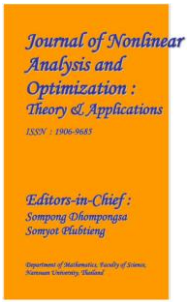
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1, No.1 : 2024
ISSN : **1906-9685**



IMPROVED OPTIMIZATION BASED TASK SCHEDULING ALGORITHM FOR CLOUD COMPUTING ENVIRONMENT

Mrs. O.DEVIPRIYA, Research Scholar, Department of Computer Science, Mother Teresa Women's University, Kodaikanal.
Dr. K.KUNGUMARAJ, Assistant Professor, Department of Computer Science,A.P.A. College Palani.

ABSTRACT

The rapid development in the computation and storage facilities of the Internet, cloud computing (CC) resources have become cheap, dominant, and globally accessible. In CC, task scheduling acts as a significant role to distribute the requested tasks to the resources with the aim of increasing the exploitation rate and minimization task execution time. Task scheduling is formulated as an optimization problem and can be resolved by metaheuristics. With this motivation, this paper designs a new seagull optimization algorithm-based task scheduling (SGOA-TS) technique for CC environment. The SGOA-TS technique models the interesting behaviour of the seagulls in ocean into the task scheduling process in CC. The SGOA-TS technique generates a fitness function involving different parameters related to the CC environment. A brief set of simulations takes place to ensure the better performance of the SGOA-TS technique. The experimental outcomes showcased the improved performance of the SGOA-TS technique over the recent techniques.

Keywords: Cloud computing, Load balancing, NP hard problem, SGO algorithm, Task scheduling, Task completion

1. INTRODUCTION

Cloud computing (CC) is one among the popular technologies that exist in the domain of distributed computing. It finds useful in application areas comprising data storage, data analytics, and IoT applications [1, 2]. It has considerably modified the conventional ways where the services are provided by companies or persons. It gives distinct kinds of services to enrolled clients as web services so that the clients do not require investing in the computation infrastructures. The CC offers services like IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service) [3]. For each kind of service, the clients are anticipated to offer requests to the service provider via the Internet medium. The CSP handles the process of resource management for fulfilling the requests created by the clients. The CSP uses scheduling techniques to allocate the requests and handle the computation resources proficiently.

Service Providers utilize scheduling techniques for scheduling the received requests (tasks) and for managing their computing resources effectually. Task Scheduling (TS) and resource management permit provider for maximizing revenue and the consumption of resources up to it restricts. Actually, with respect to the efficiency of CC resources, the scheduling and distribution of resources are essential hurdles. Therefore, the investigators are concerned with analysis of TS in CC. TS is the model of organizing incoming requests (task) in particular approach and so the accessible resource is appropriately employed. As CC has the technologies which deliver service with medium of Internet, services user can submit its request online. Since all services have a amount of users, the amount of requests (tasks) can be created at a time [4]. The system which doesn't utilize scheduling can feature longer waiting period to task furthermore, a few short-term tasks can terminate, because of waiting period. Fig. 1 demonstrates the overview of TS model.

