I. J. Computer Network and Information Security, **2018, 4, 38-47**
Published Online April 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2018.04.05

Linear Improved Gravitational Search Algorithm for Load Scheduling in Cloud Computing Environment (LIGSA-C)

Divya Chaudhary*, Bijendra Kumar
Department of Computer Engineering, Netaji Subhas Institute of Technology, Dwarka, New Delhi, 110078, India
E-mail: *divyadabas@gmail.com, bizender@gmail.com

Received: 24 September 2017; Accepted: 08 January 2018; Published: 08 April 2018

Abstract—The load scheduling is one of the prime concerns for the computation of tasks in a virtual distributed environment. Many meta-heuristic swarm based optimization methods have been developed for scheduling the load in cloud computing environment. These swarm intelligence based algorithms like PSO play a key role in determining the scheduling of the cloudlets on the VMs in the datacenter. Gravitational Search algorithm based on law of gravity schedules the load in an effective manner. Its potential has not been utilized in cloud for load scheduling. This paper proposes a linear improved gravitational search algorithm in Cloud (LIGSA-C). This presents a new linear gravitational function and cost evaluation function for cloudlets using gravitational search approach in cloud. The results are computed by particles for scheduling 10 cloudlets on 8 VMs in the cloud. The detailed analysis of the result is performed. This paper states that LIGSA-C outperforms the existing algorithms like GSA and PSO for minimized cost.

Index Terms—Cloud Computing, Load Scheduling, GSA, Swarm Intelligence, PSO, Gravity.

I. INTRODUCTION

The cloud computing is one of the incremental domains in the area of distributed computing. It works on the principle of virtualization. Distributed computing offers the benefit of communicating over a number of systems located over a wider area. It helps in remote area communication over the networks. The computing refers to the processing of the tasks in the system [1]. The objective of computation on the resources and tasks is to perform efficient utilization of the resources for the users. With the advent of the new advancements prime focus is on increasing the computational power day by day. It overcomes the drawback of peer to peer networks by using a central repository for storage and processing of the resources [7, 9]. The cloud extends the pay-as-you-use model to software and their applications. It provides us with a list of resources for managing the tasks provided by the users [26]. The paper explains the cloud computing framework for load scheduling of the cloudlets using the meta-heuristic algorithms.

Load scheduling is the process of assigning, processing and managing the tasks (cloudlets) on the virtual machines for the resources [6, 8]. The resources are scheduled in an efficient manner such that the tasks (cloudlets) take minimum response time, waiting time and processing time. The minimization or reduction of cost of computation is the other objective that needs to be fulfilled [4, 5]. The cost of computation and response time are interdependent. The lesser the response and waiting time of the computation the lesser is the cost involved. The meta-heuristic techniques are used for scheduling the load. This paper discusses some of the heuristic techniques using the swarm intelligence and physical significance. The load scheduling is performed using the physical laws that are universal in nature. The swarm intelligence deals with the swarm of objects or particles, ants, bees etc. [12, 14]. The concept of gravity given by Newton is also discussed for optimization of results. The law of gravitation is used for locating the best possible particle in the search space based on the force acting on the masses and the distance between the masses. The gravitational search algorithm is discussed along with proposing a new linear improved gravitational search algorithm in cloud (LIGSA-C). The cloudlets are scheduled in such a manner so that the tasks minimize the cost of computation. This dynamic nature of the algorithms is used for processing as it includes a larger search space. The randomized selection of the particle is discussed in the algorithm using improved gravitational search algorithm such that the total cost is minimized. This is evident from the results of the proposed scheme (LIGSA-C) with the existing schemes like PSO (Particle Swarm Optimization) [17-19] and GSA (Gravitational Search Algorithm) [21-23]. Thus, the prime goal of minimizing the cost of computation or the total cost in scheduling the cloudlets on the virtual machines (VMs) on the datacenter is to be fulfilled. The CloudSim Simulator [2] is used as a tool for performing the scheduling using the algorithms on the network.

This paper provides an insight on Load scheduling using gravitational search algorithm in Section 2. Section 3 elaborates the proposed scheme Linear Improved

Gravitational Search Algorithm approach in cloud (LIGSA-C) environment in detail. Section 4 discusses the experimental setup, simulator, results and analysis for scheduling the load by implementing the LIGSA-C and existing methods using a tabular and graphical approach. Finally Section 5 provides the conclusion & future scope of the paper.

II. Load Scheduling based on Swarm Intelligence and Law of Gravitation

The distributed computing has evolved into cloud computing using the virtual framework of data storage [3]. It has evolved from the grid computing framework i.e., from gridlets to cloudlets for processing the resources. It uses the power of client-server architecture of the cloud to meet the needs of the end users. The benefits of cloud computing are large by taming them for larger computational power, scalability and lesser time for processing of cloudlets [27]. All these capabilities lead to reduced computational cost of the system [12, 14]. The cloud behaves like a centralized server that stores huge amount of data in the system. The data can be processed by the server by the various clients located at distributed locations. The requests sent by users are scheduled on the basis of Newton's laws of gravity [10]. The requests are received as cloudlets by the virtual machines on the data center for processing. These are scheduled on the basis of the real world scenario using the bees, ants or particles over the virtual machines in swarm intelligence based algorithms like Genetic Algorithm (GA) [26], PSO (Particle Swarm Optimization) and physical laws like Newton's law of gravitation. The law of gravity is followed by the GSA (Gravitational Search Algorithm) [11, 13] for calculating the force and acceleration between the particles. They follow the nature scenario for locating the food efficiently. The load scheduling is performed in PSO using stochastic nature of the particles and the convergence of the values in the search space [16, 20]. The goal of assigning the workflow of the tasks is such that the total cost of computation is reduced. For this the *pbest* (best position among all particles) and *gbest* (best particle in the entire population) values are calculated for the particles. These help in determining the velocity and position of the particles for further processing based on the values and fitness of the particle. In GSA, the masses are specified for each particle. On the basis of fitness value the next particle to be executed on the basis of the force and mass is computed. The Gravitational Constant used is exponential in nature which leads to larger set of values of search space [15, 24]. This is the drawback of the GSA for locating the next best particle [25]. The force and acceleration are calculated so the next best particles velocity and position is determined. This algorithm has been defined on the mathematical set of equations thereby resulting in lesser cost of computation than the particle swarm optimization technique. The next section overcomes these drawbacks of GSA by the proposed algorithm (LIGSA-C).

