheduled length ratio, VM utilization, CPU utilization. The performance results shows better efficiency than the existing algorithms.

Maintaining quality and other parameters is a challenging problem [21]. To solve this problem and effective dynamic scheduling algorithm is introduced. This algorithm balances the load on all virtual machines with elastic resource provisioning. Algorithms are further checked on various values to reach better scalability. There are two main steps included in this load balancing algorithm: task scheduling and monitoring the virtual machine. Virtual machines are monitored to check if they perform the former operation. The introduced algorithm is designed such that even when the scope of the problem is increased it fits. Effective dynamic scheduling algorithm also ensures elasticity in cloud environment. Using optimal k-interval values, the algorithm starts to degrade when the value of k is more than 15. Experimental proofs are used to show the comparison and working of the cloud environment. Better elasticity and reduced rejection ratio results have been proved with the load balancing algorithm. Various other parameters can be extended using this model to ensure high priority requests.

Scheduling problems are addressed in multi cloud systems [22]. Load scheduling is the majorly faced problem in cloud. Hence designing an algorithm which considers multiple intrinsic characters is challenging and important. A dynamic scheduling strategy (DSS) is introduced that groups all the load theory and techniques. Intensive experiments are conducted to check DSS algorithm and evaluate the performance efficiency many large scale applications have been migrated for processing big amount of data. Multi cloud systems solve heavy cloud requirements. Scheduling multiple division loads with arbitrary release is done using this architecture. Rigorous stimulation experiments are performed to test this architecture. More than the baseline scheme the stimulation experiments perform best. The results also provide the applicability of the proposed architecture and there are various range of cloud based realistic systems. These systems are further into the designing of the sophisticated strategies of handling the big data.

Cloud computing is the most blooming high performance computing environment [23]. Multiple resource management techniques are used to improve the efficiency of the full cloud environment. The most significant part of computing the cloud is resource scheduling. Optimised scheduling of these jobs on the most appropriate virtual machine is considered as the NP hard problem and many other algorithms have been presented to solve it. Since work flows have arose as a complexity scheduling of these work flows is planned in the algorithm. A unique algorithm which prolongs the natural based intelligent water drops (IWD) algorithm. This enhances the scheduling of work flow in the cloud. Meta heuristics are used in this study for the scheduling. The various scheduling algorithm MINMAX, round robin results were compared with the original algorithms to optimize the result value. This comparison shows table behaviour and costs of work flow. Heterogeneous VMs are used to make preference over any other machine. Improvements over algorithms are made to optimize the solution and include various performance objectives with the cloud computing and energy usage of resources.

Cloud computing trend offers utility related services [24]. Cloud computing has recently been extensively used and adopted by healthcare industries due to the cost nature. Medical services to users are adopted using the cloud tool. This tool acts as a game changer in most of the operations of the health care industry. In the cloud computing, the process of balancing the workload among multiple servers and application include scaling, routing traffic. The need to improve the utility of cloud resources the balancing approach is used. Hence this balancing is necessary. A hybrid fruit fly optimization technique is used based on load balancing. Various experiments are compared to the original FOA with the Hybrid FOA. Compared values produce optimum solution for balancing. The FOA–SA–LB uses the sleeping strategy to reduce the energy consumption. Improvisation to the algorithms are done for increasing the efficiency of resources.

Cloud computing is the computing capability that provides platform, hardware infrastructure and software applications as mode of service [25]. It reduces the investment on these resources drastically by offering Service Level Agreements (SLA's) to user's required Quality of Service (QoS) as it is hosted in service or third party infrastructure. This providing of cloud is categorized into public, private and hybrid cloud based on the usage of customers and habitually as user and system level. An algorithm named hybrid self-adaptive learning global search algorithm (HSLGSAFA) and firefly algorithm (FA) is done to realise and converge task scheduling for optimization. This method replaces worst solution to improve quality. Self-adaptive learning gravitational search algorithm (SLGSA) is hybrid to gravitational search algorithm velocity updating stage. Scheduling enhances server and resource performance of the system so HSLGSAFA represents a task and solution as whole by using SLGSA and FA to schedule task from DCTS problems and for enhancing certain difficulties.

Cloud computing which acts as the platform and software for users, services of infrastructure and supplies the on-demand services to users through Internet is emerging as one of the potential research directions following utility computing [26]. It employs physical hosts to meet the needs of the supply user services. The task cannot be assigned to physical host as the resource amount is high and hence it is assumed that the task from users deploy to physical host randomly. This may cause deployment failure if the amount incurred for submitted resource is greater than the amount for the remaining resource imbalance may occur due to repeated request for tasks. The data centre assigns tasks to highest load demands utilising the physical host, remain idle comparatively wasting the computing resources. Selection of appropriate physical host is necessary for load balancing in data centres. There is a limitation for the overly concentrating on the optimal load balancing policy for the current deployment problem and thus it decreases the efficiency and users waiting time is prolonged without any purpose. The satisfactory service performance can be obtained by using available amount of computing resource in data centres, larger than the requested amount though this kind of requirements is more realistic and firm. A heuristic approach is deployed to find the optimal physical hosts for tasks deployment by accomplishing a strategy using long-term algorithm process and thereby obtain optimal performance by the allocation of each requested task a value which is constraint, and where the computing resources are found to be higher than the constraint value of the tasks in the cloud data center The deployment services would then be carried out by services offered externally. Load Balancing based on Bayes and Clustering (LB–BC) is done by the load balancing strategy which is proposed.

These days cloud computing has made rapid development as technologies emerge commercially [27]. This enables user to pay for service they need and get the resources. This technology involves virtualization techniques. Factors such as resources, software, and hardware can be virtualized in cloud computing platform. Research of task scheduling is introduced by genetic algorithm, particle swarm algorithm, and simulated annealing and colony optimization algorithms. Genetic algorithm is especially hard because of its large number of Parameters. Biological social background is very good in particle swarm optimization and has strong global search capabilities for problems in nonlinear and multimodal techniques. This paper is about the enhanced binary particle swarm algorithm to adapt to cloud task scheduling and reduce resource consumption. GCTA simplifies the matrix to piping number and redefines velocity and position is cloud and solves the problem in using BPSO. The results can be made local optimal or global optimal solution by numbering after sorting. The future is expected to turn in the direction of virtual machines memory and bandwidth on execution time.

