Hindawi
Mathematical Problems in Engineering
Volume 2024, Article ID 3136546, 11 pages
https://doi.org/10.1155/2024/3136546

Research Article

A New Integrated Approach for Cloud Service Composition and Sharing Using a Hybrid Algorithm

Jayaudhaya J.,[1] Jayaraj R.,[2] and Ramash Kumar K.[3]

[1]*Department of Electronics and Communication Engineering, RMD Engineering College, Chennai, India*
[2]*Department of Data Science and Business Systems, SRM Institute of Science and Technology, Chennai, India*
[3]*Department of Electrical and Electronics Engineering, Dr. NGP Institute of Technology, Coimbatore, India*

Correspondence should be addressed to Ramash Kumar K.; ramashkumar@drngpit.ac.in

Received 13 February 2023; Revised 30 September 2023; Accepted 30 January 2024; Published 12 February 2024

Academic Editor: Azeem Irshad

The concept of a "Smart City" emphasizes the need to employ information and communication technologies to strengthen the quality, connectivity, and efficiency of various municipal services. Cloud computing and the Internet of Things are shaping future tech. Both ideas greatly impact smart city application and solution development. Cloud computing is amazing at managing and storing remote service access. Several companies have switched to cloud leasing to reduce local resource burden. Due to the intricacy and flexibility of cloud-maintained services, selecting jobs that best suit client needs should be optimized. Quality of service criteria for each cloud service are the best tools for choosing and optimizing cloud carriers. Genetic algorithms (GAs) and ant colony optimization (ACO) are combined to make cloud computing. It is discovered that the recommended ACO + GA obtains an accuracy of 82% when compared to existing methods of energy- and reliability-aware multiobjective optimization method and the hybrid cuckoo particles swarm, artificial bee colony optimization (CPS + ABCO) where accuracy is 68% and 75%, respectively.

1. Introduction

Cloud computing is the next step in the evolution of internet-based computing, and it enables the usage of information technology skills as a utility [1]. The Internet of Things (IoT) has the potential to improve productivity, throughput, and efficiency when smart devices leave the cloud infrastructure environment. Smart cities are populated areas that systematically endeavor to understand the new locations of records and modern communications to attain environmental sustainability, urban system authority, enhanced health, knowledge growth, and network-driven advancement. The proposed algorithm based on Kalman filtering aims to intelligently suggest software and applications to users in the cloud computing network. It utilizes user log files and interests to estimate and predict the best recommendations. The algorithm offers advantages such as accurate estimation, adaptability to changing environments, and efficient resource allocation. Further research is needed to address scalability, privacy and security, and performance optimization for improved effectiveness [2]. The transmission of information and communications technology (ICT) resources across a network is made possible by cloud computing, which the next stage in the development of internet-based is computing. The IoT can gain from improved efficiency, effectiveness, and throughput in cloud infrastructure. The paper proposes an improved ant colony optimization (IACO) algorithm for load-aware service discovery in cloud computing [3]. The IACO addresses the load balancing problem of the traditional ACO algorithm and demonstrates improved performance in terms of reduced energy consumption, mitigated response time, and better service level agreement (SLA) compliance in cloud environments.

The primary emphasis of the IoT is on the problems that arise in an environment that is both changing and shared. The IoT is a broad categorization that encompasses a wide range of adaptable and unusual devices that have storage, power source, and performance characteristics that are limited [4]. These constraints, which include complex issues such as interoperability, efficiency, complete functionality, and availability, act as an obstacle and an obstruction to the development of IoT systems. Computing in the cloud is one of the most essential strategies that could be used in combination with the IoT to overcome these limitations. The cloud provides common resources that are notable for their accessibility, affordability, and stylistic attractiveness [5]. These resources include the network, storage, computers, and software. As a result, the difficulty of cloud computing is to create a composition approach that meets the needs of the clients in terms of quality of service (QoS) parameters, this is referred to as the QoS-aware cloud service composition challenge. The cloud computing and service selection, which also encourages dealing with the CSC problem, enhanced the service collaboration. The goal of CSC is to fulfill the client's request by locating an improved service or a group of providers that adheres to their QoS specifications. The client's requirement in CSC is broken down into a series of tasks called a workflow. Then, a group of services known as the recruitment services list is created by retrieving the services from various providers that are functionally comparable to the client's request but have varying QoS criteria for each activity from the cloud pool [6]. Both the number of suppliers and the quantity of services offered by these operators are growing. As a result, there are several different resources for each work from various providers, which makes it quite difficult to select a superior products combination. This issue fits the definition of an NP-hard optimization method. Many nature-inspired methods, such as ACO, ABCO, PSO, and cuckoo search, etc., were developed to address the CSC problem. The sophisticated behavior of the insects is inspired by swarm-based algorithms. This behavior results from a cooperative effort to look for food sources. A class of algorithms known as ACO mimics the way ant colonies forage for food.

The ACO family of algorithms contains a number of different algorithms, such as the ant system, ACS, and the max–min ant system. The worker ants in a colony communicate with one another in order to plan their foraging trips in such a way that they take the path that is the shortest distance between the colony and the food source. Pheromone is the channel through which they communicate with one another. Pheromone is a molecular substance that is left behind by traveling insects and steadily evaporates over time. The concentration of the fragrance is utilized in order to persuade additional ants to follow in the footprints of the original ant. In ACO algorithms, mechanical ants work together to discover a better solution, imitating the behavior of ant communities as they go about their business. In this research, a combined metaheuristic algorithm that is built on ACO (ACS principles were used), as well as GA, is recommended as a way to successfully resolve the CSC challenge. In addition to that, the evolutionary algorithm was utilized so that the ACS parameters could be independently tuned GA. Metropolisville, a fictional city, implemented a smart city initiative addressing traffic management, energy efficiency, waste management, public safety, connectivity, and data analytics. Results included reduced congestion, lower energy consumption, efficient waste collection, enhanced safety, universal high-speed internet access, and informed decision-making.

The main objectives of this article are as follows:

(1) Investigate both network and non-network QoS variables that play a significant role in determining service selection and composition.
(2) Develop a method that enforces restrictions on the number of clouds involved while optimizing the selection of the best services from multiple cloud providers.
(3) Utilize a multicriteria decision-making approach to evaluate and compare the QoS attributes of similar candidate services, aiding in the decision-making process.