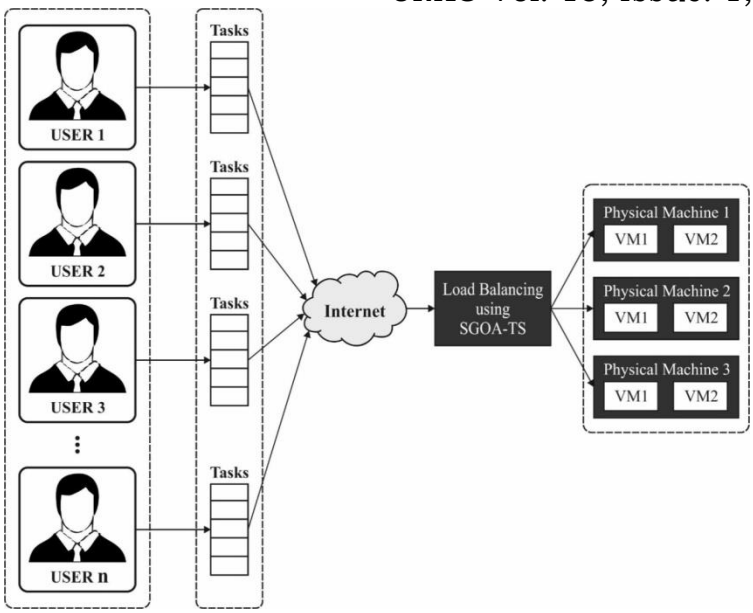


Fig. 1. Overview of task scheduling process

The Cloud Service (CS) end users could utilize the total stack of computing services that range from hardware to application. The service in CC utilizes a pay-as-you-go beginning. The CS end users reduce or improve the accessible resource, per the demand of application. The most benefits of CC, then services user could be responsible to pay further cost for this benefit. The CS users will rent the resource at some time and released them without effort. In CS users is the freedom for employing some service dependent on applications required. The freedom of services optimal for users have managed to problem which is the next user request could not be exactly forecasted. So, the TS and resource distributions are needed parts of CC analysis. The performance of resources utilizes dependent upon scheduling as well as load balancing (LB) techniques before the arbitrary distribution of resources. In CC was extremely utilized to resolve difficult tasks (user request). In resolving difficult task problems, utilize of scheduling techniques is suggested. While the scheduling techniques leverage the resource.

In [5], a novel enhanced scheduling in WRR (Weighted Round Robin) for cloud framework service is proposed which regards length of job and resource potential. It can be utilized to reduce the response time by effectual application of VM (Virtual Machine) in the application of static and dynamic scheduling procedures by determining the job length and resource capabilities and effectual forecast of VM and removing the excess of VM. The multi-level and autonomous task is considered. LB (load Balancing) in extremely load case for task migration is unconsidered. In [6–8], scheduling technique depends on task in grid is proposed. An effectual mapping of DAG (Directed Acyclic Graph) depends on applications are presented by [9, 10]. It can be dependent upon the list scheduling technologies. The non-critical path earliest-finish scheduling technique was presented as [11]. It suggests that maximum efficiency is obtained utilizing heterogeneous computing platforms. Similar kinds of problems are determined as [12]. The stochastic hill climbing technique is implemented for load distributions in CC [13], in which soft computing depends on technologies are related to RR and FCFS. The dynamic workflows scheduling manner for grid and CC platforms is projected [14] that decreases the workflow execution time and minimizes the scheduling works. In Genetic Algorithm (GA) called Chaos GA as projected by [15] for resolving the scheduling problems by considering user budget as well as deadline. It creates optimum outcomes with restricted time period.

With this motivation, this paper designs a new seagull optimization algorithm-based task scheduling (SGOA-TS) technique for CC environment. The SGOA-TS technique models the interesting behaviour of the seagulls in ocean into the task scheduling process in CC. The SGOA-TS technique generates a fitness function involving different parameters related to the CC environment. A brief set of simulations takes place to ensure the better performance of the SGOA-TS technique.

2. PROPOSED SGOA-TS APPROACH

The projected SGOA-TS approach goals for reducing the calculation difficulty and assign the loads to VM effectively. The calculation cost contains the transfer cost and implementation cost in cloudlet. It gives enhanced search space consumption and client approval on the other techniques. The projected SGO manner develops the FF. The cloudlet and VM parameters are bandwidth, MIPS, execution cost, and transfer cost that is employed to validate the value of FF. The distribution of cloudlet on VMs is implemented. The data centre hold$(VMs)^{Cloudlets}$ possible techniques of applying the cloudlets on respective VM. When 3 cloudlets are located on 2 VMs, then the probability develops 8. The butterflies S endure initialization at CloudSim tool as deword menstruated in Eq. (1):

$$S_i = \left(s_i^1, s_i^2, \dots\dots, s_i^n, \dots\dots\dots, s_i^d\right)$$
$$\forall i = 1\ to\ 25\ and\ n = 1\ to\ 10 \quad (1)$$

The FF estimates the fitness value of butterflies in the exploring area. An initial butterfly endures initialization and the selection of successive butterflies occurs by utilizing of optimum fitness value. It is mostly dependent upon bandwidths, MIPS, implementation cost, and transfer cost of cloudlet and VM. Let $\mathrm{C}t_{exec}(M)_j$ implies the entire cost of implementation of butterfly assigned for computing the VM resource PCj. It can be calculated as summing up of every weight allocated to maps of butterfly S of the cloudlets assigned to all resources. Assume $\mathrm{C}t_{frons}(M)_j$ represents the total transfer cost among the cloudlet assigned for determining the VM resources PCj. The outcome would be product of output file size and broadcast cost. The average cost of data amongst the pair of resources of transmission is provided as $dS(k1), S(\mathrm{k2})$ and the butterfly does not rely on one another. The whole cost was joined to every butterfly S by implementation and transfers cost and afterward, the cost is besides reduced for computing the fitness value that defined as:

