Available online at http://ijim.srbiau.ac.ir/
Int. J. Industrial Mathematics (ISSN 2008-5621)
Vol. 14, No. 4, 2022 Article ID IJIM-1606, 14 pages
DOI: http://dx.doi.org/10.30495/ijim.2022.68023.1606
Research Article
Science and Research Branch (IAU)

A Systematic Review of the Combinatorial Methods of Service Composition in the Cloud Computing Environment

M. B. Karimi *†

Received Date: 2022-06-12 Revised Date: 2022-08-21 Accepted Date: 2022-09-08

Abstract

In the latest decade, the concepts of service delivery and software-as-a-service have evolved into two important evolution paradigms that have affected the information systems area. A paradigm shift in software development has been triggered by this change, in which software is developed through the use of ready-made services in the cloud. Since service composition in the cloud environment must be done on-the-fly, realizing this requires a trade-off between the optimality of the composite service and the time it takes. As QoS-aware service composition has several potential solutions, some of which are usually considered optimal, which should be considered an NP-hard problem. A growing number of services leads to a larger problem search space, which is why in recent years many researchers have looked into methods that use meta-heuristic algorithms to solve the problem of service composition in a cloud environment. Thus, it is crucial that researchers have access to up-to-date and specialized review articles. Based on a systematic review of the research literature, the paper aims to extract important questions that are relevant to meta-heuristic QoS-aware service composition methods. Then, after classifying the studies and studying the proposed methods, goals, and priorities of researchers in articles, useful results and statistics for future research in this field are presented.

Keywords : Systematic literature review; Cloud computing; QoS-aware service composition; Combinatorial composition methods.

1 Introduction

In the last decade, cloud computing has attracted a lot of attention in both academia and industry, which of course will continue. The major developments and evolutions in the realm of IT including pervasive computing, grid computing, service-oriented architecture, and Web 2.0 have led to the advent of cloud computing [1, 2, 3]. The rationale behind cloud computing was to provide all things as a service. The statistic provided by IDC and Fobes indicates that, in 2021, 67% of organizations have used cloud technology and 81% of all enterprises have reported that they have a multi-cloud strategy already laid out or in the works.

Cloud computing provides three different types of services to meet the needs of various IT ar-

*Corresponding author. karimi@iaut.ac.ir, Tel:+98(914)3039915.
†Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran.

eas. The first service type includes SaaS (software as a service) through which a software application is delivered online to a customer as an on-demand service. The second type of services within cloud computing consists of a PaaS (platform as a service) through which computational platform, solution stack, and software development tools are provided as services within a cloud computing environment. Finally, the third type of services encompasses IaaS (infrastructure as a service) through which computational infrastructures such as processer, network bandwidth, and storage tools are virtually provided within the form of a service [4, 5, 6, 7].

In a cloud computing environment, users can voice their needs for a SaaS service on the Internet, which can include both functional and non-functional requirements. In many cases, users may request complex services in a cloud environment that cannot be provided with a single service. Hence, in such cases, the user request should be provided as a composite service including composition of individual services that are either pre-built or provided by a third party [8].

As several IT companies are keen on providing cloud services, the number of these services is remarkably increasing and will duplicate even more rapidly in the near future. Thus, there are many candidates for each abstract service from any workflow based on demand, which are the same in terms of performance, but differ in terms of service quality (QoS) features such as cost, response time, and availability. So, the service composition system must choose services from the highest quality concrete services, according to quality specifications specified by users and documented in the Service Level Agreements (SLA). This allows the user to access optimal services [8, 9, 10].

QoS-aware service composition could be thought of as a multi-constraint optimization problem (MCOP), which can be mapped into the multi-dimensional multiple-choice knapsack problem (MMKP) [4, 11]. Several methods have been proposed to solve the above problem. However, since this problem is known as an NP-Hard problem [11, 12], any solution to this problem must meet the inherent conditions of such an environment including the issue as well. Therefore, the inherent features of the cloud computing environment such as long-term service contracts, support for SLA contracts, economic aspects, environmental dynamics, and the need for process automation should be considered in the composition of services [5, 13].

Due to the novelty and growth of cloud applications, in recent years, many researchers have become interested in researching on service composition problems. Depending on the field of research and the preferences of the researchers, specific aspects of the service composition problem in the cloud computing environment have been targeted in previous researches. In this respect, some researchers have attempted to review previous works.