III. Proposed Linear Improved Gravitational Search Algorithm in Cloud Computing (LIGSA-C)

The proposed method Linear Improved Gravitational Search Algorithm in Cloud Computing (LIGSA-C) is an improved gravitational search algorithm technique in cloud computing environment. It is meta-heuristic swarm intelligence based technique depicting convergence of the search space which locates the next probable position of the particle. It focuses on the objective of parallel computation where the tasks are computed in parallel. The main focus of this approach is to reduce the total cost of computation incurred by the cloudlets. The cost of computation includes the execution and transfer cost of the particles. This approach is a new fresh method applied on the cloud for scheduling the load. The basis of this law of gravity and an improved method for cost calculation such that parallel execution is depicted. The law of gravity states that every particle using force gets attracted to other particle in the universe. The force is directly proportional to the masses of the particles and inversely proportional to the distance between the particles. It removes the drawbacks of gravitational search algorithm having greater search space using a linear gravitational constant function *G* and imbibes parallel computation using an improved cost evaluation function. It generates the better results having reduced computation time than the particle swarm optimization and basic gravitational search algorithm. It also achieves maximum utilization of the resources by the particle or objects. Once the optimal positions are located based on the fitness values calculated by the fitness equation, they are assigned to the cloudlets. The fitness value depends on the masses of the particles. The cloudlets further run on the virtual machines present in the datacenter using the datacenter broker. This is explained using a flowchart given in figure 1 using the updated cost evaluation function.

In a cloud environment, N cloudlets are taken for performing load scheduling. The cloudlets are allocated to specific number of virtual machines (V). The total possible solutions in the search space to allocate cloudlets is $(V)^N$ for the corresponding VMs. Each particle having d dimension is defined for a specific number of particles, X in search space having value of any number of virtual machine (V-1). LIGSA-C algorithm is applied to a number of particles to return the position of the global best in the search space. The particles are defined as:

$$X_i = \left(x_i^1, x_i^2, \dots \dots, x_i^n, \dots \dots \dots, x_i^d\right) \quad (1)$$

The particles are distributed in the search space. The fitness function for each particle is calculated on the basis of the equation. The fitness value is dependent on the transfer $Co_{transfer}(M)_j$ and execution cost $Co_{exec}(M)_j$ of the virtual machines and cloudlets. The minimized total cost of computation of the particles is calculated on the basis of equation (2) to (7).

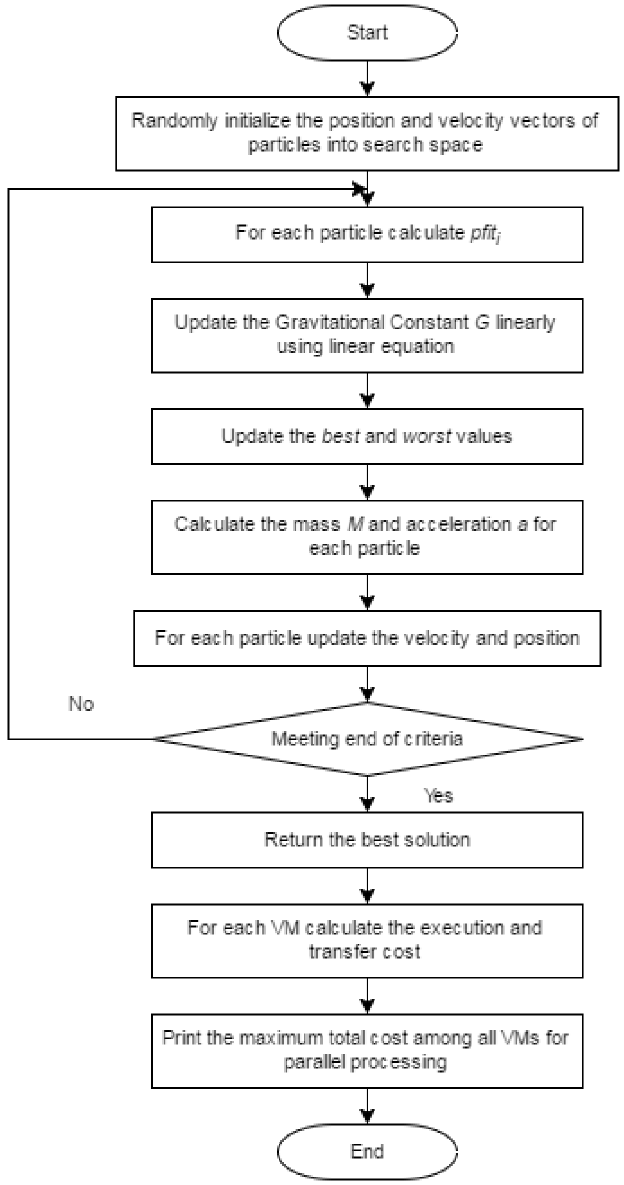

Fig.1. Flowchart Depicting the LIGSA-C Algorithm in Cloud Computing Environment

$$C_{exec}(M)_j = \sum_k w_{kj} \quad (2)$$

$$\forall M(k) = j \quad (3)$$

$$C_{transfer}(M)_j = \sum_{k1\in T} \sum_{k2\in T} d_{M(k1),M(k2)} * e_{k1,k2} \quad \forall M(k1) = j \text{ and } M(k2) \neq j \quad (4)$$

