 doi:10.20944/preprints202109.0407.v1

Article

A Cloud Computing-Based Modified Symbiotic Organisms Search Algorithm for Ecosystem Sustainability

Ajoze Abdulraheem Zubair[1,*], Shukor Bin Abd Razak[2] and Md. Asri Bin Ngadi[3]

[1] School of Computing, Faculty of engineering Universiti Teknologi Malaysia,81310 Skudai Johor Bahru, Johor Malaysia
[2] School of Computing, Faculty of engineering Universiti Teknologi Malaysia,81310 Skudai Johor Bahru, Johor Malaysia; shukorar@utm.my
[3] School of Computing, Faculty of engineering Universiti Teknologi Malaysia,81310 Skudai Johor Bahru, Johor Malaysia; dr.asri@utm.my
* Correspondence: ajozeoziada1968@gmail.com

Abstract: The search algorithm based on symbiotic organisms' interactions is a relatively recent bio-inspired algorithm of the swarm intelligence field for solving numerical optimization problems. It is meant to optimize applications based on the simulation of the symbiotic relationship among the distinct species in the ecosystem. The modified SOS algorithm is developed to solve independent task scheduling problems. This paper proposes a modified symbiotic organisms search based scheduling algorithm for efficient mapping of heterogeneous tasks to access cloud resources of different capacities. The significant contribution of this technique is the simplified representation of the algorithm's mutualism process, which uses equity as a measure of relationship characteristics or efficiency of species in the current ecosystem to move to the next generation. These relational characteristics are achieved by replacing the original mutual vector, which uses an arithmetic mean to measure the mutual characteristics with a geometric mean that enhances the survival advantage of two distinct species. The modified symbiotic organisms search algorithm (G_SOS) aimed to minimize the task execution time (Makespan), response, degree of imbalance and cost and improve the convergence speed for an optimal solution in an IaaS cloud. The performances of the proposed technique have been evaluated using a Cladism toolkit simulator, and the solutions are found to be better than the existing standard (SOS) technique and PSO.

Keywords: cloud computing; cloud resource management; task scheduling; ecosystem; geometric mean; symbiotic organisms search algorithm; convergence speed

1. Introduction

Cloud computing is a modern computing model that offers virtualization computing services as a utility to Cloud service users [1–4]. It is a concept for obtaining resources from a customizable shared resource, such as a group of networks, servers, storage, utilities, and applications, in instantaneously and based on request. Cloud service providers use virtualization technologies to better use resources by allowing multiple virtual machines (VMs) to operate on top of a single physical computer. Consumers of cloud services are automatically provisioned based on Service-Level Agreements (SLA), which are usually formed by negotiations between Cloud service providers and Cloud service users/consumers. Issues related to inefficient mapping of tasks to cloud resources often occur in a cloud environment [5–8].

Task scheduling, therefore, refers to efficiently scheduling computational activities and rational allocation of computing resources under some restrictions in the IaaS cloud environment. Scheduling's job is to assign tasks to the most suitable resources in order to achieve one or more goals. Therefore, selecting an appropriate work scheduling algorithm to increase cloud computing resource efficiency while keeping high quality of service (QoS) guarantees is an important issue that continues to attract research attention [9–11].

 doi:10.20944/preprints202109.0407.v1

As a result of the broad solution space and complex existence of heterogeneous resources in cloud computing, the task scheduling problem falls into the group of NP-hard problems [12–15].

Heuristic and metaheuristic scheduling techniques have been used to address the task scheduling problem in cloud computing [2,16]. For more trivial concerns, heuristic methods provide optimal results; but, as the size of the problem rises, the solutions generated by these algorithms will be less optimal. Metaheuristic algorithms, on the other hand, have shown impressive effectiveness in delivering near-optimal scheduling solutions for a complex large size problem. Nevertheless, metaheuristic algorithms continue to suffer from being trapped in local optima, premature convergence, delayed convergence, and imbalance between the search methods [17–19].

In recent years, an increasing number of independent scholars have investigated the quality of service provided by task scheduling techniques. Several techniques such as PSO, ACO, GA, SOS, CS, and FPA have been advanced to address various issues like cost, execution time and SLA parameters to be obeyed by the cloud service provider as specified by the cloud users. However, these algorithms have problems such as being entrapped in a local optimal, having a high computational cost, a slow convergence rate, and being unsuitable for certain types of decision variables. Initially, the standard SOS was intended to solve numerical optimization problems [20]. This algorithm's implementation has been found in the literature, given in references [18,21–27]. SOS's capacity to find a globally accepted solution to optimization problems makes it appealing for further exploration and development. Thus, the quest for global solutions to optimization problems would have a reasonable likelihood of success by further modifying the ecosystem's mutual characteristics between the distinct species.

Therefore, this study presents a G_SOS algorithm, a scheduling method based on a modified SOS for the best feasible mapping of different tasks to cloud resources, with the goal of minimizing task processing time, cost of resource usage, and optimizing resource utilization.

The following are the paper's main contributions:

- The modified SOS procedures are described in a more explicit and more precise manner.
- The modified SOS algorithm tagged G_SOS was designed and implemented for job scheduling in IaaS cloud computing environment.
- The technique's performance indicators include makespan, responsiveness, and the degree of imbalance among VMs.

Further discussions in this paper are arranged as follows: Section 2 introduces the related literature. Section 3 presents the problem formulation. Section 4 briefly introduces the Geometric Mean-Based Symbiotic Organisms Search (G_SOS) algorithm. The simulation and results analysis are presented in Section 5. This paper's work is concluded in Section 6.