In [4], through a systematic study of the research literature, researchers have reviewed 34 works done on service composition in cloud computing environment from 2009 to 2013. In the research, the previous works are classified into 5 groups: groups: machine-based approaches, classic and graph-based approaches, framework area research, and combinatorial approaches. Furthermore, papers are evaluated in different aspects such as user needs support, quality attributes support, automation of the composition process, etc. However, the classification presented in this paper, despite the passage of time, still has research value. Despite this, the researchers have not considered some important aspects of the composition of services, such as the distribution degree of composition methods and the economic considerations.

In [14], researchers reviewed 50 published research papers on cloud services composition by 2017 through a systematic study. In the paper, the methods of service composition have been classified into two groups, single-cloud and multi-cloud environments, and multi-objective and single-objective optimization methods. Previous researches has been examined in this study from aspects such as the type of composition method, in addition to characteristics of service quality, which provide useful results for re-

searchers. However, service composition in the cloud environment requires some important factors, including economic aspects, the distribution or centralization of composition methods, the degree of automation of service composition processes, and the dynamic nature of the cloud environment that have not been noticed by researchers.

In previous work in the field of this research [4, 14] appropriate categories for structured study of papers have been done and useful results have been created for researchers interested in the research area. However, some gaps in those studies have arisen over time, necessitating the need for newer studies with new features and goals:

Previous review studies cover the research work done up to 2017, so new researches are necessary to review recent works.

In previous systematic reviews, researchers have attempted to cover all the different categories of service composition methods. Since each category of service composition methods has particular objectives and considerations, and since the number of publications has increased significantly in every field of research, it is imperative to conduct specialized review studies for each category of service composition methods.

Cloud service providers are increasing day by day. Since service composition is an NP-Hard problem, when the number of services increases, the search space of the problem expands and previous methods become less effective. Consequently, there is a growing number of researchers interested in this field who would like to find more efficient solutions to the problem of service composition whit reasonable time complexity.

Due to the above reasons, this paper focuses on conducting a specialized review of meta-heuristic service composition methods. This study reviews 20 recent papers published between 2012 and 2021 including state-of-the-art approaches concerning cloud service composition. Furthermore, in the paper, the real aspects of a cloud system, such as the economic aspects of service delivery, long-term composite services, and support service level agreements, have been considered.

The rest of the paper is organized as follows: In section 2, we formally describe the service composition problem and its requirements in the cloud environment. In section 3, recent meta-heuristic service composition approaches will be investigated. In section 4, a detailed discussion of the studied works will be done. Finally, in section 5, the conclusion of the study and directions for further research are mentioned.

2 Service Composition in Cloud Computing Environment

The issue regarding the service composition process is selecting an appropriate service among many similar services for executing a task and ultimately producing a composite service (figure 1). Since the service composition component is the core of supplying cloud applications, hence, due to its vital significance, there is an essential need for embedding a section in the middleware of cloud computing systems as the service composition unit.

2.1 Service quality model

As shown in equation (2.1), the global quality of a composite service should be measured based on the workflow structure of that composite service. In many previous studies [13, 15], the following four pattern types are considered in the internal structure of composite services which are supported by BPEL (business process execution language) [16]:

- Sequence: sequential execution of services
- AND: parallel execution of services
- Branch: branching or conditional execution of services
- Loop: repetitive execution of services

Table (1) indicates the manner of measuring global values for QoS parameters in a composite service [13].

Table 1: Method of calculating qualitative values.

2*QoS parameters	Composition Patterns			
	Sequence	AND	Branch	Loop
Response time	$\sum_{i=1}^{n} q^1 s_i$	$\max_{i=1}^{n} q^1 s_i$	$\max_{i=1}^{n} q^1 s_i$	$\sum_{i=1}^{n} q^1 s$
Price	$\sum_{i=1}^{n} q^2 s_i$	$\sum_{i=1}^{n} q^2 s_i$	$\max_{i=1}^{n} q^2 s_i$	$\sum_{i=1}^{n} q^2 s$
Availability	$\prod_{i=1}^{n} q^3 s_i$	$\prod_{i=1}^{n} q^3 s_i$	$\max_{i=1}^{n} q^3 s_i$	$\prod_{i=1}^{n} q^3 s$
Successability	$\mathrm{Avg}_{i=1}^{n} q^4 s_i$	$\mathrm{Avg}_{i=1}^{n} q^4 s_i$	$\mathrm{Avg}_{i=1}^{n} q^4 s_i$	$q^4 s$

