Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 7, July 2021: 6401- 6415

Research Article

Design of Oppositional Glow worm Swarm Optimization based Resource Scheduler on Big Data Environment

[1]J. Saravana Kumar and [2]L.R. Aravind Babu

Abstract

Big data is commonly used to support significant exploitation of processing resources, concentrating on on-demand services and resource scalability. With numerousmethods available, controlling massive quantity of data in several data centers is still a tedious task. Particularly, resource scheduling (RS) is treated as a way of distributing resources by an efficient decision-making process with the aim of assisting desired tasks over time. The combination of heterogeneous computing resources using the Big Data users enables the chance of minimizing the energy utilization and maximizing resource efficiency. But the state of art RS techniques needs to boost the scheduling performance in the big data environment. In this aspect, this paper designs an Oppositional Glowworm Swarm Optimization based Resource Scheduling (OGSO-RS) scheme for big data environment. The proposed OGSO-RS technique aims to allocate the resources proficiently in the big data platform. The searching area and a large amount of data are provided as input to the geo-distributed datacenter, where the population initialization of glowworms takes place. In addition, the MapReduce function computes the optimal resource, and thereby the efficiency can be improvised. Moreover, the load can be allotted to the datacenters by minimizing the computational cost and storage area. In order to showcase the improved performance of the proposed OGSO-RS technique, a series of experiments were carried out. The simulation results highlighted the betterment of the RS efficiency of the OGSO-RS technique compared to other existing approaches.

Keywords: Big data, Resource scheduling, GSO algorithm, MapReduce, Oppositional learning

1. Introduction

The development of data intensive applications has been increasing rapidly the variety, velocity, and volume of the produced data [1]. Effective computing of data generated at the time its lifespan is further than the ability of present software and hardware techniques [2]. This representsthe main challenge for several organizations and is called a Big Data problem [3]. Streaming Big Data is associated with the velocity dimensions of Big Data and represents how faster data are produced and how faster they should be analyzed. Analyses of streaming Big Data is the latter and primary phase of the streaming Big Data lifespan in systematic application [4].

[1]Research Scholar, Department of Computer and Information Science Annamalai University, Annamalainagar – 608002, Tamilnadu, India
[2]Assistant Professor, Department of Computer and Information Science Annamalai University, Annamalainagar – 608002, Tamilnadu, India
Correspondence: er.arvee@rediffmail.com

The real time instance of application could be analyses of healthcare data, networks traffic, and web-clicks/financial transactions. For developing this application; higher level declarative query languages such as SQL are generally chosen on coding them straight in algorithmically languages like Java [5]. Hence, all the public domain distributed stream computing environments (like Flink, Storm, and Heron platforms) also assist declarative language for developing and executing analyses queries on the cluster of computing nodes. Effective implementation of this query is significantly impacted by scheduling decisions made by the platform. Present public domain Big Data stream computing platforms don't sufficiently tackle the problems of effectively scheduling systematic queries. Such as Storm platforms use a round robin approach i.e., not attentive to resources/tasks features (such as data stream traffic of task, task structure) or the resource accessibility. Current research has presented solutions to enhance the efficacy of the platforms scheduler by considering this aspect. But, almost all considers each aspect or don't attain satisfying result.

The main goal of scheduling in Big Data Processing fully emphasizes the strategy of completing and processing as varied tasks as possible depending upon limited amount of data handlings and modification attained in successful ways [6]. Generally, dissimilar approaches are extremely preferred for allocating resources as they possess specialized framework properties. Regarding this, recognizing an optimal scheduling technique for all certain data processing is deliberated as a significant problem. Such challenges are highly complicated as the Big Data processing is deliberated as the major batch task which runs on a Higher Performance Computing (HPC) cluster by separating a task into small taskswith the aim of distributing the task to the clustering nodes. But, the Big Data processing method should be attentive to area where the data exist in case of transmitting the data to the nodes utilized for computations [7]. Presently, the task is essentially assigned to all computation nodes depending upon the 2 procedures. The 2 procedures the static/dynamic scheduling applied in Map Reduce cluster and process examining accurate context of using resource. Furthermore, the task scheduling procedure could calculate the resource utilization related to every assigned job that mayn't be attained with the investigation of finished tasks.

Huge work was introduced for solving the problems in scheduling concepts in several platforms. In this study, it is tackled using an emerged method employed scheduling in CC and grid methods. The main crises included in scheduling are resolved using Swarm Intelligence (SI) and Meta heuristic methods. In [8], numerous processing methods were proposed by means of Artificial Bee Colony (ABC) approach. Subsequently, optimal outcomes are achieved than Ant Colony Optimization (ACO) and Genetic Algorithm (GA) methods through memory space for bees in ABC model. After that, task scheduling is employed in lime modern grid, shared networks, to optimize Quality of Service (QoS) parameter which leads to span. Moreover, it is deliberated significant factors of scheduling problems and shared processing.

This paper designs an Oppositional Glowworm Swarm Optimization based Resource Scheduling (OGSO-RS) scheme for big data environment. The proposed OGSO-RS technique intends to allot the resources competently in the big data platform. The searching area and the huge quantity of data are fed as input to the geo-distributed datacenter, where the population initialization of glowworms takes place. Moreover, the MapReduce function calculates the optimal resource and thus the efficacy can be improvised. Furthermore, the load can be selected for the datacenters by minimalizing the computational cost and storage area. A wide-spread experimental analysis is performed to point out the better performance of the OGSO-RS technique.