The benefits of cloud computing and IoT in smart cities:

(1) Cloud computing:
    (a) Cost savings: Cloud computing eliminates the need for cities to invest in expensive infrastructure, reducing upfront costs, and ongoing maintenance expenses.
    (b) Scalability: Cloud platforms can easily scale up or down based on demand, ensuring that smart city applications can handle increasing data volumes without disruptions.
    (c) Accessibility: Cloud-based services are accessible from anywhere, allowing city officials and residents to access information and services on various devices.
(2) IoT:
    (a) Real-time data: IoT devices generate real-time data, enabling cities to monitor and respond to changes or issues promptly.
    (b) Efficient resource management: IoT sensors can track and optimize resource usage, such as energy consumption, reducing waste, and enhancing sustainability.
    (c) Improved quality of life: IoT-enabled systems, such as smart lighting or intelligent transportation, enhance safety, convenience, and overall quality of life for residents.

2. Literature Survey

Zhang et al. [7] developed a four-layer model of an IoT applications with edge computing. With regard to technology, we present a multiobjective optimization approach that is energy and serviceability and can maximize the utilization of resources of edge servers, the energy consumption of smart devices, and the system's resilience all at the same time while abiding by security constraints.

In Yu et al.'s [8] study, goal of this effort is to address the problem of resource and cost management in cloud infrastructure to offer cloud applications for smart cities a high service rate with sustainable infrastructure. The suggested model uses artificial neural networks and algorithms drawn from nature to lower execution costs, average start, and end times, and enhance system utilization while also making the system more strength.

In Dahan et al.'s [9] study, detection rate for smart cities remains a difficult challenge. This study suggests a score-level fusion-based improved multimodal biometric method for a smart city. In particular, the suggested method offers a solution for the issues by combining a multimodal fusion method with an improved fuzzy genetic algorithm (GA) that offers improved performance. Studies using various biometric settings show that existing methods can be significantly improved.

Alayed et al. [10] described the proposed structure's many steps, including the cloud broker, network administrator, customer devices, and customer requests (tasks). The cuckoo search algorithm, PSO, and ABCO are three optimization techniques that are used in the study to reduce the amount of time required to execute the stakeholder's demand. The three fields that make up the fitness function are CPU usage, turnaround time, and time required.

In Sefati and Halunga's [11] study, a platform for universal healthcare called Ube Health is proposed. To overcome the aforementioned problems, UbeHealth makes use of edge devices, deep learning, big data, HPC technology, and IoT. The application's three primary components and four layers offer an improved network QoS. The Cloudlet and network layers employ the reignited network traffic to optimize data rates, data caching, and routing decisions. Deep learning, big data, and HPC are used to make these predictions. The network layer can better address the communication needs of apps and report suspicious traffic and anomalous data when it comes to the division of network protocols into traffic flows. To distinguish between the many types of data coming from the same application protocols, clustering is used. Using the framework, a proof-of-concept UbeHealth system has been created.

To minimize the size of the information collected from the internet-of-energy ecosystem in a smart city, Ahanger et al. [12] provided a make a positive impression big data management method. Through the use of higher order single-value decomposition, the core data are recovered from the collected data utilizing tensor operations like matriculation, vectorization, and temporization. Then, the condensed version of this fundamental data is kept in the cloud. In smart cities, information is used for numerous services after its dimensionality has been reduced, and this study has examined how it might be applied to offer DR services [13].

For the purpose of computing and evaluating the significance degrees of various criteria, an innovative PLBW technique that is centered on the score value has been proposed. In order to evaluate IoT platforms, an innovative integrated probabilistic linguistic MCDM model that is based on the TODIM method has been suggested. This model is based on the PLBW method, two-tuple distance measure, and two-level probability degree [14].

In Huang et al.'s [15] study, we proposed a new score function for approximately representing PFSs, which is founded on the degrees of determinacy and indeterminacy. The original MULTIMOORA method is then modified to include the Dice distance and score function, and this modified version of the method is applied to the solution of the multicriteria decision-making problems that arise within the PFS information context.

In Lin et al.'s [16] study, an innovative technique known as LPF-TOPSIS is suggested as a solution to challenges involving the decision-making of numerous attributes. In conclusion, the LPF-TOPSIS technique is utilized in order to deal with a situation involving the selection of firewall manufacturing.

In Lin et al.'s [17] study to tackle the drawbacks of the existing correlation coefficients between PFSs, some novel directional correlation coefficients are put forward to compute the relationship between two PFSs by taking four parameters of the PFSs into consideration, which are the membership degree, nonmembership degree, strength of commitment, and direction of commitment. After that, there are two real-world cases that show how the proposed directional correlation coefficient can be used to diagnose diseases and how the proposed weighted directional correlation coefficient can be used in cluster analysis.

In Lin et al.'s [18] study, we suggested the IOLs to determine PFNs, which can show how the ADs and NDs in two PFNs interact with each other and how the ADs and ODs in two PFNs interact with each other.

In Vijayaraj et al.'s [19] study, TS-copy is a TOPSIS-based copy placement algorithm. First, a multiattribute grid that fully represents node efficiency and load is defined. Next, a TOPSIS-based program calculates data node efficiency scores. For combining numerous trait weights, the entropy weight method is presented. Next, copy location is based on the complete load values of each Spark cluster data node, rack, and cluster [20].

Yang et al.'s [21] Score-ESCA, a new score function based on the exponential semantic value and confidence estimate value for the PLTS-CI, is used to compare nodes with distinct voting attitudes. This method accurately selects the main node with the highest score by using complex judgment attitudes.

Hosseini Shirvani's [22] introduced BOTV-PSO, a biobjective optimization algorithm, to address cloud security risks in multicloud environments. BOTV-PSO outperforms other methods in terms of convergence, diversity, fitness, and scalability.

Abedpour et al. [23] addressed efficient resource allocation for IoT applications in fog computing. It combines k-means clustering and a GA to minimize errors and reduce latency. Simulation results show its superiority over other methods in terms of performance and delay rates.

Shirvani et al. [24] introduced a multiobjective task scheduling algorithm for efficient resource management in distributed systems. It leverages the fuzzy TOPSIS tool for multicriteria optimization, focusing on reducing power consumption and operational costs. The results demonstrate effective compromise between criteria based on their assigned weights.

The end-users (commercial and residential) are categorized into normal, overloaded, and under loaded categories using the core data using a support vector machine (SVM)-based classifier.