From the literature review, most of the scheduling and load balancing problems are solved by using hybrid meta-heuristic algorithms. The hybridization algorithms have demonstrated powerful results when compared with standard algorithms. GSA has successfully resolved the problem of slow searching speed in the last iterations. The factors like, faster convergence rate and little controlling parameter inspires the usage of LOA. These advantages of the two algorithms namely GSA and LOA motivates the usage of hybridization of both algorithms which further leads to achieve near optimal solution for scheduling and load balancing problems.

## 3 Defining the Problem and Offering Solution Framework

The important objective of the proposed method us optimise the task cost efficiently, using low memory and conserving energy. This paper suggests the scheduling in a parallel method i.e., all the tasks performed simultaneously. Scheduling is by itself a prominent in controlling task within the cloud. The amount of resources needed to complete the task is approximated by the Scheduling process and it also settles upon the task that should be allocated to particular computing component. Before processing the subtasks in parallel sequence they can be divided into smaller subtasks. By splitting a computation into smaller subtasks and implementing these subtasks on various processors, the overall advantage of the implementation is increased. Also, the aim of the task scheduling algorithm is not applicable to schedule the complete task into the available processor in order to widen the profit (profit here indicates the combo of low cost, low memory utilization and energy conservation) with affecting the principal necessities. It is a challenge in task scheduling. To traverse these hardships present in the scheduling process in this job we perform the optimization approach the given task contains a number of extremely parallel and self-deciding derived tasks.

By splitting a computation into smaller subtasks and implementing these subtasks on various processors, the overall advantage of the implementation is increased. Also, the aim of the task scheduling algorithm is not applicable to schedule the complete task into the available processor in order to widen the profit (profit here indicates the combo of low cost, low memory utilization and energy conservation) with affecting the principal necessities. It is a challenge in task scheduling. To traverse these hardships present in the scheduling process in this job we perform the optimization approach the given task contains a number of extremely parallel and self-deciding derived tasks.

The execution of each task has to be done in a single VM instance type. If pm = {$pm_1$, $pm_2$, … $pm_n$) is a amalgamation of physical machine. Vmi = {$Vm_1$, $Vm_2$ … $Vm_I$} which is the sequence followed in a virtual machine VMI types and T = {$T_1$, $T_2$ … $T_m$} is defined as the set of task. Every task considers a set which is derived, Ti = {$t_1$, $t_2$, … $t_n$}. The unique cost of every task be $C_1$, memory $m_i$, and energy $E_i$. Problem factors are defined as a solution to this problem. In this paper, the utmost function depends on three factors such as cost, energy consumption and memory usage. Here each task has individual moving charge. To put this problem in a formula, control factors are defined. The problem is formulated below.

In this paper, the utmost function depends on three factors such as cost, energy consumption and memory usage. Here each task has individual moving charge. To put this problem in a formula, control factors are defined. The problem is formulated below.

$$Objective\ function = \sum_{a=1}^{n} P \times \text{Cost} + Q \times Energy\ consumption + R \times Resource\ utilization \tag{1}$$

$$Cost = \frac{1}{Physical\ machine_i} \sum_{a=1}^{n} \left( \frac{No\ of\ evolving\ in\ virtual\ machine}{Total\ no\ of\ virtual\ machines} \right) \tag{2}$$

$$Energy\ consumption = \frac{1}{Physical\ machine \times virtual\ machine} \left[ \sum_{a=1}^{Physical\ Machine} \sum_{b=1}^{Virtual\ Machine} \rho_{ab} E_{Maximum} + (a-N)\gamma_{ab} E_{Maximum} \right] \tag{3}$$

$$\gamma_{ab} = \frac{1}{2}\left[\left(\frac{CPU\ utilized_{ab}}{CPU_{ab}}\right) + \left(\frac{Memory\ utilized_{ab}}{Memory_{ab}}\right)\right] \quad (4)$$

$$Resource\ Utilization = \frac{1}{Physical\ machine \times virual\ machine}\left[\sum_{a=1}^{Physical\ Machine}\sum_{b=1}^{Virual\ Machine}\frac{1}{2}\left(\frac{CPU\ utilized_{ab}}{CPU_{ab}}\right) + \left(\frac{Memory\ utilized_{ab}}{Memory_{ab}}\right)\right] \quad (5)$$

In Eq. 1 the major aim of our research is highlighted concerning the first term of (1) it indicates the cost of the job and the second term specifies the energy consumption and the third term specifies the resource consumed. Equation 2 depicts the formula for cost. The cost should be reduced in order to attain a way good profit Eq. 3 gives the formula for energy utilized. This equation describes the extent of energy utilized in this process. Equation 4 gives the formula for energy that is being used. The definition states that the challenge of the job as scheduling. The problems that is inclusive of a numerical programming method are considered, it consequently ends up in a lot of computational time for a large sized problem discussed in this paper. The major objective of this paper is to reduce those problems mentioned above.

### 3.1 Proposed LGSA Based Scheduling Approach

Task based on LGSA algorithm are scheduled using the proposed scheduling approach. This hybrid optimization algorithm is compiled between the Lion search algorithm and gravitational search algorithm.

Then LS and GSA is a current algorithm and has a parcel of preferred standpoint. The main drawback of LS technique is the iterations to find an optimal solution and moreover it takes maximum time to derive at an optimal solution. GSA has the disadvantage that it's union speed backs offing the later inquiry stage and it is difficult to fall into the local optimum solution. The approach uses the upsides of the gravitational search algorithm and Lion search algorithm thereby avoiding their disadvantages. Hybridizing both algorithms, their weaknesses are overcome and certain benefits the effectively the acknowledging and rapidly focalizing with the goal that the planning methodology can get an idea or imperfect arrangement in a shorter computational time. The outcome segment demonstrates that the proposal streamlining of LGSA accomplished preferable streamlining execution over the individual advancement (Fig. 1).

