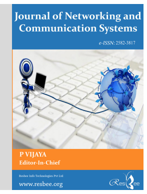
Vol. 5 No.3 July 2022

Journal of Networking and Communication Systems

Received 19 April, Revised 15 June, Accepted 10 July



A Novel Hybrid Optimization Model for Task Scheduling in Cloud Environment

Seelam Mrudula Naidu

Department of CSE

MVSR Engineering College, Hyderabad, Telangana, India

Abstract: Resource management is a fundamental design issue for big data processing systems in the cloud". Various resource allocation strategies pose diverse impacts on fairness and performance. Thus, task scheduling is said to be an effective and classic method to enhance resource efficiency in the cloud. Since several big data appliances with diverse resource requirements are deployed in the cloud, there were huge prospects for allocating resources. However, no clear solution was developed for tackling the issues that occur during task scheduling. Therefore, this paper aims to introduce a new task scheduling model using a novel hybrid optimization algorithm termed Sea Lion Updated Grey Wolf Optimization Algorithm (SU-GWO). The newly developed model takes account of multi-objective functions like security and makespan that enhances the allocation of tasks in virtual machines (VM). In the end, the performance of the presented approach is examined by determining make span, security analysis, and total cost as well.

Keywords: Cloud Computing; Make Span; Security; Task Scheduling; SU-GWO Model.

Nomenclature

Abbreviation	Description
ANN	Artificial Neural Network
IaaS	Infrastructure As A Service
CSSA	Chaotic Squirrel Search Algorithm
VM	Virtual Machine
SLnO	Sea Lion Optimization
SU-GWO	Sea Lion Updated Grey Wolf Optimization Algorithm
QoS	Quality of Service
PSO	Particle Swarm Optimization
GA	Genetic Algorithm
EDA	Estimation of Distribution Algorithm

1. Introduction

"Cloud computing is a large-scale distributed computing concept and it includes the development of the conventional grid computing and distributed computing". Task scheduling exists as a remarkable issue in cloud surroundings [1] [2]. It is deployed for scheduling the tasks for utilizing the resources in an enhanced manner. Accordingly, particular tasks should be allocated to specific resources for the better scheduling of the jobs [6] [7]. The major objective of task scheduling is to improve the QoS and performance and in addition, it maintains the efficacy among the tasks and thus minimizes the cost. The available virtual resources are optimally deployed in task scheduling [8] [9] [10] [11]. By effective scheduling of resources, a better cloud computing environment can be created. The different constraints that are regarded in scheduling methods are task completion cost, and completion time [12] [13].

Moreover, the traditional task-scheduling approaches concern more with CPU memory, task-resource needs, execution cost, and execution time. In cloud computing, task scheduling is considered an "NP-hard problem", which can be computed through certain heuristic schemes namely, PSO. Nowadays, the scheduling of various tasks using the available resources is attaining larger attention in cloud computing owing to the improved performance in task execution. Unluckily, scheduling approaches in conventional distributed computing could not work effectively since it is large-scale and dynamic [14].

To date, evolutionary algorithms like GA and EDA were developed for resolving numerous mapping as well as scheduling problems [15] [16]. Single-objective optimization approaches mostly employ conventional scheduling schemes like "Min-Min, Max-Min, and Sufferage models". However, usually, the

https://doi.org/10.46253/jnacs.v5i3.a1

existing scheduling schemes include low extensibility and adaptability. The majority of the traditional techniques do not consider security factors when attempting to diminish the task achievement time. Therefore, it is necessary to develop a new task scheduling model that should take account of the security constraints as well. The major contributions of this research work are:

- Introduces a new task scheduling framework for efficient load balancing based on a 2-fold multi-objective function.
- Develops a hybrid optimization algorithm termed SU-GWO for solving the optimization issues on diverse perspectives defined.

The rest of the paper is organized as: Section 2 addresses the recent works accomplished under task scheduling. Section 3 depicts a short description of the developed task scheduling framework. Section 4 describes the proposed multi-objectives and solution encoding: a novel SU-GWO-based optimization for task allocation. The results are discussed in Section 5 and the conclusion is presented in Section 6.