$$C_{total}(M)_j = C_{exec}(M)_j + C_{transfer}(M)_j \quad (5)$$

$$Cost\ (M) = \max\left(C_{total}(M)_j\right) \quad \forall_j \in P \quad (6)$$

$$Minimize(Cost\ (M) \quad \forall\ M) \quad (7)$$

The fitness value of each particle in the search space $pfit_i^t$ is calculated. The masses M_{acti}, M_{pasi} and M_{ineri} namely active, passive and inertial masses respectively are calculated on the basis of the fitness function $FM_i(t)$. $pfit_i(t)$ is fitness value at specific instant of time. The values $best(t)$ and $worst(t)$ values represent the best and worst fitness of the particle array.

$$M_{acti} = M_{pasi} = M_{ineri} = M_i, \qquad i = 1,2,3, \ldots\ldots, N \quad (8)$$

$$m_i(t) = \frac{pfit_i(t) - worst(t)}{best(t) - worst(t)} \quad (9)$$

$$FM_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \quad (10)$$

$$best(t) = \frac{min}{j\in\{1,\ldots,N\}} fit_j(t) \quad (11)$$

$$worst(t) = \frac{max}{j\in\{1,\ldots,N\}} fit_j(t) \quad (12)$$

The force depends on the value of gravitational constant, *G* which is to be calculated at a time instant and reduced to increase the system efficiency. The force helps in locating the next optimal particle. The gravitational constant *G(t)* helps in specifying the next position of the particle. *G(t)* is the function of G_0 (initial value)and time *t*.

$$G(t) = G_0\left(1 - \left(\frac{t}{\alpha * iter_{max}}\right)\right) \quad (13)$$

α here represents a random value and $iter_{max}$ represents maximum number of iterations. The force calculated on the basis of mass and distance helps in locating the potential of execution on the virtual machines. So, the force acting on mass *i* with respect to mass *j* particular instance of time t is computed as:

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon}\left(x_j^d(t) - x_i^d(t)\right) \quad (14)$$

The gravitational force is inversely proportional to the Euclidean distance between the two particles X_i and X_j in the search space. $R_{ij}(t)$ is calculated by the equations (15) and (16).

$$R_{ij}(t) = \left\|X_i(t), X_j(t)\right\|_2 \quad (15)$$

$$R\left(X_i, X_j\right) = R\left(X_j, X_i\right) = \sqrt{\sum_{i=1,\ j=1}^{N}(X_j - X_i)^2} \quad (16)$$

So, the total force acting on a specific particle in respect to other particles in the search space is calculated. The value of the function $rand_j$ lies in the interval [0, 1) as a static value.

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t) \quad (17)$$

The acceleration by which the particle *i* is executed at a specific time t in the d-dimensional space depends on their force and the inertial mass of the cloudlet:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ineri}(t)} \quad (18)$$

The next particle to be chosen for execution depends on the velocity of the next particle to be calculated. The velocity of the next iteration of the particle $v_i^d(t+1)$ is calculated on the basis of a uniform random function $rand_i$ having value between [0, 1] along with the existing velocity and acceleration.. The next position of the particle $x_i^d(t+1)$ to be executed depends on the original particle position and velocity of the particle. The velocity and position are computed as:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \quad (19)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (20)$$

The process continues till the condition of maximum iterations is fulfilled. The cloudlets are assigned to respective virtual machines VMs on the basis of fitness value. Thus, the cloudlets are scheduled on the VMs. The results and the experimental setup are discussed in the next section. This algorithm reduces the cost of computation considerably.

IV. Experimental Results & Analysis

The proposed Linear Improved Gravitational Search Algorithm in cloud computing based on scheduling heuristics is evaluated in this section. The CloudSim Simulator is used for implementation of the algorithm for obtaining the results in the system. The Network CloudSim Simulator is based on the CloudSim. The platform is used for running the proposed LIGSA-C and existing approaches PSO and GSA. These algorithms are implemented on the simulator using the Swarm package JSwarm. The total cost of computation by the number of cloudlets on the number of iterations incurred after scheduling the load are elaborated in a tabulated and graphical manner on the basis of the new and existing algorithms. A set of 25 particles are distributed into 10, 15 and 20 cloudlets on 8 VMs in the simulator. Each cloudlet includes the fitness value, execution cost and transfer cost that needs to be handled by the cloudlet and execution cost on the VMs. The total cost is calculated based on these values.

The algorithms LIGSA-C, GSA and PSO are executed on a number of iterations ranging from 10 to 1000. The tabulated results of the total cost incurred using the proposed (LIGSA-C) and existing algorithms (GSA and PSO) on the 10, 15 and 20 cloudlets on 8 VMs in respect of no of iterations are provided in Table 1, Table 2 and Table 3.

