International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Improved Method for Selecting Web Services with Normalized Qos in Cloud Environments Using Optimization Algorithms

L. Thenmozhi[1]* **N. Chandrakala[2]**

[1]*M.G.R. College, Hosur, Tamilnadu, India*
[2]*SSM College of Arts and Science, Salem, Tamilnadu, India*
*Corresponding author's Email: *thenmozhilakshmanan@gmail.com*

Abstract: Cloud Computing makes use of the Internet to deliver standard and virtualized resources, software, and data as a service. Users will have the option to take use of all of these advancements in information technologies with cloud computing approach without needing to be extremely informed or competent in each one separately. An enormous number of services are currently available. One of the main challenges in the cloud computing environment is the lack of a workable method for developing online web services that may achieve high potential for Quality of Service while following to service level agreement (SLA) boundaries. Choosing or composing those combined services is referred to as nondeterministic polynomial time optimization problem. Many metaheuristic algorithms have been used in the past due to the NP-hard complexity of service composition. The reliability and QoS (Quality of Service) issues with other grid computing models are addressed. This paper presents the Coaching Based Multi-Verse Optimization (CBMVO) methodology in Web services, covering their composition and implementation methods as a metaheuristic algorithm to achieve the desired goals. The application of optimization techniques to choose web services in cloud settings with a normalised quality of service (QoS). By considering various QoS metrics, employing a hybrid optimization approach, and taking user preferences into account throughout the selection process, the method outperforms previous methods. These developments lead to a more effective and customized choice of web services, eventually enhancing user experience across cloud environments. The proposed methodology is evaluated and the results are validated the efficiency and performance of the proposed approach in relation to reliability, availability and cost. The new algorithm for quality of service stands out from other methods because it has a higher score of 0.89 while the MFO method only scored 0.77. This means that the suggested technique has improved the quality of service by 13% compared to the CBMVO method.

Keywords: Cloud computing, Metaheuristic algorithm, Quality of service (QoS), Service level agreement (SLA), Coaching based multi-verse optimization algorithm (CBMVO).

1. Introduction

Today's cloud computing services provide a wide range of possibilities, from the essentials of storage, networking, and processing power to cutting-edge innovations and technologies like artificial intelligence and natural language processing as well as conventional office applications. It is a distributed cloud-based system where a number of devices are inter connected together in order to make use of the resources offered by the service provider [1]. More than simply "lifting and transferring" outdated IT infrastructures to the cloud for cost and convenience's sake represent digital transformation. We are fully aware of the demands placed on modern technology and the necessity for constant innovation. Because of this, businesses use Cloud to alter their infrastructure and overcome their most difficult problems [2]. In addition to the expansion of services as a whole, a number of factors have also influenced cloud computing. Here are some of these patterns most significant explanations.

Containerization

Containerization is the bundling of software code with only the OS libraries and dependencies necessary to run the code in order to generate a single, lightweight executable, or container, which reliably functions on any infrastructure. Containers have replaced virtual machines (VMs) as the standard compute unit for contemporary cloud-native apps because they are more portable and resource-efficient [3].

Serverless computing

Computing that is performed on controlled infrastructure is known as serverless computing. It frees up analysis, software development, Operations and deployment teams from having to worry about back-ends and allows engineers to concentrate only on the functionality of applications or code. Google workspace cloud functions and microsoft azure functions are a few instances of serverless services [4].

Edge computing

The expectations for latency and accessibility have been significantly altered by the distributed nature of cloud services. Users and data transfers between cloud services have had to grow both in number and volume. By bringing computation and data analytics processes closer to people and devices, edge computing has contributed to this expansion. With the growth and aggregation of big data and the internet of things (IoT), edge computing is creating much more opportunity [5, 6].

High performance computing

All of the major cloud computing provides HPC (high performance processing), which is substantially parallelized and offers massive computing capabilities at an affordable price. Businesses employ these HPC cloud services for computationally heavy applications including analysis the big volume of data, risk operation and structural genomics [7]. Even while the cloud has made capacity provisioning much simpler, managing quality-of-service (QoS) now presents a number of fresh issues. This motive of service is essential for both cloud providers and customers, who must achieve the correct balance between operating expenses and service levels. Users expect providers to fulfil the claimed quality features [8]. With a study of new QoS modelling techniques that are suitable to cloud computing and a description of their initial implementation in cloud resource management, this paper intends to help such efforts [9-42].

Modelling of CC workload

For QoS models to have strong predictive capability, correct workload models must be defined. Here, we review studies on workload characterisation and related modelling methods. To use benchmarking to describe the QoS displayed by cloud deployment scenarios. For example, network bandwidth variance, virtual machine (VM) startup timings, and start failure rates, need to be estimated with realistic values. For several VM instance types, observations of performance variability have been made and reported [10].

Workloads of cloud computing application

The quality-of-service (QoS) applies also to Traffic control methods that strive to either give predictable or guaranteed performance to applications, sessions or to distinguish performance in accordance with application or network operator requirements. Packet delay and other types of losses are fundamental quality-of-service phenomena. [11-12]. A flow is a series of packets travelling from a source to a destination. A flow is said to strive for a certain level of quality of service. All of the packets in a flow in a connection-oriented network follow the same order. Every packet in a network without connections may take a different path. The main parameters can be used to categorise each flow's requirements are Reliability, Delay and Bandwidth [13, 14]. In order to mix different services or select the best services from those that have similar functionality but varied Quality of Service to fulfil the user's request, a service composition algorithm is required Which service was tasked with that duty has to be identified [11-14].

Currently, with y tasks and x related services of function things for each type of task, there are now xy with composition of service options. An NP-hard optimization issue is the service composition problem [12]. As a result, a service composition that offers high quality of service while meeting the service level agreement (SLA) is needed [15]. In a short amount of time, metaheuristic algorithms are effective instruments for finding a solution that is close to ideal. As a result, a few metaheuristic algorithms have been developed to handle the NP-hard problem of service composition. To attain higher performance, however, it is necessary to find algorithms whose performances can be enhanced.