2. Literature Review

2.1 Related Works

In 2020, Wilczyński and Joanna [1] have presented a method for scheduling tasks depending on the blockchain mechanism. In contrast to traditional schemes, the execution of blockchain modules was offloaded. In addition, a "proof–of–schedule consensus algorithm (instead of 'proof–of–work')" was developed, and "Stackelbekrg games" were used for enhancing the authorization of the created schedules. The experimentations revealed that the presented method significantly enhanced the effectiveness of created schedules with improved makespan.

In 2019, Abazari *et al.* [2] have designed an efficient technique that considered both interaction and task security while scheduling the tasks in the cloud. Moreover, for improving security, a heuristic model was formulated for task scheduling based on security requirements and the completion time of tasks. In addition, a novel attack response method was presented for reducing definite security risks in the cloud. Furthermore, the outcomes demonstrated that the designed "attack response algorithm" has reduced the threats, thus enhancing the security of the network.

In 2019, Pang *et al.* [3] have developed an EDA and GA-based hybrid algorithm for task scheduling. Initially, the sampling and probability models of EDA were exploited for generating specific viable solutions. As the final step, the optimum scheduling approach for allocating tasks to VMs was realized. The investigational results illustrated that the presented scheme has efficiently improved the load-balancing capability and reduced the completion time.

In 2019, Sanaj and Joe [4] have explored CSSA for optimal multitask scheduling in an IaaS cloud environment. The technique generated job plans continuously, which makes the present schemes more cost-efficient. For guaranteeing better global convergence, the earlier network was generated with chaotic optimization for well-organized task allocation. In the end, the established CSSA model has revealed better outcomes when compared to the traditional SSA model.

In 2020, Ismayilov and Haluk [5] have investigated a prediction-oriented approach named as "NN-DNSGA-II algorithm" that incorporated NSGA-II with the ANN framework. In addition, five foremost non-prediction-oriented dynamic approaches were adopted for solving the workflow scheduling issue. The presented work was concerned with six objectives namely, enhancement of utilization and reliability reduction of energy, cost, makespan, and level of imbalance.

3. A Short Description of Proposed Task Scheduling Framework

In a cloud paradigm, a resource pool involves a group of servers, here every server is indicated by K_i , and the server resources group of the cloud state $K_i; i = 1,2,...,n$. At the initial state k , VMs are passed on to the server K_i at every node, which is pointed out by $K_i = VM_{i,1},...,VM_{i,k}$. Accordingly, $VM_{i,k}$ is a VM on the server K_i . Each server K_i includes several equipment assets, E.g. network, memory, and CPU. In the cloud, the determination of the host for VM allocation is a noteworthy task.

The client offers a server demand for service as $\{T_i; i = 1,2,..,n\}$ and consequently every sort of administration has the requirement for better QoS. VM groups enhance the task scheduling process and accessibility evaluation of VM resources. Every task $T_i; i = 1,2,..,n$ is characterized by varied constraints. Among them, security (Sec) and makespan (Ms) are considered the most significant aspects for proficient load balancing during VM migration. This is regarded as the optimization issue and a novel method is introduced for solving it. Fig 1 describes the adopted task scheduling model.

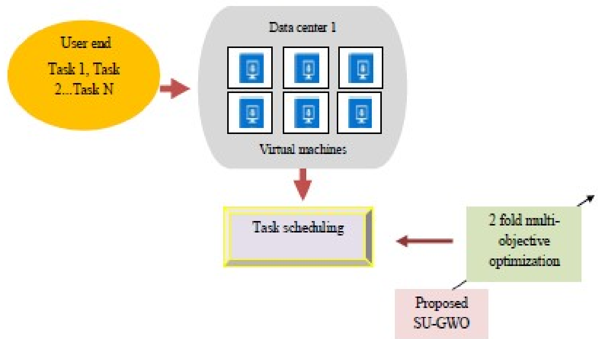


Fig 1. *Diagrammatic representation of the proposed task scheduling framework*

4. Proposed SU-GWO Algorithm for Task Allocation: Determination of Multi-Objective Function

4.1 Multi-objective Function

The most important objective of this work lies in the minimization of the 2 fold multi-objectives that are numerically delineated in Eq. (1), which OF indicates the objective function.