2. Related Works

2.1. Metaheuristic Techniques used in Cloud Task Scheduling

Metaheuristics techniques are based on analogues of biological concepts. It has been demonstrated that metaheuristic-based strategies can achieve near-optimal solutions in a reasonable amount of time for some complex problems [2,28]. Metaheuristic approaches were used to address resource scheduling difficulties such as makespan and response time reduction. The methods have been shown to find an optimal mapping of tasks to resources, reduce computation costs, improve service quality, and increase computing resource utilization. Metaheuristic algorithms have demonstrated exceptional effectiveness in delivering near-optimal scheduling solutions for a complex large size problem and as such have piqued the interest of various researchers [8,29]. Nevertheless,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 September 2021 doi:10.20944/preprints202109.0407.v1

metaheuristic algorithms continue to suffer from being trapped in local optima, premature convergence, delayed convergence, and imbalance between the search methods [17–19,30].

2.2. Symbiotic Organisms Search Technique (SOS)

Cheng and Prayogo [20] applied the SOS (Symbiotic Organisms Search) algorithm to handle challenges involving statistics, engineering design and optimization problems. Symbiotic Organisms Search algorithm emulates the symbiotic relationship that organisms must uphold to survive and grow in the ecosystem. The results show that the algorithm performs exceptionally well in several complex numerical problems.

The SOS algorithm has been continually modified since its conception to offer optimal and efficient solutions to diverse optimization challenges [31]. It is worth noting that the majority of the improvements to the original SOS algorithm were achieved by reworking either the mutualism phase or the commensalism phase, or both. Except under exceptional circumstances is a fourth phase added to the original three.

In Abdullahi et al. [16] a discrete variant of the Symbiotic Organism Search (SOS) algorithm termed DSOS has been proposed. The proposed technique used the Symbiotic Organisms Search (SOS) in the cloud to find the best schedules for tasks based on the available cloud resources. The results of the simulation process indicate that the algorithm outperforms PSO and can be used to solve large-scale scheduling issues.

Nama S. et al. [32] suggested an Improved Symbiotic Organisms Search (I-SOS) to solve various dynamic unconstrained global optimization problems. The technique employs a random weighted reflective parameter and a predation phase to improve its performance. The improved SOS experimental outcomes proved better efficiency compared to PSO, DE and SOS algorithms.

A Modified Symbiotic Organisms Search (MSOS) algorithm has been proposed by S Banerjee and S. Chattopadhyay [22,33]. The algorithm changes the composition of the organism and chooses a set of parameters for a newly created symbiotic organisms search (SOS) algorithm to improve the convergence rate and its accuracy. MSOS splits the ecosize into three inhabitants, and the combined inhabitant is executed based on predefined probabilities. A new relationship is introduced into the phases to enhance the capacity to locate a stable and high-quality solution in a shorter period. MSOS algorithm is endowed with a chaotic element created by the logistic map to find the most promising zones.

Tejani et al. [34] proposed an algorithm for optimizing structural design using adaptive symbiotic organisms search (SOS). The proposed technique modifies the benefit factors of SOS based on the fitness value of the current organism and the best organism's fitness value. Furthermore, the benefit factor improved the exploration capability, especially when the two distinct organisms X_i or X_j (i≠j) are far from the best (X_{best}) in the search space. In the same vein, when the two distinct interaction organisms are closer to the best organism, it strengthened its exploitation capability.

A revised symbiotic organisms search algorithm for the problem of unscrewed combat aerial vehicle (UCAV) route planning has been presented by F. Miao et al. [35]. Under a multi-constrained global optimization problem, the proposed technique modified the standard SOS based on the simplex method (SMSOS). The method improves population diversity, thereby overhauling intensification and diversification to avoid premature convergence and local optima entrapment.

3. Problem Formulation

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 September 2021 doi:10.20944/preprints202109.0407.v1

In a cloud computing environment, users' various tasks are dynamically assigned to the desired virtual machine (VM) or cloud resources. It is strenuous to assign jobs to available VMs based on their requirements. Also, tasks will not be allocated uniformly among virtual machines because of their heterogeneous nature, thus increasing the makespan of specific VMs while reducing the makespan of others. The overall system performance will suffer as a result of this. The primary purpose of this work is to identify the best schedule for executing a group of tasks in VMs in the shortest time possible while also improving overall performance.

3.1. Mathematical Model

This paper introduces a discrete algorithm for scheduling independent tasks based on Improved Symbiotic Organism Search (G_SOS). This scenario uses a weighted sum of all objective functions as an objective function, as shown in Eq.1. The primary goal is to reduce the task completion time and computing expenses on accessible virtual machines. Table 1 depicts the estimated completion time and cost.

Minimize

$$f(z) = \delta \times f_1(z) + (1 - \delta) \times f_2(z) \quad (1)$$

Where δ is assuming the weight factor, such that $0 \leq \delta$ 1

The objective function for makespan is $f_1(z)$, which is described as

$f_1(z) = \max\{Time^{ij}\} \forall$ task T^i mapped on VM^j. Similarly

$f_2(z)$ is the objective function for computing cost, which is defined as

$$f_2(z) = \sum_{j=1}^{m} C_j \times \alpha_j \qquad \forall \text{ VMs}$$

Table 1. Expected time of completion (ETC) and corresponding cost

T / VM	T^1	T^2	-	-	T^n
VM^1	T^1/VM^1	T^2/VM^1	-	-	T^n/VM^1
VM^2	T^1/VM^2	-	-	-	T^n/VM^2
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
VM^m	T^1/VM^m	T^2/VM^m	-	-	T^n/VM^m

Consider a set of m virtual machines $V = \{VM^1, VM^2, VM^3, \ldots, VM^m\}$ and a set of n tasks or applications $T = \{T^1, T^2, T^3, \ldots, T^n\}$, ETC matrix is an (n × m) that keeps track of how long each task takes i.e., T^i on each VM^j. It is measured as the ratio of Task length to the virtual machine capacity, as shown in Eq.2.

Let ETC_{ij}, $i = 1,2,3, \ldots, n, j = 1,2,3, \ldots, m$ be the processing time of executing task T^i on each virtual machine VM^j.