Table 1. Comparison of Total Cost for 10 cloudlets in PSO, GSA and LIGSA-C algorithms

Iteration	PSO	GSA	LIGSA-C
10	144670.354	22033.564	27842.761
20	154718.388	22231.076	22364.729
30	146023.003	22646.835	22364.729
40	151117.032	25236.798	29819.639
50	153934.604	21804.132	28550.078
60	146671.563	28238.387	22977.632
70	150630.778	24134.520	25685.220
80	144320.104	27754.698	26092.184
90	150043.444	28324.327	27754.698
100	155538.952	29378.085	27309.156
200	143757.944	22364.729	25289.422
300	155469.762	26501.040	23337.076
400	154534.972	27754.698	29023.443
500	148303.411	22239.871	24134.520
600	143482.749	22364.729	25410.000
700	148181.188	33288.008	25229.587
800	145980.430	26069.204	25705.824
900	145713.846	19760.956	22927.794
1000	150183.341	19879.760	18100.890

Table 2. Comparison of Total Cost for 15 cloudlets in PSO, GSA and LIGSA-C algorithms

Iteration	PSO	GSA	LIGSA-C
10	251281.517	37857.076	32257.321
20	248417.269	35963.567	28937.135
30	248960.089	29040.000	28961.424
40	247411.824	38166.082	38475.073
50	249097.514	32648.053	33050.346
60	252863.341	38681.683	36268.231
70	233301.076	41898.990	36731.930
80	240275.467	31897.645	35784.151
90	238414.151	36863.288	30352.662
100	246693.713	38786.142	31826.259
200	257046.548	31383.812	37946.693
300	241913.925	30846.270	36969.745
400	250686.566	43246.331	38615.232
500	255247.494	30421.034	34466.448
600	255640.518	41051.117	38202.020
700	240621.622	42062.432	32052.996
800	234732.638	35731.239	31027.555
900	245081.605	37408.506	38177.916
1000	245697.967	34995.054	40039.803

Table 3. Comparison of Total Cost for 20 cloudlets in PSO, GSA and LIGSA-C algorithms

Iteration	PSO	GSA	LIGSA-C
10	368649.602	44876.475	44087.031
20	354478.049	41712.666	45876.723
30	345540.580	46517.096	45524.083
40	356553.370	44648.863	43818.193
50	353046.854	50765.556	49868.660
60	355353.891	47062.315	46108.799
70	351187.753	44648.863	45980.000
80	357996.221	36958.024	42285.248
90	371324.589	47707.209	39759.519
100	368448.353	43551.325	42368.312
200	359849.885	48400.000	35498.519
300	347655.150	44876.475	42082.111
400	350306.997	44648.863	46404.602
500	365131.362	39521.912	45492.859
600	366225.261	51872.510	39434.343
700	369217.780	47914.295	39521.912
800	366704.318	42284.396	46171.691
900	362957.682	47195.216	50498.534
1000	363249.969	52452.208	51730.622

The figures 2, 3 and 4 depict the graphical analysis of the total cost versus number of iterations. This comparison is performed for GSA and LIGSA-C algorithms for 10, 15 and 20 cloudlets respectively.