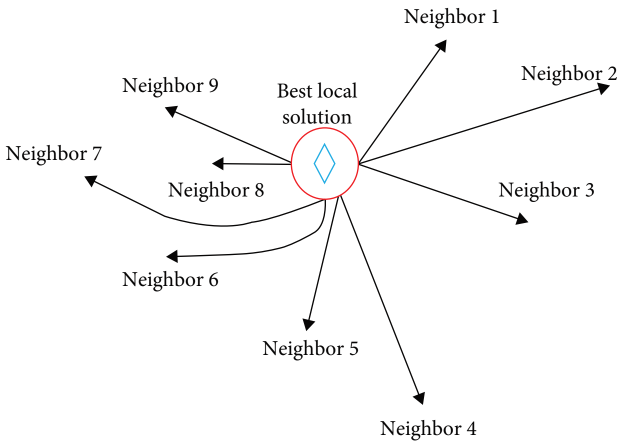

Figure 1: Flying process to get nearest neighbors.

3. An Adapted Ant-Inspired Algorithm for Enhancing Web Service Composition

This article suggests a method that has been motivated by ants for this kind of issue.

They termed it IACO. Figure 1 shows the flying ants added pheromone on additional nodes in addition to the ones directly in their path, increasing the possibility that those nodes will be investigated in later rounds [25]. When two nodes are close to one another, the amount of pheromone released on those nodes is inversely proportional to the distance between the nodes on the trail.

According to the authors, if pheromones are placed on surrounding nodes, FACO might analyze a larger population of solutions. This might avoid stagnation. Empirical studies reveal that FACO outperforms ACO for the WSC issue in order to improve its performance even if it requires a bit more time to run.

3.1. Method. FACO algorithm's objective is to maintain a healthy balance between exploitation and exploration. Imagine a hypothetical flying ant that injects its pheromone along its path like a typical ant and that explains the basic principle. Though numerous nodes might receive some of the implanted pheromone because it is administered when the ant is flying away. The nodes nearby would also receive pheromone, so it was not just the nodes along the path. The amount of pheromone delivered to those nodes would be inverse proportion to how far away they are from the nodes along the path. This increases the likelihood that these nearby nodes will be investigated in subsequent cycles. Assuming that the central node in picture has numerous neighbors and is located on the best path (local solution) during a certain iteration [26]. Initially, our flying ant applies pheromone to this node, and then, depending on the distance between each of its k closest neighbors and the central node, spreads the same amount of pheromone among them.

FACO, for the WSC QoS issue, can deal with two challenges in order to discover a solution for WSC utilizing FACO: first, we must identify the optimal nearest neighbors using QoS criteria. Second, we must apply the proper quantity of pheromone to the nearby nodes. There are numerous phases in the FACO algorithm, some of which resemble the classic ACO [27] FACO, on the other hand, differs from ACO in that it has to look for nearby nodes and inject pheromone into them. During the initial task, each ant gets originally randomly placed in a WS and then navigates using heuristic value and prior knowledge [28]. Equation is used to implement the localized version for the pheromone for every created solution. Finest ant uses equation to modify the global pheromone at the conclusion of each iteration. We just used the Euclidean distance to locate the closest WSs to the optimal solution discovered depending on the QoS values as in equation C, RT, A, and R in order to determine the closest neighbors for the QoS-aware composition problem: cost, reaction speed, accessibility, and dependability, in that order; x: the best WS outperforms the physically fittest in the current task; y: a member of the group of services performing the same task as x, and $x \neq y$. The distribution of the world-wide upgrade among the k neighbors is inversely related to how far they are from the optimal WS for each job. The following formula is created to accomplish that, where ij $(t+1)$, the pheromone originates from Equation (1). In the very same job, the standardization for the distances between W_{Sj} and its neighbor l, and we assume that just a part of ants were flying ants and the remainder are typical (waking) ants in order to include the advantages of both types of ants. The paper presents a new intelligent algorithm that utilizes Kalman filtering for estimation and prediction in cloud computing environments. The algorithm aims to identify servers with lower traffic loads and transfer high-level processes onto them. This approach prioritizes high-level processes and ensures QoS in the network. By leveraging Kalman filtering, the algorithm provides intelligence in resource allocation and optimization for efficient cloud computing [29].

The paper proposes a new system that addresses the challenge of selecting IT service providers among numerous companies in the market. The system gathers information about service providers, as well as the needs and system requirements of clients or organizations seeking services. It then utilizes this data to recommend the best and reliable service providers based on the specific needs and requirements of customers. This system aims to simplify the selection process and ensure that customers are matched with suitable service providers that meet their needs effectively [30].

4. Existing System

By offering a vast array of services via the clouds, the providers aim to satisfy the needs of the businesses and users. A service (basic task) or group of services may be able to fulfill these requirements (composite task). A function that may be broken down into a number of primitive tasks based on its functionality is referred to as a collective effort. The collective effort can be divided into a process for this breakdown, with each subtask defining a basic task. This study explores the impact of COVID-19 on HRM roles and the role of technology in empowering HRM. It emphasizes the importance of authentic leadership and prioritizing employee well-being.

Web-based solutions like cloud computing and teleworking are discussed as practical solutions. The study highlights the need for business innovation and remote working opportunities. Overall, the research contributes to advancing knowledge in the field of HRM in the context of COVID-19 [31].

The primary function of the ACO is to resolve issues of the kind of optimization algorithms that are defined as graphs $G = (V, E)$, where V denotes the collection of vertexes (nodes) and E denotes the collection of edges [32]. Although the Z ants traverse around the network nodes to identify the initial collection of solutions, each of the graph edges originally contains the same quantity of pheromone trail T 0. Equation contains the initial solution transition rule. Then, a series of steps are repeated until a stop condition is satisfied. This article explores the challenges of cloud-based IoT systems and the potential of edge computing to address them. It highlights the importance of node selection in IoT-edge environments and proposes a novel approach for block chain-enabled edge IoT that utilizes fuzzy logic and TOPSIS for node selection. The experimental results demonstrate the effectiveness of the proposed framework in improving relevant parameters [33].

When looking for novel solutions, the ACO family of algorithms uses many algorithms with various selection/updating method to choose the best node and modify pheromone trails. In this paper, we concentrate on ACS to address the CSC issue.