It is calculated as:

$$ETC_{ij} = length\ of\ task\ T_i / capacity\ of\ Vm_j = MI_i / MIPS_j \quad (2)$$

Eq. (3), which measures the strength of the organism's degree of adaptation to the ecosystem, is used to calculate the fitness value of each organism.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 September 2021 doi:10.20944/preprints202109.0407.v1

$$Objective\ Function = max\left\{\sum_{j=1}^{m}\frac{f(Vm_j)}{m}\right\} \quad (3)$$

$$f(Vm_j) = \frac{\zeta}{makespan} \quad (4)$$

$$\zeta = \sum_{j=1}^{m}\frac{r_j}{m} \quad (5)$$

$$r_j = \frac{Task\ T_i}{makespan} \quad (6)$$

$$Makespan = \max\{ETC_{i,j} | i \in T, i = 1,2,3,\dots,n\ and\ j \in VM, j = 1,2,3,\dots,m\} \quad (7)$$

While Eq.4. represents the virtual machine $VM^{j}s$ fitness value, Eq.5 shows the average usage of virtual machines employed for task execution denoted by ζ., The utilization of virtual machine VM^{j} is defined by r_j, as indicated in Eq.6. Equation (7) defines the maximum time taken by virtual machines working in parallel to accomplish the task.

Let $\{c_1, c_2, c_3, \dots, c_m\}$ represent the unit cost of virtual machine VM^{j} per time quantum[36,37]

The unit cost of executing a task T^{i} on a VM^{j}in this study is considered per-second basis. In this situation, the costs of data transmission and retrieval are ignored[38,39]

The i^{th} task in the task sequence is represented by T^{i}, and its length is millions of instructions (MI). The j^{th} virtual machine in the data center or cloud environment is designated by VM^{j} and the rate at which information is processed is expressed in millions of instructions per second (MIPS). The cost matrix is a 1 × n matrix that contains each virtual machine's cost.

The tasks are placed on the available VMs, and they are executed in the order in which they arrive. Our goal is to schedule tasks on virtual machines (VMs) so that they can be fully utilized in a short period of time (makespan). The values of the expected time of completion (ETC) matrix and the cost matrix are normalized by dividing them by their respective maximum values. [40,41].

Therefore, since the virtual machines are heterogeneous in nature, the related cost of using each varies.

The goal of the scheduling problem, according to the essential characteristics of task scheduling in a cloud computing context, is to map every task T^{i} vigorously i = 1, 2, 3,..., n to a suitable cloud virtual machine VM^{j} j = 1, 2, 3,..., m to reduce overall execution time and resource utilization costs.

4.The Geometric Mean-Based Symbiotic Organisms Search (G_SOS) Algorithm

The improvement of the SOS algorithm was informed by the relationship characteristics of the mutual vectors between two distinct organisms $Z^{i}$and Z^{j} in the Mutualism phase is calculated using the arithmetic mean, which signifies equality between the two species. But in a heterogeneous cloud environment, the cloud resources likened to two distinct species can never have the same structure in terms of the processor's speed, memory size, and storage capacity. Therefore, the mutual vector which signifies relationship characteristics between organisms Z^{i} and Z^{j} can be best computed using the geometric mean, which signifies equity between the two species.

A mutual vector (MV) denotes a shared trait shared by two different species in order to improve their chances of survival or sustainability.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 September 2021 doi:10.20944/preprints202109.0407.v1

Where $MV = \frac{(Z^i + Z^j)}{2}$, i.e. the Arithmetic mean.

Therefore, the Geometric mean (Gm) for n number of organisms denoted by gm is shown in Eq. 8.

$$gm = \sqrt[n]{\{Z^1 * Z^2 * Z^3 * \ldots * Z^n\}} \quad (8)$$

The geometric mean for two distinct species, Z^i and Z^j is calculated as shown in Eq.9.

$$gm = \{Z^i * Z^j\}^{\frac{1}{2}} \quad (9)$$

Therefore, Eq.10. shows the Mutual vector between the two distinct species, calculated as the square root of the absolute value of the product of Z^i and Z^j.

$$Z^{MV} = \sqrt[2]{|\{Z^i * Z^j\}|} \quad (10)$$

Gm ensures equity in each species contribution for their survival. Each VM contributes its resource by its capability and strength to improve its mutualistic characteristic, increasing its survival advantage, increasing the exploration capability of the technique, thereby increasing the convergence speed of the search process to achieve a globally optimal solution.

Figure 1 depicts the symbiotic interaction between various species in an ecosystem

(a) *(b)* *(c)*

Figure.1. ***Examples of symbiotic relationships***: (a) Mutualism between Bee and flower; (b) Commensalism between Elephant and Bird (c) Parasitism between Fish and Cymothoa exigua (A parasite that eats a fish's tongue and takes its place)

So, the new algorithm will be operating based on the new structure.

4.1. Mutualism Phase

During this phase, two species Z^i and Z^j, where (i ≠ j) are randomly chosen to interact with each other. The two separate species were mutually beneficial. Equations (11) and (12) represent the mathematical description of this technique, respectively.

$$A = Z^i + rand(0,1) \times [Z^{best} - Z^{MV} \times B^{f-1}] \quad (11)$$

$$B = Z^j + rand(0,1) \times [Z^{best} - Z^{MV} \times B^{f-2}] \quad (12)$$

where A and B represent the two organisms selected randomly from the ecosystem. The Mutual Vector (Z^{MV}) and Benefit factors (B^{f-1} *and* B^{f-2}) are derived from the respective mathematical formulation in Eq.10. and equations (4) and (5) as described in the work of [30]

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 September 2021 doi:10.20944/preprints202109.0407.v1

The new species Z'_{i_new} *and* Z'_{j_new} are generated by moulding their structure from Z^{MV} and BF (Benefit factors) corresponding to the best organism (Z^{best}) of the current population as shown in equations (13) and (14).