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

$$\mathrm{C}t_{trans}(S)_j = \sum_{\mathrm{k1}\in T} \sum_{\mathrm{k2}\in T} d_{S(\mathrm{k1}),S(\mathrm{k2})} * e_{k1,\mathrm{k2}}, \quad (3)$$

$$\forall S(k1) = jandS(1\langle 2) \neq j \quad (4)$$

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

$$Cost_{Total}(S) = \max\left(\mathrm{C}t_{\mathrm{total}}(S)_j\right), \forall j \in S \quad (6)$$

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

Seagull is omnivorous and eats insects, fish, reptiles, amphibians, earthworms, etc. The seagull has a special pair of glands exact on its eye that is specially planned for flushing the salt to its system with openings in the bill. The migration is determined as seasonal drive of seagulls from one place to another for finding the richest and most abundant feed source which gives suitable energy [16]. The seagulls regularly attack migrate birds on the sea if they can be migrating from one place to another. It creates its spiral natural shape effort in attack. These performances are formulated in manner which it is connected with objective functions that exist optimized. This creates it feasible for formulating a novel optimization technique. This paper concentrates on 2 natural performances of seagulls. The mathematical methods of migrating as well as attack the prey. In the migration, the technique pretends the set of seagulls move nearby one place to another. During this stage, the seagull must fulfil 3 situations:

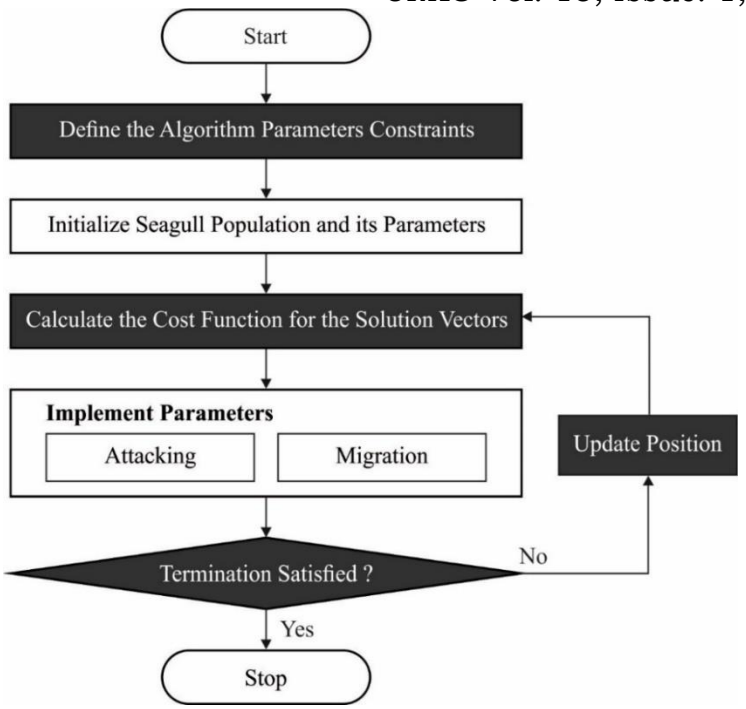


Fig. 2. Flowchart of SGO

For avoiding the collision amongst neighbors (for instance, other seagulls), the additional variable A was utilized for computation of novel search agent place. Fig. 2 illustrates the flowchart of SGO technique.

$$\vec{C}_s = A \times \vec{P}_s(x) \quad (8)$$

where $\vec{C}_s$ implies the place of search agent that doesn't collide with other search agents, $\vec{P}_s$ signifies the current place of search agents, x represents the present iterations, and A stands for the movement performance of search agents in provided search spaces.

$$A = f_c - \left(x \times (f_c / Max_{iteration})\right) \quad (9)$$

where: $x = 0, 1, 2, \ldots, \text{Max}_{iteration}$, f_c represents the established for controlling the frequency of utilizing variables A that is linearly reduced from f_c to 0. Afterward, the avoiding the collision amongst neighbors, the search agents are move near the way of optimum neighbor.

$$\vec{M}_s = B \times \left(\vec{P}_{bs}(x) - \vec{P}_s(x)\right) \quad (10)$$

where $\vec{M}_s$ implies the places of search agents $\vec{P}_s$ near the optimum fit search agents $\vec{P}_{bs}$ (for instance, fittest seagull). The performance of B has the randomized that is responsible for appropriate balancing amongst exploration and exploitation. B has computed as:

$$B = 2 \times A^2 \times rd \quad (11)$$

where rd represents the arbitrary number lies in range of 0 and 1. Finally, the search agents upgrade their place interms of optimum search agents.