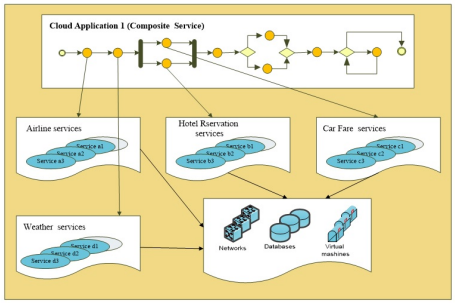

Figure 1: An example of a composite service [5]

2.2 The formal statement of the problem

In composing the services, the issue which should be considered is the selection of individual and atomic services with high-value qualitative parameters. Indeed, this issue should be taken into consideration so that the total parameters should not exceed the constraints specified by the users. The service composition issue can be formalized by equations (2.1) and (2.2) [5].

$$\begin{cases} \max \sum_{l=1}^{p} Q_l(cs) \cdot W_l \\ \text{Subject To } Q_j(CS) \leq QC_j(CS) \\ \qquad \text{for negative attributes} \\ \text{Subject To } Q_j(CS) \geq QC_j(CS) \\ \qquad \text{for positive attributes} \end{cases} \tag{2.1}$$

$$\sum_{l=1}^{p} W_l = 1 \tag{2.2}$$

The workflow of a composite service is defined by the vector $WF = (T_1, T_2, \ldots, T_n)$ and a candidate composite service is denoted by $CS = (S_1, S_2, \ldots, S_n)$ in which T_i refers to a task with an abstract service in the workflow and S_i denotes a selected concrete service for the T_i service in the candidate composite service. The qualitative constraints specified by the customer are represented by the vector $QC = (QC_1, QC_2, \ldots, QC_p)$. In equation (2.1), QC_j (CS) function determines the value of the j_{th} global qualitative parameter specified by user for the CS composite service. Also, Q_j (CS) function determines j_{th} qualitative parameter for the CS composite service and W_l indicates the weight of the L_{th} qualitative parameter. By taking equations (2.1) and (2.2) into account, optimal composite service should be achieved.

2.3 Research scope and questions

In this study, we aim to review the research studies that have addressed the problem of QoS-aware service composition in cloud computing environments using combinatorial methods. To accomplish this goal and identify how researchers solved a problem, studies are covered in which state-of-the-art methods are presented and results have been published in reputable journals. The research is focused on answering the following research questions (RQs):

RQ 1: What are the main goals of the current researches?

RQ 2: What simulation tools, datasets, or benchmarks are used, and what case studies are considered?

RQ 3: What QoS parameters are accounted for?

RQ 4: Which of the challenges of the cloud computing environment has been addressed?

RQ 5: to what extent has the SLA-based service composition been considered in the proposed methods?

RQ 6: Which types of cloud services are involved in the service composition process?

To conduct this study, we searched for articles published in 2013 and 2022 containing the keywords "Cloud", "Service Composition" and "QoS", and 92 articles were found. The next step was to apply the filter to the method presented in the article. The number of articles was reduced to 45 by selecting articles whose method was combinatorial and had significant innovation. Finally, we selected 20 articles for evaluation based on the Q1 and Q2 ranks of the journals and their impact factors exceeding 1.0.

3 Cloud Service Composition Methods based on combinatorial algorithms

Prior to cloud computing, service composition was performed in service-oriented architectures (SOAs) and grid computing environments. In SOA, service composition allows a workflow to be realized with concrete web services. Service composition is also a term for resource scheduling in a grid computing environment. As a result, it can be said that the service composition in cloud computing is a compound inheritance of service composition in SOA and grid computing environments. The process involves discovering and selecting software services that are deployed on infrastructure services. The selection of such services is based upon the non-functional needs of customers and the constraints specified in the service level agreement (SLAs). The composition of services in a cloud computing environment is an optimization problem, and one salient feature of this type of problem is dealing with a very large search space. This problem is inherently an NP-Hard problem. Since the number of services is increasing, it leads to a larger search space, which forces researchers to take synthetic and discovery algorithms seriously to obtain the best solution in a reasonable amount of time. There are many papers in which researchers have tried to use different models of combinatorial and met heuristic algorithms to solve the problem of service composition some of those are reviewed below.