Modify the new organisms to reflect the discretization of the algorithm using

$$Z'_{i\_new} = \lceil A \rceil \, mode \, m + 1 \tag{13}$$

$$Z'_{j\_new} = \lceil B \rceil \, mode \, m + 1 \tag{14}$$

Fitness values of new species Z'_{i_new} *and* Z'_{j_new} are evaluated and compared to each predecessor to select the fittest in the population.

If $f(Z'_{i\_new}) < f(Z^i)$

$Z^i \leftarrow Z'_{i\_new}$

Similarly, if $f(Z'_{j\_new}) < f(Z^j)$

$Z^j \leftarrow Z'_{j\_new}$

Note that worst fitness values are replaced or rejected.

4.2. Commensalism phase

Select a random organism Z^j, where $Z^i \neq Z^j$. let C be the new status of the organism Z^i as shown in Eq.15. Modify the new organisms to reflect the discretization of the algorithm using Eq.16.

$$C = Z^i + rand(0,1) \times [Z^{best} - Z^j] \tag{15}$$

Also

$$Z'_{i\_new} = \lceil C \rceil \, mode \, m + 1 \tag{16}$$

If $f(Z'_{i\_new}) < f(Z^i)$

$Z^i \leftarrow Z'_{i\_new}$

4.3. Parasitism phase

Select a random organism Z^j, where $Z^i \neq Z^j$, let D be the new status of the parasite vector created from organism Z^i as shown in Eq.17. Modify the new parasite vector D to reflect the discretization of the algorithm using Eq.18.

$$D = rand(0,1) \times Z^j \tag{17}$$

$$Z^p = \lceil D \rceil mode \, m + 1 \tag{18}$$

If $f(Z^p) < f(Z^j)$

$Z^j \leftarrow Z^p$

The steps of the modified SOS algorithm (G_SOS) are described in Algorithm 1

Algorithm 1: *Modified Symbiotic Organism Search algorithm (G_SOS) pseudocode*
Input: Size of Population (ecosize), maximum number of iterations (Maxitern)
Output: Z^{best} which is the optimal solution.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 September 2021 doi:10.20944/preprints202109.0407.v1

The Looping of G_SOS begins:

While itern<maxitern

For i = 1: Population (ecosize)

For each Spesies in the ecosystem Z^i, i = 1, 2, 3, . . ., ecosize, search for the organism with the best fitness valus Z^{best}

Mutualism Phase

Randomly select organisms Z^i and Z^j $(i \neq j)$

Calculate the mutual vector (Z^{MV}) eq. (10) and the Benefit factors (B^{f-1} *and* B^{f-2}) using equations (4) and (5) as described in the work of [30]

Using Eq. (13) and Eq. (14) to generate the new organisms Z'_{i_new} & Z'_{j_new} and evaluate their fitness values.

If the new organisms' fitness value are higher, then replace the predecessors

Commensalism phase

Select organism Z^j randomly $(i \neq j)$

Using Eq. (16) to generate a new organism Z'_{i_new} and evaluate its fitness value

If the new organisms' fitness value are higher, then replace the predecessor.

Parasitism Phase

Select organism Z^j randomly $(i \neq j)$

Generate parasite vector Z^p by modifying Z^i in **Eq. (18)**

Evaluate the fitness value

If the parasite vector (Z^p)s' fitness value is higher, then replace Z^j with Z^p

End for

Update best organism Z^{best} of the current population (ecosize)

End while

5. Simulation and Results

Simulations were run using the cloudsim-3.0.3 toolkit simulator to test the performance of the proposed technique [42]. The cloud environment is characterized by the heterogeneity of tasks and virtual machines. A single data center with two hosts was established. Each host had 1 TB of storage, 20 GB of RAM 10 Gbps of bandwidth, and a time-shared virtual machine scheduling technique. Twenty (20) virtual machines (VMs) were

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 September 2021 doi:10.20944/preprints202109.0407.v1

created, ten (10) each per host, each with a 10 Gigabyte image size, 0.5 Gigabyte memory, 1 Gigabyte per second bandwidth, and one processing element. The VMs ranged in processing power from 100 to 5000 MIPS, with prices ranging from 0.05 to 0.25 USD. All the VMs were run on a time-shared cloudlet scheduler with Xen VMM. In 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1000 examples, a consistent distribution of task sizes was generated, resulting in an equal number of large, medium, and small assignments. We can better grasp the algorithms' scalability and performance when dealing with large problem sizes by using more significant instances. For each algorithm, the simulation was repeated for 30 experimental runs. Table 2 lists the algorithm parameter settings for the simulation experiment, while the cloudsim experimental settings are listed in Table 3. Figures 2-6 show simulation results for the scheduling of G_SOS, SOS, and PSO algorithms with a task count ranging from 100 to 1,000. As illustrated in Figure.2.- 4, G_SOS achieves the shortest makespan, lowest cost and minimum response time.

Table 2. Parameter Settings for SOS and PSO

Algorithm	Parameter	Value
SOS	Ecosize	100
	Number of iterations	1000
PSO	Particle size	100
	Inertia weight, w	0.9 – 0.4
	Coefficients C_1 and C_2	2
	Number of iterations	1000

Table 3. Parameter Settings for CloudSim

Cloud Entity	Parameter	Value
Datacenter	Number	1
Host	Number	2
	Processing speed	1000000 MIPS
	RAM	20 GB
	Storage	1 Terabyte (TB)
	Bandwidth	10 GB/s
	Operating system	Linux
	Architecture	x86
	VMM	Xen
VM	Number	20
	Bandwidth	1GB/s
	Memory	0.5 GB
	Image size	10 GB
	Processing speed (MIPS)	100 – 5000
	Scheduler	Time-shared
Task	Number of tasks	100 - 1000

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 September 2021 doi:10.20944/preprints202109.0407.v1