2. Related works

Mortazavi-Dehkordi and Zamanifar [9] introduce Bframework, an effective resource scheduling architecture utilized streaming Big Data analyses applications depending upon cluster resource. Bframework proposes a query method with DG and presents operator allocation and operator scheduling algorithm depending upon a new separating method. Bframework is very adaptable for the variation of streaming Big Data and accessibility of clustering resources.Zhang [10] propose a technique of scheduling network data resource in CC platform depending upon PSO. Initially, this technique dynamically clusters CC network data resources. With the consideration of networks bandwidth, bandwidth utilization, the present networks run state, the technique presents the PS apportion method in the procedure of creating the data resources scheduling method, and predict the implementation speed of tasks depending upon arrange, and schedule and node tasks set of distinct data assets.

In Seethalakshmi et al. [11], HGDSMO algorithms are presented for effective resource scheduling that deals with the challenges and issues in the Hadoop heterogeneous platform. The presented HGDSMO algorithms use the social behaviour and Gradient Descendant foraging of spider monkey optimization algorithms included in the objectives of efficient resources sharing. This is developed as an effective tasks scheduling method which balancesa load of cloud by assigning it to proper VM based on their needs. Also, it is presented as a dynamic resources management system for effectively assigning the cloud resource for efficient implementation of client tasks.

Zhao et al. [12] proposed automated and scalable admission control and profit optimization resources scheduling algorithm, that maximize profit for AaaS providers, efficiently confess data analytic requests, and dynamically provision resources, when fulfilling QoS requirement of queries using SLA assurances. Furthermore, the presented algorithm enables users to tradeoff precision for fast respond time and lesser resources cost for query processing on larger datasets.Enayet et al. [13] proposed a mobility aware optimum resources sharing framework such as Mobi Het, for remote big data tasks implementation in MCC which provides high efficacy in reliability and timeliness. The key component and system architecture of the presented resources sharing services are evaluated and presented. The outcomes of experiment and simulation have shown the efficiency and effectiveness of the projected Mobi-Het framework for mobile big data applications.

Madni et al. [14] proposed a novel HGDCS method depending upon GD and CS algorithms to optimize and resolve the problem associated with resource scheduling in IaaS CC. This study relates the throughput, makes pan, performance improvement rate, and load balancing of present Meta heuristic algorithm using presented HGDCS algorithms appropriate to CC. Madni et al. [15] presented advanced MOCSO algorithms to handle resource scheduling challenges in CC. The major aim of resources scheduling challenge to decrease the cloud user costs also improve the efficient by minimizing make span time, that assist for increasing the profit/revenue for cloud provider with maximal resource utilization. Thus, the presented MOCSO method is a novel technique to solve multiobjective resource scheduling problems in IaaS CC platform.

Consider the real instance of the enterprise, the RBF NN and AHP methods are utilized widely in [16]. Initially, the AHP is utilized for obtaining the weights of all calculation indexes in the human post matching index scheme. Simultaneously, the ANN concept is self-adapting. Learning is useful for solving the problems that the AHP technique is also personal. The 2 learns from one another robust point and integrate their weakness gradually for increasing the effectiveness and convenience of evaluations.Peng et al. [17] solve the conflicts among CSPs aims to minimize energy cost and search for optimizing services quality. Depending upon the outstanding environment attentiveness and online adaptive decision making capability of DRL, they presented an online resource scheduling architecture depending upon the DQN method. The architecture can create a tradeoff of 2 optimization purposes of energy utilization and tasks make span by altering the proportions of reward of distinct optimization purposes.

Tao et al. [18] proposed a novel DHCI on IaaS framework that consists of 4 main components: scheduling, monitoring, VM migration, and management methods. Loads of VM and physical host is gathered with the monitoring method and is utilized for designing data locality solution and resources scheduling. Next, they presented a simpler load feedback dependent resources scheduling system. The RA is evaded on overloaded physical host or the stronger scalability of virtual clusters could be attained with fluctuating the amounts of VM. For improving the flexibility, they adapt the separate placement of storage and computation VMs in the DHCI framework that negatively impacts the data regions.

In Ramamoorthy et al. [19],MCAMO method is presented for cloud resources scheduling specially handles framework dependent cloud service. The technique handles multiobjective through multi limitations when resources scheduling in framework cloud service. The presented technique is new as it handles the limitations of submit tasks and satisfying the objective of cloud services. For a strong arrangement, fitness value worth takes base worth value and the enhanced determinations of assets based on MCAMO evaluation.

3. The Proposed Model

The proposed OGSO-RS method follows the concept of GSO & OBLE techniques. The goal of OGSO-RS method is to decrease the entire operation costs and schedule the loads on resources efficiently. The operation costs contain the transmitting and implementation costs of cloudlet. It gives user satisfaction efficient and searching region exploitations by acquiring a FF. The parameter involves in the FF of resource & cloudlet is bandwidth, MIPS, transmission and implementation costs. The cloudlet scheduling at the resource is executed. The datacenter contains $(VMs)^{Cloudlets}$ feasible techniques of performing the cloudlet on corresponding resources. When executing 3 cloudlets on 2 resources, the likelihood turns into 8. The glowworm S undergoes initialization at CloudSim tools are given below:

$$S_i = \left(s_i^1, s_i^2, \dots \dots, s_i^n, \dots \dots \dots, s_i^d\right)$$

$$\forall i = 1 to\ 25\ and\ n = 1\ to\ 10 \tag{1}$$

The FF defines the fitness value of glowworm in the searching region. The early glowworms use CE followed by the selection of consequent glowworm through optimal fitness value. Consider a $Ct_{exec}(M)_j$ represents the whole implementation costs of each glowworm assigned to estimate the resource PCj. It is created by adding the weight allocated to the nodes in the mapping of glowworm of each cloudlet assigned to separate resource.