The papers in this review have been classified into two categories, single cloud, and multi-cloud, based on the type of deployment environment. As opposed to the single cloud model, where the services are located in one cloud, in the multi-cloud model integrated concrete services are located on a variety of clouds that can be geographically distributed.

3.1 Single-Cloud methods

In this category of composition methods, researchers have developed solutions to realize the optimal composite services according to the quality attributes determined by the user, by selecting the most appropriate services located on a single cloud.

As a service composition problem in distributed environments, one of the key challenges is to provide high-performance converged network cloud services that include both network services(NaaS) and cloud services(CaaS, IaaS) with guaranteed end-to-end performance. In [17], the service composition problem in the network cloud environment has been modeled as an adaptive multi-constrained optimal path problem, and then an approximation algorithm called SCA is proposed as a solution. In the algorithm, network cloud services are represented as a graph and, from there, an optimal or near-optimal solution between input and output is found in an acceptable time. In [18], a special version of the SLO algorithm is used to solve the problem of QoS-aware service composition in the cloud environment. Named S-SLO, the algorithm composes the differential evolution algorithm and the SLO. Using its learning power to increase convergence, thereby improving the composite service formation process in terms of results accuracy and time efficiency. The algorithm has performed better than other correspondences in terms of the degree of compliance of the composite service with the quality demands of users, but it is not better than the genetic and IDE algorithms in terms of time complexity. Also, it does not have good time complexity due to a lack of local search.

In [5], the genetic algorithm is used to achieve global optimization according to the SLA contract. In addition, service clustering has been

used to reduce problem search space, and association rules for a hybrid service based on their histories were used to increase the efficiency of the service combination. For this purpose, before using the genetic algorithm, two data mining algorithms K-means were used for clustering and Apriori for rule mining as local selection. In paper [19], it has discussed that as the scope and scale of cloud services increase, traditional combinatorial algorithms such as PSO and genetic algorithms become less effective. Therefore, in this study, the knowledge-based differential evolution algorithm has been used to solve the service composition optimization problem. The algorithm accelerates convergence velocity by importing structure-based knowledge including internal structural patterns of composite services such as AND, OR, and Sequence relations. In paper [20], researchers have addressed two important aspects of cloud service composition, which include not only covering all QoS parameters but also taking into account the connectivity constraints of cloud services. The paper describes a new hybrid method called Optimal Fitness Aware Cloud Service Composition (OFACSC) that uses modified invasive weed optimization (MIWO) to increase convergence rate and decrease computational complexity. In the paper, the authors propose a novel approach to assessing cloud service fitness and composition fitness by considering all possible QoS parameters, a prime QoS parameter satisfying connectivity constraints, and balancing QoS parameters using skewness-based approaches.

In [21], the authors presented an optimal combination method that combines optimal fit with a kind of genetic algorithm based on adaptive genotype evolution. In this approach, multiple QoS parameters can be supported and the QoS parameters and connectivity constraints (the connection between any two constraints that depend on each other) are balanced within the service composition. A novel method is presented in the paper for assessing cloud service fitness and composition fitness using Discrete Uniform Rank Distribution (DURD) and Discrete Uniform Service Rank Distribution (DUSRD). Furthermore, an adaptive genotype evolution-based genetic algorithm (AGEGA) has been used as part of an optimal fitness-aware cloud service composition (OFASC) to increase convergence rates and compute complexity. The basis of the paper [22] is the fact that if the population's diversity is low, it leads to early convergence and the possibility of reaching a global optimization is lost, and if the diversity is high, it leads to a slower convergence. The paper presents a new algorithm that uses the Eagle strategy to balance these two realities. This algorithm allows the optimization process of the algorithm to be optimized using a method of exploration and optimal exploitation. According to the results, the performance of this algorithm is better than that of classical combinatorial algorithms such as genetic and PSO. In paper [23], a hybrid meta-heuristic algorithm is used to solve the QoS-aware service composition problem. By using the Ant Colony algorithm (ACO), the paper proposes a method for finding the optimal or near-optimal composite service. Earlier studies have shown that ant colony algorithms with self-adaptive parameters tuning perform better than static parameters tuning, and in this research, genetic algorithms have been used to automatically adjust ACO parameters. The main contribution of the research is to help prevent stagnation in the ACO algorithm while helping to improve the ACO's overall performance, which may be affected by the value of ACO parameters. However, parameter tuning time is an overhead cost.