$$OF = \min\{Sec, Ms\} \tag{1}$$

Make span: "Make span or completion time is the total time taken to process a set of jobs for its complete execution". This is evaluated as the maximum wall time (W_{time}) for overall blocks in the tasks as given in Eq. (2).

$$Ms(tasks) = \max(\sum_{B=1}^{N} W_{time}) \tag{2}$$

Security: The migration of the VM has to be secure for effective and secured task scheduling. The security of a VM $(Y(i))$ is determined by computing the variation between the security of the physical machine and the task security. This valuation is carried out using the risk probability (P^{risk}) model. The pseudocode is revealed in Algorithm 1.

```
Algorithm 1: Security based on risk probability
for i = 1:length(Y)
    if Y(i) <=0
        P^risk = 0;
    elseif Y(i) >0 && Y(i) <=1
        P^risk = 1-exp((-1/2)* Y(i));
    elseif Y(i) >1 && Y(i) <=2
        P^risk = 1-exp((-3/2)* Y(i));
    else
        P^risk = 1;
    end
 P^risk(i) = P^risk;
end
Security = mean(P^risk);
```

4.2 Solution Encoding

Fig. 2 demonstrates the solution encoding. The chromosome length is two times the multiple of the number of blocks $B_{i_n}; i = 1...N$ assigned for overall tasks. Every block involves two parts: the PM that carries out the task and the id of VM (VM^{id}) to whom the task is assigned by the PM. Usually, the block size gets varied for each task.

For example: if the number of blocks is 42, then the chromosome length=84. Here, the 1st 42 chromosomes are filled by PM and the rest of the chromosomes are filled by VM.

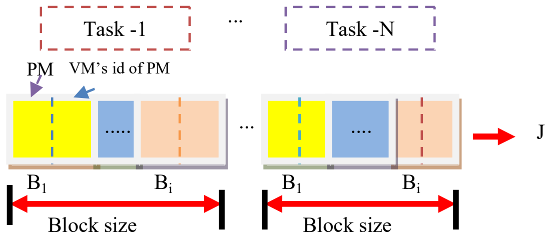


Fig 2. *Solution Encoding*

4.3 Proposed SU-GWO Algorithm

Even though the GWO [17] algorithm provides good qualities like good stability and flexibility, it suffers from a slow convergence rate. To overcome this, the SLnO algorithm is hybridized with the GWO algorithm that offers improved outputs. Hybrid optimization algorithms have been reported to be promising for certain search problems. They take the benefits of different optimization algorithms for fast convergence [19]. The wolves α, β and γ are said to be the major wolves that focus on the process of hunting. Among these wolves, α is considered as the leader that makes decisions relating to hunting process, sleeping location, time to awake, etc. whereas, β and γ holds the 2nd and 3rd level that helps α in taking decisions. In addition, the final level of wolves is concerned as ζ, which concerns on eating. The encircling characteristics are modeled as per Eq. (3) and Eq. (4), where G and L denotes coefficient vectors, J_p indicates prey's position vectors, J denotes position vectors of grey wolves, and it specifies current iteration. Eq. (5) and Eq. (6) denote the model for G and L, where $\hat{a}$ is a parameter which is minimized steadily from 2 to 0 in entire iterations. Here, ra_1 and ra_2 specifies the random vectors that lie among [0, 1] and it_{max} denotes the maximum iteration.

$$Z = \left|F.J_p(it) - J(it)\right| \tag{3}$$

$$J(it+1) = J_p(it) - G..Z \tag{4}$$

$$G = 2\hat{a}.ra_1 - \hat{a} \tag{5}$$

$$L = 2ra_2 \tag{6}$$

The arithmetical formula for describing the hunting character of wolves is given from Eq. (7) to Eq. (12). Here, the proposed hybrid evaluation takes place during the computation of Z_α, Z_β and Z_γ. In this context, the evaluation of Z_α, Z_β and Z_γ takes place using the distance computation of SLnO update, where the vector position of the sea lion is pointed out by $\vec{X}(t)$ and the arbitrary vector is given by $\vec{R}$. The final position updating evaluation is specified in Eq. (13).