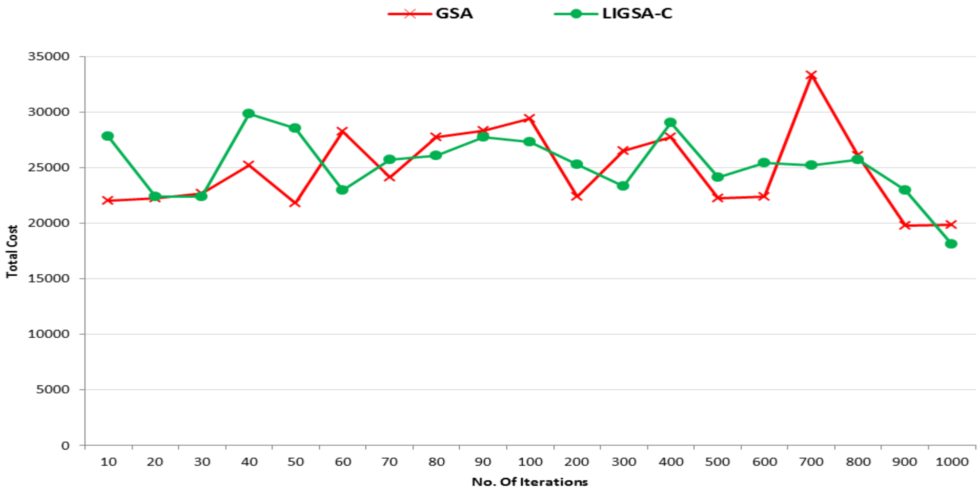

Fig.2. Comparison of Total Cost for 10 cloudlets in GSA and LIGSA-C algorithms

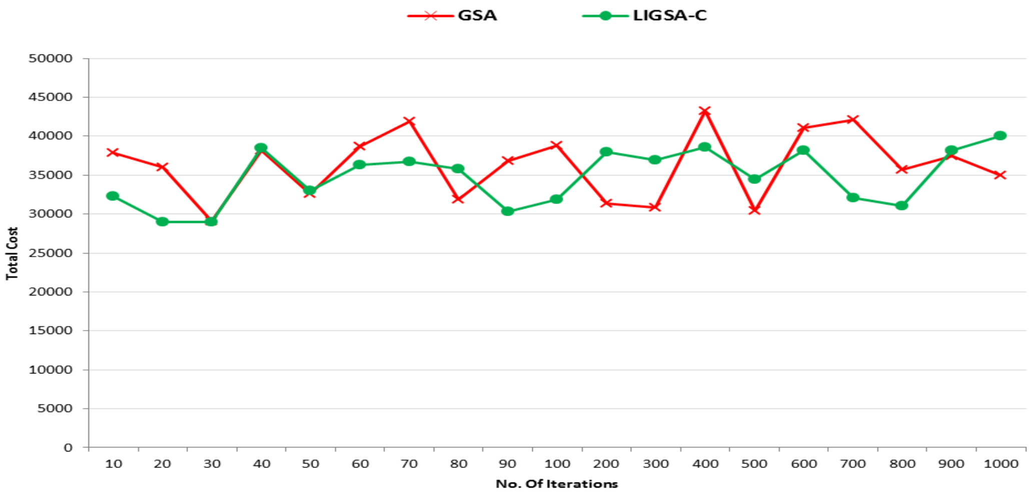

Fig.3. Comparison of Total Cost for 15 cloudlets in GSA and LIGSA-C algorithms

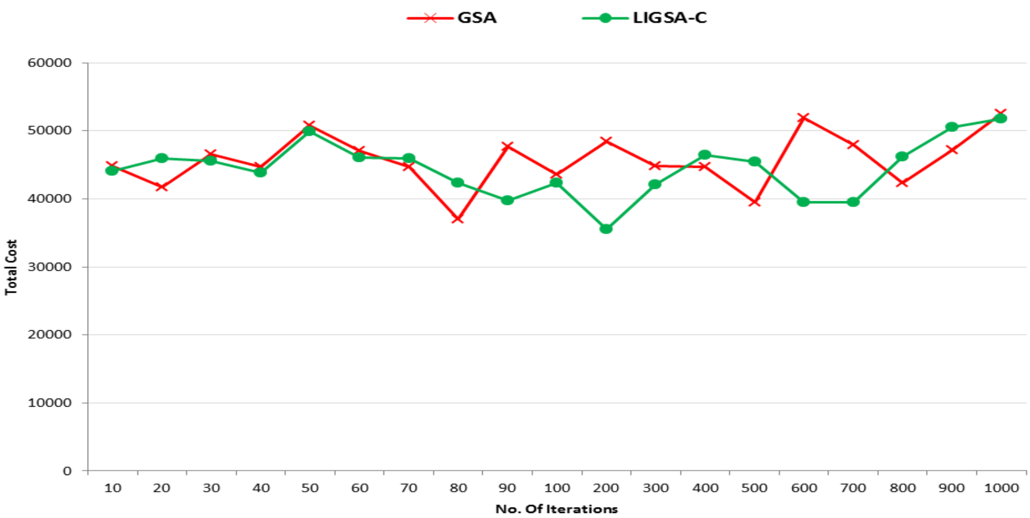

Fig.4. Comparison of Total Cost for 20 cloudlets in GSA and LIGSA-C algorithms

The comparison of the proposed Linear Improved Gravitational Search Algorithm in Cloud (LIGSA-C) approach with the Gravitational Search Algorithm (GSA) and Particle Swarm Optimization (PSO) are depicted in Figure 5 for 10 particles, Figure 6 for 15 particles and Figure 7 for 20 particles respectively.

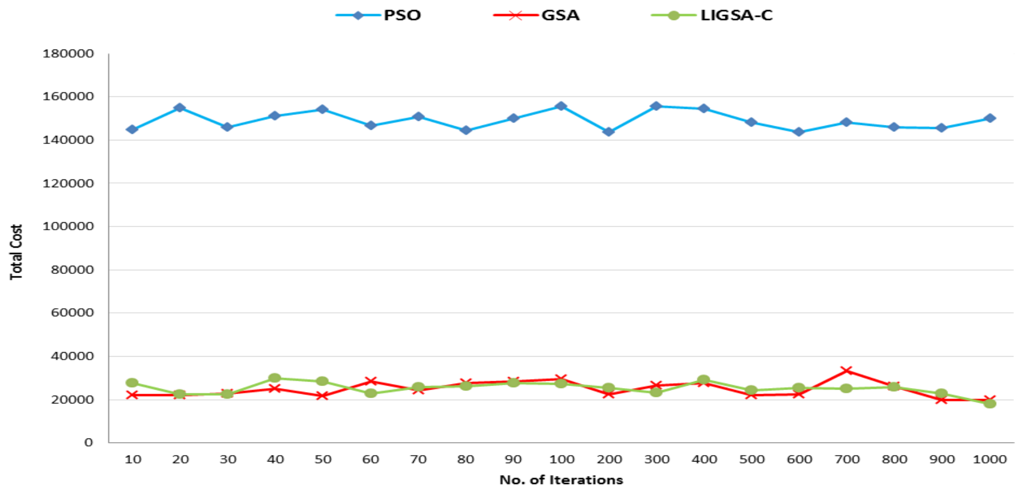

Fig.5. Comparison of Total Cost for 10 cloudlets in PSO, GSA and LIGSA-C algorithms

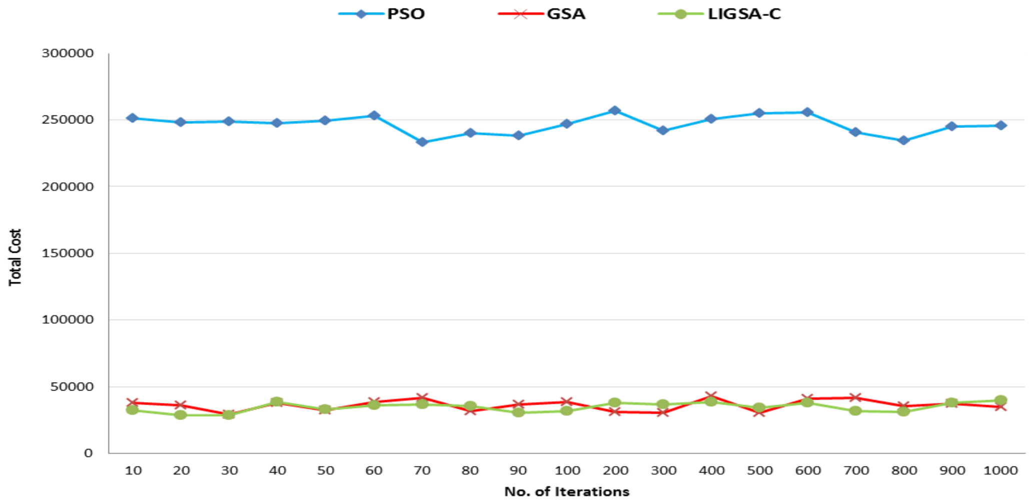

Fig.6. Comparison of Total Cost for 15 cloudlets in PSO, GSA and LIGSA-C algorithms