Where $\mathrm{C}t_{frons}(M)_j$ specifies the amount of transmission costs that occurred between the cloudlets assigned to estimate the resources $P\mathrm{C}j$. The output specifies the product of output file size and transfer costs. The average cost of data's between a group of 2 assets of transfer is determined as $dS(k1), S(\mathrm{k}2)$ and the glowworm is autonomous of one another. The entire costs are involved for all glowworms using the inclusions of implementation and transmission costs also it is decreased to estimate the FF.

$$\mathrm{C}t_{exec}(S)_j = \sum_k \omega_{kj}, \forall S(\mathrm{k}) = j \tag{2}$$

$$\mathrm{C}t_{trans}(S)_j = \sum_{\mathrm{k}1 \in T} \sum_{\mathrm{k}2 \in T} d_{S(\mathrm{k}1), S(\mathrm{k}2)} * e_{k1,\mathrm{k}2}, \tag{3}$$

$$\forall S(k1) = j \ and \ S(1\langle 2) \neq j \tag{4}$$

$$\mathrm{C}t_{\mathrm{total}}(S)_j = \mathrm{C}t_{exec}(S)_j + \mathrm{C}t_{trans}(S)_j \tag{5}$$

$$Cost_{Total}(S) = \max\left(\mathrm{Ct}_{\mathrm{total}}(S)_j\right), \forall j \in S \tag{6}$$

$$Minimize\ (\mathrm{C}ost_{Total}(S), \forall S) \tag{7}$$

As above mentioned, the CEGSO-LB models integrate the concept of GSO & CE, the calculation functions are shown in Eq. (8):

$$Evaluation\ function = 1 - \left(\alpha_{\mathrm{t}i} \times \frac{\mathrm{t}_i - \mathrm{t}_{\min}}{t_{\max - \mathrm{t}_{\min}}} + \alpha_{ci} \times \frac{c_i - c_{\min}}{c_{\max - c_{\min}}}\right) \tag{8}$$

Now, all feasible processes are calculated for selecting the optimal order of implementation. For efficient scheduling of resources, the OGSO-RS model is employed. Additionally, different processes are deliberated as input including 2 variables such as run time and arrival time. Finally, the calculation functions are estimated using:

$$1 - \left(\alpha_{\mathrm{t}i} \times \frac{t_i - \mathrm{t}_{\min}}{\mathrm{t}_{max} - t_{\min}} + \alpha_{ci} \times \frac{c_i - c_{\min}}{c_{max} - c_{\min}}\right) \tag{9}$$

In which $\mathrm{f}_{\min}$ &$\mathrm{f}_{\max}$ represents minimal & maximal run times, $c_{\min}$ and c_{max} indicates the minimal & maximal input times correspondingly.

In GSO method, a collected group of glowworms undergoes early arbitrary placement in the solution space. Each glowworm represents a resolution of objective functions in the searching space and holds a specific number of luciferins. The number of luciferins is related to the fitness levels of current location of the agent. The bright levels of glowworm represent the best solution. Using likelihood depends on methods, the agent is paying attention to the nearby agents where luciferin intensities exceed the individual inside the position decision domains and next move towards it. The density of glowworms Neighbour affects the influence of decision radius and computes the size of local decision domains. If the adjacent densities are established to be lower, the local decision domains get larger to identify various neighbours; otherwise, it decreases the enable of swarm separation to a small set of groups. This process gets iterated until the GSO algorithms reach the ending conditions. Now, all the individual gathers on the bright individual [20]. Briefly, a group of 5 main phases is included in the GSO algorithms such as neighborhood select, luciferin update, movement, decision radius update, and the moving probability computer. The luciferin upgrade phase is mostly depending upon the fitness value and early luciferin value, and the rules are given below

$$l_i(t+1) = (1-\rho)l_i(t) + \gamma\, Fitness\left(x_j(t+1)\right). \tag{10}$$

whereas $l_i(t)$ denotes the luciferin value of glowworm i in time t, ρ represents the luciferin decay constant, γ indicates the luciferin improvement constant; $x_i(t+1) \in R^M$ implies the location of glowworm i in time $t+1$, and Fitness $\left(x_j(t+1)\right)$ signifies the fitness value of glowworm i's location in time $t+1$.

In the Neighbour selection phase, the neighbours $N_i(t)$ of glowworm i in t time includes bright individual as follows

$$N_i(t) = \left\{j: d_{ij}(t) < r_d^i(t);\ l_i(t) < l_j(t)\right\}. \tag{11}$$

whereas $d_{ij}(t)$ signifies the Euclidean distances between the glowworms $i\&j$ i time t, and $r_d^i(t)$ defines the decision radius of glowworm i in time t. In Moving Probability Computer phase, the glowworms utilize a likelihood rule to move in the direction of another glowworm with maximal luciferin levels. The likelihood $P_{ij}(t)$ of glowworm i move in the direction of neighbour, j as follows:

$$P_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)}. \tag{12}$$

In movement phase, consider the glowworm i selects a glowworm $j \in N_j(t)$ with $P_{ij}(t)$; the discrete time method of i is given below

$$x_i(t+1) = x_i(t) + s\left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|}\right). \tag{13}$$

Whereas $\|\cdot\|$ denotes the Euclidean norm operator, and s defines the step size. Lastly, in the decision radius upgrade phase, the decision radius of glowworms i are denoted by:

$$r_d^i(t+1) = \min\left\{r_s, \max\left\{0, r_d^i(t) + \beta\left(n_t - \left|N_j(t)\right|\right)\right\}\right\}. \tag{14}$$

In which, β represents a constant, r_s implies the sensor radius of glowworm i, and n_t means a regulatory parameter of neighbour.