$$Z_\alpha = \left|2\vec{R}.J_\alpha - \vec{X}(t)\right| \tag{7}$$

$$Z_\beta = \left|2\vec{R}.J_\beta - \vec{X}(t)\right| \tag{8}$$

$$Z_\gamma = \left|2\vec{R}.J_\gamma - \vec{X}(t)\right| \tag{9}$$

$$J_1 = J_\alpha - G_1.(Z_\alpha) \tag{10}$$

$$J_2 = J_\beta - G_2.(Z_\beta) \tag{11}$$

$$J_3 = J_\gamma - G_2.(Z_\gamma) \tag{12}$$

$$J(it+1) = \frac{J_1 + J_2 + J_3}{3} \tag{13}$$

4.4 Database Creation

The database collected for VM-based task scheduling is partially synthetic, that is, the data respective to physical machines are downloaded from: "https://www.kaggle.com/discdiver/clouds [Access data: 2020-05-03]". By exploiting this database, the data related to VM are generated. The database of PM involves certain constraints as given in Table 1.

Table 1: Database Constraints

1	Count of PM	Original **data**
2	Number of tasks	
3	Wall time	
4	Cost of PM	
5	Security	Synthetic data
6	Count of CPU	

The database for VM is generated using the PM database. It also comprises similar constraints to the PM. The database formation for every constraint is described below:

Assume the task to be carried out in PM as $T_i; i = 1,2,3,4$. The block size of Task 1, task 2, task 3, and Task 4 is 5, 10, 15, and 12, correspondingly. The task from the user is assigned to the PM that splits the tasks and allocates them to separate VMs. Here, 13 PM are exploited and each PM includes 10 VMs. As a result, there is a total of 130 VMs to carry out the tasks.

Security: The security range of PM is fixed randomly among 1 to 4. For example, if the number of tasks to be accomplished is 6, then 6 security values are created within the bounds from 1 to 4. Here, the task with security value 1 is considered a highly secured one, whereas the least secured task holds a higher security value (4).

The security being the significant constraint, it is determined for VM by computing the variation between the PM security and task security. As described earlier, the security value of VM and PM is fixed from 1 to 4. For example, if the security value of PM is 1 and task security is 1, then the security of VM is 0. Therefore, the security requirement is fulfilled as per the risk mechanism.

Wall time: "Elapsed real-time, real-time, wall-clock time, or wall time is the actual time taken from the start of a computer program to the end. In other words, it is the difference between the time at which a task finishes and the time at which the task started. Wall time is thus different from CPU time, which measures only the time during which the processor is actively working on a certain task. The difference between the two can arise from architecture and run-time dependent factors, e.g. programmed delays or waiting for system resources to become available". The PM has a pre-defined wall time. The wall time of VM is computed using the "normrnd function". The syntax of the function is represented in Eq. (14).

$$W_{time} = \mathrm{normrnd}(\mu, \delta) \tag{14}$$

In general, this function creates a random number from the normal distribution with mean parameter μ and standard deviation parameter sigma (δ). In this context, δ is fixed as 0.1 and μ points out the wall time of PM.

5. Results and Discussions

5.1 Simulation Procedure

The presented task scheduling model using the SU-GWO approach was implemented in **MATLAB** and the analysis was held. Here, the dataset was downloaded from the link "https://www.kaggle.com/discdiver/clouds". The performance of the presented technique was compared with the other conventional methods like GWO [17] and SLnO [18] schemes. Accordingly, analysis was carried out concerning make span, security, and total cost by varying the number of blocks from [20 25 30 35 40]. Moreover, statistical analysis was performed by varying the number of VM from [10 20 30 40]. Further, sub-blocks will be created from the above block variation, and here T1, T2, T3, and T4 are tasks