4.1. Multi-Agent ACS For Qos-Aware CSC Problem. Most of the time, the CSC problem's high level of complexity makes it challenging to find an ideal solution in a fair amount of time. Using searching decomposition and live population disruption, the suggested multi-agent ACS (MAACS) model can lessen the difficulty of the problem. The multiagent system, which consists of autonomous agents working together to simultaneously achieve shared goals, is frequently employed in distributed and parallel applications [34]. Additionally, it offers a crucial feature for solving optimization issues via metaheuristic algorithms, allowing one agent to locally resolve a suboptimization issue with the aim of 32 pursuing to achieve the global optimum by intersecting with other agents. In this paper, we take advantage of this feature to create a multiagent method for the CSC issue based on ACS, which includes a diffident agent with a specific function called MAACS, describes the suggested algorithm's architecture in this section, and aims for a better MAACS architecture in this design with regard to the number of agents, their function, and the maintenance of a good trade-off between the mechanisms for exploration and exploitation. One SA, two execution agents, and a third agent make up the proposed MAACS, EAs, and one MA, each of which is designed to perform a certain action [35]. Figure depicts the design of the system MAACS, and the sections that follow explain each agent's function.

4.2. Synchronization Agent (SA). This agent, which has many functions, is made up of all the agents in the system that it is acquainted. It is in charge of splitting the population in half and distributing the halves to the EAs. Additionally, when the limit condition is reached, it is in charge of signaling the MA to obtain the best EAs solution. Finally, it is in charge of selecting the best overall solution from the several EAs. This meta-analysis investigated the association between central obesity and LTL in adults. The analysis included five articles and found that individuals with higher telomere length had a significantly lower waist circumference. The association between LTL and waist-to-hip ratio was not significant. All included studies were of moderate or high quality [36].

4.3. Execution Agents (EAs). The optimization process that the ACS algorithm uses is carried out by EAs. Each of the two separate EAs has a unique initialization strategy. While the second agent (EA2) initializes the subpopulation greedily, the first agent (EA1) initializes the subgroup randomly [19]. The ACS algorithm is run when the subpopulation has been initialized, and at the same time, each agent sends the MA their ideal method to date. A single service at a time, each representing a step along the solution path, is built by the EA agents. According to the stepwise transition rule in Equation (4), each ant begins with Task 1 (T_1) and chooses a web service CSj11 from among the candidate services of T_1 to be included in the aspects of the work. Then advances to Task 2 (T_2), chooses a new web service (CSj12), and so forth. The heuristic data represents the sum of the QoS property values N (qosr), where N (qosr) has the exact meaning as in equation, and the primary sources, which symbolize the quantity of pheromone laid by the ants, are both used by the transition rule to progressively create the solutions. This building process goes on until completing Task 2 (T_2) and choosing a web service. The local pheromones update rule is employed by ants to keep the pheromones for discovered treatments up to date. The strongest candidate ant chooses the best solution discovered so far at the conclusion of the solution building, and this best candidate then uses a global update rule to update the pheromone on the best solution discovered so far. As illustrated in figure, both EA agents build a solution during each iteration, save the best one so far, call it gbest, and dispatch the MA entity to the SA entity if the limit condition is reached.

4.4. Monitorization Agent (MA). The service's EAs make up this entity that it is acquainted with. It adjusts the population's distribution based on how well EAs are performing [21]. The population of the SA is split equally in half, and the two halves are then distributed to the EAs. To promote the utilization of the effective EA agent, MA, on the other hand, favors a good EA that discovers a superior solution by dividing the population is divided into two unequal portions. The system execution ends when this limit condition is met, and the SA agent then compares the EAs' responses and displays the result as the system's permanent conclusion.

5. Proposed System

In its simplest form, QoS-aware proposed framework is the issue of choosing the optimal combination from all of the potential services indicated by S $(i = 1, 2, \ldots, n)$. It is anticipated that a process will be executed that includes the specified task of $T = T_1, T_2, \ldots, T_n$ and satisfies user expectations.

```
Output: Pbest
For (i = 1 To n)
Consumer List ⟵
Random Services chosen (Dataset, m)
Data set ⟵ Delete Services (Candidate List)
For (j = 1 To m)
  For (q = 1 To Q)
    Workflow [i, j, k]
      = Return QoS Value
(Consumer List [j, K])
End
End
Graph ⟵ Graph Building (Workflow);
Graph ⟵ Normalization (Graph)
[Pbest. Pbestcost] ⟵ GACS (Graph, Parameters);
// Algorithm (4)
Return (Pbest. Pbestcost)
```

Algorithm 1: Input: Data, n, m, Q, Attributes

There are various methods to evaluate a weighted graph, and ultimately, service composition will be used to determine the shortest route at the lowest cost. The directed acyclic graph $G = (S_i; P_i)$, where S represents for services and P for service attributes, served as the inspiration for this concept. Additionally, S_i includes services that inputs produce the service from the preceding tier, and P_i is made up of service parameters. The usage of vectors to describe services (S) and user requirements is a more well-liked variant of this strategy. SC in input vectors has indeed been developed by selecting a collection of utility vectors that meet the user's performance standards. Given a demand indicated by R, the goal of service synthesis is to identify an operationally valid composite service that satisfies RS by enhancing the following qualitative standards:

$$\mathrm{SC} = \{\mathrm{Rs}, 1, \&, \mathrm{Rs}, n\}. \tag{1}$$

Since positive attributes are different from negative numbers, Equation (1) illustrates how such QOS value is normalized into a number between 0 and 1:

$$\begin{aligned} \mathrm{QoS} &= \frac{Q(S) - Q_{\min}}{Q_{\max} - Q_{\min}}, \quad \text{if } Q \text{ is positive attribute}, \\ \mathrm{QoS} &= \frac{Q_{\max} - Q(S)}{Q_{\max} - Q_{\min}}, \quad \text{if } Q \text{ is negative attribute}, \end{aligned} \tag{2}$$

where proposed framework is determinable. The issue has been modeled using a general technique as multistage graph planning.

The optimization of a QoS-aware service selection is expressed as Equation (3), where n denotes quality attribute scores and j is the number of qualitative factors:

$$\Sigma m W_n = 1 \text{ withstand Optimize QoS } (S) = \Sigma j W_n.\mathrm{Qn}(S). \tag{3}$$

The following are important factors for the proposed method and flowchart, as shown in Figure 2.

5.1. Challenges

5.1.1. Meeting QoS Requirements. Selecting cloud services that can meet diverse QoS requirements, such as low latency and high availability, can be challenging.

5.1.2. Ensuring Consistent Performance. Dealing with performance variability in cloud services, caused by factors like resource sharing and network congestion, is a challenge.

5.2. Techniques

5.2.1. Performance Measurement. Regularly measure and monitor the performance of cloud services using metrics like response time and availability to assess their QoS capabilities.

5.2.2. SLA Evaluation. Evaluate the SLAs offered by cloud providers to ensure they align with the desired QoS requirements.

5.2.3. Multicloud and Hybrid Cloud. Distribute workloads across multiple cloud providers to mitigate risk and optimize QoS by leveraging different provider strengths.