OBL is a type of optimization procedure i.e., usually utilized in many studies to enhance the quality of early solutions with the diversification of the solution. The OBL method occurs by seeking all directions in the searching region. The 2 direction comprises the opposite and real solutions. Eventually, the OBL system was considered the best solution from the present solution.

Opposite number: x is determined by an actual amount in the interval of $x \in [lb, ub]$. The opposite amount of x is given as $\tilde{x}$ also, it defined in Eq. (15):

$$\tilde{x} = lb + ub - x \tag{15}$$

Eq. (15) is applied to searching regions with multiple dimensions. For generalization, all search agent's location and the opposite position is shown in Eqs. (16)-(17):

$$x = [x_1, x_2, x_3, \dots x_D] \tag{16}$$

$$\tilde{x} = [\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \dots, \tilde{x}_D] \tag{17}$$

The value for all elements in $\tilde{x}$ is given in Eq. (18):

$$\tilde{x}_j = lb_j + ub_j - x_j \ where \ j = 1,2,3, \dots, D \tag{18}$$

Now, the FF is deliberated as $f(.)$. Hence, if the fitness value $f(\tilde{x})$ of opposite solutions exceed the $f(x)$ of new solution, then $x = \tilde{x}$; else $x = x$.

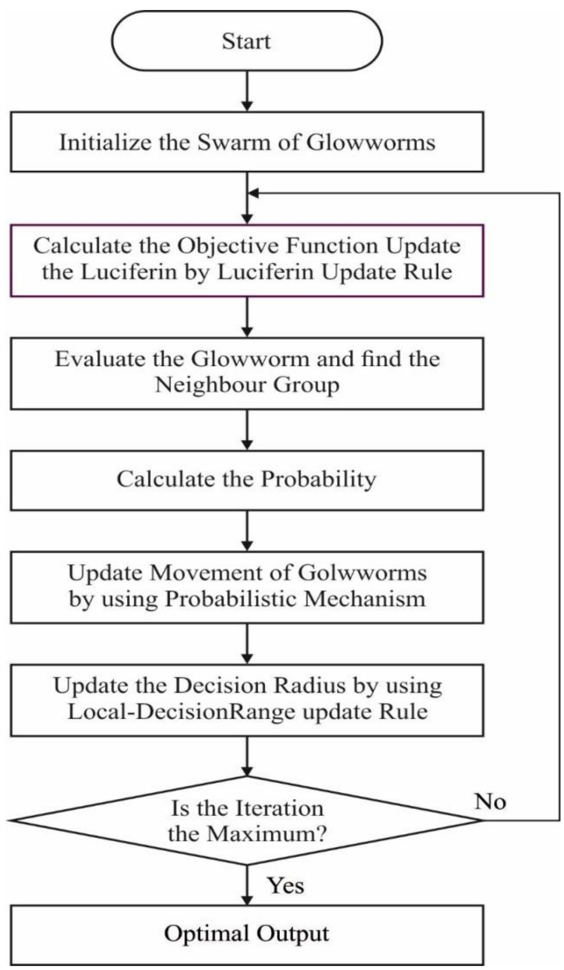

Fig. 1. Flowchart of GSO

The process involves in the incorporation of OBL and GSO is given below.

- Initialization of glowworm populationX as $x_i in\ which(i = 1,2, \dots, n)$.
- Calculate the opposite position of glowworm population OX as $\tilde{x}_i in\ which(i = 1,2, \dots, n)$.

Select n optimum glowworm from $\{X \cup OX\}$ and it is applied to the novel initial population of GSO. Fig. 1 illustrates the flowchart of GSO.

4. Performance Validation

This section investigates the scheduling performance of the OGSO-RS technique under distinct dimensions. Table 1 and Fig. 2 demonstrate the DAE analysis of the OGSO-RS technique with other existing RS techniques under distinct data. The table values portrayed that the OGSO-RS technique has accomplished improved performance with the higher DAE. For instance, with 15 data, the OGSO-RS technique has obtained an increased DAE of 88% whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS techniques have offered a reduced DAE of 84%, 80%, 67%, and 73% respectively. Likewise, with 45 data, the OGSO-RS approach has attained a higher DAE of 89% whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS manners have existed a minimum DAE of 86%, 84%, 71%, and 79% correspondingly.

Table 1: Result Analysis of Existing with Proposed OGSO-RS in terms DAE vs number of data

No. of data	**Data Allocation Efficiency (DAE) (%)**				
	OGSO-RS	**IABOA-RS**	**MM-MGSMO**	**JOA-RAS**	**GT-DRAS**
15	88.00	84.00	80.00	67.00	73.00
30	93.00	89.00	87.00	73.00	80.00
45	89.00	86.00	84.00	71.00	79.00
60	94.00	90.00	87.00	75.00	82.00
75	90.00	87.00	85.00	73.00	80.00
90	96.00	92.00	89.00	78.00	83.00
105	96.00	94.00	90.00	79.00	85.00
120	95.00	91.00	89.00	81.00	83.00
135	96.00	93.00	91.00	79.00	85.00
150	94.00	92.00	89.00	80.00	83.00