As per the collected database

- block variation 1 encompasses: [4 7 6 3]-[T1 T2 T3 T4]=total blocks 20
- block variation 2 has [6 8 6 5]-[T1 T2 T3 T4]=total blocks 25
- block variation 3 has [6 10 7 7]-[T1 T2 T3 T4]=total blocks 30
- block variation 4 has[7 11 8 9]-[T1 T2 T3 T4]=total blocks 35
- block variation 5 [5 10 15 10]-[T1 T2 T3 T4]=total blocks 40

5.2 Makespan Analysis

Fig. 3 shows the make-span analysis of the presented SU-GWO model over existing models by varying the number of blocks from 20, 25, 30, 35, and 40. Here, makespan is evaluated for four variations namely, 10, 20, 30, and 40. While noticing the results, the proposed SU-GWO model has attained the minimum makespan. Especially, from Fig. 3(b), the presented model has reached a minimum makespan, which is 70% and 30% better than GWO and SLnO models when the number of blocks is 25. Similarly, from Fig. 3(d), the presented SU-GWO scheme has achieved a minimum makespan of 160; whereas, the makespan attained by existing GWO and SLnO models are 270 and 210 when the number of blocks is 40. That is, the presented scheme is 68.75% and 37.5% superior to GWO and SLnO models. This shows the enhanced performance of the presented SU-GWO model in terms of makespan.

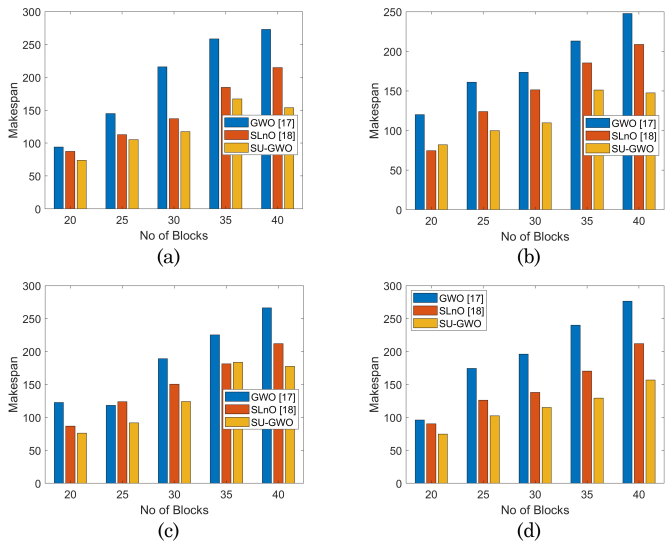


Fig 3. *Analysis of makespan for the proposed model over conventional schemes in case of varying count of VMs allocated by a PM for load balancing (a) VM=10 (b) VM=20, (c) VM=30 and VM=40*

5.3 Security Analysis

The security analysis of the implemented SU-GWO model over the conventional models by varying the number of blocks from 20, 25, 30, 35, and 40 are described in this section. Here, the security is computed for four variations namely, 10, 20, 30, and 40 in terms of risk. On observing the attained graphs, the security (risk) of using the SU-GWO model has reached the minimal value for all variations. More particularly from Fig. 4(c), the proposed model attains secured task scheduling with a minimal risk of 0.41, whereas, the compared schemes such as GWO and SLnO models have attained a higher risk of 0.45 and 0.49 respectively when the number of blocks is 20. Likewise, from Fig. 4(d), the suggested model attains secured task scheduling with a minimal risk of 0.3, whereas, the traditional GWO and SLnO models have attained a comparatively higher risk of 0.48 and 0.56 respectively when the number of blocks is 25. That is, the presented model reveals a betterment of 37.5% and 46.43% over the compared GWO and SLnO models. Thus, the security enhancement of the presented SU-GWO method in terms of risk has been established from the analysis outcomes.

5.4 Total Cost Analysis

The total cost analysis of the presented SU-GWO model by varying the number of blocks from 20, 25, 30, 35, and 40 is explained in this subdivision. Fig. 5 demonstrates the performance of the presented SU-GWO approach over the conventional models is evaluated for varied costs namely 10, 20, 30, and 40. Here, all the attained outcomes for the presented SU-GWO model have accomplished optimal cost values when evaluated over the compared models. Particularly, from Fig. 5(b), the total cost of the presented scheme is 60% and 28.57% better than existing GWO and SLnO models when the number of blocks is 20. In addition, from Fig. 5(d), the total cost of the presented scheme has attained minimal value, which is 41.43% and 35.71% better than existing GWO and SLnO models when the number of blocks is 40. Thus, the enhancement of the adopted SU-GWO scheme in terms of the total cost has been confirmed from the analysis outcomes.