5.3. Decision-Making

5.3.1. Analytic Hierarchy Process (AHP). Use AHP to assign weights to different QoS criteria and compare cloud providers based on these criteria to make informed decisions.

5.3.2. Quality Function Deployment (QFD). Prioritize QoS criteria based on customer requirements to guide cloud service selection.

5.4. Illustration. The cloud service selection framework (CSSF) considers QoS, cost, security, scalability, and provider reputation. It evaluates and ranks cloud carriers based on these criteria to guide decision-making. By addressing challenges, employing techniques such as performance measurement and SLA evaluation, and utilizing decision-making frameworks like AHP or CSSF, smart city stakeholders can optimize cloud service selection based on QoS criteria.

6. Hybrid Metaheuristic Algorithm

In this study (see Figure 2), we present the hybrid method, where we use GA on top of heuristic to dynamically optimize the parameters. The three operators that makeup GA are selection, crossover, and mutation. As a result, the computerized tuning of the homogeneous or individual GA operators (crossover and mutation) adjusts the parameters of the ACS algorithm in the suggested technique. Running time adjustments are made to these two operators' probabilities (C_p for a crossover and M_p for a mutation) based on the three biggest fitness values of the mating parents (P_m), the population's maximum ($P_{\max}$), and its mean (P_{mean}). Algorithm 1 demonstrates the suggested hybrid algorithm, which calls

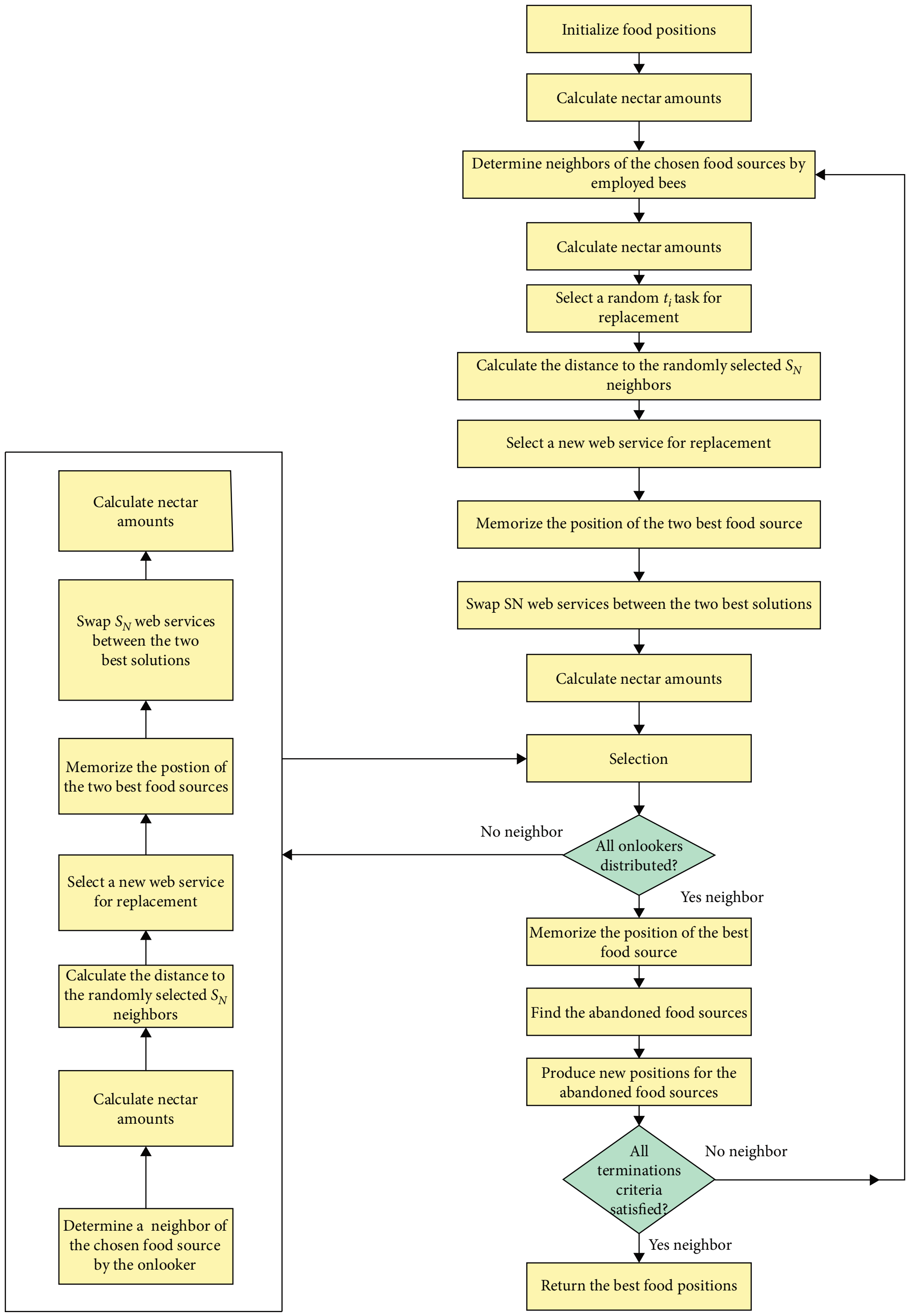

Figure 2: Flowchart of the proposed method.

Algorithm 1 to adjust the parameters. Utilizes a population of individuals I_i (chromosomes) with stable gene lengths in order to optimize the parameters. Every gene offers a potential scope of model parameters given proximity; hence the amount of parameters in a person is equal to the number of genes. The indicated and revised acceptable ranges again for parameters utilized in this work. The GA community is randomly initialized, and each person's gene value is chosen at random from a spectrum of conceivable values. The number of generations affects the stop condition. The Select Parent () function in Algorithm 1 uses stochastic universal sampling to choose a parent, and the Survivor Selection () function uses the best survivor selection approach. A two different weighted sum equation, which is a variant of the four component expression provided as follows, is used to determine each person's fitness value:

$$F = \Sigma_{i=1}^{2} C_i F_i. \tag{4}$$

7. Experimental Parametrs

We generated our dataset using the Quality of Web Service (QWS) framework, incorporating both quantitative and qualitative features. We categorized attributes like security, usability, and flexibility into low, medium, and high classifications, making the dataset user-friendly. After sorting based on QWS, we had 50 services evaluated against six QoS criteria. For real-world use, service providers should adjust QoS features accordingly.