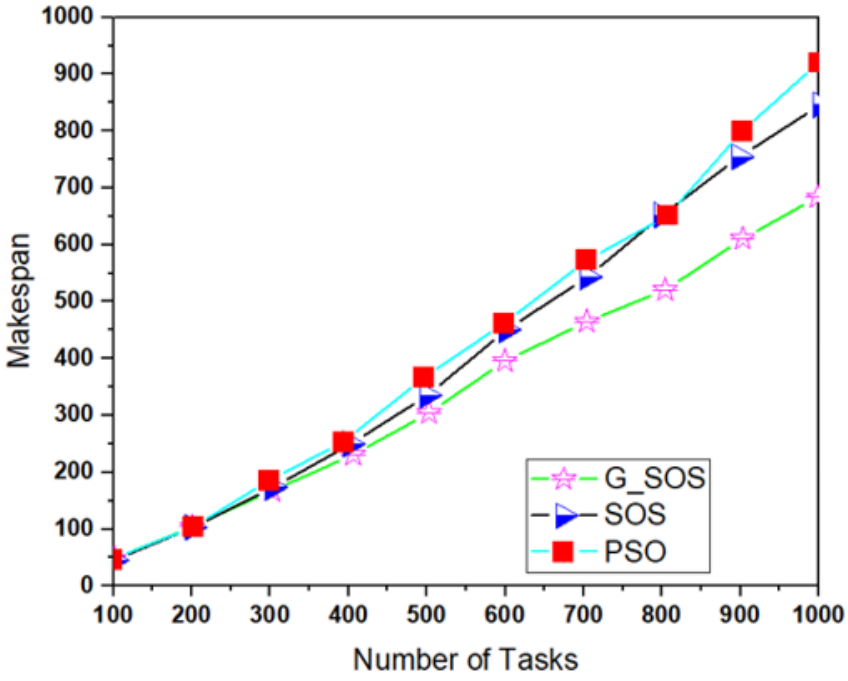

Figure.2. Makespan comparison between G_SOS, SOS and PSO (Uniform Distribution)

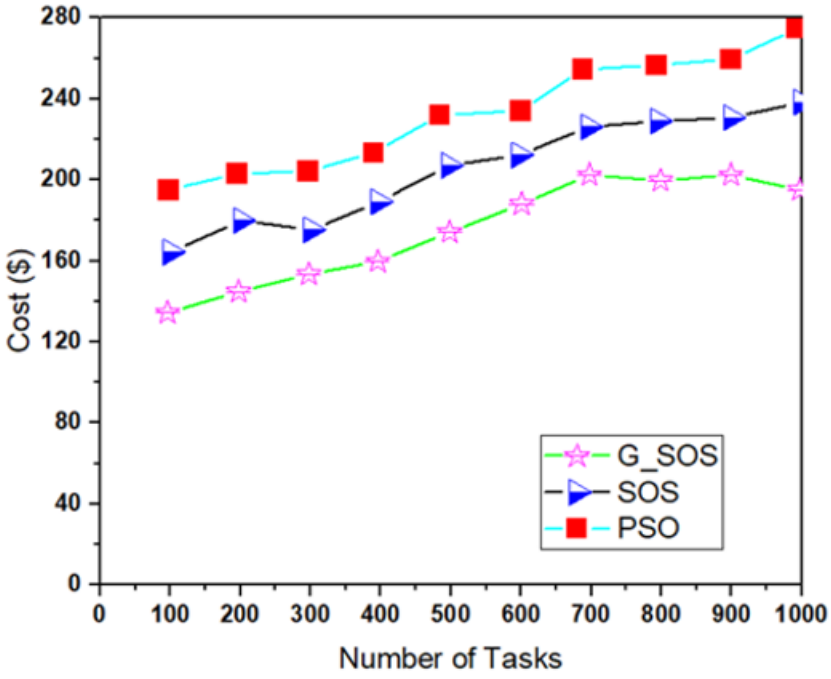

Figure.3. Average cost corresponding to the number of tasks

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 September 2021 doi:10.20944/preprints202109.0407.v1

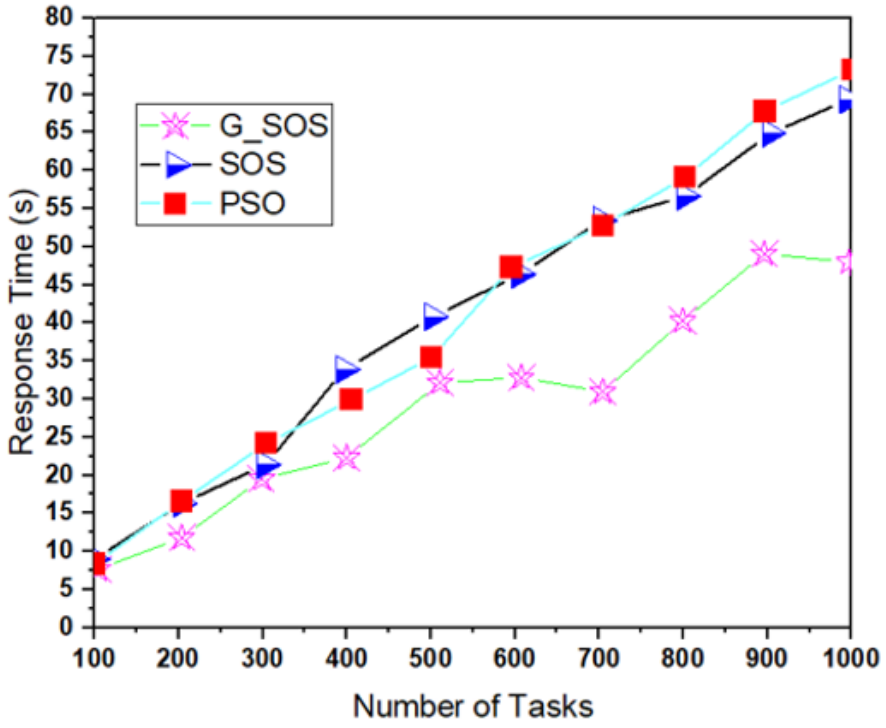

Figure.4. Response comparison between G_SOS, SOS and PSO (Uniform Distribution)