$$\vec{D}_s = |\vec{C}_s + \vec{M}_s| \quad (12)$$

where $\vec{D}_s$ signifies the distance amongst the search agent and optimum fit search agents (for instance, the optimum seagull whose fitness value is minimum). The exploitation proposes exploiting the history and experience of search processes. The seagulls are altering the angle of attack continuously and speed in migration. It can continue its altitude utilizing its wings as well as weight. But, the attacked the prey, the spiral movement performance arises in the air. This performance, y, and z planes are explained as:

$$x' = r \times cos(k) \quad (13)$$
$$y' = r \times sin(k) \quad (14)$$
$$z' = r \times k \quad (15)$$
$$r = u \times e^{kv} \quad (16)$$

where r implies the radius of all turns of spiral, k represents the arbitrary number in range $[0 \leq k \leq 2\pi]$. u and v are constants for defining the spiral shape, and e represents the base of the natural logarithm. The upgraded place of search agent is computed utilizing Eqs. (12)-(16).

$$\vec{P_s}(x) = \left(\vec{D_s} \times x' \times y' \times z'\right) + \vec{P}_{bs}(x) \qquad (17)$$

where $\vec{P_s}(x)$ avoids the optimum solutions and upgrades the place of other search agents. The presented SOA begins with arbitrary created populations. The search agents are up grading their places in terms of optimum search agents in the iteration processes. A implies the linearly reduced from f_c to 0. In order to smooth change amongst exploration and exploitation, variable B is responsible. Therefore, the SOA is regarded as global optimization due to its optimum exploration as well as exploitation ability.

3. PERFORMANCE VALIDATION

This section examines the task scheduling performance of the SGOA-TS technique in terms of different dimensions. Table 1 showcases the execution time analysis of the SGOA-TS technique with other techniques under varying iterations. Fig. 3 assesses the execution time of the SGOA-TS technique with other methods under PM = 5, VM = 15, and Task = 50. From the figure, it is showcased that the SGOA-TS technique has gained least execution time over the other methods. For instance, with 10 iterations, the SGOA-TS technique has offered a minimum execution time of 0.29 whereas the FA, DA, ADA, and QBMBO algorithms have provided a maximum execution time of 0.63, 0.42, 0.38, and 0.32 respectively. Besides, with 50 iterations, the SGOA-TS technique has obtained a decreased execution time of 0.35 whereas the FA, DA, ADA, and QBMBO algorithms have offered an increased execution time of 0.69, 0.48, 0.42, and 0.37 respectively.

Fig. 4evaluate the execution time of the SGOA-TS method with other approaches under PM = 10, VM = 30 and Task = 75. From the figure, it can be portrayed that the SGOA-TS manner has reached worse execution times over the other algorithms. For instance, with 10 iterations, the SGOA-TS technique has offered a minimal execution time of 0.61 whereas the FA, DA, ADA, and QBMBO methods have providing a superior execution time of 0.93, 0.86, 0.73, and 0.64 correspondingly. Finally, with 50 iterations, the SGOA-TS algorithm has attained a reduced execution time of 0.66 whereas the FA, DA, ADA, and QBMBO methodologies have offered a maximum execution time of 0.95, 0.89, 0.78, and 0.70 correspondingly.

Table 1 Execution Time Analysis

PM = 5, VM = 15 and Task = 50					
Iterations	**FA**	**DA**	**ADA**	**QBMBO**	**SGOA-TS**
10	0.63	0.42	0.38	0.32	0.29
20	0.65	0.43	0.39	0.33	0.30
30	0.67	0.45	0.40	0.34	0.32
40	0.68	0.47	0.41	0.35	0.34
50	0.69	0.48	0.42	0.37	0.35

PM = 10 and VM = 30 and Task = 75					
Iterations	**FA**	**DA**	**ADA**	**QBMBO**	**SGOA-TS**
10	0.93	0.86	0.73	0.64	0.61
20	0.95	0.87	0.75	0.65	0.62
30	0.95	0.87	0.76	0.66	0.65
40	0.95	0.88	0.77	0.76	0.73
50	0.95	0.89	0.78	0.70	0.66

PM = 20 and VM = 50 and Task = 100					
Iterations	**FA**	**DA**	**ADA**	**QBMBO**	**SGOA-TS**
10	0.95	0.90	0.78	0.71	0.68
20	0.98	0.90	0.79	0.73	0.69
30	0.99	0.93	0.80	0.75	0.71
40	0.99	0.95	0.80	0.76	0.74
50	0.99	0.96	0.81	0.78	0.75

Fig. 5consider the execution time of the SGOA-TS manner with other methods under PM = 20, VM = 50 and Task = 100. From the figure, it is outperformed that the SGOA-TS algorithm has attained minimum execution time over the other methods. For sample, with 10 iterations, the SGOA-TS approach has offered a lower execution time of 0.68 whereas the FA, DA, ADA, and QBMBO methods have provided an improved execution time of 0.95, 0.90, 0.78, and 0.71 respectively. Moreover, with 50 iterations, the SGOA-TS manner has obtained a minimal execution time of 0.75 whereas the FA, DA, ADA, and QBMBO algorithms have offered a superior execution time of 0.99, 0.96, 0.81, and 0.78 correspondingly.