### 3.2 Lion Optimization Algorithm

Describe the main operator of Lion Algorithm in a way where Mating refers to deriving new solutions and Territorial Defence and Territorial Takeover intend to find and replace the worst solution by the best solution. Similar to Lion Algorithm, Lion pride optimizer is established on fighting between individuals and mating. Not only do lions mate and fight, they also exhibit other behaviour such as the one unique way of capturing their prey, marking of territory and migration which on the whole depict the difference between life style of nomad and resident lions. Thus, the projected algorithm is inspired by the simulation of the self-contained and cooperative traits of lions which are completely different from the previous algorithm.

The LSA is a population-based meta-heuristic algorithm in which the first step is to arbitrarily generate the population across the solution space. In this algorithm, every single solution is referred to as "Lion". In a N var dimensional optimization problem, a Lion is represented as follows:

$$Lion_x = [X1, X2, X3, \ldots Xvar] \quad (6)$$

Cost (fitness value) of each Lion is c by evaluating the cost function, as:

$$Fitness\ value\ of\ Lion = Lion_x = (Lion) = F(X1, X2, X3, \ldots Xvar) \quad (7)$$

#### Hunting

In each pride the food required for the pride is searched by some females in the group and they provide food for the pride. Certain strategies are adopted by the hunters to trap the prey and capture it. Standard divides the roles adopted by lion into seven different stalking roles, and they are classified into Left Wing, Centre and Right Wing positions. Each lioness changes its position based upon the relative position

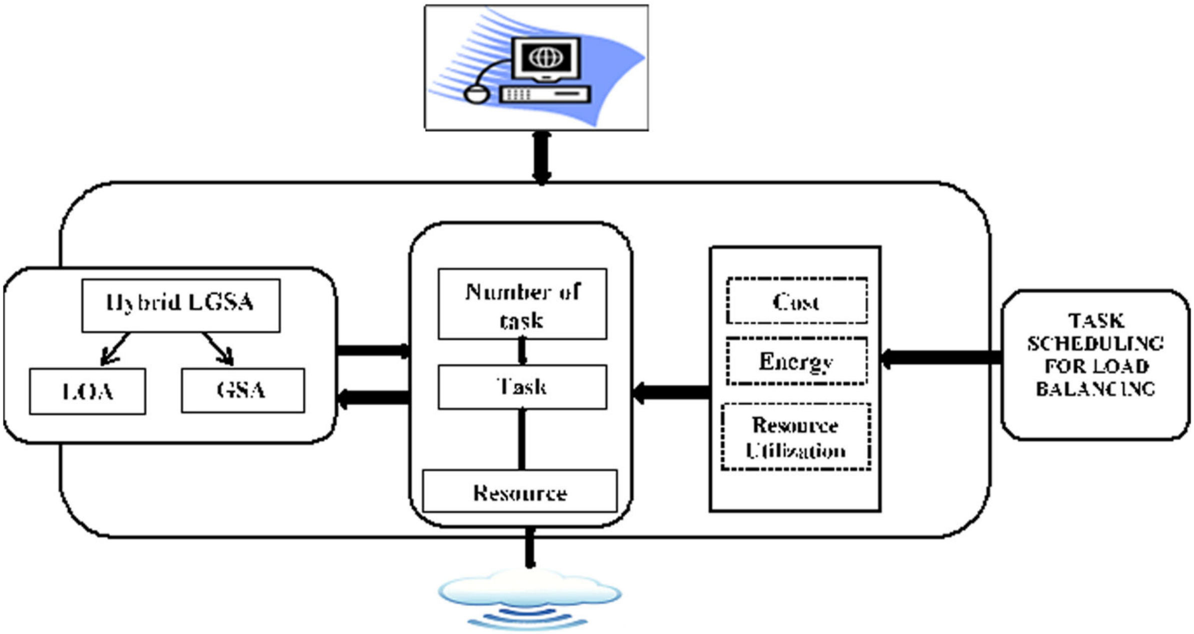


**Fig. 1** Proposed hybrid LGSA architecture

of the other members of the pride. It is inferred from the facts given above; predators are normally divided into three random sub groups. The group which encompasses the highest cumulative skill of the pride is considered as center whereas the other two groups are considered wings. An Assumed prey ($Prey_X$) is considered to be at the centre of hunters.

$$\left( Prey_X = \sum \frac{Hunters(X1, X2, X3, \ldots Xvar)}{No\ of\ hunters} \right) \tag{8}$$

The hunters are chosen randomly one after another and the hunters selected from group pounces on the dummy prey. This process would be defined after a while based upon the group the hunting lion belongs to. If the ability and subtlety of the hunter is enhanced in the due course $PREY_X$ will escape from hunter and new position of $Prey_X$ is obtained as follows:

$$Prey_Y = Prey_X + Random(0,1) \times p_i \times (Prey_X - Hunter) \tag{9}$$

where $Prey_X$ is the present position of prey, hunter attacks the prey in the new position and pi is the percentage of enhancement in the skill of the hunter. The formulae are devised to imitate the encompassing the prey mentioned in the hunter groups. The new positions of hunters which belong to the left and right wing are produced as follows:

$$Hunter_i = \begin{Bmatrix} Random((2 \times Prey_X - Hunter), Prey_X), \\ Random(Prey_X, (2 \times Prey_X - Hunter)), \\ (2 \times Prey_X - Hunter) < Prey_X \\ (2 \times Prey_X - Hunter) > Prey_X \end{Bmatrix} \tag{10}$$

where $Prey_X$ is the new location of the prey, Hunter the current position and Hunter's is the new location of hunter. The new position of the center Hunter is produced as follows:

$$Hunter_i = \{Random(Hunter, Prey_X), Random(Prey_X, Hunter), Hunter\langle Prey_X, Hunter\rangle Prey_X, Prey_X\} \tag{11}$$

The random numbers generated in the above equations rand (a, b) which lies between 'a' and 'b', in which, the 'a' and 'b' are upper and lower bounds, respectively.

**Roaming**

Each male lion in the pride roams in the territory of that pride for some reasons. Toimitate this behaviour of resident males, %P of pride territory is selected at random such that it is visited by that lion. While roaming, if resident male visits a new spot that is considered better than its current best spot, his top visited solution is updated. This roaming is a robust native search and supports Lion Optimization Algorithm (LOA) to search for a solution to improve it.