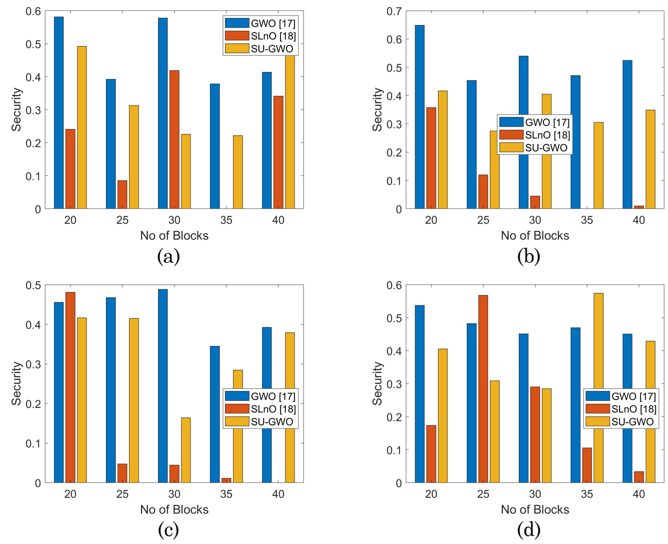


Fig 4. *Analysis of security for the proposed model over conventional schemes in case of varying count of VMs allocated by a PM for load balancing (a) VM=10 (b) VM=20, (c) VM=30 and VM=40*

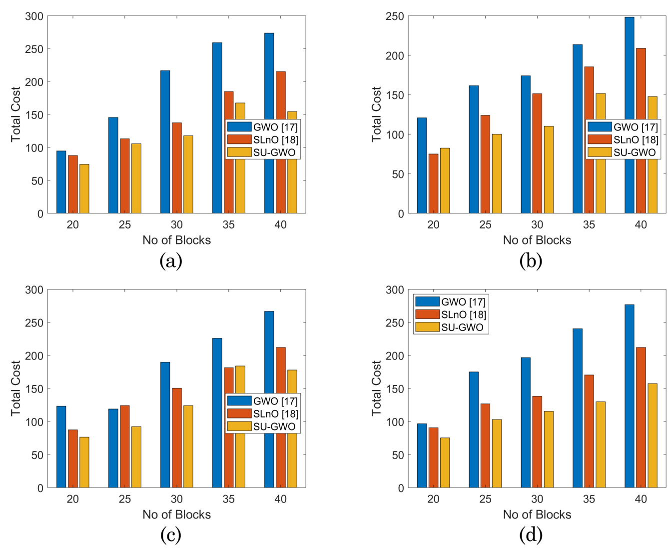


Fig 5. *Analysis of the total cost for the proposed model over conventional schemes in case of varying count of VMs allocated by a PM for load balancing (a) VM=10 (b) VM=20, (c) VM=30 and VM=40*

5.5 Overall Analysis: Security, Makespan, and Total Cost

The overall analysis of the suggested SU-GWO model over existing models to security, makespan, and the total cost is summarized in this section by varying the counts of VM from 1, 2, 3, 4, and 5. For better performance of the system, the considered factors (security, makespan, and total cost) have to be minimal, which is attained by the presented model. From Table 2, the security of the suggested method is 29.29% and 66.35% superior to traditional GWO and SLnO models when the number of VM is 5. Furthermore, the makespan of the implemented scheme has obtained a minimal value of 152.56; whereas, the compared GWO and SLnO schemes have obtained comparatively higher makespan values of 272.5 and 214.16 when the number of VM is 1. Thus, the development of the presented SU-GWO model is verified from the investigation outcomes.