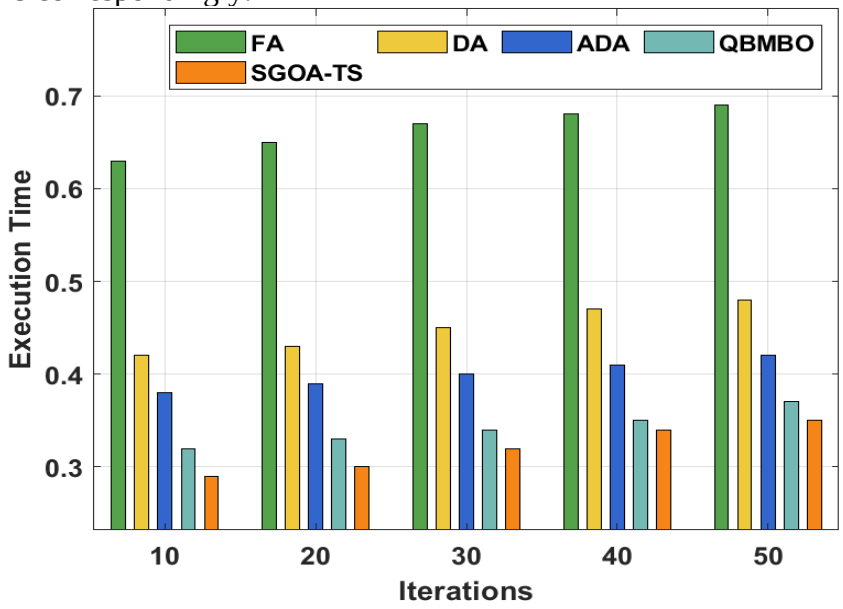


Fig. 3. Execution time analysis of SGOA-TS model under PM = 5, VM = 15 and Task = 50

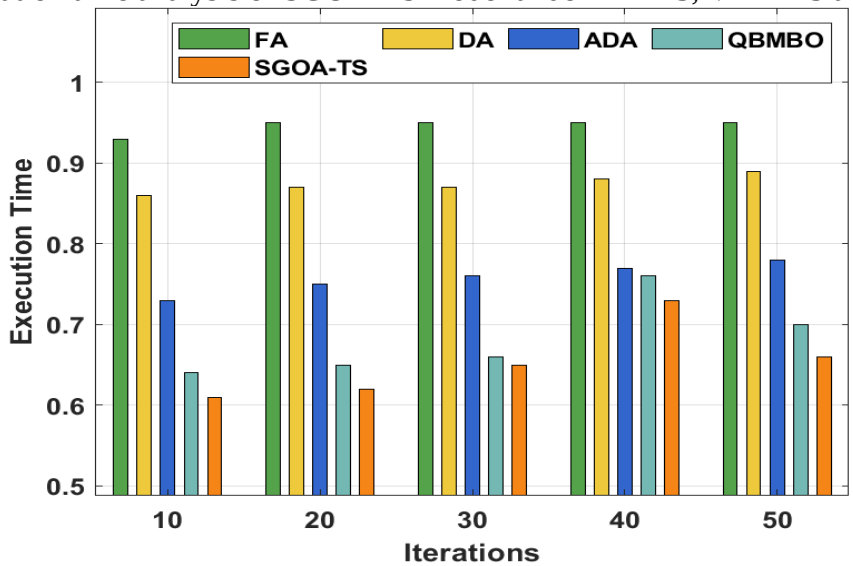


Fig.4. Execution time analysis of SGOA-TS model under PM = 10 and VM = 30 and Task = 75