Paper [23] has proposed an agent-based ant colony optimization (ACO) algorithm to solve the cloud service composition problem. Since the complexity of the CSC problem is NP-Hard, it is almost impossible to find a solution in a reasonable amount of time, so multi-agent technology has been used to increase the efficiency of the algorithm. By decomposing the search process and incorporating population re-disruption in real-time, the proposed multi-agent ACS (MAACS) method can reduce problem complexity. Research reported in the paper [24] aims to perform QoS-aware service composition concerning automated composition processes, dynamic adaptivity, increasing flexibility and scalability, and en-

hancing resource utilization. The proposed model uses the SMO algorithm to perform classification at two different levels to achieve the above objectives: classification of services based on QoS attributes and grouping of services based on functionalities. This algorithm then identifies a local leader for each class of services, which is a load balancer virtual machine that will monitor service compositions within the class. After that, a dependency graph is generated using the Multistage Forward Search (MFS) algorithm, which is used to find the best service composition in the graph. According to the proposed model, the SLA contract guarantee is also performed using the SMO algorithm. In other words, whenever the workload of integrated services on a virtual machine increases, the virtual machine is divided into two or more virtual machines to be able to adapt to the contracted quality.

3.2 Multi-cloud methods

In these types of service composition methods, infrastructure quality attributes such as communication costs and network latency, etc., are considered in the service composition, and researchers try to provide methods that reduce such costs as much as possible. Challenging the problem, In [25], a special form of chaos optimization algorithm called FC-PACO-RM is used to solve the service composition problem. Considering the large and irregular selecting space of the Multi-Objective Optimal Service Combination (MOSCOS), a chaotic adaptive optimization is presented by combining two methods of matching turbulent sequences and selecting a roulette wheel to make high-quality decisions on sequences. To make the search more efficient, coarse-grained parallelism with a complete connection topology on the MPI communication interface is used to reduce the execution time of the service composition in the algorithm. Also, to reduce communication overhead costs, a fully connected topology and a new migration method called reflective migration have been used. In [26], the genetic algorithm is used to compose services in a geo-distributed multi-cloud environment in which user demands and constraints are submitted and specified in SLA contracts. In this work, Skyline and roulette wheel operators have been used to improve the quality of the composite service and to avoid biasing some quality attributes by others. In this method, the service composition time is significantly increased by enlarging the workflow and increasing the concrete services of each of the abstract services. The results show that this method works better in finding the optimal solution as well as the cost of execution time. Also, the overall optimality of the composite service and the coverage of the desired quality attributes of the users determined in SLAs are better covered, but as the number of services increases, the efficiency of the proposed method in terms of combination time decreases. Due to the spread of services on multiple clouds, they significantly increase the communication costs and financial costs of composite services. To meet this challenge, a trade-off must be made between the number of clouds and the number of examined services.

An algorithm was proposed in [27] for services composition that could compose services from different clouds. This algorithm selects the cloud with the maximum number of services, which increases the possibility of composing with the least time overhead. The paper proposes a new combinatorial optimization algorithm for cloud service composition (COM2) that can compose services with small numbers of examined services and combined clouds. The proposed algorithm ensures that the cloud with the maximum number of services is selected before selecting clouds with fewer services, which increases the possibility of meeting service requests with minimal overhead costs. However, in the method, the details of the simulation and the QoS attributes have not been specified. Providing composite services in the cloud environment is challenging due to the growing number of cloud service providers, which increases the search space. The cloud computing environment becomes less scalable as it becomes harder to find an optimal composite service within a reasonable time complexity. In Article [28], the CSSICA method has presented as a method for solving the service composition

problem in large-scale cloud computing environments. In this method, an Imperialist Competition is used along with a search space classification. The method consists of dividing the search space (cloud service providers) into three categories based on the total time they have been operational and assigning a probability to each of those categories based on the average service time values of all providers assigned to each class. With this ranking, the Imperialist Competition algorithm is now able to choose the most appropriate service provider to realize a special request. A solution to the challenge of finding the optimal cloud combination is also presented in [29]. The paper presents a hybrid algorithm (ACO-WSC algorithm) that uses greedy and ACO algorithms for selecting a feasible cloud combination using a minimum number of clouds. With greedy algorithms (Greedy-WSC), the most suitable cloud is selected over and over again until it covers all the requirements. In this algorithm, the ant colony iteratively finds the best solution. This method achieves a superior trade-off between quality and time and is a practical solution for providing web services in multi-cloud environments. But in the paper, other requirements of the cloud environment such as adaptability and dynamic environment have not been considered.