An effective approach for handling numerical optimization issues is the multi-verse optimization (MVO) algorithm [16]. A new, coaching based multiverse optimization (CBMVO) approach is presented on this work to address SLA restrictions while solving the service composition problem. The following list of MVO's enhancements is provided [17]. The search space around the best answer is determined by the travelling distance rate in order to locate a better solution. The travelling distance rate, which drops linearly with each iteration, is the algorithm's initial flaw. While in the suggested technique, iterations that fail to produce the new, improved answer are increased while adaptively decreasing iterations are increased. Instead of using the single best option, with the collection best optimization solutions p has been produced [43]. In order to maintain diversity in the algorithm's first step, the wormholes mechanism used by MVO is modified. In order to reach the best outcome, the chance of blackhole existence increases as more things are exchanged between worlds via white and black holes [44].

The further part of this paper can be structured as given below: In part 2 discussed an overview of the literature on stochastic optimization approaches, part 3 addresses multiverse theory topics and coaching based multiverse optimization methodology (CBMVO) for QoS in cloud based web services, part 4 reveals the performance of results achieved with the newly proposed methodology and part 5, concludes the proposed work and the possibility of enhancements.

2. Related works

The problem of service composition is NP-hard and has a broad range of possible solutions. As a result, numerous meta-heuristic techniques have been used to resolve this issue. For the purpose of resolving the Quality of Service - aware service composition challenge, a genetic algorithm-based solution was designed and evaluated against comprehensive study along with randomized selection genetic algorithms [18, 19].

To solve the issue of web service composition with quality-of-service awareness, an orthogonal GA has been proposed; in this algorithm, the initial population is constructed using an orthogonal design. It performs better than a conventional genetic algorithm. In multi-cloud environments, the CloudPick framework and ontology-based services was proposed for the deployment of quality of service. It offers a utility computing service repository so users may access the quality of service of several clouds. For optimization, it employed a GA built selection method. Nevertheless, this methodology shows certain limitations corresponding to time of search, a poor convergence rate, and a local optimum caused by a fixed, planned crossover between two parents [20].

The cloud manufacturing service composition employed a hybrid artificial bee colony algorithm. This method generates individuals using both chaos and probabilistic operators. Compare to particle swarm optimization and genetic algorithm it performs better. ABC algorithm (artificial bee colony), however, exhibits delayed convergence properties [21, 22]. For the composition of cloud services, a fuzzy-based bat algorithm was presented. In this study, relationships between QoS components have been taken into account to evaluate the effectiveness of the bat algorithm using a fuzzy method. The system performance is dependent on the users' mined rules with required membership functions, which is a drawback of built fuzzy logic before realisation, more adjusting [23]. The difficult issue of scheduling user workloads while taking into account users Quality of Service (QoS) criteria with the aim of lowering physical machine energy consumption [24].

To address the shortcomings of conventional methods, we employ a backtracking-based algorithm and a GA extension for enhancing quality online service design adhering to agreement requirements. An evolutionary algorithm and simulated tempering were coupled to build the algorithm has taken to improve the on-demand service design while simultaneously taking SLA limitations into consideration in the cloud environment. The local optimum possible solutions are reached by using the single point crossover [25].

To resolve the QoS cloud service composition, a new method called social learning optimization that mimics the process of human intellect evolution has been implemented. It performs well in terms of search ability. However, the fact that it operates on three co-evolution spaces makes it more difficult. An algorithm built on the principles of ant colony optimization has been proposed, using the fewest possible utility computing in dispersed on demand environments whereas taking response time and service are to be consider into account [26].

The artificial bee colony (ABC) algorithm is then introduced with a set of services to pick from, based on the needs of each user, if the fitness function of the services chosen by the genetic algorithm (GA) is suitable. The system is tested using Cloud SIM simulation, and the numerical outcomes demonstrate its effectiveness in terms of dependability, availability, and cost [27, 28]. In cloud environments for web service composition, a new method meta-heuristic optimization cuckoo algorithm was presented. Fuzzy linear programming and rough set

Received: March 2, 2023. Revised: April 28, 2023.

Table 1 Studies of QoS criteria

Study	Optimization algorithm	QoS terms	Algorithms Associated	Results	Study contribution
[18]	Linear Programming	Time, cost, and reliability	Greedy, GA_WSC, CSA_WSC	Reduces the web service composition search space.	Improving QoS
[19]	GA	Time, Fitness, Random selection	GA VS. Integer Programming Approach, Culture Algorithm	Improve the performance	Robust service composition
[20]	GA-Based Heuristic Algorithms	Execution time	GA, BSMA, Sun GA	Performance and Efficiency	Quality of Service (QoS) for multimedia application
[21]	Hybrid artificial bee colony (HABC)	Time complexity	GA, PSO, BABC	Service Optimal Selection	Performance in CMfg service optimal selection
[23]	Fuzzy-based bat algorithm	Availability, Response Time, Reliability, Cost	Simple Additive Weighting (SAW), Fuzzy Approach	QoS Preference	Cloud Service Selection
[24]	HPSO Algorithm	Load balancing, Time, Scheduling	PSO, Greedy algorithm, GSA	Reduce the system's cost consumption.	Improvement in the prediction's reliability
[25]	Extended GA	Generalized composite service	backtracking algorithm, Genetic Algorithm	Outperforms the standard one	Choosing a cloud service to achieve an improved solution
[31]	ESWOA	Evaluation of Service	GA, WOA, DGABC, HGA, and GRASP	QoS factors	Ensures an optimal balance

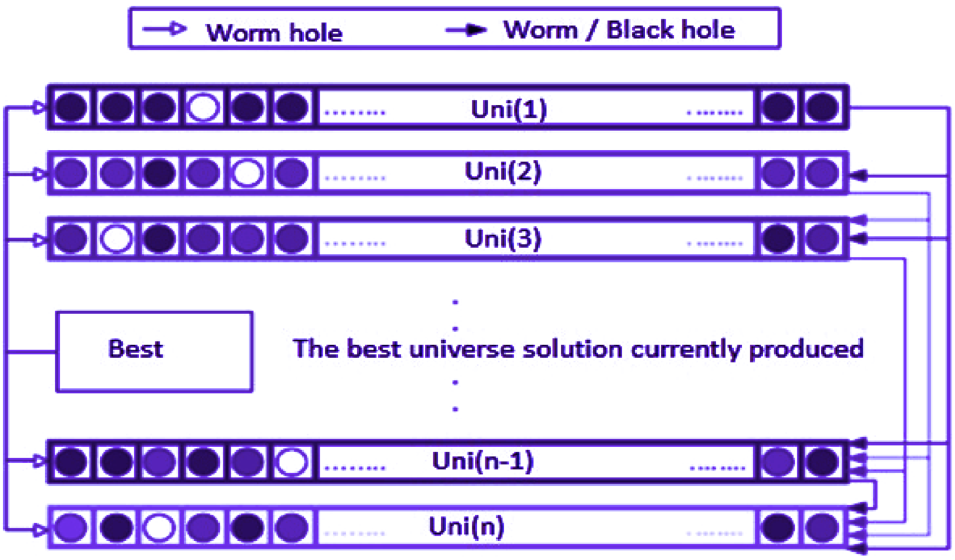


Figure 1. Presents conceptual model of the suggested algorithm (CBMVO)