**Mating**

It is a vital process that not only ensures the survival of the lions, but also offers an opportunity for sharing of information among members. In every pride, %Na of female lions mates with one or many resident males. These males are selected randomly from the same pride by the female to produce offspring. For nomad lions it is different in that a nomad female only mates with one of the males which are selected arbitrarily. The mating operator is a linear combination of parents giving rise to two new offspring

**Defence**

When a male lion in a pride is matured, they turn aggressive and contest other males in their pride. Once the lion is beaten, it leaves its pride and becomes a nomad. However, if a nomad male lion is stout enough to take control of a pride by fighting its males, the beaten resident male lion is driven out of the pride and converts a nomad. Defence operator in LOA divided into two main steps:

1. Defence against new mature resident males.
2. Defence against nomad males.

### 3.3 Lion Optimization Algorithm

1. **Initialization**
   Number of Lions, VMs and Iterations are a Set value of parameters. Arbitrary solution for each Lion is created and pride and Nomad lions are initiated.
2. **for each Pride**
   Arbitrarily select females for hunting
   Move remaining females toward best spots of territory.
   Roam every male %R of the territory.
   Mate one or more resident males with %M of females.
   Drive out the feeblest male from the pride.
   Become Nomad.
3. **for each Nomad lion**
   Move male and female unsystematically in the search space
   Mate %M of females with only one male
   Attack prides with Nomad males
4. **for each pride**
   %I(Immigrate) of females Become Nomad
5. **Do**
   Sorting of the Gender of nomad lion is done based on their fitness value
   The best females are selected and distributed to prides filling empty places
   Nomad lions with least fitness are removed based on the max acceptable quantity of each gender.
6. **If (t < Iterations)**
   Proceed to second step

### 3.4 Gravitational Search Algorithm

In GSA all objections, comprehends the areas and situations of the opposite objects by the gravitational force, At GSA, we can conclude the answer by determining the stance of every specialist at the location (for the operator). In a random orders the items are put in search space, during the principal, at every objection, the mass estimation is resolved by the fitness value, given in Eq. (11)

$$H(\mathrm{x}) = H_0 e^{\left(-\frac{\alpha x}{R}\right)} \tag{12}$$

$$p_{j=ff_i} - \frac{worst(X)}{best(X) - worst(X)} \tag{13}$$

$$\mathrm{P_j(x)} = \frac{\mathrm{p_{j(x)}}}{\sum_{\mathrm{i=1}}^{\infty} \mathrm{p_i(x)}} \tag{14}$$

The increasing speed or a specialist is processed by adding powers from an composition or heavier masses, because, it has to be viewed as the low or gravity, that is travelled calculation or agent acceleration which uses the low of moment as in (13).After a brief period of time the following speed of an agent is ascertained as a minimum amount of its present speed added to its acceleration using (14).At that mark, its stance at the location can be calculated by using(15).

$$d_i^j = \sum_{j \in gbest, j \neq i} Ram_j H(s) \frac{P_i(x)}{E_{ij} + \in} \left( S_j^d(x) - S_i^d(x) \right) \tag{15}$$

Thus, we have

$$A_i^d(x+1) = Ram_i \times A_i^d(x) + a_i^d(x) \tag{16}$$

$$B_i^d(x+1) = B_i^d(x) + A_i^d(x+1) \tag{17}$$

H(x) is represents the gravity constant, $E_{ij}(x)$ is the Euclidian distance between two agents I and j, $\in$ represents a small value $S_i$ and $S_j$ are two random numbers in the interval [0, 1], that guarantee the stochastic characteristics of the algorithm.

### 3.5 Gravitational Algorithm of Search

1. Identification of Search Space
2. For i = 1, 2, … N Randomly initialize X, = (x, x}, …, X.
3. Agents are evaluated for Fitness.
4. Update worst(t), best(t), Kbest, M(t)
5. Choosing centre of well by a Probabilistic procedure is used to choose
6. for I = 1, … N Update agent's position to yield X/(t + 1).
7. For stopping criteria repeat steps c to g.
8. End of the program.

### 3.6 Proposed Hybrid Algorithm LGSA

***Input:***

Number of task Tm, Number of subtask TN, Number of physical machines PMn, Number of virtual machine VMi, Parameters of lion algorithm, Parameters of gravitational search algorithm

***Output***:

Optimized scheduled work from **LGSA**

1. ***Initialize***
   a. Assign parameter values for Total Lions, Virtual Machines, and Repetitions
   b. Solution produced for every Lion are random
   c. Initialize Prides and Nomad Lions
2. ***For every Pride***
   a. Hunting carried out through random selection of female Lions.
   b. Un-selected females start moving to appropriate locations of the region.
   c. Every male wanders in R% of region.
   d. M% of female lions carries out mating with at least one of non-roaming residential male.
   e. Unfit Male lions move out of pride and turn out to be nomad.
3. ***For Nomad lion***
   a. Female and Male lions wander around randomly in their region
   b. M% of female lions carries out mating with exactly one male
   c. Wandering nomads begin to attack prides.
4. ***For every pride***
   a. I% of female lions move out from pride position and become nomad.
5. ***Do***
   a. Every male and female lions under nomad category are sorted according to their fitness score.
   b. Female lions faring above fitness score are selected and disseminated to prides, filling out the empty positions.
   c. Lions faring below fitness score are taken out in accordance to maximum permissible count under each gender
6. ***If (t < Repetitions)***
   Go to step 7.//***Hybridization of lion and Gravitational search algorithm***.
7. Identify the search domain.
8. Initialize randomly values for $X_i = (x_{i,}^1 \ldots)$, for i = 1, 2, … N
9. Evaluate fitness score for participants.

10. Based on fitness score obtained designate Kbest, best(t), worst(t), M(t).
11. Highly fit centre score is chosen according to probability.
12. Participants are reorganized to produce $A_i^d(x+1)$ for i = 1, 2 … N.
13. Iterate thru 9 to 13 till satisfying condition is reached.
14. Stop.

**Fitness Function Manipulation for GSA**

The fitness function is used to reduce the cost, energy consumption and resource of all the tasks. The best way in performing this function is by using a number of iterations.

$$Fitnes\,function = \sum_{a=1}^{n} T_a (P * COST + Q * ENERGY + R * VOLUME) \quad (18)$$

**Updating the Solution with GSA Algorithm**

Next to the fitness function, the solution is to be updated with GSA algorithm.