In [30], a flexible and QoS-aware service composition method called EDMOEA is presented based on multi-objective optimization. Because many QoS constraints are often not precisely defined by users, it is possible neither to find the best composite service nor to reject many composite services due to deviations from the constraints. Consequently, in the method proposed in this paper, researchers have considered selection based on variance and deviation from user constraints as an independent goal that provides information to the user with more flexibility. Where a completely desirable service isn't available, the user may choose from among the combined services with a lesser variance than his QoS constraints. In [31], a new method is proposed using a combination of two strategies of local selection and global optimization. To reduce the time complexity, a local search is performed using fruit fly optimization, and for global optimization, a genetic algorithm is used along with a new roulette wheel method to avoid premature convergence. In the paper, the problem of service composition with multi-constraint optimization is reduced to a single-objective optimization problem using a simple weighting method. In this work, interdependence and correlation between cloud services have been considered. The results of evaluations show that this method has a good performance in terms of service execution time due to the reduction of search space using fruit fly optimization. However, in this method, some traits can be biased by some other qualitative traits. Also, the effect of the distribution of cloud services in the experiments has been eliminated. In [32], the authors have addressed the issue of distributed cloud data centers. In the paper, a method is presented called CSA-WSC, which uses a cuckoo search algorithm to compose the cloud services. The algorithm presented in the paper has considered the distributed network environment. Also, the algorithm has considered not only the QoS attributes of the SaaS services but also the network QoS attributes of IaaS services. The cuckoo algorithm used has improved the convergence speed, but as the number of services increases, the time cost of the algorithm increases with a greater slope.

An algorithm called LS-NSGA-II-DE has been proposed in [33] to optimize service composition in a multi-cloud environment for improved service diversity and convergence. In other words, it is for dealing with the interference of some desirable qualitative features within the constraints imposed at one time, to increase complexity. A major achievement of this research is its replacement of the mutation and crossover operators of the NSGA-II algorithm with an adaptive differential evolution algorithm that combines two mutation strategies and changes evolutionary parameters continuously in the process of creating new generations. In Paper [34], the authors employ several promising strategies of differential evolution to solve multi-objective problems and overcome the weaknesses of the ABC algorithm. In this way, they have improved both the behav-

ior of bees in search of food as well as the operator control parameters for optimum offspring reproduction. In addition, they used a quality-indicator-based fitness assignment approach and an external archive to better identify search areas and guide search operators. In paper [35], a hybrid Shuffled Frog Leaping Algorithm and Genetic Algorithm (SFGA) is used to optimize mobile cloud service composition. The regular SFLA has been designed to optimize continuous problems. Therefore, it is not possible to apply a service composition problem as a discrete problem. To accomplish this, the encoding of individuals is established in the applied SFLA to provide proper and valid solutions. As well, GA algorithms are embedded in SFLA to evolve the individuals.

4 Discussion

Based on the results of reviewing the papers, we attempt to answer the research questions in this section.

4.1 Research Objectives (RQ1)

The first research question was to determine what objectives the authors generally pursued in the research described in the reviewed papers. In general, the following goals have been considered:

- GOAL1: Increasing the performance of service composition: In the combinatorial methods, the main goal is to reduce the time of the composition process, along with increasing the fitness of the composite service.
- GOAL2: QoS-aware service composition: In most articles reviewed, service composition has been performed with regard to quality requirements and constraints announced by users.
- GOAL3: Cloud environment features: Most of the reviewed articles have covered some features and parameters of the cloud computing environment to provide real and practical methods, as shown in Table 3.

Figure 2, shows how much attention has been given to each of the objectives.

4.2 Datasets and tools (RQ2)

In order to evaluate the methods presented in the papers, dataset QWS was used in 55% of the articles [36], dataset OWL-S [37, 38] and WS-Dream [39] were used in 10% of the articles and the data were generated at random(RG) in 35% of the articles. Additionally, in the past, some researchers used Cloudsim simulation tools, but in recent years many researchers have implemented simulation environments using Python, Java, and .NET languages, as well as MATLAB.

4.3 QoS attributes

Based on heuristic algorithms, methods for composing services are QoS-aware. Quality attributes identified in most articles are a subset of the QWS database's 9 quality attributes. In other works in the field of service composition for cloud manufacturing or resource management, energy consumption has also been considered. Figure (3) illustrates the attention paid to each qualitative attribute.