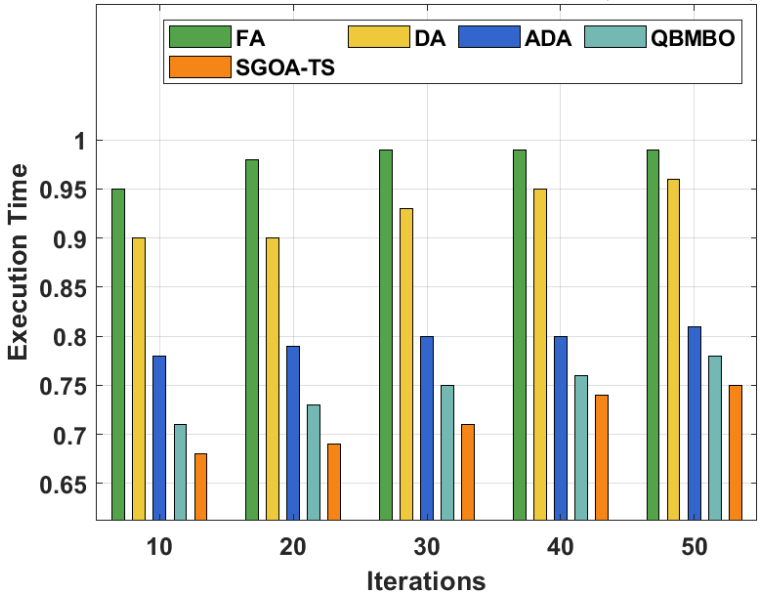


Fig. 5. Execution time analysis of SGOA-TS model under PM = 20 and VM = 50 and Task = 100

Table 2 illustrates the execution cost analysis of the SGOA-TS method with other approaches under different iterations.

Fig. 6 assesses the execution cost of the SGOA-TS method with other algorithms under PM = 5, VM = 15, and Task = 50. From the figure, it can be exhibited that the SGOA-TS technique has reached worse execution costs than the other techniques. For sample, with 10 iterations, the SGOA-TS method has offered a minimal execution cost of 0.64 whereas the FA, DA, ADA, and QBMBO algorithms have given an increased execution cost of 0.80, 0.76, 0.75, and 0.68 correspondingly. Furthermore, with 50 iterations, the SGOA-TS technique has obtained a decreased execution cost of 0.64 whereas the FA, DA, ADA, and QBMBO methods have offered an increased execution cost of 0.80, 0.77, 0.76, and 0.67 correspondingly.

Fig. 7calculate the execution cost of the SGOA-TS technique with other techniques under PM = 10, VM = 30 and Task = 75. From the figure, it can be shown that the SGOA-TS manner has attained least execution cost over the other techniques. For instance, with 10 iterations, the SGOA-TS technique has offered a minimum execution cost of 0.04 whereas the FA, DA, ADA, and QBMBO methods have provided a maximal execution cost of 0.19, 0.12, 0.08, and 0.05 correspondingly. Followed by, with 50 iterations, the SGOA-TS technique has obtained a lower execution cost of 0.07 whereas the FA, DA, ADA, and QBMBO methodologies have offered a maximum execution cost of 0.28, 0.16, 0.13, and 0.09 respectively. measures the execution cost of the SGOA-TS manner with other techniques under PM = 20, VM = 50, and Task = 100.

Table 2 Execution Cost Analysis

PM = 5, VM = 15 and Task = 50					
Iterations	**FA**	**DA**	**ADA**	**QBMBO**	**SGOA-TS**
10	0.80	0.76	0.75	0.68	0.64
20	0.70	0.67	0.60	0.62	0.60
30	0.79	0.78	0.74	0.65	0.62
40	0.80	0.77	0.74	0.64	0.63
50	0.80	0.77	0.76	0.67	0.64
PM = 10 and VM = 30 and Task = 75					
Iterations	**FA**	**DA**	**ADA**	**QBMBO**	**SGOA-TS**
10	0.19	0.12	0.08	0.05	0.04
20	0.22	0.13	0.08	0.05	0.04
30	0.26	0.14	0.08	0.05	0.04

40	0.28	0.15	0.09	0.06	0.05
50	0.28	0.16	0.13	0.09	0.07
PM = 20 and VM = 50 and Task = 100					
Iterations	**FA**	**DA**	**ADA**	**QBMBO**	**SGOA-TS**
10	0.25	0.15	0.11	0.07	0.05
20	0.28	0.16	0.12	0.08	0.07
30	0.29	0.17	0.13	0.09	0.08
40	0.32	0.18	0.13	0.09	0.08
50	0.33	0.19	0.14	0.10	0.09

Fig. 8 From the figure, it can be demonstrated that the SGOA-TS method has attained worse execution costs than the other methods. For instance, with 10 iterations, the SGOA-TS method has offered a minimal execution cost of 0.05 whereas the FA, DA, ADA, and QBMBO algorithms have provided a maximum execution cost of 0.25, 0.15, 0.11, and 0.07 respectively. Besides, with 50 iterations, the SGOA-TS technique has obtained a minimum execution cost of 0.09 whereas the FA, DA, ADA, and QBMBO algorithms have offered a higher execution cost of 0.33, 0.19, 0.14, and 0.10 correspondingly.