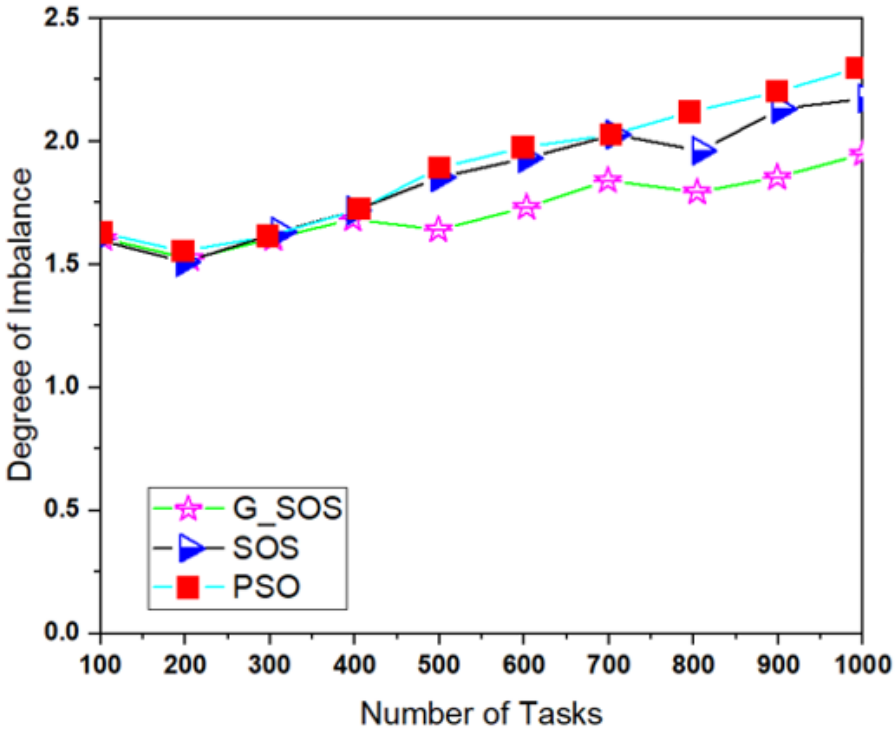

Figure.5. Degree of Imbalance comparison between G_SOS, SOS and PSO

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 September 2021 doi:10.20944/preprints202109.0407.v1

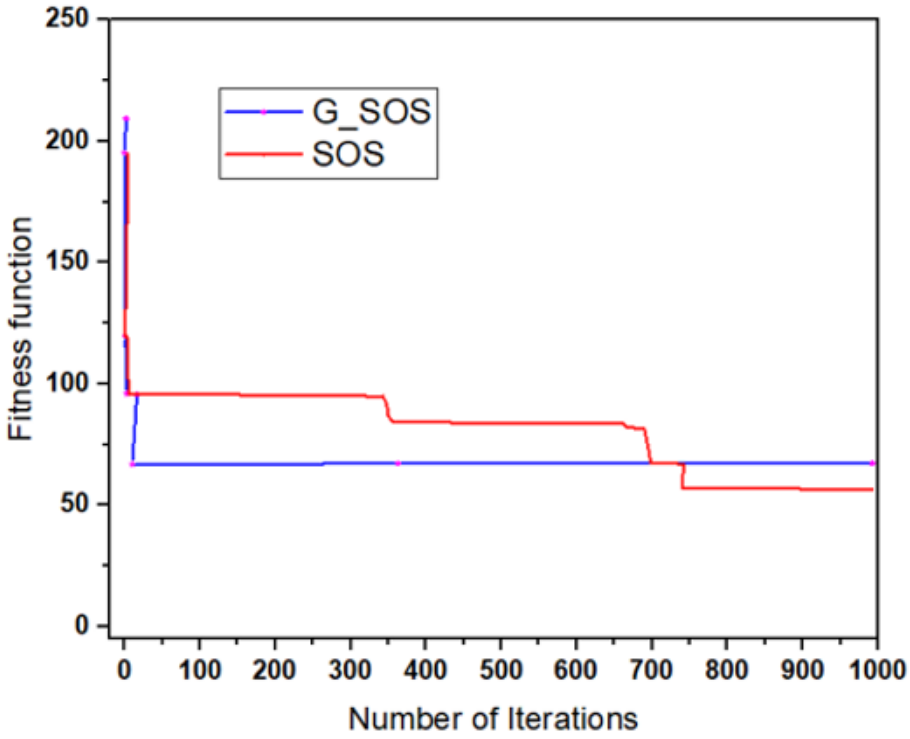

Figure.6. Convergence graph with sample tasks

As observed from the experimental results, the techniques demonstrated an initial improvement in the quality of solutions, but G_SOS proved the ability to improve the quality of solutions at a later point of the search process. G_SOS produces higher-quality solutions than regular SOS and PSO, especially when the problem size is enormous. Figure.5. depicts G_SOS having the lowest degree of imbalance among the heterogenous virtual machines than the traditional SOS and PSO. Similarly, G_SOS's search direction tends to converge to a stable position after fewer iterations, as shown in Figure.6. The new technique can be used alone or in combination with other metaheuristic algorithms to address various optimization problems in cloud computing and other disciplines.

6. Conclusions and Recommendations

This paper presents a modified symbiotic organism search optimization algorithm called G_SOS motivated by the symbiotic relationship in organisms. The proposed method uses a modified SOS algorithm that changes the relation characteristic of species from the arithmetic mean, which signifies equality, to the geometric mean, which signifies equity, to address the heterogeneous nature of the cloud resources. According to simulation results, the proposed approach, G_SOS, outperforms regular SOS in terms of convergence speed, cost, and makespan. There are various other optimization issues in cloud computing systems that can be solved using the proposed method and other discrete optimization problems in different domains. In the future, we intend to hybridize G_SOS with other effective local search and metaheuristic algorithms.