For the net (x + 1)th iteration, the velocity is calculated using following equation.

$$A_i^d(x+1) = Ram_i \times A_i^d(x) + a_i^d(x) \quad (19)$$

$$B_i^d(x+1) = B_i^d(x) + A_i^d(x+1) \quad (20)$$

Similarly the position of the next $(x + 1)$th iteration can be calculated using,

Here, $Ram_i$ is the Random number between [0, 1].$A_i^d(x+1)$ is velocity of $i$th particle at $d$th dimension at ith repetition. $B_i^d(x+1)$, position of the ith particle at $d$th dimension at $x$th iteration.

**Cessation Criteria**

The method stops its operation when maximum cycles is accomplished and the arrangement which has the best fitness value is shortlisted and marked as the best component to the task scheduling. The point where the best fitness is accomplished with the assistance of the LGSA strategy, and that picked assignment is distributed for the cloud computing methodology. The LGSA subordinate task scheduling strategy Pseudo code and proposed LGSA algorithm (Fig. 2).

**Hybridization**

While comparing the GSA and LION, if GSA has the best fitness value and LGSA results in a lesser value than the LION fitness value, then the LGSA's best position of the GSA is replaced by that of the LION. Else, if the LION fitness value is less than the GSA fitness value, then the position is replaced by LION solution.

## 4 Results and Discussion

This section discusses the result obtained from the proposed task scheduling based technique referred as LGSA algorithm using Java (jdk 1.6) with cloudSim 3.0 tools. A series of experiments were performed on PC with Windows 7 OS, at 2 GHz dual core, and with 8 GB main memory running a 64-bit version of Windows 2007.

**Performance Study**

The basic concept of our research algorithm is multi objective-based task scheduling using line and gravitational search algorithm. The performance of target is mainly evaluated using profit, cost, and energy. The experimental results are made from study using three different configurations such as

$$\text{(i) PM} = 5 \text{ and VM} = 20,\ \text{(ii) PM} = 10 \text{ and VM} = 30,\ \text{(iii) PM} = 15 \text{ and VM} = 45$$

Proposed LGSA algorithm formed on multi-objective scheduling is set in difference with GSA, LION, PSO and GA. In finding, a gravitational search algorithm, GSA works on the principle Newtonian gravity; cuckoo bird behaviour, a principle that enables lion work; and bird's behaviour PSO, is a particle swarm optimization algorithm. Two operations: velocity updating and position updating are available in the PSO. Similarly, GA is an efficient optimization algorithm with two operations: mutation and crossover. All four algorithms got some problems. Hence, with the aim to improve the difficulties present

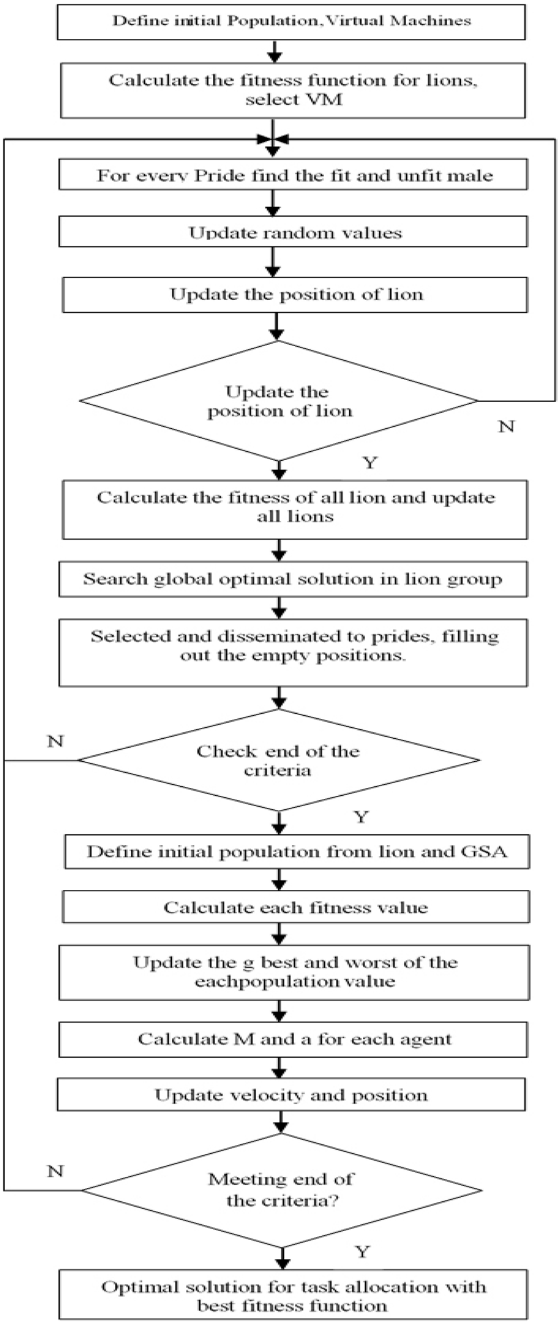


**Fig. 2** Proposed LGSA Flow Chart

in the individual LION and GSA, we hybrid LION algorithm with GSA algorithm. The Fig. 3 shows the performance of proposed approach based on a profit function. The good system has the maximum profit like minimum cost function, minimum energy consumption, and less resource use. On analysing Fig. 3,

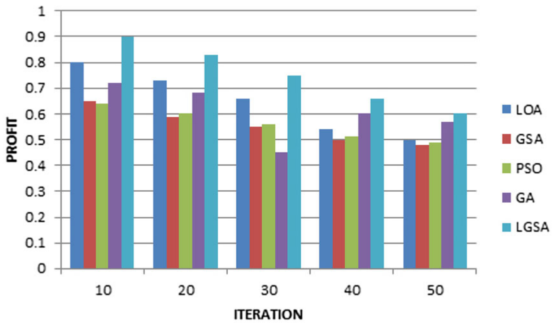


**Fig. 3** Performance analysis of proposed against existing using profit

our proposed approach obtained the maximum profit out of 0.8 and it is only 0.73 for using GSA, 0.65 for using LION, 0.7 for using PSO and 0.64 for using GA and it revealed that LGSA has the highest profit rate. Figure 4 shows the performance study of the proposed algorithm against existing one using cost function. Here, we analysed the cost of requirement for task scheduling and found that our algorithm achieved a minimum cost of 0.011$ when compared to all the other iterations. Figure 4 show the performance analysis of the proposed algorithm against existing one using energy function, FOR LGSA it is only 0.04$ where all the other iterations has the higher cost rates. Here, we the amount of energy utilized is analysed to the scheduling the task. When analysing Fig. 5, our proposed approach utilized minimum amount of energy compare to other approaches. Thus it is proved that our proposed algorithm achieved better results compared to other results. Comparing all the iteration with LGSA, minimal cost production is executed only in LGSA. When energy consumption is less, resource