4.4 Pay attention to the challenges of the cloud environment

Combinatorial service composition papers seek to select the optimal or near-optimal composite services in a reasonable amount of time. However, these works also take into consideration some of the major challenges of cloud computing. A few of these challenges have been addressed in the papers:

- Scalability (CH1): Both the number of cloud service providers and users are growing. As the scale increases, previous solutions become inappropriate for large-scale environments, so researchers have begun looking at alternatives.
- Flexibility (CH2): Due to the uncertainty of the cloud environment and the impossibility of meeting users' quality demands, some researchers have considered the flexibility of decision-making in the choice of services.

- Automation of the composition process (CH3): Due to the increasing number of users and composite service requests for cloud systems, current methods of handling service requests are not responsive and there is a need to manage the service composition process on the fly, so some researchers are paying attention to it.

- Adaptability (CH4): Due to the nature of the cloud, the quality of the infrastructure network can change and affect the quality of the composite services provided, so this challenge has been addressed in some articles.

Figure (4), shows how much attention has been given to each of the challenges, and Table (3) shows coverage details.

4.5 Support for SLA contracts (RQ5)

Since cloud services, like other IT services, are provided based on the quality requirements and constraints outlined in the service level agreements, some researchers have addressed how to incorporate service level agreements into the method in 15% of papers.

4.6 Type of services (RQ6)

In some articles, the service composition in the SaaS layer is only considered. However, due to the nature of the cloud, in 15% of papers, the deployment of composite services on IaaS services has also been considered.

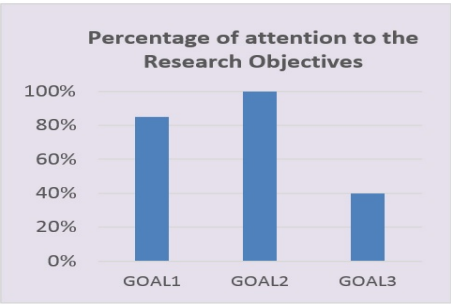

Figure 2: Percentage of attention to the Research Objectives

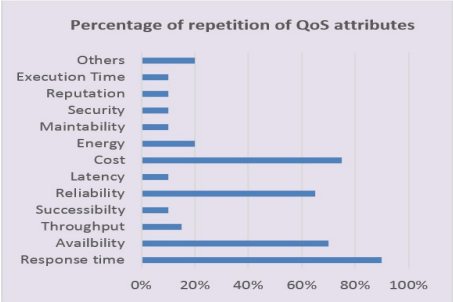

Figure 3: Percentage of repetition of QoS attributes

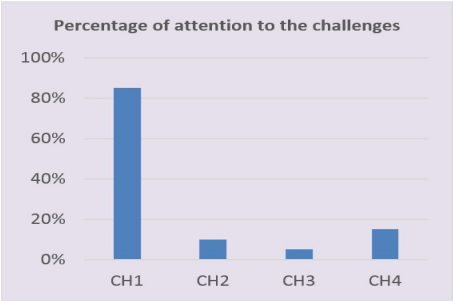

Figure 4: Percentage of attention to the challenges

5 Conclusion

In the paper, a systematic literature review was performed on papers published in the field of QoS-aware service composition using combinatorial methods. To accomplish this, 92 articles published from 2013 to 2022 were filtered, and 20 papers with high journal impact factors and important innovations were selected and reviewed in two categories of multiple clouds and single clouds. Based on a comprehensive and detailed review of selected articles, the main focus of each article, as well as the advantages and disadvantages of the proposed methods, were determined.

Based on analyzing the selected articles, 100% aimed to reduce composition time and improve fitness, 45% aimed to provide services based on the quality requests of users, and 40% claimed to be targeting cloud environment characteristics. Further, researchers have largely focused on scalability, flexibility, automation, and adaptability in their papers to address the challenges of a cloud environment. In 85% of the papers, they have addressed scalability, in 10% they have ad-

Table 2: A summary of the findings of the article review.