methodology are proposed in the direction of resolve competing constraints for online services. But before realisation, fuzzy logic's rules and membership functions still need some fine tweaking. Results demonstrate the employed algorithm's great performance. Modified invasive weed optimization was used to balance multiple QoS characteristics, meet connectivity requirements for cloud service composition, and optimise multiple QoS parameters [29].

A multi-criteria decision-making process called simple additive weighting (SAW) has been presented, which can assist customers in selecting the existing cloud service that best suits their needs from a pool of available options. In addition, an improved eagle strategy with whale optimization (ESWOA) methodology to keep the local and global optimums in balance. From a selection of readily available cloud services, our system can suggest time-saving, reliable, trustworthy cloud services [30]. As part of the eagle strategy to develop the algorithm, the whale optimization algorithm's exploitation method was applied in this algorithm. But in this work, SLA restrictions were not taken into account [31]. To choose the most effective service per request in a geographically dispersed cloud environment for web service composition problem, we present a linear optimization programming solution, known as "LP-WSC." This approach will help us to increase the performance criteria for the on demand QoS [32].

3. CBMVO method for QoS in cloud based services

3.1 Inspiration

A recently developed optimization technique that draws inspiration from nature is called "multi-verse optimization algorithm. For thoughtful three planetary notions are used to represented as black, white and worm holes help as the substance for the multiverse optimizer (MVO). These notions have respective mathematical models for exploration, exploitation and local search. Bestowing in the direction of big bang theory, the cosmos began as a huge explosion. Another current and well-known hypothesis among physicists is the multi-verse theory [33]. This idea, there were multiple big bangs, and each one resulted in the creation of a new universe. Multiverse, which suggests the existence of worlds other than the one we all now occupy, is the opposite of universe. According to the multiverse theory, each universe may have its own unique set of physical laws [34]. White, black and worm holes are the primary ideas from the multi-verse theory that we used as models for the MVO algorithm. Physicists think that the big bang can be compared to a white hole and may have been the driving force behind the birth of the universe, despite the fact that a white hole has never been seen in our universe [35]. White holes behave radically differently from black holes, which have been found regularly [36].

They have a tremendously strong gravitational pull that attracts everything, including light beams [37]. Wormholes are the openings that link various regions of the universe together. According to the multiverse theory, wormholes serve as time-and-space conduits that enable items to immediately travel between any two points in a universe [38]. Every cosmos experience perpetual inflation, which results in the expansion of the universe through space [39]. It is important for a universe's pace of inflation to create stars, planets, asteroids, black holes, white holes, wormholes, physical laws, and be conducive for life. One of the cyclic multi-verse theories suggests that worm, black and white holes allow interaction across several universes to get to a stable state [40].

3.2 MVO algorithm

The algorithm is made to address numerous optimization issues that crop up in the fields of engineering, science, and other fields. The main goal of MVO is to simulate a large-scale multiverse made up of several smaller worlds. MVO explores the idea of multiverses and worlds. The multiverse is an optimization problem with various solution sets, and the universe is a set of solutions to that issue. The multiverse is first initialised by the algorithm with a random distribution of universes. Every universe stands for a potential answer to the issue at hand. The programme then rates each universe in the population according to its fitness. After assessing the original population's fitness, MVO goes on to carry out a series of cosmic events. and work to raise the standard of applicants in the field. MVO is superior to conventional optimization methods because it can mimic cosmic events, which aid in escaping local optima and successfully explore the solution space. Also, it is a global optimization technique, which enables us to tackle issues for which there are several viable solutions or none at all.

In summary, the multi-verse optimization (MVO) method is a swarm-based metaheuristic optimization algorithm that uses the behaviour of the real world to tackle different optimization issues. The technique, which is regarded as a reliable and fast method for resolving global optimization issues, makes advantage of cosmic occurrences to enhance the quality of the solution set.

This illustration demonstrates between universes how white/black hole tunnels enable items to travel. When a white/black tunnel forms between the two worlds, it is assumed that the universe with the greater inflation rate has a white hole, while the universe with the lower inflation rate is assumed to have black holes. Then, the items are transferred from the source universe's white holes to the destination universe's black holes. The universes can swap items with ease thanks to this technique. We make the assumption that universes with high inflation rates are very likely to include white holes in order to improve the overall inflation rate of the universes. Contrarily, black holes are more likely to exist in worlds with modest rates of inflation. We used a roulette wheel method to exchange the elements of universes and mathematically describe the white/black hole tunnels. According to the inflation rates for each iteration, the universes are sorted and a white hole is randomly selected in one of them using a roulette wheel. To do this, the following actions are taken.
Consider that

$$Universe\ (U) = \begin{bmatrix} a_1^1 & a_1^2 & \cdots & \cdots & a_1^d \\ a_2^1 & a_2^2 & \cdots & \ldots & a_2^d \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_n^1 & a_n^2 & \ldots & \cdots & a_n^d \end{bmatrix} \quad (1)$$

Where n denotes the universe in numbers and d represents the variable parameters.

$$a_i^j = \begin{Bmatrix} a_k^j & r1 < Norm(Ui) \\ a_i^j & r1 \geq Norm(Ui) \end{Bmatrix} \quad (2)$$

Where a_i^j denotes the jth parameter for the ith position Universe, random number r1 in the range [0, 1], a_k^j denotes the jth parameter for the kth position Universe chosen using a selection procedure roulette wheel method, $Norm(Ui)$ denotes the Universe ith normalization with inflation rate.