Author Contributions: Conceptualization, A.A.Z., S.B.A.R. and M.A.B.N.; methodology A.A.Z., S.B.A.R. and M.A.B.N.; formal analysis, A.A.Z.; investigation, A.A.Z.; data curation, A.A.Z.; writing—original draft preparation, A.A.Z.; writing—review and editing A.A.Z., S.B.A.R. and M.A.B.N.; visualization, A.A.Z.; supervision, S.B.A.R. and M.A.B.N.; project administration, S.B.A.R.; All authors have read and agreed to the published version of the manuscript

Data Availability: The outcomes of this study are supported by the data that are available on request from the corresponding author.

Conflicts of Interest: The authors have all declared that they have no conflicts of interest.

References

1. Usman MJ, Ismail AS, Chizari H, Abdul-Salaam G, Usman AM, Gital AY, et al. Energy-efficient Virtual Machine Allocation Technique Using Flower Pollination Algorithm in Cloud Datacenter: A Panacea to Green Computing. J Bionic Eng 2019;16:354–66. https://doi.org/10.1007/s42235-019-0030-7.
2. Samee NMA, Ahmed SS, Seoud RAAAA. Metaheuristic algorithms for independent task scheduling in symmetric and asymmetric cloud computing environment. J Comput Sci 2019;15:594–611. https://doi.org/10.3844/jcssp.2019.594.611.
3. Zhou Z, Li F, Zhu H, Xie H, Abawajy JH, Chowdhury MU. An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments. Neural Comput Appl 2020;32:1531–41. https://doi.org/10.1007/s00521-019-04119-7.
4. Seth S, Singh N. Dynamic heterogeneous shortest job first (DHSJF): a task scheduling approach for heterogeneous cloud computing systems. Int J Inf Technol 2019;11:653–7. https://doi.org/10.1007/s41870-018-0156-6.
5. Zhao Y, Calheiros R, Gange G, Bailey J, Sinnott R. SLA-Based Profit Optimization Resource Scheduling for Big Data Analytics-as-a-Service Platforms in Cloud Computing Environments. IEEE Trans Cloud Comput 2018;PP:1. https://doi.org/10.1109/TCC.2018.2889956.
6. Lavanya M, Shanthi B, Saravanan S. Multi objective task scheduling algorithm based on SLA and processing time suitable for cloud environment. Comput Commun 2020;151:183–95. https://doi.org/10.1016/j.comcom.2019.12.050.
7. Stephen A, Benedict S, Kumar RPA. Monitoring IaaS using various cloud monitors. Cluster Comput 2019;22:12459–71. https://doi.org/10.1007/s10586-017-1657-y.
8. Choe S, Li B, Ri I, Paek C, Rim J, Yun S. Improved hybrid symbiotic organism search task-scheduling algorithm for cloud computing. KSII Trans Internet Inf Syst 2018;12:3516–41. https://doi.org/10.3837/tiis.2018.08.001.
9. Nandhini JM, Gnanasekaran T. Enhanced fault identification and optimal task prediction (EFIOTP) algorithm during multi-resource utilization in cloud-based knowledge and personal computing. Pers Ubiquitous Comput 2019. https://doi.org/10.1007/s00779-019-01265-6.
10. Sindhu S, Mukherjee S. An evolutionary approach to schedule deadline constrained bag of tasks in a cloud. Int J Bio-Inspired Comput 2018;11:229–38. https://doi.org/10.1504/IJBIC.2018.092799.
11. Elaziz MA, Xiong S, Jayasena KPN, Li L. Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution. Knowledge-Based Syst 2019;169:39–52. https://doi.org/10.1016/j.knosys.2019.01.023.
12. Shirani MR, Safi-Esfahani F. Dynamic scheduling of tasks in cloud computing applying dragonfly algorithm, biogeography-based optimization algorithm and Mexican hat wavelet. vol. 77. Springer US; 2021. https://doi.org/10.1007/s11227-020-03317-8.
13. Zubair AA, Razak SBA, Ngadi MAB, Ahmed A, Madni SHH. Convergence-based task scheduling techniques in cloud computing: A review. vol. 1073. 2020. https://doi.org/10.1007/978-3-030-33582-3_22.