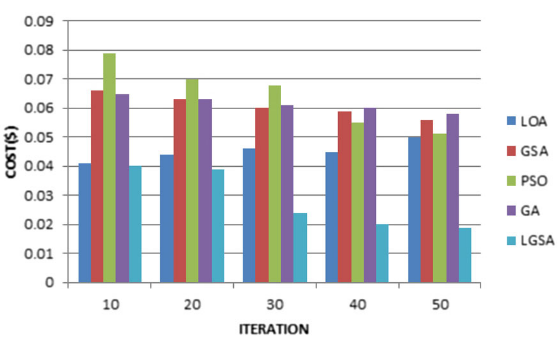


**Fig. 4** Performance analysis of proposed against existing using cost

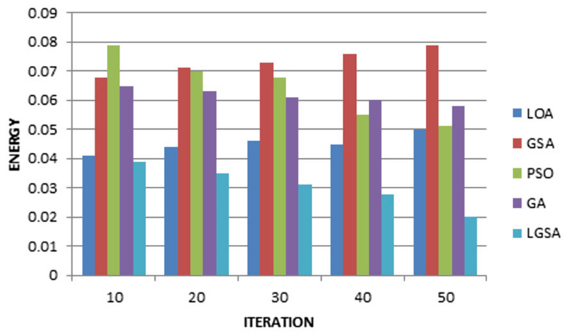


**Fig. 5** Performance analysis of proposed against existing using energy

utilization is high and this proves to be a great advantage in LGSA.

The Fig. 3 shows the details about LOA, GSA, PSO, GA, and LGSA in terms of points. For the ration 10, the results of LOA, GSA, PSO, GA, and LGSA are 0.8, 0.65, 0.64, 0.72, 0.9 and for the ratio 20, the findings of LOA, GSA, PSO, GA, and LGSA are 0.73, 0.59, 0.6, 0.68 and 0.83 respectively. The variations in LOA, GSA, PSO, GA, and LGSA for the ration 30 are 0.66, 0.55, 0.56, 0.45, 0.75 and the ratio 40 the readings are 0.54, 0.5, 0.551, 0.6, 0.66 distinctly. The decree for the ration 50 for the iteration LOA, GSA, PSO, GA, and LGSA are 0.5, 0.348, 0.49, 0.57, 0.6. By comparing all the iterations with the ratios, the LGSA has produced better results above the all and the below bar chart explains the same in terms of x axis as iteration and y axis as profit.

Figure 4 displays the details, given in terms of iteration and ratio. With the ration as 10, 20, 30, 40, and 50 with the iteration as LOA, GSA, PSO, GA, and LGSA, the cost measures are 0.041, 0.066, 0.079, 0.065, 0.04 for the ratio 10, 0.044, 0.063, 0.07, 0.063, 0.039 for the ratio 20, 0.046, 0.06, 0.068, 0.061, 0.024 for the ratio 30, 0.045, 0.059, 0.055, 0.06, 0.02 for the ratio 40 and 0.05, 0.056, 0.051, 0.058, 0.019 for the ration 50 respectively. Comparing the ratio with the iterations of LOA, GSA, PSO, GA, with LGSA, the cost value for LSGA is less appropriately. These findings are also given in the chart to have a detailed clarification on the same cost value.

The consumption of energy has been variantly low of LGSA when compared with LOA, GSA, PSO and GA. For give a detail energy output in terms of ration, the LOA consumes an energy of 0.042, 0.045, 0.047, 0.046, 0.051 for the ratio 10, 20, 30, 40, and 50. The energy consumption of GSA are 0.068, 0.071, 0.073, 0.076, 0.079 and for PSO are 0.079, 0.07, 0, 0.068, 0.055 and 0.051. For GA, the consumption of energy is 0.065, 0.063, 0.061, 0.06 and 0.058. Comparing all these iterations with LGSA for the ratios, 0.039, 0.035, 0.031, 0.028 and 0.02. All these findings prove that LGSA consumes less amount of energy comparing all the other four. The same result is also given in bar chart to have a better clarification on the consumption of energy.

**PM = 10 and VM = 30**

10 physical machine and 30 virtual machines were used in our plan to reach our required task. Figures 5, 6 and 7 shows the functional process of the implemented approach using this configuration. The above Figs. 6, 7 and 8 shows the performance of implemented procedure based on parallel machine scheduling using PM 10 and VM 30. Figure 6 shows the Performance analysis of implementation against present scenario using profit. Here, the x-axis displays the iteration and y-axis displays the profit. When analysing Fig. 6, our proposed implementation aim is to achieve the maximum profit of 0.91 which is 0.66 for using GSA algorithm based scheduling, 0.46 for using lion based scheduling, 0.6 for using PSO based scheduling and 0.65 for using GA based scheduling. Figure 7 shows the functional analysis of implementation against present using cost function. Here, our key concept is to approach utilized minimum cost to achieve the goal. In case if scheduling has used minimum cost means we can get the maximum profit. Figure 8 shows the operation analysis of proposed against existing concept using energy. When analysing Figure our idea of implantation approach utilized minimum of 25% energy which is 33% for using GSA, 45% for using lion, 35% for using PSO and 41% for using GA.