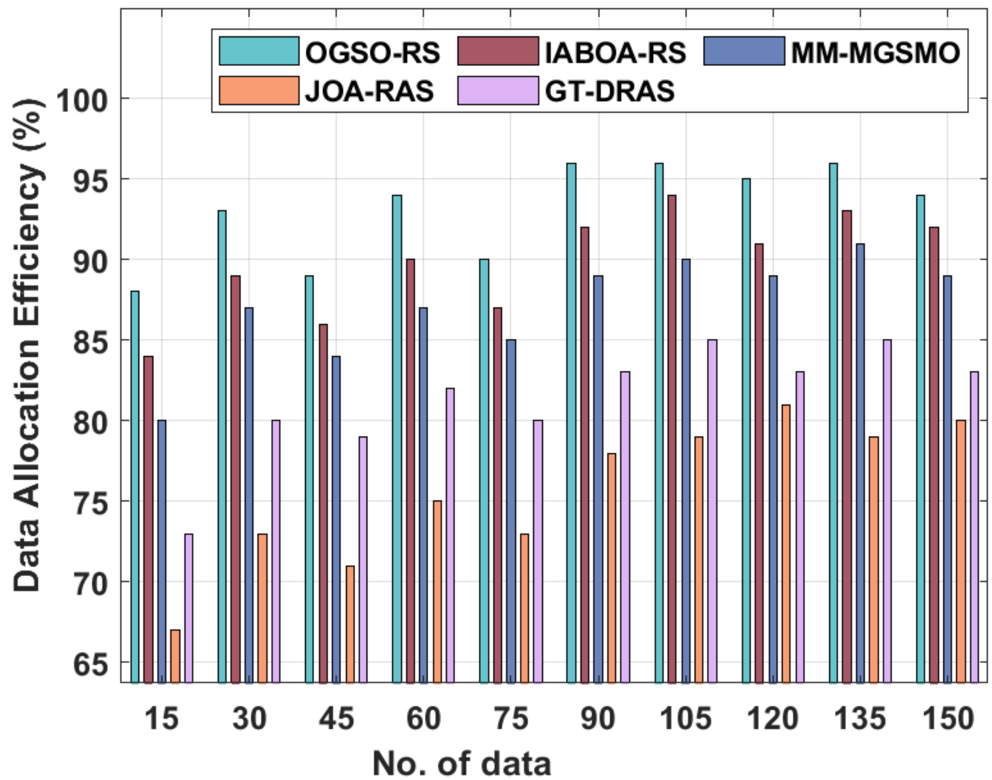

Fig. 2. DAE analysis of OGSO-RS model with different data

Meanwhile, with 60 data, the OGSO-RS approach has gained an improved DAE of 94% whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS manners have presented a minimal DAE of 90%, 87%, 75%, and 82% correspondingly. In line with, with 90 data, the OGSO-RS method has gained a maximum DAE of 96% whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS techniques have offered a decreased DAE of 92%, 89%, 78%, and 83% respectively. Along with that, with 120 data, the OGSO-RS method has achieved an improved DAE of 95% whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS manners have presented a lesser DAE of 91%, 89%, 80%, and 83% correspondingly. Simultaneously, with 150 data, the OGSO-RS technique has reached an increased DAE of 94% whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS algorithms have offered a lower DAE of 92%, 89%, 80%, and 83% correspondingly.

Afalse positive rate (FPR) analysis of the OGSO-RS manner takes place with distinct amount of data in Table 2 and Fig. 3. The outcomes demonstrated that the OGSO-RS approach has accomplished maximal efficiency with lesser FPR over the other approaches. For sample, with 15 data, the OGSO-RS technique has resulted in a least FPR of 14% whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS algorithms have reached a maximum FPR of 16%, 20%, 33%, and 27% correspondingly. In line with, 45 data, the OGSO-RS technique has resulted in a lesser FPR of 11% whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS methods have obtained a higher FPR of 13%, 16%, 29%, and 22% respectively.

Table 2: Result Analysis of Existing with Proposed OGSO-RS in terms False Positive Rate (%) versus number of data

No. of data	FPR (%)				
	OGSO-RS	IABOA-RS	MM-MGSMO	JOA-RAS	GT-DRAS
15	14.00	16.00	20.00	33.00	27.00
30	7.00	11.00	13.00	27.00	20.00
45	11.00	13.00	16.00	29.00	22.00
60	8.00	10.00	13.00	25.00	18.00
75	9.00	12.00	15.00	27.00	20.00
90	5.00	09.00	11.00	22.00	17.00
105	5.00	08.00	10.00	21.00	15.00
120	6.00	10.00	11.00	19.00	17.00
135	4.00	07.00	9.00	21.00	15.00
150	5.00	09.00	11.00	20.00	17.00