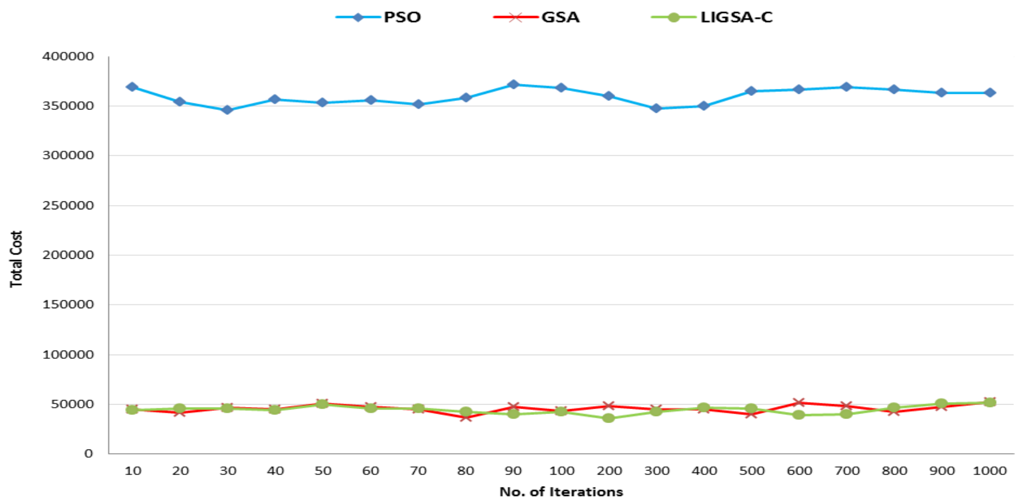

Fig.7. Comparison of Total Cost for 20 cloudlets in PSO, GSA and LIGSA-C algorithms

The experimental study on the total cost values of 10, 15 and 20 cloudlets is performed. The statistical analysis of the results generates the mean, standard deviation, minimum and maximum values are presented in Table 4.

Table 4. The Statistical Analysis of the total cost of the Cloudlets on Different Particles.by Different Algorithms

Parameters	No. of Cloudlets	PSO	GSA	LIGSA-C
Mean	10	149119.782	24842.390	25258.914
	15	246493.939	36260.437	34744.365
	20	359677.771	45663.908	44342.724
Standard Deviation	10	4187.166	3618.570	2863.5521
	15	6849.873	4286.245	3562.211
	20	7944.934	3956.376	4093.485
Minimum	10	143482.749	19760.956	18100.890
	15	233301.076	29040.000	28937.135
	20	345540.580	36958.024	35498.519
Maximum	10	155538.952	33288.008	29819.639
	15	257046.548	43246.331	40039.803
	20	371324.589	52452.208	51730.622

The figures 8 and 9 present the graphs of mean of the total cost versus number of cloudlets. This comparison is performed for GSA and LIGSA-C in Figure8 and for PSO, GSA and LIGSA-C in Figure 9. Figure 10 shows the standard deviation of total cost versus number of cloudlets for the algorithms.

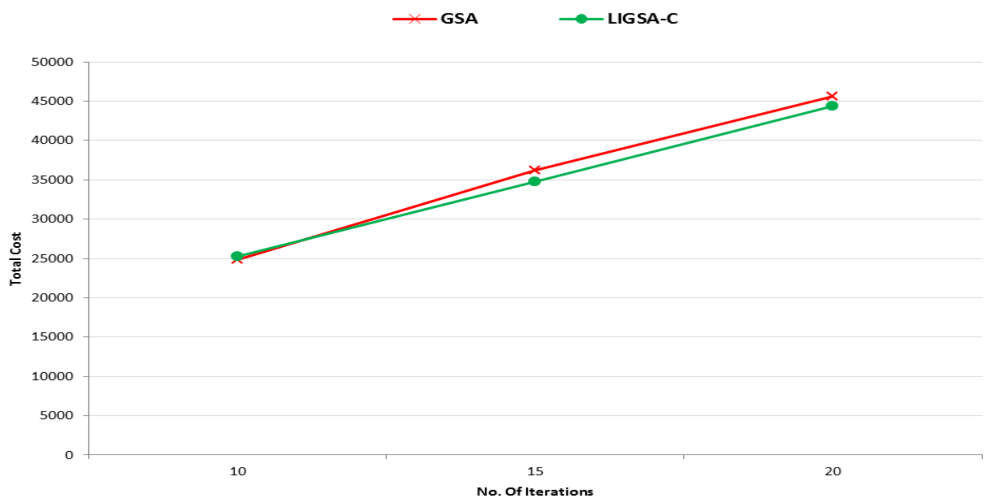

Fig.8. Comparison of Mean of Total Cost for number of cloudlets in GSA and LIGSA-C algorithms