3.3 Coaching based multi-verse optimization algorithm (CBMVOA)

Three fundamental components of the initial algorithm were enhanced in the modified MVO. The algorithm's travelling distance rate (TDR) reveals the first flaw in the design. The original algorithm's TDR reduces linearly with each iteration. To put it another way, the original algorithm's Travelling Distance Rate (TDR) is as follows:

$$\mathrm{TD}_{\mathrm{Rate}_{original}} = 1 - (Iteration/Max\_Iteration) \quad (3)$$

Travelling distance rate (TDR) starts off large and then gradually diminishes throughout the repetitions. When the finest answer may not be best after a certain number of iterations, the suggested algorithm's TDR is adaptive and can rapidly decrease. Since this affects the search space size surrounding the finest solution to discover one that is better than the best solution already found, failure to improve the best solution indicates that Travelling Distance Rate is inappropriate and that TDR should be reduced even more dramatically. In this research, the method was cut in semi and the procedure was repetitive after there was no better solution than the one that had already been found after several iterations.

The successful pattern search algorithms that were used to solve optimization issues are where this concept originated. If a solution cannot be improved using the pattern of points surrounding it, the pattern's scale is lowered (frequently by partial), and new design of arguments is created about the existing result. This process is repeated until no better solution can be found. The term "unsuccessful iteration" refers to the number of attempts made by an algorithm to improve upon the best answer already in existence.
The wormhole-transferred objects are shown by white points. It is possible to notice that wormholes randomly alter the universes items without taking into account their rates of inflation. In order to provide local modifications for each universe and have a high possibility of accelerating the inflation rate via wormholes, we assume that wormhole tunnels are continuously constructed between a universe and the best world created thus far. This mechanism's formulation is as follows:

$$a_i^j \begin{cases} \begin{cases} a_{best}^j + TD_{Rate} * \left((ubound_j - lbound_j) * r4 + lbound_j \right) \\ \qquad r3 > 0.5 \;\; r_2 < WE\_Prob \\ a_{best}^j - TD_{Rate} * \left((ubound_j - lbound_j) * r4 + lbound_j \right) \\ \qquad r3 \leq 0.5 \;\; r_2 < WE\_Prob \end{cases} \\ \qquad a_i^j \qquad r_2 \geq WE\_Prob \end{cases} \quad (4)$$

Where r2 < WE_Prob & r2 ≥ WE_Prob respectively, where aj designates the jth value of parameter for the finest universe yet formed, TD_Rate and WE_Prob are coefficients, lower bound (lboundj) denotes the jth value of variable, upper bound (uboundj) denotes the jth value of variable, a_i^j denotes the jth parameter

value of the universe ith, and r2, r3, and r4 (random numbers) in [0, 1].
The below logic shows the adaptive coefficients:

$$WE\_Prob = \max + cur\_iter \ * \ (\frac{\max - min}{\max \ \_iter}) \quad (5)$$

Where cur_iter is the current iteration, min denotes the minimum (in this case, 0.2), max denotes the maximum, and maximum iterations denotes max_iter.

$$TD\_Rate \ = 1 - \frac{cur\_iter^{1/p}}{max\_iter^{1/p}} \quad (6)$$

Where p is the iteration accuracy of the exploitation. The faster and extra precise the exploitation/local search is the greater p. In the MVO method, the formation of a collection of random universes is first step of optimization process. At each iteration, matter from high-inflation worlds tends to travel through black or white holes to universes with rates of low inflation. While this is happening, wormholes randomly transfer stuff from every universe to the finest universe. The shown methods are repeated till the requirement is satisfied (for example, after a predetermined maximum number of repetitions). The complexity of the suggested methods depends on the number of iterations, the universe count, the roulette wheel mechanism, and the universe sorting technique.

3.4 Proposed approach

The following observations help illustrate how the suggested technique theoretically could be used to resolve optimization issues:

- Since white holes are more likely to originate in worlds with high inflation rates, they can transport matter to other universes to help them increase their inflation rates.
- Black holes are more likely to form in the presence of low inflation rates, which enhances the possibility that they will absorb matter from other universes. This increases the likelihood that inflation rates may
- increase once again in worlds with low rates.
- White/black hole tunnels commonly transport things from universes with high inflation rates to those with low inflation rates, improving the overall/average inflation rate of all universes over the course of iterations.
- Wormholes often develop at random in any universe, independent of inflation rate, preserving the variety of worlds throughout time.
- Sudden changes can be used to break up local optima stagnations.

The composite service's QoS is dependent on the following pattern. Sequential modes, Parallel modes, Probabilistic modes, and Circular modes of composition patterns are discussed in 41.It contain the QoS estimations for various patterns. Where rtj, ayj, ryj, ctj of service jth. probj remains chance of implementation. These four Quality of Service serve as the foundation for the SLA restrictions. The method takes as inputs a task list T = {T1, T2, T3, T4, ….,Tn} their process workflow, consider z atomic service as S = {s1, s2, s3, s4, ..., sz} for each job, QoS for every atomic service SL is denoted as <rtl, ayl, ryl, ctl>. As a result, there are several ways to allocate an atomic service from z services to n separate jobs and SLA vector SLA = <SLArt, SLAay, SLAry, SLAct>. The atomic service that maximises the best service with satisfying service agreement constraints are chosen as output for each job from a list of z atomic services.

Coaching based multiverse optimization algorithm (CBMVO) is continuous enhanced method. In this study, the chosen fissionable facility for the job jth & ith web design result is regarded to be the continuous value of a_j^i computed by the procedure converted to a numerical value of a_j^i. $indx_j^i$ = rnd(a_j^i).

Cost and benefit criterion are two types of quality-of-service characteristics. Higher value of attributes is equivalent to lower quality. Greater attribute values (such as dependability and availability) are equivalent to higher quality in terms of a benefit criterion. The following relationship can be used to normalise quality of service attributes.

$$Val^{norm(k)} = \begin{cases} \frac{val^k - min_k}{max_k - min_k} \ val^k \in benefit\_criteria \\ \frac{max_k - val^k}{max_k - min_k} \ val^k \in cost\_criteria \end{cases} \quad (7)$$

Where valk is the QoS attribute's value before normalisation, mink and maxk are the attribute's minimum and maximum values. The normalisation value for cost and benefit attributes is valnorm(k).
In the event that the user's selections for the kth Quality of Service attribute is $\gamma^k \in [0,1] \ in \ \sum_{k-1}^{0} \gamma^k = 1$ for 0 parameters. The fitness value can be constrained by attributes of the complex service that satisfy the users can be taken as

Received: March 2, 2023. Revised: April 28, 2023.