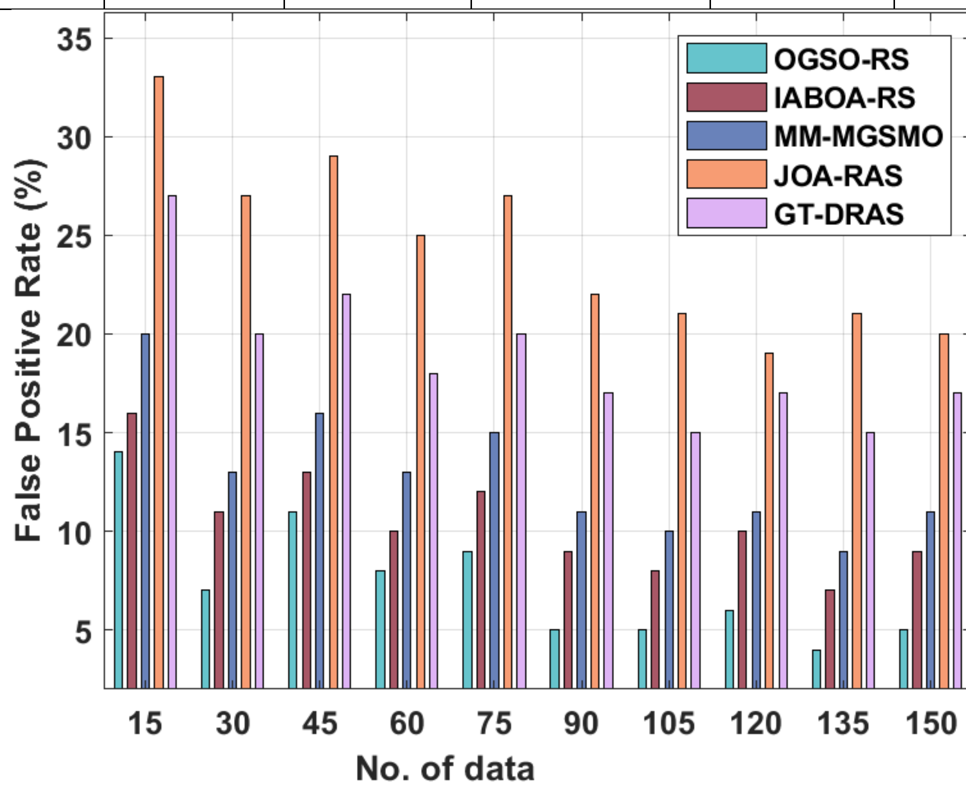

Fig. 3. FPR analysis of OGSO-RS model with different data

Additionally, with 90 data, the OGSO-RS manner has resulted in a minimal FPR of 5% whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS methods have attained a superior FPR of 9%, 11%, 22%, and 17% correspondingly. Moreover, with 120 data, the OGSO-RS approach has resulted in a minimum FPR of 6% whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS algorithms have reached an improved FPR of 10%, 11%, 19%, and 17% correspondingly. Finally, with 150 data, the OGSO-RS method has resulted in the least FPR of

5% whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS techniques have reached an increased FPR of 9%, 11%, 20%, and 17% correspondingly.

Table 3: Result Analysis of Existing with Proposed OGSO-RS in terms Storage Capacity versus number of data

No. of data	**Storage Capacity (MB)**				
	OGSO-RS	**IABOA-RS**	**MM-MGSMO**	**JOA-RAS**	**GT-DRAS**
15	11.00	14.00	18.00	24.00	21.00
30	14.00	16.00	21.00	30.00	27.00
45	15.00	19.00	23.00	36.00	32.00
60	21.00	23.00	26.00	39.00	36.00
75	24.00	27.00	31.00	39.00	37.00
90	26.00	28.00	32.00	45.00	39.00
105	30.00	32.00	38.00	48.00	44.00
120	33.00	36.00	40.00	50.00	46.00
135	36.00	39.00	43.00	53.00	49.00
150	39.00	42.00	45.00	54.00	51.00

Eventually, a storage capacity (SC) analysis of the OGSO-RS manner takes place with different amounts of data in Table 3 and Fig. 4. The results showcased that the OGSO-RS method has accomplished maximal performance with the least SC over the other techniques. For instance, with 15 data, the OGSO-RS algorithm has resulted inthe least SC of 11MB whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS manners have reached a superior SC of 14MB, 18MB, 24MB, and 21MB correspondingly. Afterward, with 45 data, the OGSO-RS technique has resulted in a lower SC of 15MB whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS techniques have been obtained a higher SC of 19MB, 23MB, 36MB, and 32MB respectively. Also, with 90 data, the OGSO-RS approach has resulted in a minimum SC of 26MB whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS methods have been obtained an increased SC of 28MB, 32MB, 45MB, and 39MB respectively. At the same time, with 120 data, the OGSO-RS algorithm has resulted in a lower SC of 33MB whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS methods have reached a superior SC of 36MB, 40MB, 50MB, and 46MB correspondingly. However, with 150 data, the OGSO-RS technique has resulted in a lesser SC of 39MB whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS techniques have been obtained a maximum SC of 42MB, 45MB, 54MB, and 51MB correspondingly.