5.6 Statistical Analysis

The statistical analysis for the presented model over the existing approaches to varied counts of VM from 10, 20, 30, and 40 are summarized in Table 3. Since the meta-heuristic models are stochastic, each algorithm is executed ten times and the “best, worst, mean, median, and standard deviation measures” were obtained. From the outcomes, the presented SU-GWO approach has attained optimal outputs under all test case scenarios. While analyzing the outcomes obtained under the best-case scenario from Table 3, the adopted scheme is 21.29% and 15.12% superior to existing GWO and SLnO models, when the number of VM is 10. Likewise, the adopted model under the mean case scenario is 33.68% and 17.08% superior to existing GWO and SLnO models, when the number of VM is 20. While examining the median case, the presented SU-GWO model is 30.79% and 21.09% superior to compared GWO and SLnO models when the number of VM is 30. Thus, the superior performance of the presented SU-GWO scheme has been validated effectively.

Table 2: Overall Analysis of the proposed model over existing models by varying the count of VM

		Security	
VM	GWO [17]	SLnO [18]	SU-GWO
10	0.10078	0.090939	0.11878
20	0.067184	0.35633	0.16629
30	0.077021	0.35633	0.19004
40	0.13845	0.33257	0.15237
50	0.11469	0.14253	0.081103
		Make span	
VM	GWO [17]	SLnO [18]	SU-GWO
10	272.5	214.16	152.56
20	246.53	213.03	144.65
30	268.85	210.71	177.32
40	276.57	208.89	153.97
50	264.36	178.41	163.89
		Total cost	
VM	GWO [17]	SLnO [18]	SU-GWO
10	272.6	214.25	152.68
20	246.6	213.38	144.82
30	268.93	211.06	177.51
40	276.71	209.22	154.12
50	264.48	178.55	163.97

Table 3: Statistical analysis of the presented model vs traditional models in terms of varied counts of VM

		VM=10	
Measures	GWO [17]	SLnO [18]	SU-GWO
Best	94.518	87.646	74.398
Worst	259.04	184.83	167.5
Mean	178.92	130.76	116.3
Median	181.06	125.28	111.64
STD	73.244	41.397	38.697
		VM=20	
Measures	GWO [17]	SLnO [18]	SU-GWO
Best	120.7	74.916	82.408
Worst	213.47	185.35	151.51
Mean	167.44	133.92	111.04
Median	167.79	137.71	105.12
STD	38.22	46.66	29.318
		VM=30	
Measures	GWO [17]	SLnO [18]	SU-GWO
Best	118.84	87.312	76.482
Worst	225.76	181.4	183.82
Mean	164.38	135.79	119.22
Median	156.45	137.23	108.28
STD	52.22	39.936	47.434
		VM=40	
Measures	GWO [17]	SLnO [18]	SU-GWO
Best	96.682	90.626	75.173
Worst	240.37	170.4	129.89
Mean	177.15	131.5	105.89
Median	185.76	132.48	109.25
STD	60.149	32.92	23.258

6. Conclusion

This paper has introduced a new VM-oriented task scheduling model based on 2-fold multi-objective functions namely, make span and security. In addition, a hybrid optimization algorithm termed SU-GWO was developed for solving the optimization issues on diverse perspectives defined. Finally, analysis was done for validating the betterment of the presented scheme. Here, the security of the suggested method was 29.29% and 66.35% superior to traditional GWO and SLnO models when the number of VM was 5. Furthermore, the makespan of the implemented scheme has obtained a minimal value of 152.56; whereas, the compared GWO and SLnO schemes have obtained comparatively higher makespan values of 272.5 and 214.16 when the number of VM was 1. Thus, the improvement of the presented SU-GWO model was confirmed by the simulated outcomes.