$$fit = \sum_{k-1}^{0} \gamma^{k}{}_{val^{norm(k)}} \tag{8}$$

When a user's constraints are not fulfilled for a result, a penalty is functional that is lower than the usual fitness to demonstrate the poor performance. Below formula shows the fitness for a non-satisfied constraint result:

$$fit = \sum_{k-1}^{0} \gamma^{k}{}_{val^{norm(k)}} - BetaVal\ X\ PenaltyVal - NumConf \tag{9}$$

Be aware that an answer's fitness depends on its quality and is strongest when the solution's is of high quality. As a result, the regular fitness should be reduced by the penalty level. A user-defined constant, the Beta. When maintaining SLA criteria is more crucial than the significance of QoS, beta should be high, and vice versa. Based on how much the characteristics deviate from the attribute limits, as can be seen below, the consequence amount is determined.

$$Uni^{norm(k)} = \begin{cases} \frac{SL^k - val^k}{max_k - min_k},\ val^k \in \\ benefit\_criteria, and\ not\ fulfilled\ constraint \\ \frac{val^k - SL^k}{max_k - min_k},\ val^k \in \\ cost\_criteria, and\ not\ fulfilled\ constraint \\ 0 \qquad Otherwise \end{cases} \tag{10}$$

$$Penalty = \sum_{k=1}^{0} Uni^{norm(k)} \tag{11}$$

To ensure that any solution with unsatisfied requirements has a lower fitness value than any solution with fulfilled constraints, the number of quality of service characteristics that could not meet service level agreement criteria, termed NoConflict, is removed from fitness benefits in relation to the penalty values. When the kth feature of quality of service in a service composition satisfies the specified service level agreement, constraint, uninorm(k) is zero. The values of maxk and values of mink represent the highest and lowest of the kth quality of service characteristic, respectively. The SLk is the service level agreement restriction that the user has requested for the kth property. In this study, the characteristics of quality of service include time of response, resources availability, cost and reliable of service. A cost attribute must have a value smaller than the user's required service level agreement for such an attribute. The provider may deliver quality service with a shorter reaction period than the user requires (service level agreement for time of response).

When the response time is smaller than the necessary response time, the service level agreement or constraint for response time is fulfilled, and uninorm(1) = 0; the value of penalty is unaffected by this property. When the response time exceeds the necessary time of response, the value of penalty is raised by uninorm(1) = (val1 – SL1) / (max1 – min1). As a result, if the value of a cost attribute exceeds the compartment service level agreements, the penalty may increase.

When the values of benefit qualities such as reliability and resources availability are less than the service level agreements restrictions, the penalty is imposed. It will rise by uninorm(k) = (SLk - valk) / (maxk – mink) as advantage features must take a greater value than Service Level Agreements to meet the same constraint. Raising the value of penalty reduces the fitness. As an outcome, the method looks for a solution that has higher fitness or a lower penalty, which is the same as a solution that resolves any conflicts between service level agreement criteria.

4. Results

Several optimization strategies, were employed to assess the suggested approach including multi-verse optimization (MVO) algorithm, particle swarm optimization (PSO) algorithm, and genetic algorithm (GA). In which algorithms parameter settings may be summarised as follows. The total population of all methods is 30. The following are the parameters that are often utilised in the literature for various methods. The genetic algorithm has a crossover probability (probc) of 0.8 and a probability of mutation (probm) of 0.01. The particle swarm optimization (PSO) algorithm parameters are 0.5 for the cognitive component, 1.25 for the social element, and 0.4 for the inertia weight, which falls gradually from the values 0.9 to 0.4. The suggested Multi-Verse Optimization (MVO) algorithm parameters are based on the features of the basic MVO that were considered in prior research and the suitable values of parameter settings were determined experimentally. In MVO and CBMVO the maximum and lowest WE_Prob, i.e. maxWE_Prob & minWE_Prob are 0.2 and 1 respectively. Upper and lower likelihood of blackhole presence, i.e. maxBProp & minBProp, are also the values of 1, 0.2.

Benchmark functions to evaluate

The suggested technique was evaluated using benchmark functions that were both unimodal and

Received: March 2, 2023. Revised: April 28, 2023.

multimodal, as shown in [2, 8] with a dimension Value of 50. The suggested algorithm's ability to be used is examined by the unimodal functions. The multimodal functions put the method's capacity to explore and depart from the local optimum to the test.

The findings for these tasks, including the mean, minimum and standard deviation are shown in Table 1. Results are acquired for each function after thirty runs and thousand repetitions for algorithms. In this study as fundamental involvement is the introduction of an algorithm for the creation of web services with service level agreement that are quality of service aware. Table 2 benchmark function findings, however, demonstrate that increased multi-verse optimization might produce the best outcomes in the majority of (F(9) out of F(13) functions. Overall, it is possible to state that the suggested algorithm's results are the optimized performance solution. The Schwefel function, often known as function F(3). For this situation, the Particle swarm optimization (PSO) could produce best outcomes. During the search space, the PSO maintains the best minimum value in terms of local associated to every person in the environment. This mechanism might work well for issues of this nature. Thus, taking into account the localized best in multi-verse optimization can also be taken into account to gain better performance. The F(5) has a fairly smooth slope surrounding the global optimum location and is known as the Rosenbrock function. Even though the minimum value for GA throughout the 30 runs was better (6.741), its standard deviation as well as mean was worse than the suggested improved MVO. The rate of inflation, based on criterion selecting white and black holes across algorithm iterations, might be adjusted to account for the smooth slope of the F(5) in order to further optimise the suggested method. The F(8) and F(9) are also referred to as the Rastrigrin and Schwefel functions, respectively. Numerous peaks and valleys are present in these functions. The global minimum may not be found by many algorithms. For these issues to be resolved, algorithms need more variety. Instead of using the best solution, the proposed approach uses the best solutions of p. Both the algorithm diversity and the diversity of best solution can rise with an increase in p.