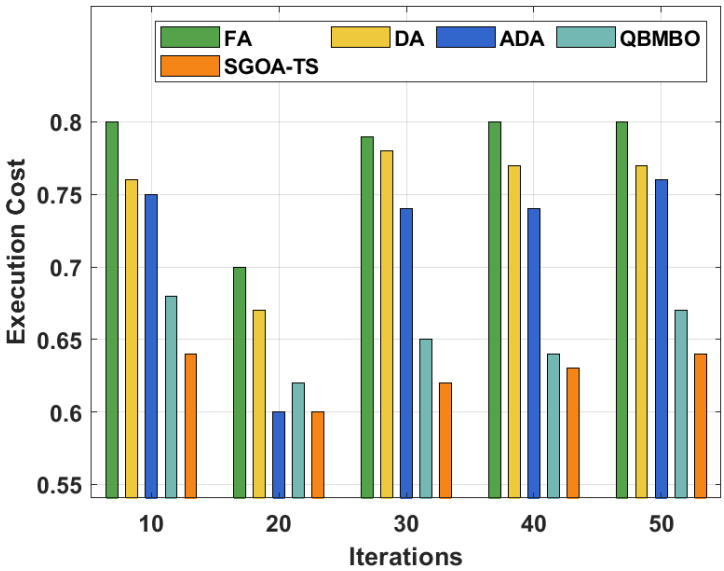


Fig. 6. Execution cost analysis of SGOA-TS model under PM = 5, VM = 15 and Task = 50

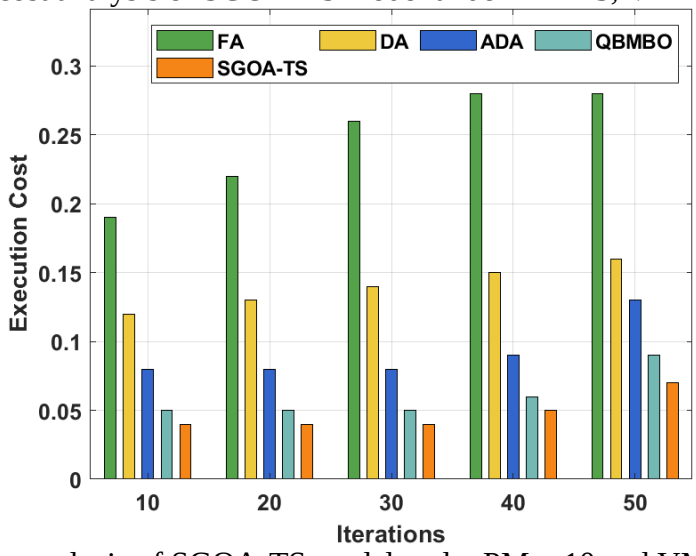


Fig. 7. Execution cost analysis of SGOA-TS model under PM = 10 and VM = 30 and Task = 75

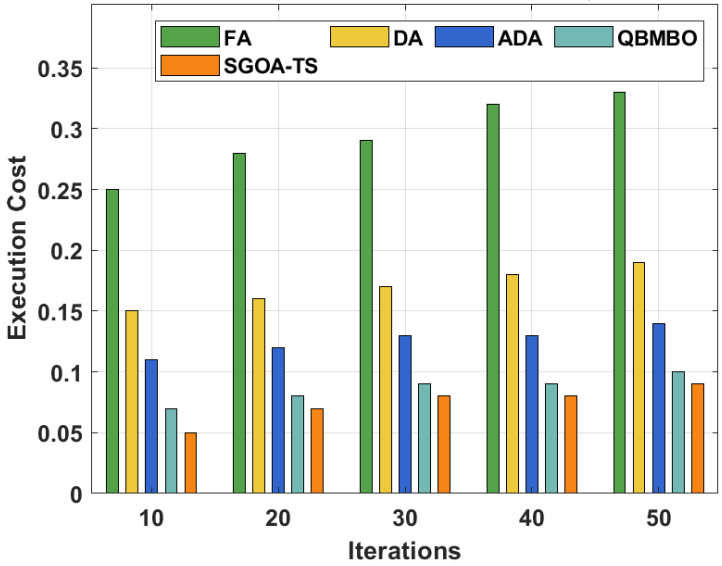


Fig. 8. Execution cost analysis of SGOA-TS model under PM = 20 and VM = 50 and Task = 100