14. Madni SHH, Abd Latiff MS, Abdulhamid SM, Ali J. Hybrid gradient descent cuckoo search (HGDCS) algorithm for resource scheduling in IaaS cloud computing environment. Cluster Comput 2019;22:301–34. https://doi.org/10.1007/s10586-018-2856-x.
15. Arul Xavier VM, Annadurai S. Chaotic social spider algorithm for load balance aware task scheduling in cloud computing. Cluster Comput 2019;22:287–97. https://doi.org/10.1007/s10586-018-1823-x.
16. Abdullahi M, Ngadi MA, Abdulhamid SM. Symbiotic Organism Search optimization based task scheduling in cloud computing environment. Futur Gener Comput Syst 2016;56:640–50. https://doi.org/10.1016/j.future.2015.08.006.
17. Gabi D, Universiti Teknologi Malaysia MIA (Faculty of C, Universiti Teknologi Malaysia MZA (Faculty of C, Universiti Teknologi Malaysia MZZ (Faculty of C, Al-Khasawneh A. Hybrid Cat Swarm Optimization and Simulated Annealing for dynamic task scheduling on Cloud Computing Environment. J Inf Commun Technol 2018;3:435–67.
18. Abdullahi M, Ngadi MA, Dishing SI, Abdulhamid SM, Ahmad BI eel. An efficient symbiotic organisms search algorithm with chaotic optimization strategy for multi-objective task scheduling problems in cloud computing environment. J Netw Comput Appl 2019;133:60–74. https://doi.org/10.1016/j.jnca.2019.02.005.
19. Gabi D, Ismail AS, Zainal A, Zakaria Z, Abraham A, Dankolo NM. Cloud customers service selection scheme based on improved conventional cat swarm optimization. Neural Comput Appl 2020;6. https://doi.org/10.1007/s00521-020-04834-6.
20. Cheng MY, Prayogo D. Symbiotic Organisms Search: A new metaheuristic optimization algorithm. Comput Struct 2014;139:98–112. https://doi.org/10.1016/j.compstruc.2014.03.007.
21. Panda A, Pani S. A Symbiotic Organisms Search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems. Appl Soft Comput J 2016;46:344–60. https://doi.org/10.1016/j.asoc.2016.04.030.
22. Banerjee S, Chattopadhyay S. Optimization of Three-Dimensional Turbo Code using Novel Symbiotic Organism Search Algorithm Subhabrata, IEEE; 2016.
23. Ezugwu AES, Adewumi AO. Discrete symbiotic organisms search algorithm for travelling salesman problem. Expert Syst Appl 2017;87:70–8. https://doi.org/10.1016/j.eswa.2017.06.007.
24. Abdullahi M, Ngadi MA. Hybrid symbiotic organisms search optimization algorithm for scheduling of tasks on cloud computing environment. PLoS One 2016;11:1–29. https://doi.org/10.1371/journal.pone.0158229.
25. Ezugwu AE, Adeleke OJ, Viriri S. Symbiotic organisms search algorithm for the unrelated parallel machines scheduling with sequence-dependent setup times. PLoS One 2018;13:1–23. https://doi.org/10.1371/journal.pone.0200030.
26. Yu VF, Redi AANP, Yang CL, Ruskartina E, Santosa B. Symbiotic organisms search and two solution representations for solving the capacitated vehicle routing problem. Appl Soft Comput J 2017;52:657–72. https://doi.org/10.1016/j.asoc.2016.10.006.
27. Truong KH, Nallagownden P, Elamvazuthi I, Vo DN. A Quasi-Oppositional-Chaotic Symbiotic Organisms Search algorithm for optimal allocation of DG in radial distribution networks. Appl Soft Comput J 2020;88:106067. https://doi.org/10.1016/j.asoc.2020.106067.
28. Yang XS. Nature-inspired optimization algorithms: Challenges and open problems. J Comput Sci 2020;46:101104. https://doi.org/10.1016/j.jocs.2020.101104.
29. Madni SHH, Latiff MSA, Ali J, Abdulhamid SM. Multi-objective-Oriented Cuckoo Search Optimization-Based Resource Scheduling Algorithm for Clouds. Arab J Sci Eng 2019;44:3585–602. https://doi.org/10.1007/s13369-018-3602-7.

30. Zubair AA, Abd Razak S Bin, Ngadi A Bin, Ahmed A. Current Perspective of Symbiotic Organisms Search Technique in Cloud Computing Environment : A Review. Int J Adv Comput Sci Appl 2021;12:446–53. https://doi.org/10.14569/IJACSA.2021.0120650.
31. Ezugwu AE, Prayogo D. Symbiotic Organisms Search Algorithm: theory, recent advances and applications. Expert Syst Appl 2019;119:184–209. https://doi.org/10.1016/j.eswa.2018.10.045.
32. Nama S, Saha AK, Ghosh S. Improved symbiotic organisms search algorithm for solving unconstrained function optimization. Decis Sci Lett 2016;5:361–80. https://doi.org/10.5267/j.dsl.2016.2.004.
33. Banerjee S, Chattopadhyay S. Power Optimization of Three Dimensional Turbo Code Using a Novel Modified Symbiotic Organism Search (MSOS) Algorithm. Wirel Pers Commun 2017;92:941–68. https://doi.org/10.1007/s11277-016-3586-0.
34. Tejani GG, Savsani VJ, Patel VK. Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization. J Comput Des Eng 2016;3:226–49. https://doi.org/10.1016/j.jcde.2016.02.003.
35. Miao F, Zhou Y, Luo Q. A modified symbiotic organisms search algorithm for unmanned combat aerial vehicle route planning problem. J Oper Res Soc 2019;70:21–52. https://doi.org/10.1080/01605682.2017.1418151.
36. Gabi D, Ismail AS, Zainal A, Zakaria Z. Quality of service task scheduling algorithm for time-cost trade off scheduling problem in cloud computing environment. Int J Intell Syst Technol Appl 2019;18:448–69. https://doi.org/10.1504/IJISTA.2019.101952.
37. Liu J, Pacitti E, Valduriez P, de Oliveira D, Mattoso M. Multi-objective scheduling of Scientific Workflows in multisite clouds. Futur Gener Comput Syst 2016;63:76–95. https://doi.org/10.1016/j.future.2016.04.014.
38. Liu Y, Shu W, Zhang C. A parallel task scheduling optimization algorithm based on clonal operator in green cloud computing. J Commun 2016;11:185–91. https://doi.org/10.12720/jcm.11.2.185-191.
39. Thirumalaiselvan C, Venkatachalam V. A strategic performance of virtual task scheduling in multi cloud environment. Cluster Comput 2019;22:9589–97. https://doi.org/10.1007/s10586-017-1268-7.
40. Panda SK, Member I, Jana PK, Member IS. A Multi-Objective Task Scheduling Algorithm for Heterogeneous Multi-Cloud Environment. 2015 Int. Conf. Electron. Des. Comput. Networks Autom. Verif., Shilong, India: IEEE; 2015, p. 82–7.
41. Srivastava D, Kalra M. Improved symbiotic organism search based approach for scheduling jobs in cloud. Lect Notes Networks Syst 2020;116:453–61. https://doi.org/10.1007/978-981-15-3020-3_39.
42. Calheiros RN, Ranjan R, Beloglazov A, And CAFDR, Buyya R. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw - Pract Exp 2011;41:23–50. https://doi.org/10.1002/spe.995.