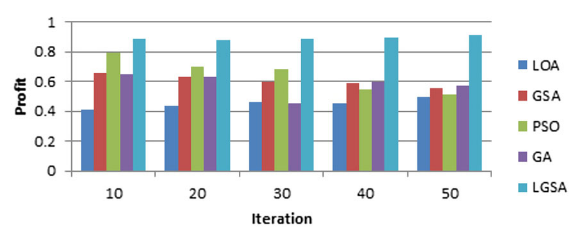


**Fig. 6** Performance analysis of proposed against existing using profit

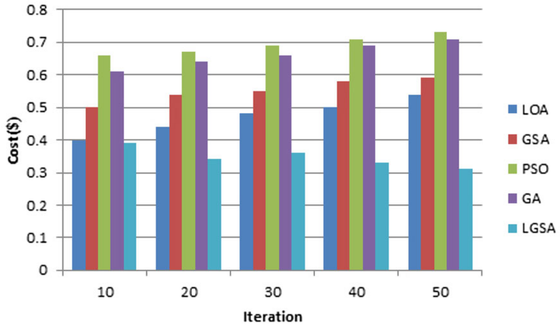


**Fig. 7** Performance analysis of proposed against existing using cost

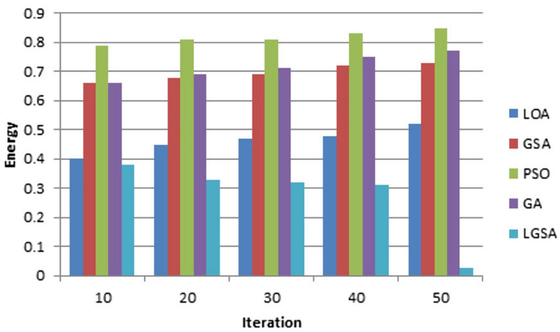


**Fig. 8** Performance analysis of proposed against existing using energy

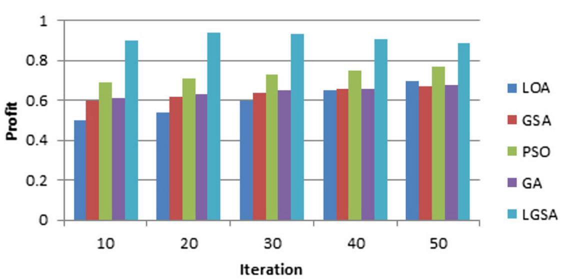


**Fig. 9** Performance analysis of proposed against existing using profit

### PM 15 and VM 45

In this section, we analyse the proposed task scheduling using 10 physical machines and 45 virtual machines. Figures 9, 10 and 11 shows the functional details of the proposed method using this configuration. The main goal of this paper is to run the complete task with high profit, less memory wastage, and less energy consumption. To achieve this objective function in this paper we have used the multi-objective

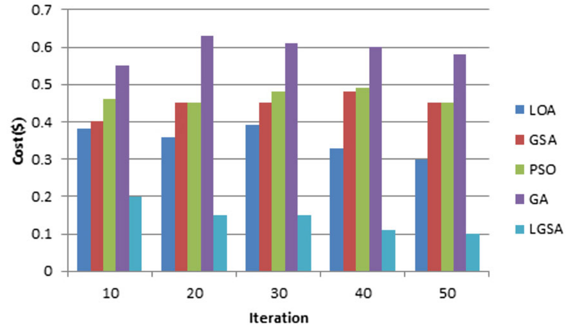


**Fig. 10** Performance analysis of proposed against existing using cost

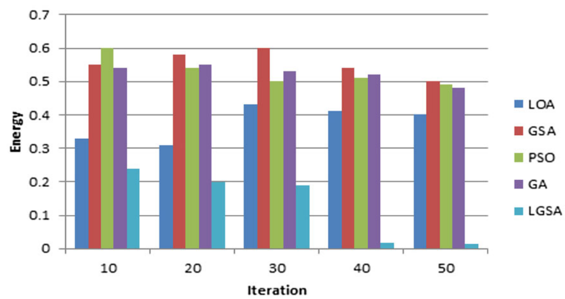


**Fig. 11** Performance analysis of proposed against existing using energy

function. Figure 10 shows the performance analysis of this implementation against present scenario approach using profit. The gain is the key parameter of scheduling. When analyzing Fig. 9, our implementation obtains the maximum profit value compare to other target. Figure 10, shows the performance of difference in variance occurred on a cost function. In this work, we obtain the minimum cost of 0.10$ which is very less compared to other two targets. Energy utilization is an heart core parameter of scheduling. The function of energy is shown in the figure. When researching Fig. 11 our proposed approach utilizes the less energy compare to other targets. In the result section, we can get a clear understanding on our implemented goal that achieve and brings a good outcome compare to other targets.

## 5 Conclusion

This paper elucidates multi-objective scheduling that is based on hybridization of gravitational search algorithm and lion optimization. The advantageous point about multi-objective optimization approach than that of single-objective function is that the former is used to improve the scheduling performance. The algorithm balances cost, energy and resources depending upon the requirement of the end-user. The vital part of scheduling process relies in assigning user's tasks in the way that it maximizes the profit of the system. In addition to tithes algorithm overcomes the over loading and under loading problems during the task-scheduling process. The optimization algorithms like Lion algorithm and GSA could hence be hybridized so as to attain high-quality solution. The results obtained through experiment which was based on three models, shows that our proposed multi-objective scheduling is best of all other approaches.

**Funding** The author, N. Manikandan and the co-author, Dr. A. Pravin declare that there has been no significant financial support for this works that could have influenced its outcome.

**Compliance with Ethical Standards**

**Conflict of interest** The author, N. Manikandan and the co-author, Dr. A. Pravin wish to confirm that there are no known conflicts of interest associated with publication.

**Ethical Approval** This article does not contain any studies with human participants or animals performed by any of the authors.