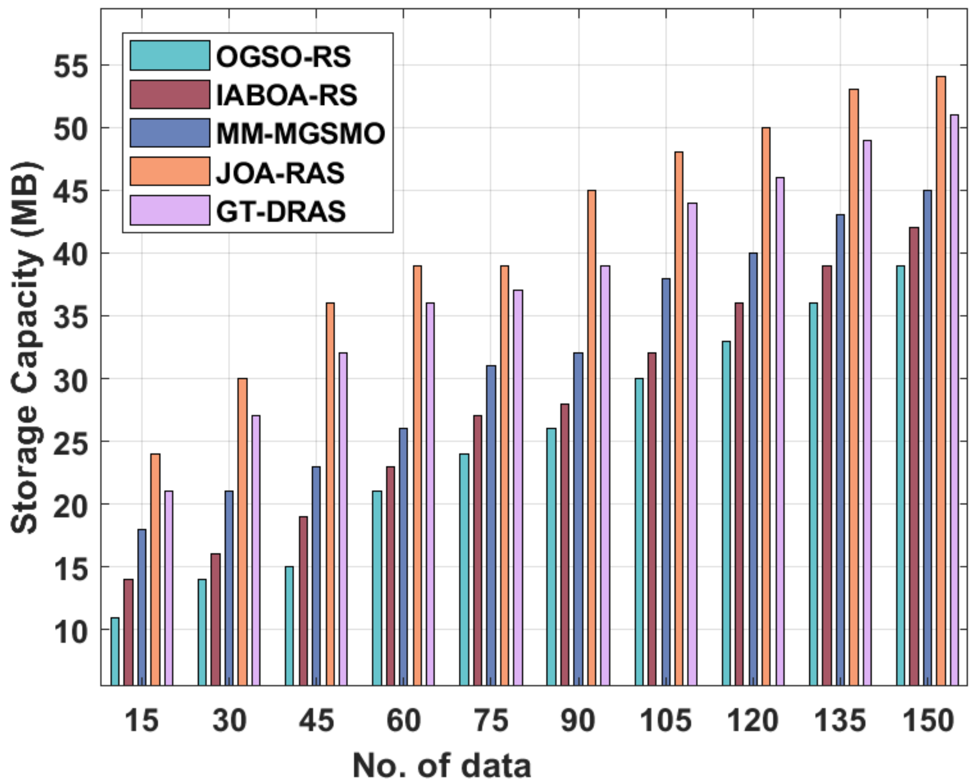

Fig. 4. SC analysis of OGSO-RS model with different data

Finally, a computational cost (CTC) analysis of the OGSO-RS technique takes place with varying amounts of data in Table 4 and Fig. 5. The results portrayed that the OGSO-RS technique has accomplished maximum performance with the minimal CTC over the other techniques. For instance, with 15 data, the OGSO-RS technique has resulted in a lower CTC of 16ms whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS techniques have obtained a higher CTC of 18ms, 21ms, 27ms, and 24ms respectively. Followed by, with 45 data, the OGSO-RS method has resulted in a minimum CTC of 22ms whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS approaches have achieved an increased CTC of 24ms, 27ms, 36ms, and 32ms correspondingly.

Table 4: Result Analysis of Existing with Proposed OGSO-RS in terms Computation Cost versus number of data

No. of data	**Computation Cost (ms)**				
	OGSO-RS	**IABOA-RS**	**MM-MGSMO**	**JOA-RAS**	**GT-DRAS**
15	16.00	18.00	21.00	27.00	24.00
30	18.00	20.00	24.00	33.00	27.00
45	22.00	24.00	27.00	36.00	32.00
60	23.00	28.00	30.00	38.00	35.00
75	26.00	30.00	34.00	44.00	40.00
90	30.00	32.00	39.00	47.00	43.00

105	34.00	38.00	44.00	53.00	50.00
120	38.00	40.00	43.00	55.00	49.00
135	37.00	42.00	47.00	58.00	53.00
150	45.00	47.00	51.00	60.00	54.00

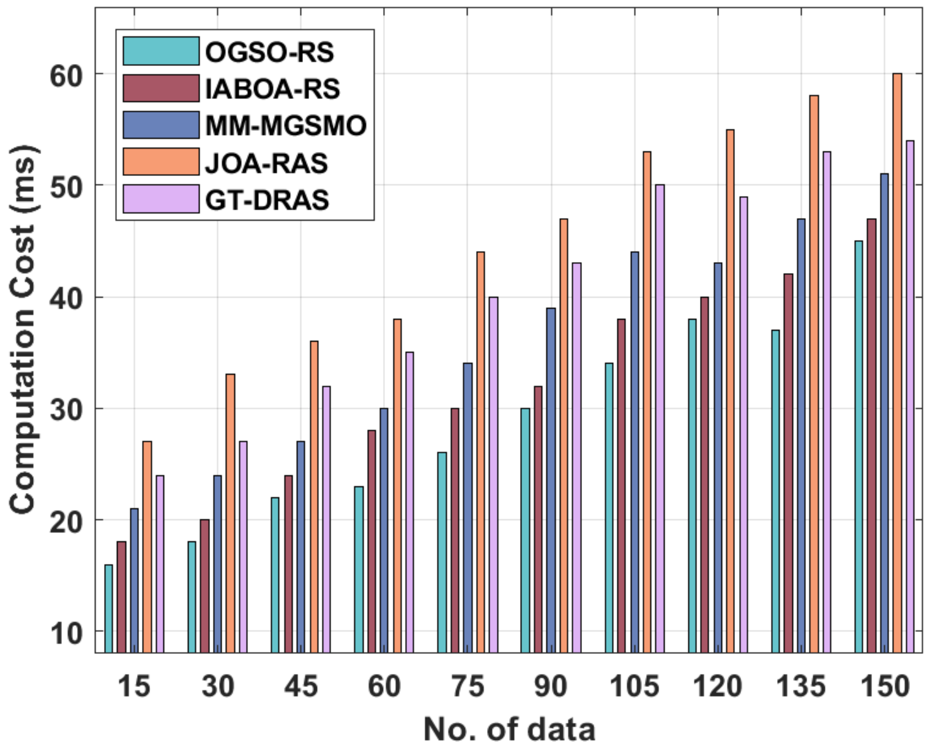

Fig. 5. CTC analysis of OGSO-RS model with different data

In addition, with 90 data, the OGSO-RS algorithm has resulted in the least CTC of 30ms whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS methods have gained an improved CTC of 32ms, 39ms, 47ms, and 43ms correspondingly. Moreover, with 120 data, the OGSO-RS technique has resulted in a least CTC of 38ms whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS techniques have attained a superior CTC of 40ms, 43ms, 55ms, and 49ms respectively. Lastly, with 150 data, the OGSO-RS manner has resulted in a minimal CTC of 45ms whereas the IABOA-RS, MM-MGSMO, JOA-RAS, and GT-DRAS methodologies have reached an increased CTC of 47ms, 51ms, 60ms, and 54ms correspondingly.By looking into the above mentioned results analyses, it is apparent that the OGSO-RS technique is found to be an effective tool to schedule resources in the big data environment.