Coaching based multiverse optimization algorithm (CBMVO)

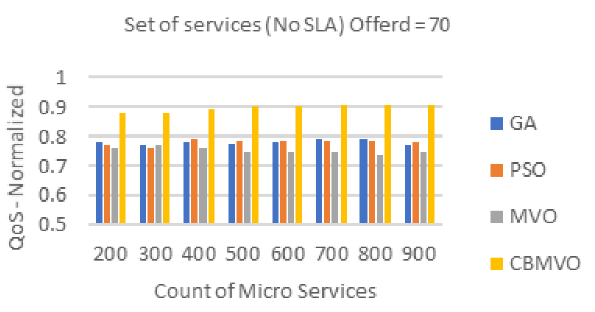
Four QoS characteristics of the atomic activities in this work were chosen from [15] in the following manner to simulate the suggested algorithm: Response time is based on a distribution with a mean value in the 20–1500 millisecond range. The value of availability is constant from the the range of 0.95 to 1. Values for cost and dependability are uniformly distributed at random within the intervals of 2–15 and 0.4–1, respectively, and the users have equal preferences. The service level agreement (SLA) restrictions for a service selection with an assumption that the QoS. For QoS attributes, web service selection with SLA = <SLArt, SLAay, SLAry, SLAct> are taken into consideration. Different circumstances are taken into account to demonstrate the performance of the suggested method. For five alternative scenarios, the number of services is taken into consideration as five different values of: 10, 30, 50, 70 & 90 respectively. The estimated number of services with atomic for every set of value is $\{200, 300, 400, 500, 600, 700, 800, 900\}$. Ten simulation runs for every situation were used to average the findings. The suggested service composition algorithm is contrasted through the previously described well-known algorithms, including GA, PSO and MVO. Normalised service for suggested technique in comparison to other algorithms is shown in Fig. 2. Every graph compares the normalised Quality of Service for a certain set of services with various numbers of services for every service. The statistics diagram demonstrates the suggested methodology performs improved compared than others. For a limited number of services, the performance of the proposed method is comparable to current algorithms but in more dimensional settings, the proposed algorithm's differences from other methods quality of service are more prominent.

In the figure, with 90 service sets and 900 micro services per service set, the normalised quality of service for the suggested technique is 0.89, whereas MFO is 0.77. This is a 13% improvement in quality of service over the CBMVO method. This is due to the proposed algorithm's strong performance in the area of global optimization and its ability to prevent convergence rate to limited cost. When there are few micro services in each group of services, there are fewer possibilities for each service, fewer potentially high-quality options are available, and quality of service will suffer as a result. The service composition problem has a larger and more challenging solution space as a result of the growing range of micro services for each services set and the total number of services sets. Comparison of quality-of-service attributes with regard to function evaluations is shown in Fig. 3. Although the suggested method has better quality of service when compared to other algorithms, this does not necessarily mean that it is superior in terms of all quality-of-service aspects. Cloud computing web

Received: March 2, 2023. Revised: April 28, 2023.

Table 2 Benchmark function findings

F	GA Algorithm			PSO Algorithm		
	M	A	SD	M	A	SD
F(1)	0.017	0.215	0.092	0.019	0.281	0.317
F(2)	1.812	5.771	2.104	0.899	3.101	0.951
F(3)	6.943	26.001	35.91	**2.408**	**7.384**	**4.034**
F(4)	2.003	2.881	0.514	0.512	0.574	0.067
F(5)	**6.948**	**104.07**	**63.99**	105.73	152.1	40.12
F(6)	0.165	0.528	0.491	0.201	1.011	0.725
F(7)	0.912	1.91	0.801	0.431	1.541	0.697
F(8)	-725	-712	44	-1621	-902	293
F(9)	**9.845**	**27.01**	**8.12**	24.03	40.12	10.91
F(10)	0.847	1.178	0.401	0.187	1.41	0.401
F(11)	**0.007**	**0.008**	**0.007**	0.002	0.024	0.015
F(12)	0.004	0.007	0.097	0.012	0.031	0.032
F(13)	0.006	0.054	0.044	0.032	0.247	0.069
F	MVO Algorithm			CBMVO Algorithm		
	M	A	SD	M	A	SD
F(1)	2.301	6980	8214	**6.40E-04**	**1.20E-03**	**2.20E-04**
F(2)	10.124	63.954	36.478	**1.40E-02**	**1.70E-02**	**2.30E-03**
F(3)	2.30E+04	5.70E+04	2.34E+04	47.8	105.2	38.1
F(4)	71.214	83.05	6.478	**0.076**	**0.351**	**0.165**
F(5)	799.102	1.30E+07	3.70E+07	35.97	66.67	32.14
F(6)	2.103	8801	10501	**0.001**	**0.001**	**0**
F(7)	0.312	22.074	24.177	**0.016**	**0.041**	**0.015**
F(8)	**-16578**	**-13207**	**1432**	-11901	-10391	524
F(9)	196.9	315.71	58.19	23.11	40.74	8.98
F(10)	18.123	19.784	0.436	**0.006**	**0.007**	**0.001**
F(11)	0.891	63.178	66.174	**0.001**	**0.006**	**0.007**
F(12)	4.314	8.71E+05	4.70E+07	**0**	**0.005**	**0.021**
F(13)	17.574	1.40E+07	7.50E+07	**0**	**0.004**	**0.007**



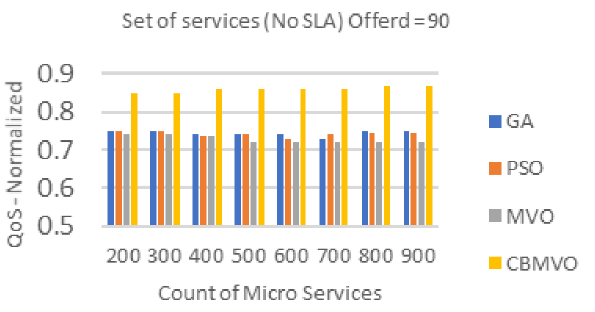


Figure 2. Normalized QoS - set of services (70,90) without SLA constraints for a range of microservices per set

service composition challenges with constraints of SLA are investigated, simulated & explored in the sections that follow. Simulations were run five times for 10, 30, 50, 70, and 90 situations, with 200, 300, 400, 500, 600, 700, 800, and 900 atomic services in each set.

5. Conclusion and future enhancements

Utility or on-demand computing provides platform to consumers over the revolutionized communications. The proposed approach for choosing web services with normalised quality of service (QoS) in cloud environments using optimization algorithms is a effective solution that incorporates a number of essential components.

Firstly, the technique takes into account the significance of normalisation, which is necessary for the accurate evaluation and comparison of QoS indicators across various service providers. A readily comparable collection of well-defined, normalised QoS values across various providers is produced as a result of this normalisation procedure, which is

Received: March 2, 2023. Revised: April 28, 2023.