## References

1. Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications, 1*(1), 7–18. https://doi.org/10.1007/s13174-010-0007-6.
2. Islam, S., Keung, J., Lee, K., & Liu, A. (2012). Empirical prediction models for adaptive resource provisioning in the cloud. *Future Generation Computer Systems, 28*(1), 155–162. https://doi.org/10.1016/j.future.2011.05.027.
3. Bala, A., & Chana, I. (2016). Prediction-based proactive load balancing approach through VM migration. *Engineering with Computers, 32*(4), 581–592. https://doi.org/10.1007/s0036.
4. Arianyan, E., Taheri, H., & Sharifian, S. (2016). Novel heuristics for consolidation of virtual machines in cloud data centers using multi-criteria resource management solutions. *The Journal of Supercomputing, 72*(2), 688–717. https://doi.org/10.1007/s11227-015-1603-9.
5. Wood, T., Ramakrishnan, K. K., Shenoy, P., Van Der Merwe, J., Hwang, J., Liu, G., et al. (2015). CloudNet: Dynamic pooling of cloud resources by live WAN migration of virtual machines. *IEEE/ACM Transactions on Networking (TON), 23*(5), 1568–1583. https://doi.org/10.1109/tnet.2014.2343945.
6. Thaman, J., & Singh, M. (2015). Improving performance of cloud datacenters using heuristic driven VM migration. In *2015 1st International conference on IEEE next generation computing technologies (NGCT)* (pp. 41–45). https://doi.org/10.1109/ngct.2015.7375079.
7. Masdari, M., Nabavi, S. S., & Ahmadi, V. (2016). An overview of virtual machine placement schemes in cloud computing. *Journal of Network and Computer Applications, 66,* 106–127. https://doi.org/10.1016/j.jnca.2016.01.011.
8. Manikandan, N., & Pravin, A. (2018). An efficient improved weighted round Robin load balancing algorithm in cloud computing. *International Journal of Engineering Technology, 7,* 110–116. https://doi.org/10.14419/ijet.v7i3.1.16810.
9. Zheng, Q., Li, R., Li, X., Shah, N., Zhang, J., Tian, F., et al. (2016). Virtual machine consolidated placement based on multi-objective biogeography-based optimization. *Future Generation Computer Systems, 54,* 95–122. https://doi.org/10.1016/j.future.2015.02.010.
10. Liu, L., Zheng, S., Yu, H., Anand, V., & Xu, D. (2016). Correlation-based virtual machine migration in dynamic cloud environments. *Photonic Network Communications, 31*(2), 206–216. https://doi.org/10.1007/s11107-015-0539-6.
11. Xu, B., Peng, Z., Xiao, F., Gates, A. M., & Yu, J. P. (2015). Dynamic deployment of virtual machines in cloud computing using multi-objective optimization. *Soft Computing, 19*(8), 2265–2273. https://doi.org/10.1007/s00500-014-1406-6.
12. Tang, M., & Pan, S. (2015). A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers. *Neural Processing Letters, 41*(2), 211–221. https://doi.org/10.1007/s11063-014-9339-8.
13. Kansal, N. J., & Chana, I. (2016). Energy-aware virtual machine migration for cloud computing—a firefly optimization approach. *Journal of Grid Computing, 14*(2), 327–345. https://doi.org/10.1007/s10723-016-9364-0.
14. Kaur, R., & Luthra, P. (2012). Load balancing in cloud computing. In *Proceedings of international conference on recent trends in information, telecommunication and computing, ITC* (pp. 374–381). https://doi.org/02.ITC.2014.5.92.
15. Yang, H., Breslow, A., Mars, J., & Tang, L. (2013). Bubble-flux: Precise online qos management for increased utilization in warehouse scale computers. *ACM SIGARCH Computer Architecture News, 41*(3), 607–618. https://doi.org/10.1145/2485922.2485974.
16. Gobalakrishnan, N., & Arun, C. (2018). A new multi-objective optimal programming model for task scheduling using genetic gray wolf optimization in cloud computing. *The Computer Journal, 61*(10), 1523–1536. https://doi.org/10.1093/comjnl/bxy009.
17. Krishna Doss, P., & Jacob, P. (2019). OLOA: Based task scheduling in heterogeneous clouds. *International Journal of Intelligent Engineering and Systems, 12,* 114–122. https://doi.org/10.22266/ijies2019.0228.12.
18. Natesan, G., & Chokkalingam, A. (2017). Opposition learning-based grey wolf optimizer algorithm for parallel machine scheduling in cloud environment. *International Journal of Intelligent Engineering and Systems, 10,* 186–195. https://doi.org/10.22266/ijies2017.0228.20.

19. Pradeep, K., & Jacob, T. P. (2018). A hybrid approach for task scheduling using the cuckoo and harmony search in cloud computing environment. *Wireless Personal Communications, 101*(4), 2287–2311. https://doi.org/10.1007/s11277-018-5816-0.
20. Adhikari, M., & Amgoth, T. (2018). Heuristic-based load-balancing algorithm for IaaS cloud. *Future Generation Computer Systems, 81,* 156–165. https://doi.org/10.1016/j.future.2017.10.035.
21. 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.
22. Kang, S., Veeravalli, B., & Aung, K. M. M. (2018). Dynamic scheduling strategy with efficient node availability prediction for handling divisible loads in multi-cloud systems. *Journal of Parallel and Distributed Computing, 113,* 1–16. https://doi.org/10.1016/j.jpdc.2017.10.006.
23. Elsherbiny, S., Eldaydamony, E., Alrahmawy, M., & Reyad, A. E. (2018). An extended Intelligent Water Drops algorithm for workflow scheduling in cloud computing environment. *Egyptian Informatics Journal, 19*(1), 33–55. https://doi.org/10.1016/j.eij.2017.07.001.
24. Lawanyashri, M., Balusamy, B., & Subha, S. (2017). Energy-aware hybrid fruit fly optimization for load balancing in cloud environments for EHR applications. *Informatics in Medicine Unlocked., 8,* 42–50. https://doi.org/10.1016/j.imu.2017.02.005.
25. Neelima, P., & Rama Mohan Reddy, A. (2018). An efficient hybridization algorithm based task scheduling in cloud environment. *Journal of Circuits, Systems and Computers, 27*(02), 1850018. https://doi.org/10.1142/S0218126618500184.
26. Zhao, J., Yang, K., Wei, X., Ding, Y., Hu, L., & Xu, G. (2016). A heuristic clustering-based task deployment approach for load balancing using Bayes theorem in cloud environment. *IEEE Transactions on Parallel and Distributed Systems, 27*(2), 305–316. https://doi.org/10.1109/tpds.2015.2402655.
27. Xu, L., Wang, K., Ouyang, Z., & Qi, X. (2014). An improved binary PSO-based task scheduling algorithm in green cloud computing. In 2014 9th *International conference on communications and networking in China (CHINACOM), IEEE* (pp. 126–131). https://doi.org/10.1109/CHINACOM.2014.7054272.