5. Conclusion

This paper has designed a novel OGSO-RS scheme for big data environment. The proposed OGSO-RS technique intends to allot the resources competently in the big data platform. The searching area and the huge quantity of data are fed as input to the geo-distributed datacenter, where the population initialization of glowworms takes place. Moreover, the MapReduce function calculates the optimal resource and thus the efficacy can be improvised. Furthermore, the load can be selected for the datacenters by minimalizing the computational cost and storage area. A wide-spread experimental analysis is performed to point out the better performance of the OGSO-RS technique. The simulation results highlighted the betterment of the RS efficiency of the OGSO-RS technique compared to other existing approaches. As a part of future extension, data clustering techniques can be modelled to handle the velocity and variety of big data.

References

[1] M. Chen, S. Mao, Y. Zhang and V.C. Leung, Big data: related technologies, challenges and future prospects, Heidelberg: Springer, 2014.

[2] J.J. Berman, Principles of big data: preparing, sharing, and analyzing complex information, Newnes, 2013.

[3] M. Barlow, Real-time big data analytics: Emerging architecture. O'Reilly Media, Inc., 2013.

[4] H. Hu, Y. Wen, T.S. Chu and L. Xuelong, Toward scalable systems for big data analytics: A technology tutorial, IEEE Access 2 (2014), 652–687.

[5] L. Fegaras, Incremental query processing on Big Data stream, IEEE Transactions on Knowledge and Data Engineering 28(11) (2016), 2998–3012.

[6] Rekha PM, Dakshayini M. Efcient task allocation approach using genetic algorithm for cloud environment. Clust Comput. 2019;1(1):43–56.

[7] Rana N, Abd Latif MS, Muhammad Abdulhamid S. A cloud-based conceptual framework for multi-objective virtual machine scheduling using whale optimization algorithm. Int J Innovat Comput. 2018;8(3):67–76.

[8] Natesan G, Chokkalingam A. Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm. ICT Express. 2018;1(1):34–45.

[9] Mortazavi-Dehkordi, M. and Zamanifar, K., 2019. Efficient resource scheduling for the analysis of Big Data streams. Intelligent Data Analysis, 23(1), pp.77-102.

[10] Zhang, Y., 2019. Classified scheduling algorithm of big data under cloud computing. International Journal of Computers and Applications, 41(4), pp.262-267.

[11] Seethalakshmi, V., Govindasamy, V. and Akila, V., 2020. Hybrid gradient descent spider monkey optimization (HGDSMO) algorithm for efficient resource scheduling for big data processing in heterogenous environment. Journal of Big Data, 7(1), pp.1-25.

[12] Zhao, Y., Calheiros, R., Gange, G., Bailey, J. and Sinnott, R., 2018. SLA-based profit optimization resource scheduling for big data analytics-as-a-service platforms in cloud computing environments. IEEE Transactions on Cloud Computing.

[13] Enayet, A., Razzaque, M.A., Hassan, M.M., Alamri, A. and Fortino, G., 2018. A mobility-aware optimal resource allocation architecture for big data task execution on mobile cloud in smart cities. IEEE Communications Magazine, 56(2), pp.110-117.

[14] Madni, S.H.H., Abd Latiff, M.S. and Ali, J., 2019. Hybrid gradient descent cuckoo search (HGDCS) algorithm for resource scheduling in IaaS cloud computing environment. Cluster Computing, 22(1), pp.301-334.

[15] Madni, S.H.H., Abd Latiff, M.S. and Ali, J., 2019. Multi-objective-oriented cuckoo search optimization-based resource scheduling algorithm for clouds. Arabian Journal for Science and Engineering, 44(4), pp.3585-3602.

[16] Wu, Y. and Sun, X., 2021. Optimization and Simulation of Enterprise Management Resource Scheduling Based on the Radial Basis Function (RBF) Neural Network. Computational Intelligence and Neuroscience, 2021.

[17] Peng, Z., Lin, J., Cui, D., Li, Q. and He, J., 2020. A multi-objective trade-off framework for cloud resource scheduling based on the deep Q-network algorithm. Cluster Computing, pp.1-15.

[18] Tao, D., Lin, Z. and Wang, B., 2017. Load feedback-based resource scheduling and dynamic migration-based data locality for virtual hadoop clusters in openstack-based clouds. Tsinghua Science and Technology, 22(2), pp.149-159.

[19] Ramamoorthy, S., Ravikumar, G., Saravana Balaji, B., Balakrishnan, S. and Venkatachalam, K., 2020. MCAMO: multi constraint aware multi-objective resource scheduling optimization technique for cloud infrastructure services. Journal of Ambient Intelligence and Humanized Computing, pp.1-8.

[20] Mohan, S. and Kumar, G., 2021. Cross Entropy with Glowworm Swarm Optimization Algorithm based Load Balancing Technique for Distributed Big Data Systems. *European Journal of Molecular & Clinical Medicine*, *7*(7), pp.4739-4752.