Figure 3. Normalized QoS - set of services (70,90) with SLA constraints for a range of microservices per set

Table 3 Notation list

Parameter	**Notations**
n	universe in numbers
U	Universe
d	variable parameters
$Norm(Ui)$	Normalization with Universe
TDR	Travelling Distance Rate
ubound	Upper bound
Lbound	Lower bound
p	iteration accuracy of the exploitation
QoS	Quality of Service
SLA	Service Level Agreements
fit	Fitness
z	atomic service
T	Task list
WE_Prob	Wormhole Existence Probability
Val	value of a cost attribute
Prob	Probability
Probc	crossover probability
Probm	probability of mutation
F(x)	benchmark function findings

carried out using multiverse optimizer (MVO) methodologies. Secondly, based on the normalised QoS values, optimization techniques are used to choose the optimal service provider for a specific or collection of tasks. The technique is very adaptive and versatile since the multiverse optimizer (MVO) and coaching based multi-verse optimization algorithm (CBMVOA) algorithms may be personalized to meet the unique demands and requirements of various end users. Finally, the approach is made to operate in cloud environments, which are growing more and more well-liked because of their scalability, adaptability, and affordability. This is taken into consideration in the suggested strategy, which guarantees that the chosen web services can operate effectively in cloud environments, minimizing performance issues and ensuring optimal resource allocation.

Overall, the proposed method for selecting web services with normalised QoS in cloud environments combines normalisation techniques, optimization algorithms, and cloud-specific considerations to make it an effective and valuable approach for end users looking to improve their web service selection process.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Author 1: Data collection, concept, analysis, methodology, writing - original draft preparation, software, and writing - review, and editing. Author 2: The supervision, review, editing, investigation.

References

[1] J. Chen, T. Du, and G. Xiao, "A multi-objective optimization for resource allocation of emergent demands in cloud computing", *Journal of Cloud Computing: Advances, Systems and Applications*, Vol. 10, No. 1, pp. 1-17, 2021.

[2] C. Jatoth, G. R. Gangadharan, and R. Buyya, "Computational Intelligence Based QoS-Aware Web Service Composition: A Systematic Literature Review", *IEEE Transactions on Services Computing*, Vol. 10, No.3, pp. 475-492, 2017.

[3] S. Hardikar, P. Ahirwar, and S. Rajan, "Containerization: Cloud Computing based Inspiration Technology for Adoption through Docker and Kubernetes", In: *Proc. of 2021 Second International Conference on Electronics and Sustainable Communication Systems, Coimbatore, India*, pp. 1996-2003, 2021.

[4] H. B. Hassan, S. A. Barakat, and Q. I. Sarhan, "Survey on serverless computing", *Journal of*

Cloud Computing: Advances, Systems and Applications, Vol. 10, No. 1, pp. 1-29, 2021.

[5] K. Cao, Y. Liu, G. Meng, and Q. Sun "An Overview on Edge Computing Research", *IEEE Access*, Vol. 8, No. 1, pp. 85714-85728, 2020.

[6] C. Sathish and K. Srinivasan, "An Artificial Bee Colony algorithm for Efficient Optimized Data Aggregation to Agricultural IoT Devices Application", *Journal of Applied Science and Engineering*, Vol. 24, No. 6, pp. 927-936, 2021.

[7] E. Roloff, M. Diener, A. Carissimi, and P. O. A. Navaux, "High Performance Computing in the cloud: Deployment, performance and cost efficiency", In: *Proc. of 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings, Taipei, Taiwan*, pp. 371-378, 2012.

[8] A. Jula, E. Sundararajan and Z. Othman, "Cloud computing service composition: A systematic literature review", *Expert Systems with Applications*, Vol. 41, No. 8, pp. 3809-3824, 2014.

[9] D. Petcua, G. Macariua, S. Panica, and C. Crăciuna, "Portable Cloud applications - From theory to practice", *Future Generation Computer Systems*, Vol. 29, No. 6, pp. 1417-1430, 2013.

[10] D. Magalhães, R. N. Calheiros, R. Buyya, and D. G. Gomes, "Workload modeling for resource usage analysis and simulation in cloud computing", *Computers & Electrical Engineering*, Vol. 47, No. 1, pp. 69-81, 2015.

[11] S. Benmakrelouf, C. S. Onge, N. Kara, H. Tout, C. Edstrom, and Y. Lemieux, "Abnormal behavior detection using resource level to service level metrics mapping in virtualized systems", *Future Generation Computer Systems*, Vol. 102, No. 1, pp. 680-700, 2020.

[12] D. Ardagna, G. Casale, M. Ciavotta, J. F. Pérez2, and W. Wang, "Quality-of-service in cloud computing: modeling techniques and their applications", *Journal of Internet Services and Applications*, Vol. 5, No. 11, pp.1-17, 2014.

[13] S. G. H. Tabatabaei, W. M. N. W. Kadir, and S. Ibrahim, "A Comparative Evaluation of State-of-the-Art Approaches for Web Service Composition", In: *Proc. of 2008 The Third International Conference on Software Engineering Advances, Sliema, Malta*, pp. 488-493, 2008.

[14] D. Ardagna and B. Pernici, "Adaptive Service Composition in Flexible Processes", *IEEE Transactions on Software Engineering*, Vol. 33, No. 6, pp. 369-384, 2007.

[15] D. Wang, H. Ding, Y. Yang, Z. Mi, L. Liu, and Z. Xiong, "QoS and SLA Aware Web Service Composition in Cloud Environment", *KSII Transactions on Internet and Information Systems*, Vol. 10, No. 12, pp. 5794-5811, 2016.

[16] M. J. Reddy and D. N. Kumar, "An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design", *Engineering Optimization*, Vol. 39, No. 1, pp. 49-68, 2007.

[17] G. I. Sayed, A. Darwish, and A. E. Hassanien, "Quantum multiverse optimization algorithm for optimization problems", *Neural Computing and Applications*, Vol. 31, No. 1, pp. 2763-2780, 2019.

[18] M. G. Arani and A. Souri, "LP-WSC: a linear programming approach for web service composition in geographically distributed cloud environments", *The Journal of Supercomputing*, Vol. 75, No. 1, pp. 2603-2628, 2019.