Regarding communication, we considered ranges between the user and the cloud data center (20–500) and between the user and the service set (50–400). We tested using Cloud SIM, a SaaS-level simulator, allowing us to create a virtual environment and manage resource allocation effectively.

Accuracy, F1-Score, memory, and precision are the characteristics that are used for analysis; in the formulations: true positive, false negative, true negative, and false positive stand for true positive, false negative, true negative, and false positive, respectively. The energy- and reliability-aware multiobjective optimization (ERMO) method and the combining of cuckoo particles swarm, artificial bee colony (CPS + ABCO) are two examples of existing approaches that are contrasted with the proposed ant colony optimization + genetic algorithm (ACO + GA).

Accuracy: The proportion of accurate forecast to all the values the classifier predicted. The categorization model is evaluated using this significant metric, which is provided by the following equation:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{5}$$

Precision: The percentage of samples that are accurately described to the total number of samples that have been recognized from Equation (6):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{6}$$

Recall: The ratio of successfully recognized samples to the total of the TP as well as FN samples is displayed:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{7}$$

F1-Score: F1-Score offers a flawless assessment of the classifying model that is calculated from Equation (8):

Table 1: Accuracy analysis.

Number of samples	EMRO	CPS + ABCO	ACO + GA
100	54	60	67
200	61	69	78
300	67	76	84
400	78	84	89
500	86	89	95

Table 2: Precision analysis.

Number of samples	EMRO	CPS + ABCO	ACO + GA
100	47	51	59
200	51	62	68
300	58	68	71
400	65	70	79
500	71	75	89

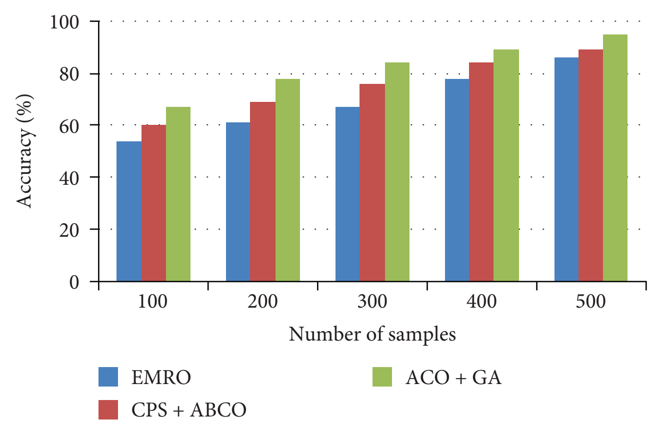

Figure 3: Comparison of accuracy analysis.

$$F1\text{-Score} = 2 \times \frac{(\text{Precision} \times \text{recall})}{(\text{Precision} + \text{recall})}. \tag{8}$$

Table 1 examines the precision of existing and planned methods.

The comparative assessment of accuracy for the suggested and existing methods is shown in Figure 2. The X and Y axes, respectively, show the number of samples and accuracy in percentage. The colors blue, red, and green, respectively, stand for ERMO, CPS + ABCO, and ACO + GA. The accuracy of the suggested method is 83%. Table 2 displays a study of precision using both existing and new methodologies.

Figure 3 depicts the results of a comparison of the recommended methods and the existing ones with regard to their levels of precision. On the X and Y axes, respectively, are the details regarding the number of samples and the precision expressed as a percentage. The acronyms ERMO, CPS + ABCO, and ACO + GA are represented, respectively, by the colors blue, red, and green. The proposed method has a 73% success rate in terms of precision. Table 3 presents an analysis of the memory of both the recommended and established techniques.

Table 3: Recall analysis.

Number of samples	EMRO	CPS + ABCO	ACO + GA
100	38	39	56
200	41	47	65
300	49	51	69
400	56	60	72
500	61	68	78

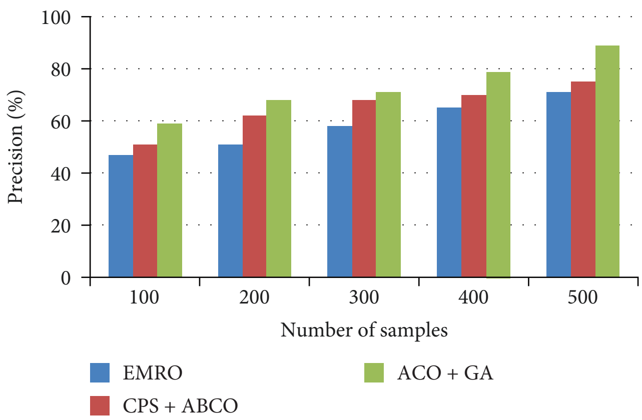

Figure 4: Comparison of precision analysis.

Table 4: F1-Score analysis.

Number of samples	EMRO	CPS + ABCO	ACO + GA
100	54	56	69
200	62	67	75
300	68	71	86
400	72	78	89
500	79	83	92

The comparative assessment of recollection for the suggested and existing methodologies is depicted in Figure 4. The number of samples and recall value were shown on *X* and *Y* axes, accordingly. The colors blue, red, and green, correspondingly, stand for ERMO, CPS + ABCO, and ACO + GA. The proposed method achieves 68% recall when compared with existing methods. Table 4 displays the examination of the F1-Score using both existing and new methodologies.

The F1-Score achieved using the conventional and suggested approaches is contrasted in Figure 5. The number of samples and F1-Score in percent are provided on the *X* and *Y* axes, respectively. The colors blue, red, and green, correspondingly, stand for ERMO, CPS + ABCO, and ACO + GA. The F1-Score achieved by the suggested strategy is 82% (Figure 6). The overall efficiency of the suggested and existing methods is shown in Table 5.

The concept of a "Smart City" centers on the integration of information and communication technologies (ICT) to enhance the quality and efficiency of municipal services, with cloud computing and the IoT as key driving forces. Cloud computing, known for its prowess in managing remote service access, has prompted numerous companies to opt for cloud leasing, relieving local resource constraints. This shift toward

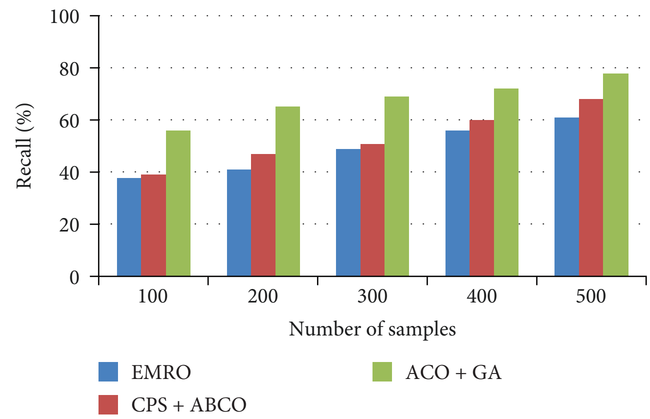

Figure 5: Comparison of recall analysis.