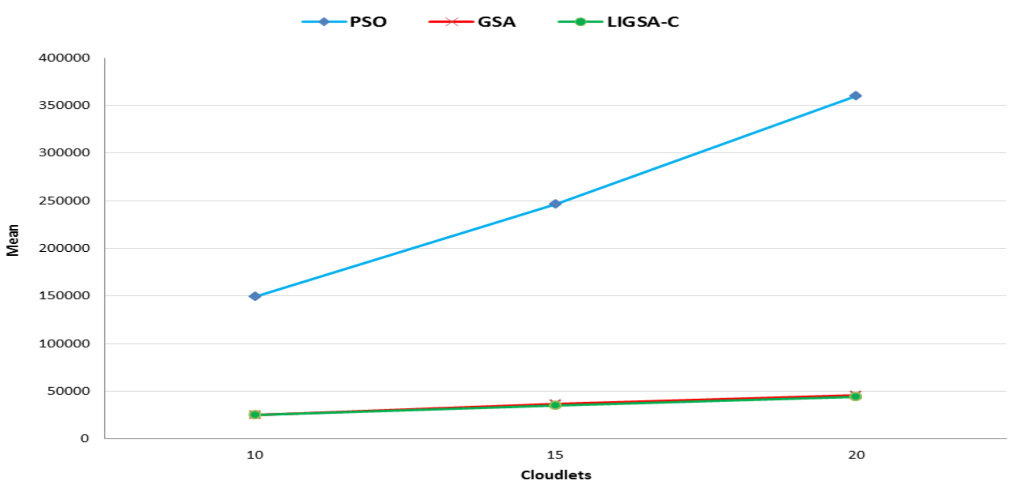

Fig.9. Comparison of Mean of Total Cost for number of cloudlets in PSO, GSA and LIGSA-C algorithms

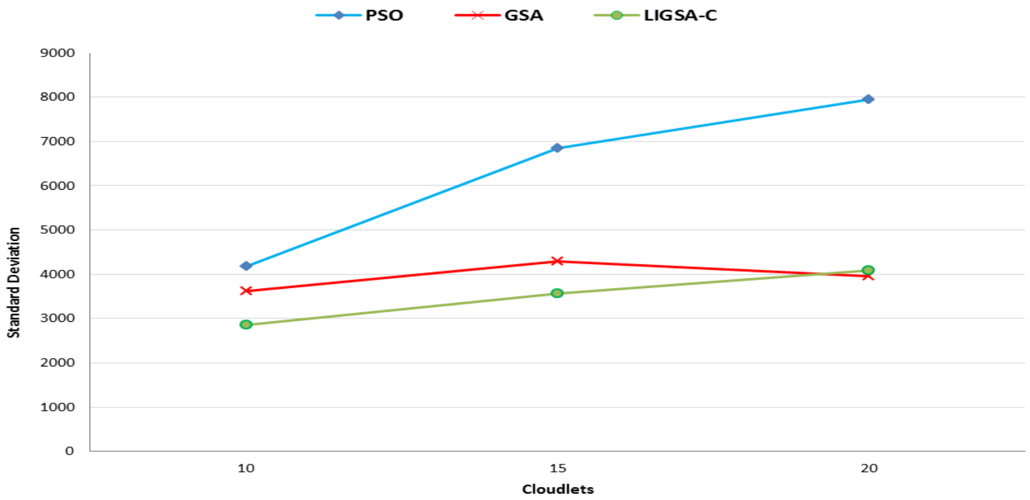

Fig.10. Comparison of Standard Deviation of Total Cost for number of cloudlets in PSO, GSA and LIGSA-C algorithms

The proposed Linear Improved Gravitational Search Algorithm in Cloud (LIGSA-C) algorithm as seen on a large set of data sets. The statistical analysis predicts that proposed algorithm performs better than the existing algorithms like gravitational search algorithm and particle swarm optimization based on the total cost of computation incurred by them. The total cost incurred in scheduling the cloudlets over the virtual machines based on set of particles. The mean and standard deviation proclaim that LIGSA-C produces lesser computational cost as seen in figure 8 and 9. Thus, by analysis we state that the Linear Improved Gravitational Search Algorithm in cloud (LIGSA-C) presented reduced total cost of computation to load scheduling problem then its counterparts like GSA and PSO.

V. Conclusion And Future Work

This paper elaborated load scheduling the prime objective in the cloud computing environment between cloudlets and VMs. The meta-heuristic based swarm intelligence techniques for load scheduling like PSO and GSA algorithm have been elaborated. The proposed Linear Improved Gravitational Search Algorithm in cloud (LIGSA-C) approach aimed to reduce the total computation cost of the system with greater VM utilization. A linear function to calculate gravitational Constant has been proposed such that search space is modified. The parallel processing of the tasks has been performed using a new cost evaluation function. The statistical analysis of the results is showcased for comparison of results. The results of the proposed approach LIGSA-C has been compared with the existing scheduling algorithms like PSO and GSA in a tabular and graphical manner. As evident, the proposed Linear Improved Gravitational Search Algorithm in cloud computing environment (LIGSA-C) produced reduced results in terms of total transfer time and total cost of execution. The future work aims to minimize the total cost by working on new swarm intelligence techniques for load scheduling.

References

[1] R. Buyya, S. Pandey, and C. Vecchiola, “Cloudbus toolkit for market-oriented cloud computing”, In CloudCom ’09: Proceedings of the 1st International Conference on Cloud Computing, volume 5931 of LNCS, pages 24–44. Springer, Germany, December 2009.

[2] J. Kennedy and R. Eberhart, “Particle swarm optimization”, In IEEE International Conference on Neural Networks, volume 4, pages 1942–1948, 1995.

[3] M. F. Tasgetiren, Y.-C. Liang, M. Sevkli, and G. Gencyilmaz, “A particle swarm optimization algorithm for makespan and total flow time minimization in the permutation flowshop sequencing problem”, In European Journal of Operational Research, 177(3):1930–1947, March 2007.

[4] C. Vecchiola, M. Kirley, and R. Buyya, “Multi-objective problem solving with offspring on enterprise clouds”, In Proceedings of the 10th International Conference on High-Performance Computing in Asia-Pacific Region (HPC Asia 2009), pages 132–139, March 2009.

[5] H. Yoshida, K. Kawata, Y. Fukuyama, and Y. Nakanishi, “A particle swarm optimization for reactive power and voltage control considering voltage stability”, In the International Conference on Intelligent System Application to Power System, pages 117–121, 1999.

[6] J. Yu, R. Buyya, and K. Ramamohanarao, “Workflow Scheduling Algorithms for Grid Computing”, volume 146, pages 173–214. Springer Heidelberg, 2008.