4. CONCLUSION

This paper has designed a new SGOA-TS technique to appropriately scheduling the tasks in the CC environment in such a way that the complete resource utilization can be accomplished based on the requested tasks. The SGOA-TS technique models the interesting behaviour of the seagulls in ocean into the task scheduling process in CC. The SGOA-TS technique generates a fitness function involving different parameters related to the CC environment. A brief set of simulations takes place to ensure the better performance of the SGOA-TS technique. In future, a hybridization of metaheuristic optimization algorithms can be derived to accomplish proficient load balancing in CC environment.

REFERENCES

[1] Gawali, M.B. and Shinde, S.K., 2018. Task scheduling and resource allocation in cloud computing using a heuristic approach. *Journal of Cloud Computing*, *7*(1), pp.1-16.

[2] Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of things (iot): a vision, architectural elements, and future directions. FuturGenerComputSyst 29(7):1645–1660

[3] Mezmaz M, Melab N, Kessaci Y, Lee YC, Talbi E-G, Zomaya AY, Tuyttens D (2011) A parallel bi-objective hybrid meta heuristic for energy-aware scheduling for cloud computing systems. J Parallel Distributed Computing 71(11):1497–1508

[4] Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I et al (2010) A view of cloud computing. Commun ACM 53(4):50–58

[5] R. Basker, V. RhymendUthariaraj, and D. Chitra Devi, "An enhanced scheduling in weighted round robin for the cloud infrastructure services," International Journal of Recent Advance in Engineering & Technology, vol. 2, no. 3, pp. 81–86, 2014.

[6] Z. Yu, F. Menng, and H. Chen, "An efficient list scheduling algorithm of dependent task in grid," in Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT '10), IEEE, Chengdu, China, July 2010.

[7] H. M. Fard and H. Deldari, "An economic approach for scheduling dependent tasks in grid computing," in Proceedings of the 11th IEEE International Conference on Computational Science and Engineering (CSE Workshops '08), pp. 71–76, IEEE, San Paulo, Brazil, July 2008

[8] W. Kadri, B. Yagoubi, and M. Meddeber, "Efficient dependent tasks assignment algorithm for grid computing environment," in Proceedings of the 2nd International Symposium on Modelling and Implementation of Complex Systems (MISC '12), Constantine, Algeria, May 2012

[9] S. Ijaz, E. U. Munir, W. Anwar, and W. Nasir, "Efficient scheduling strategy for task graphs in heterogeneous computing environment," The International Arab Journal of Information Technology, vol. 10, no. 5, 2013

[10] Y. Xu, K. Li, L. He, and T. K. Truong, "A DAG scheduling scheme on heterogeneous computing systems using double molecular structure-based chemical reaction optimization," Journal of Parallel and Distributed Computing, vol. 73, no. 9, pp. 1306–1322, 2013.
[11] L.-T. Lee, C.-W. Chen, H.-Y. Chang, C.-C. Tang, and K.-C. Pan, "A non-critical path earliest-finish algorithm for interdependent tasks in heterogeneous computing environments," in Proceedings of the 11th IEEE International Conference on High Performance Computing and Communications (HPCC '09), pp. 603–608, Seoul, Republic of Korea, June 2009.
[12] B. Xu, C. Zhao, E. Hu, and B. Hu, "Job scheduling algorithm based on Berger model in cloud environment," Advances in Engineering Software, vol. 42, no. 7, pp. 419–425, 2011
[13] B. Mondal, K. Dasgupta, and P. Dutta, "Load balancing in cloud computing using stochastic hill climbing-a soft computing approach," Procedia Technology, vol. 4, pp. 783–789, 2012.
[14] M. Rahman, R. Hassan, R. Ranjan, and R. Buyya, "Adaptive workflow scheduling for dynamic grid and cloud computing environment," Concurrency and Computation: Practice and Experience, vol. 25, no. 13, pp. 1816–1842, 2013.
[15] G. Gharooni-fard, F. Moein-darbari, H. Deldari, and A. Morvaridi, "Scheduling of scientific workflows using a chaosgenetic algorithm," Procedia Computer Science, vol. 1, no. 1, pp. 1445–1454, 2010, International Conference on Computational Science, ICCS 2010.
[16] Dhiman, G. and Kumar, V., 2019. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowledge-Based Systems*, *165*, pp.169-196.