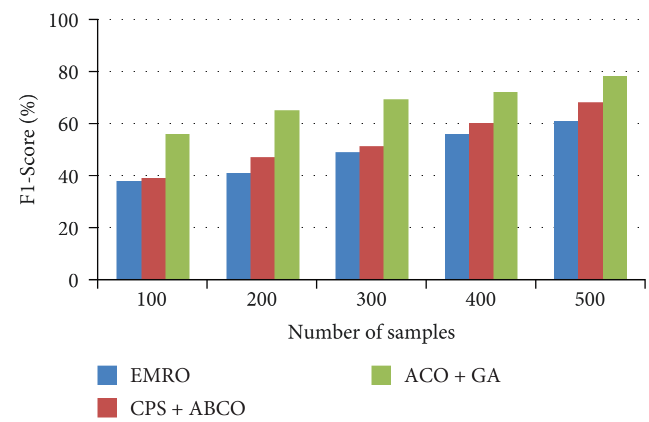

Figure 6: Comparison of F1-Score analysis.

Table 5: Overall comparison analysis.

Parameters	EMRO (%)	CPS + ABCO (%)	ACO + GA (%)
Accuracy	68	75	82
Precision	57	64	72
Recall	49	52	67
F1-Score	66	70	81

cloud-based services underscores the importance of optimizing service selection to align with client needs, guided by QoS criteria. Combining GAs and ACO in cloud computing has yielded remarkable results, achieving an accuracy rate of 82%, surpassing existing methods like ERMO and the hybrid cuckoo particles swarm and artificial bee colony optimization (CPS + ABCO) at 68% and 75%, respectively. These findings underscore the significance of cloud technologies in advancing smart city solutions.

8. Conclusion and Future Direction

This paper introduces a novel approach to combinatorial optimization, combining ACO with GAs to autonomously adjust ACS settings. The aim is to enhance cloud service composition by reducing latency and response times during service discovery. ACO and GA are known for their adaptability and complexity, making them effective in optimizing various problems. However, the paper acknowledges the need for larger scale experiments to validate the approach fully. The primary focus is on solving the combinatorial service composition (CSC) problem in cloud computing, meeting client expectations while considering QoS characteristics. A unique ACO-GA hybrid is introduced, automatically adjusting ACO settings when performance stagnates. The efficiency of this self-adaptation is demonstrated through performance assessments. Future research can further refine these concepts, conduct larger experiments, and explore applications in other combinatorial optimization problems.

Abbreviations

ICT:	Information and communication technologies
IoT:	Internet of things
QoS:	Quality of service
GA:	Genetic algorithms
ACO:	Ant colony optimization
ERMO:	Energy- and reliability-aware multiobjective optimization
CPS:	Cuckoo particles swarm
ABCO:	Artificial bee colony optimization
CSC:	Cloud service composition
PSO:	Particle swarm optimization
ACS:	Ant colony system
HPC:	High-performance computer
PLBW:	Probabilistic linguistic best-worst
PFS:	Probability distribution functions
DR:	Demand response
LPF-TOPSIS:	Linguistic pythagorean fuzzy TOPSIS
IOLs:	Interactional operational laws
PFNs:	Picture fuzzy numbers
FACO:	Flying ant colony optimization
WSC:	Web service composition
SC:	Service composition
MAACS:	Multi-agent ACS
SA:	Synchronization agent
EAs:	Execution agents
MA:	Monitorization agent
SUC:	Stochastic universal sampling
CSSF:	Cloud service selection framework
AHP:	Analytic hierarchy process
QFD:	Quality function deployment
IACO:	Inverted ant colony optimization
SLA:	Service level agreement
LTL:	Leukocyte telomere length.

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

It was performed as a part of the employment of Jayaudhaya J, Department of Electronics and Communication Engineering, RMD Engineering College, Chennai, India; Ramash Kumar K, Department of Electrical and Electronics Engineering, Dr. NGP Institute of Technology, Coimbatore, Tamil Nadu, India; and Jayaraj R, Department of Data Science and Business systems, SRM Institute of Science and Technology, Chennai, India.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-oriented computing: state of the art and research directions," *IEEE Computer Society*, vol. 40, no. 11, pp. 64–71, 2007.

[2] M. Darbandi, "Proposing new intelligence algorithm for suggesting better services to cloud users based on Kalman filtering," *Journal of Computer Sciences and Applications*, vol. 5, no. 1, pp. 11–16, 2017.

[3] A. Heidari and N. Jafari Navimipour, "A new SLA-aware method for discovering the cloud services using an improved nature-inspired optimization algorithm," *PeerJ Computer Science*, vol. 7, Article ID e539, 2021.

[4] S. Schmid, T. Chart, M. Sifalakis, and A. C. Scott, "Flexible, dynamic, and scalable service composition for active routers," in *Active Networks. IWAN 2002*, J. Sterbenz, O. Takada, C. Tschudin, and B. Plattner, Eds., vol. 2546 of *Lecture Notes in Computer Science*, pp. 253–266, Springer, Berlin, 2002.

[5] M. Swan, F. Witte, and R. P. dos Santos, "Quantum infromation science," *IEEE Internet Computing*, vol. 26, no. 1, pp. 7–14, 2022.

[6] M. H. Shirvani, "Web service composition in multi-cloud environment: a bi-objective genetic optimization algorithm," in *2018 Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1–6, IEEE, Thessaloniki, Greece, July 2018.

[7] W. Zhang, C. K. Chang, T. Feng, and H.-Y. Jiang, "QoS-based dynamic web service composition with ant colony optimization," in *2010 IEEE 34th Annual Computer Software and Applications Conference*, pp. 493–502, IEEE, Seoul, Korea (South), July 2010.

[8] Q. Yu, L. Chen, and B. Li, "Ant colony optimization applied to web service compositions in cloud computing," *Computers & Electrical Engineering*, vol. 41, pp. 18–27, 2015.

[9] F. Dahan, K. El Hindi, and A. Ghoneim, "An adapted ant-inspired algorithm for enhancing web service composition," *International Journal on Semantic Web & Information Systems*, vol. 13, no. 4, pp. 181–197, 2017.