[7] A. E. M. Zavala, A. H. Aguirre, E. R. Villa Diharce, and S. B. Rionda, “Constrained optimisation with an improved particle swarm optimisation algorithm”, In Intl. Journal of Intelligent Computing and Cybernetics, 1(3):425–453, 2008.

[8] L. Zhang, Y. Chen, R. Sun, S. Jing, and B. Yang, “ A task scheduling algorithm based on pso for grid computing”, In International Journal of Computational Intelligence Research,4(1), 2008.

[9] http://en.wikipedia.org/wiki/Cloud_computing

[10] http://en.wikipedia.org/wiki/Load_balancing_(computing)

[11] A.Khiyaita, El Bakkali, M.Zbakh, Dafir El Kettani, “Load Balancing Cloud Computing: State of Art”, In IEEE Transactions on Software Engineering, 978-1-4673-1053-6, 2012, Pages 106-109.

[12] Suraj Pandey, Rajkumar Buyya et al, “A Particle Swarm

Optimization based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments", In 24th IEEE International Conference on Advanced Information Networking and Applications, 20-23 April 2010, Pages 400-407.

[13] Chun-Wei Tsai and Joel J. P. C. Rodrigues, "Metaheuristic Scheduling for Cloud: A Survey", In IEEE Systems Journal, Vol. 8, No. 1, March 2014, Pages 279-291

[14] Chaudhary D., Chhillar R.S., "A New Load Balancing Technique for Virtual Machine Cloud Computing Environment" in International Journal of Computer Applications 69(23) Pages 37-40, May 2013.

[15] Chaudhary, D., Kumar, B., "Analytical study of load scheduling algorithms in cloud computing", In IEEE International Conference on Parallel, Distributed and Grid Computing (PDGC), 2014, Pages: 7 - 12, DOI: 10.1109/PDGC.2014.7030706

[16] Chaudhary, D., Kumar, B., "An analysis of the load scheduling algorithms in the cloud computing environment: A survey", In IEEE 9th International Conference on Industrial and Information Systems (ICIIS), 2014, Pages:1-6, DOI:10.1109/ICIINFS.2014.7036659

[17] Mathiyalagan P, Dhepthie U, Sivanandam S., "Grid scheduling using enhanced PSO algorithm", Int Journal Computer Science Engineering 2010;02(02):140–5.

[18] Liu H, Abraham A, Hassanien A., "Scheduling jobs on computational Grids using a fuzzy particle swarm optimization algorithm", In Future Generation Comput Syst 2010;26(8):1336–43.

[19] Kang Q, He H., "A novel discrete particle swarm optimization algorithm for meta-task assignment in heterogeneous computing systems", In Microprocessor Microsystems 2011, 35(1):10–7.

[20] Izakian H, Ladani B, Abraham A, Snasel V., "A discrete particle swarm optimization approach for Grid job scheduling", In. Int J Innovative Computing Inform Control 2010;6(9):4219–33.

[21] Elina Pacini, Cristian Mateos, Carlos García Garino, "Distributed job scheduling based on Swarm Intelligence: A survey", In Computers and Electrical Engineering, 40 (2014), 252–269, 2013 Elsevier Ltd.

[22] Garg S. K., and Buyya R., "Network CloudSim: Modelling Parallel Applications in Cloud Simulations", In 4th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2011, IEEE CS Press, USA), Melbourne, Australia, 2011.

[23] Rashedi E, Nezamabadi-pour H, Saryazdi S, "GSA: A Gravitational Search Algorithm", In Information Sciences, 179 (2009) 2232–2248, Elsevier.

[24] Rashedi E, Nezamabadi-pour H, Saryazdi S, "Filter modeling using gravitational search algorithm", In Engineering Applications of Artificial Intelligence 24 (2011) 117–122, Elsevier.

[25] Dinesh Kumar, Zahid Raza, "A PSO Based VM Resource Scheduling Model for Cloud Computing", In IEEE International Conference on Computational Intelligence & Communication Technology (CICT), 2015, Pages: 213 – 219, DOI: 10.1109/CICT.2015.35.

[26] Shaminder Kaur, Amandeep Verma, "An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment", In International Journal of Information Technology and Computer Science, vol. 10, 74-79, 2012.

[27] Sumit Goyal, "Public vs Private vs Hybrid vs Community -Cloud Computing: A Critical Review", In International Journal of Computer Network and Information Security, vol. 3, 20-29, 2014.

[28] Md. Imran Alam, Manjusha Pandey, Siddharth S Rautaray, "A Comprehensive Survey on Cloud Computing", In International Journal of Information Technology and Computer Science, vol. 02, 68-79, 2015.

Authors' Profiles

Divya Chaudhary received her Bachelor of Technology in Information Technology in 2011 and her Master of Technology in Computer Science in 2013. She is pursuing her Ph.D. in Department of Computer Engineering, Netaji Subhas Institute of Technology, New Delhi, India. Her research interest includes cloud computing and optimization analysis.

Bijendra Kumar did his Bachelor of Engineering from H.B.T.I. Kanpur, India. He has done his Ph.D. from Delhi Universi ty, Delhi, India. Presently he is working as Professor in Computer Engineering Division, Netaji Subhas Institute of Technology, University of Delhi, India. His areas of research interests are Video applications, Watermarking, Design of algorithms, Wireless Sensor Networks and Cloud Computing.

How to cite this paper: Divya Chaudhary, Bijendra Kumar,"Linear Improved Gravitational Search Algorithm for Load Scheduling in Cloud Computing Environment (LIGSA-C)", International Journal of Computer Network and Information Security(IJCNIS), Vol.10, No.4, pp.38-47, 2018.DOI: 10.5815/ijcnis.2018.04.05