[19] M. Zhang, L. Liu, and S. Liu, "Genetic Algorithm Based QoS-aware Service Composition in Multi-cloud", In: *Proc. of 2015 IEEE Conference on Collaboration and Internet Computing, Hangzhou, China*, pp. 113-118, 2015.

[20] A. T. Haghighat, K. Faez, M. Dehghan, A. Mowlaei, and Y. Ghahremani, "GA-Based Heuristic Algorithms for QoS Based Multicast Routing", *Knowledge-Based Systems*, Vol. 16, No. 5, pp. 305-312, 2003.

[21] J. Zhou and X. Yao, "A hybrid artificial bee colony algorithm for optimal selection of QoS-based cloud manufacturing service composition", *The International Journal of Advanced Manufacturing Technology*, Vol. 88, No. 1, pp. 3371–3387, 2017.

[22] C. Zhang and B. Zhang, "A Hybrid Artificial Bee Colony Algorithm for the Service Selection Problem", *Discrete Dynamics in Nature and Society*, Vol. 2014, No. 1, pp. 1-13, 2014.

[23] A. Ezenwoke and M. Adigun, "Integrating fuzzy theory and visualization for QoS-aware selection of SaaS in cloud e-Marketplaces", *Cogent Engineering*, Vol. 8, No. 1, pp. 1-22, 2021.

[24] M. A. Rakrouki and N. Alharbe, "QoS-Aware Algorithm Based on Task Flow Scheduling in Cloud Computing Environment", *Sensors*, Vol. 22, No. 7, pp. 1-20, 2022.

[25] Q. Wu, F. Ishikawa, Q. Zhu, and D. H. Shin, "QoS-Aware Multigranularity Service Composition: Modeling and Optimization", *In IEEE Transactions on Systems, Man, and*

Cybernetics: Systems, Vol. 46, No. 11, pp. 1565-1577, 2016.

[26] Z. Z. Liu, D. H. Chu, C. Song, X. Xue, and B. Y. Lu, "Social learning optimization (SLO) algorithm paradigm and its application in QoS-aware cloud service composition", *Information Sciences*, Vol. 326, No. 1, pp. 315-333, 2016,

[27] S. S. Sefati and S. Halunga, "A Hybrid Service Selection and Composition for Cloud Computing Using the Adaptive Penalty Function in Genetic and Artificial Bee Colony Algorithm", *Sensors*, Vol. 22, No. 13, pp. 1-22, 2022.

[28] M. B. Karimi, A. Isazadeh and A. M. Rahmani, "QoS-aware service composition in cloud computing using data mining techniques and genetic algorithm", *The Journal of Supercomputing*, Vol. 73, No. 4, pp. 1387-1415, 2017.

[29] M. G. Arani, A. A. Rahmanian, M. S. Aslanpour, and S. E. Dashti, "CSA-WSC: cuckoo search algorithm for web service composition in cloud environments", *Soft Computing*, Vol. 22, No. 1, pp. 8353-8378, 2018.

[30] S. S. Mohapatra, R. R. Kumar, and J. Pradhan, "Hybrid eagle strategy for QOS-based cloud service composition", *Journal of Information and Optimization Sciences*, Vol. 43, No. 5, pp. 1047-1059, 2022.

[31] S. K. Gavvala, C. Jatoth, G. R. Gangadharan, and R. Buyya, "QoS-aware cloud service composition using eagle strategy", *Future Generation Computer Systems*, Vol. 90, No. 1, pp. 273-290, 2019.

[32] M. G. Arani and A. Souri, "LP-WSC: a linear programming approach for web service composition in geographically distributed cloud environments", *The Journal of Supercomputing*, Vol. 75, No. 1, pp. 2603-2628, 2019.

[33] O. K. Erol and I. Eksin, "A new optimization method: Big Bang–Big Crunch", *Advances in Engineering Software*, Vol. 37, No. 2, pp. 106-111, 2006.

[34] K. Gurumurthy and M. V. Selvadurai1, "Intelligent Resource Scheduling with Neutrosophic Knowledge and Optimized Cache Management Using Cuckoo Search Method in Cloud Computing", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 3, pp. 327-338, 2020, doi: 10.22266/ijies2020.0630.30.

[35] D. M. Eardley, "Death of White Holes in the Early Universe", *Physical Review Letters*, Vol. 33, No. 7, pp. 442-444, 1974.

[36] P. J. Steinhardt and N. Turok, "A Cyclic Model of the Universe", *Science*, Vol. 296, No. 5572, pp. 1436-1439, 2002.

[37] P. C. W. Davies, "Thermodynamics of Black Holes", *Reports on Progress in Physics*, Vol. 41, No. 8, pp. 1313-1355, 1978.

[38] M. S. Morris and K. S. Thorne, "Wormholes in spacetime and their use for interstellar travel: A tool for teaching general relativity", *American Journal of Physics*, Vol. 56, No. 5, pp. 395-412, 1988.

[39] A. H. Guth, "Eternal inflation and its implications", *Journal of Physics A: Mathematical and Theoretical*, Vol. 40, No. 25, pp. 6811-6826, 2007.

[40] P. J. Steinhardt and N. Turok, "The cyclic model simplified", *New Astronomy Reviews, Elsevier*, Vol. 49, Nos. 2-6, pp. 43-57, 2005.

[41] W. Liu, B. Liu, D. Sun, Y. Li, and G. Ma, "Study on multi-task-oriented services composition and optimisation with the 'Multi-Composition for Each Task' pattern in cloud manufacturing systems", *International Journal of Computer Integrated Manufacturing*, Vol. 26, No. 8, pp. 786-805, 2013.

[42] P. U. Maheswari, L. Thenmozhi, and P. Ganeshkumar, "Distributed Detection of Clone Attacks in Wireless Sensor Networks using RED-ANT Algorithm", *Journal of Computer Applications*, Vol. 6, No. 3, pp. 43-46, 2013.

[43] L. Thenmozhi and N. Chandrakala, "Developed Modified Particle Swarm Optimization for Feature Selection on Learning Based Big Data in Cloud Computing", *Journal of Algebraic Statistics*, Vol. 13, No. 1, pp. 310-320, 2022.

[44] L. Thenmozhi and N. Chandrakala, "Weighed Quantum Particle Swarm Optimization Technique to Measure the Student Performance", *The Seybold Report*, Vol. 17, No. 106, pp. 984-995, 2022.