Compliance with Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows the ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

References

[1] Andrzej Wilczyński, Joanna Kołodziej, “Modelling and simulation of security-aware task scheduling in cloud computing based on Blockchain technology”, Simulation Modelling Practice and Theory, vol. 99, pp. 102038, 2020.
[2] Farzaneh Abazari, Morteza Analoui, Hassan Takabi, Song Fu, “MOWS: Multi-objective workflow scheduling in cloud computing based on heuristic algorithm”, Simulation Modelling Practice and Theory, vol. 93, pp. 119-132, 2019.
[3] S. Pang, W. Li, H. He, Z. Shan and X. Wang, "An EDA-GA Hybrid Algorithm for Multi-Objective Task Scheduling in Cloud Computing," IEEE Access, vol. 7, pp. 146379-146389, 2019.
[4] M. S. Sanaj, P. M. Joe Prathap, “Nature inspired chaotic squirrel search algorithm (CSSA) for multi objective task scheduling in an IAAS cloud computing atmosphere”, Engineering Science and Technology, an International Journal, vol. 23, no. 4, pp. 891-902,2020.
[5] Goshgar Ismayilov, Haluk Rahmi Topcuoglu, “Neural network based multi-objective evolutionary algorithm for dynamic workflow scheduling in cloud computing”, Future Generation Computer Systems, vol. 102, pp. 307-322, 2020.
[6] H. Zhang, J. Shi, B. Deng, G. Jia, G. Han, and L. Shu, "MCTE: Minimizes Task Completion Time and Execution Cost to Optimize Scheduling Performance for Smart Grid Cloud," IEEE Access, vol. 7, pp. 134793-134803, 2019.
[7] Sobhanayak Srichandan, Turuk Ashok Kumar, Sahoo Bibhudatta, "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm", Future Computing and Informatics Journal, vol.3, no. 2, pp. 210-230, 2018.
[8] Yingfeng Zhang, Jin Wang, Yang Liu, “Game theory based real-time multi-objective flexible job shop scheduling considering environmental impact”, Journal of Cleaner Production, vol. 167, pp. 665-679, 2017.
[9] Mohamed Abd Elaziz, Shengwu Xiong, K. P. N. Jayasena, Lin Li, "Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution", Knowledge-Based Systems, vol. 169, pp. 39-52, 2019.
[10] [10] R.K. Jena, “Energy Efficient Task Scheduling in Cloud Environment”, Energy Procedia, vol. 141, pp. 222-227, 2017.
[11] Yi-wen Zhang, Cheng Wang, Jin Liu, “Energy aware fixed priority scheduling for real time sporadic task with task synchronization”, Journal of Systems Architecture, vol. 83, pp. 12-22, 2018.
[12] Keqin Li, “Scheduling parallel tasks with energy and time constraints on multiple manycore processors in a cloud computing environment”, Future Generation Computer Systems, vol. 82, pp. 591-605, 2018.
[13] Toktam Ghafarian and Bahman Javadi, "Cloud-aware data intensive workflow scheduling on volunteer computing systems", Future Generation Computer Systems, vol. 51, pp.87-97, 2015.
[14] Jiachen Yang, Bin Jiang, Zhihan Lv and Kim-Kwang Raymond Choo, "A task scheduling algorithm considering game theory designed for energy management in cloud computing", Future Generation Computer Systems, vol. 105, pp.985-992, 2020.
[15] Hend Gamal El Din Hassan Ali, Imane Aly Saroit and Amira Mohamed Kotb, "Grouped tasks scheduling algorithm based on QoS in cloud computing network", Egyptian Informatics Journal, vol.18, no. 1, pp. 11-19, 2017.
[16] Mohammed Abdullahi, Md Asri Ngadi and Shafi’i Muhammad Abdulhamid, "Symbiotic Organism Search optimization based task scheduling in cloud computing environment", Future Generation Computer Systems, vol. 56, pp. 640-650, 2016.
[17] Seyedali Mirjalili a, Seyed Mohammad Mirjalili, Andrew Lewis, "Grey Wolf Optimizer", Advances in Engineering Software, vol. 69, pp.46–61, 2014.
[18] Masadeh, Raja & Mahafzah, Basel & Sharieh, Ahmad. "Sea Lion Optimization Algorithm", International Journal of Advanced Computer Science and Applications, vol. 10, pp. 388-395, 2019.
[19] M. Marsaline Beno, Valarmathi I. R, Swamy S. M and B. R. Rajakumar, “Threshold prediction for segmenting tumour from brain MRI scans”, International Journal of Imaging Systems and Technology, vol. 24, no. 2, pages 129-137, 2014.