[10] H. Alayed, F. Dahan, T. Alfakih, H. Mathkour, and M. Arafah, "Enhancement of ant colony optimization for QoS-aware web service selection," *IEEE Access*, vol. 7, pp. 97041–97051, 2019.

[11] 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, Article ID 4873, 2022.

[12] T. A. Ahanger, F. Dahan, and U. Tariq, "Hybridizing artificial bee colony with bat algorithm for web service composition," *Computer Systems Science and Engineering*, vol. 46, no. 2, pp. 2429–2445, 2023.

[13] F. Dahan, H. Mathkour, and M. Arafah, "Two-step artificial bee colony algorithm enhancement for QoS-aware web service selection problem," *IEEE Access*, vol. 7, pp. 21787–21794, 2019.

[14] M. Lin, C. Huang, Z. Xu, and R. Chen, "Evaluating IoT platforms using integrated probabilistic linguistic MCDM

method," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 11195–11208, 2020.
[15] C. Huang, M. Lin, and Z. Xu, "Pythagorean fuzzy MULTIMOORA method based on distance measure and score function: its application in multicriteria decision making process," *Knowledge and Information Systems*, vol. 62, no. 11, pp. 4373–4406, 2020.
[16] M. Lin, C. Huang, and Z. Xu, "TOPSIS method based on correlation coefficient and entropy measure for linguistic pythagorean fuzzy sets and its application to multiple attribute decision making," *Complexity*, vol. 2019, Article ID 6967390, 16 pages, 2019.
[17] M. Lin, C. Huang, R. Chen, H. Fujita, and X. Wang, "Directional correlation coefficient measures for pythagorean fuzzy sets: their applications to medical diagnosis and cluster analysis," *Complex & Intelligent Systems*, vol. 7, no. 2, pp. 1025–1043, 2021.
[18] M. Lin, X. Li, R. Chen, H. Fujita, and J. Lin, "Picture fuzzy interactional partitioned Heronian mean aggregation operators: an application to MADM process," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 1171–1208, 2022.
[19] Vijayaraj, M. Sumathi, M. Rajkamal, U. Vijayaraj, D. T. Kamaleshwar, and D. Rajalakshmi, "Decision trees to detect malware in a cloud computing environment," in *2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC)*, pp. 299–303, IEEE, Chennai, India, April 2022.
[20] J. Liu, M. Xie, S. Chen, G. Xu, T. Wu, and W. Li, "TS-REPLICA: a novel replica placement algorithm based on the entropy weight TOPSIS method in spark for multimedia data analysis," *Information Sciences*, vol. 626, pp. 133–148, 2023.
[21] Y. Yang, B. Yang, S. Wang, F. Liu, Y. Wang, and X. Shu, "A dynamic ant-colony genetic algorithm for cloud service composition optimization," *The International Journal of Advanced Manufacturing Technology*, vol. 102, no. 1–4, pp. 355–368, 2019.
[22] M. Hosseini Shirvani, "Bi-objective web service composition problem in multi-cloud environment: a bi-objective time-varying particle swarm optimisation algorithm," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 33, no. 2, pp. 179–202, 2021.
[23] K. Abedpour, M. Hosseini Shirvani, and E. Abedpour, "A genetic-based clustering algorithm for efficient resource allocating of IoT applications in layered fog heterogeneous platforms," *Cluster Computing*, 2023.
[24] M. H. Shirvani, N. Amirsoleimani, S. Salimpour, and A. Azab, "Multi-criteria task scheduling in distributed systems based on fuzzy TOPSIS," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1–4, IEEE, Windsor, ON, Canada, April 2017.
[25] M. Xie, J. Liu, S. Chen, G. Xu, and M. Lin, "Primary node election based on probabilistic linguistic term set with confidence interval in the PBFT consensus mechanism for blockchain," *Complex & Intelligent Systems*, vol. 9, pp. 1507–1524, 2023.
[26] M. Abdel-salam, I. M. El-hasnony, M. Elhoseny, and Z. Tarek, "Intelligent PSO approach for QoS-aware web service selection," in *The 3rd International Conference on Distributed Sensing and Intelligent Systems (ICDSIS 2022), Hybrid Conference*, pp. 229–239, IET, Sharjah, United Arab Emirates, October 2022.
[27] H. Xia, Y. Chen, Z. Li, H. Gao, and Y. Chen, "Web service selection algorithm based on particle swarm optimization," in *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pp. 467–472, IEEE, Chengdu, China, December 2009.
[28] K. Kofler, I. U. Haq, and E. Schikuta, "A parallel branch and bound algorithm for workflow QoS optimization," in *2009 International Conference on Parallel Processing*, pp. 478–485, IEEE, Vienna, Austria, September 2009.
[29] M. Darbandi, "Kalman filtering for estimation and prediction servers with lower traffic loads for transferring high-level processes in cloud computing," *HCTL Open International Journal of Technology Innovations and Research (IJTIR)*, vol. 23, no. 1, pp. 1–20, 2017.
[30] M. Darbandi, "Proposing new intelligent system for suggesting better service providers in cloud computing based on Kalman filtering," *HCTL Open International Journal of Technology Innovations and Research (IJTIR)*, vol. 24, no. 1, pp. 1–9, 2017.
[31] S. Vahdat, "The role of IT-based technologies on the management of human resources in the COVID-19 era," *Kybernetes*, vol. 51, no. 6, pp. 2065–2088, 2022.
[32] T. Stützle and H. H. Hoos, "MAX–MIN ant system," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, 2000.
[33] B. B. Gardas, A. Heidari, N. J. Navimipour, and M. Unal, "A fuzzy-based method for objects selection in blockchain-enabled edge-IoT platforms using a hybrid multi-criteria decision-making model," *Applied Sciences*, vol. 12, no. 17, Article ID 8906, 2022.
[34] Q. Wu and Q. Zhu, "Transactional and QoS-aware dynamic service composition based on ant colony optimization," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1112–1119, 2013.
[35] J. Zhou and X. Yao, "Multi-population parallel self-adaptive differential artificial bee colony algorithm with application in large-scale service composition for cloud manufacturing," *Applied Soft Computing*, vol. 56, pp. 379–397, 2017.
[36] F. A. Zadeh, D. O. Bokov, G. Yasin, S. Vahdat, and M. Abbasalizad-Farhangi, "Central obesity accelerates leukocyte telomere length (LTL) shortening in apparently healthy adults: a systematic review and meta-analysis," *Critical Reviews in Food Science and Nutrition*, vol. 63, no. 14, pp. 2119–2128, 2023.