Authors	Publisher	Journal Rank	year	Tools and Dataset	QoS parameters	
Ghobaei-Arani, Rahmanian et al [32]	Springer	IF=3.643 Rank: Q2	2018	Tools: - Dataset: RG	Time Cost Availability Reliability	Multi-Cloud
Tao, F. et al [25]	IEEE	IF=10.15 Rank: Q1	2013	Tools: - Dataset: RG	Time Cost Energy Reliability Maintainability Unexpected Conditions Trust Function Similarity	Multi-Cloud
Kurdi, Al-Anazi et al [27]	Elsevier	IF=3.818 Rank: Q1	2015	Tools:- Dataset: OWL-S XPlan Package	Unknown	Multi-Cloud
Wang, D., et al [26]	Elsevier	IF=3.818 Rank: Q1	2015	Tools: - Dataset: RG	Price Response Time Availability Reputation	Multi-Cloud
Chen, F., et al [30]	Elsevier	IF= 5.431 Rank: Q1	2016	Tools: Matlab7.1 Dataset: RG	Cost Latency Time Reliability Availability	Multi-Cloud
Yu, Chen et al [29]	Elsevier	IF=3.818 Rank: Q1	2015	Tools: VC++ 6.0 Dataset: QWS	Unknown	Multi-Cloud
Seghir, F. and A. Khababa [31]	Springer	IF=6.85 Rank: Q1	2018	Tools: MATLAB Dataset: RG	Price Response Time Availability Reliability	Multi-Cloud
Karimi, Isazadeh et al [5]	Springer	IF=2.6 Rank: Q2	2017	Tools: VC#.Net Dataset: QWS	Price Response Time Availability Successability	Single- Cloud
Liu, Chu et al [18]	Elsevier	IF= 6.795 Rank: Q1	2016	Tools: C++ Dataset: RG	Response Time Availability Reliability Successability	Single-Cloud
Jatoth, Gangadharan et al [21]	Elsevier	IF= 7.187 Rank: Q1	2019	Tools: Java Dataset: QWS	Availability Security Accessibility Cost Integrity Throughput Response Time Reliability	Single- Cloud
Liu, Gu et al [33]	KSII	IF= 1.06 Rank: Q3	2018	Tools: - Dataset: QWS	Price Response Time Reliability	Multi-Cloud
Zhou, J., et al [34]	Elsevier	IF=5.524 Rank: Q1	2018	Tools: - Dataset: RG	Price Response Time Availability Reliability Reputation Energy Maintainability Eco-impact	Multi-Cloud
Huang, Duan et al [17]	IEEE	IF=4.714 Rank: Q1	2015	Tools: - Dataset: QWS	Response Time Cost Delay	Single- Cloud
Jula, Othman et al [28]	Elsevier	IF= 6.954 Rank: Q1	2015	Tools: VC#.Net Dataset: WSDream	Unknown	Multi-Cloud
Qi, Xu et al [19]	Springer	IF= 4.81 Rank: Q1	2018	Tools: - Dataset: RG	Response time Availability Reliability	Single- Cloud
Jatoth, Gangadharan et al [20]	Elsevier	IF= 7.177 Rank: Q1	2019	Tools: - Dataset: QWS	Availability Cost Execution Time Response Time Reliability	Single- Cloud
Gavvala, Jatoth et al [22]	Elsevier	IF= 7.187 Rank: Q1	2019	Tools: MATLAB Dataset: QWS	Availability Cost Execution time Response Time Throughput Reliability	Single- Cloud
Dahan, F. [23]	IEEE	IF= 3.367 Rank: Q1	2021	Tools: - Dataset: QWS	Price Response Time Availability Reliability	Single-Cloud
Ibrahim, Rashid et al [35]	Elsevier	IF= 3.734 Rank: Q1	2020	Tools: VC#.Net Dataset: QWS	Cost Response Time Energy	Multi-Cloud
Tarawneh, H., et al [24]	MDPI	IF= 3.114 Rank: Q2	2022	Tools: VB.Net Dataset: QWS	Response Time Throughput Availability	Single-Cloud

dressed flexibility, and in 15% of them, they have addressed the adaptability of composition methods.

Additionally, most research has focused on the composition of services for large-scale environments and the adaptability of methods. Furthermore, recent research has paid more attention to multi-cloud environments. This research was limited to combinatorial methods of service composition.

Software engineering for the cloud environment will be a dominant paradigm in software development in the future due to the development of cloud computing and software as a service. This

Table 3: Covering goals and challenges by papers.

2*Papers	Goals			Challenges			
	GOLA01	GOLA02	GOLA03	CHO1	CHO2	CHO3	CHO4
[32]	√	√	√	√			
[25]	√	√	√	√			√
[27]	√	√	√	√			
[26]	√	√	√	√			
[30]		√	√		√		
[29]	√	√	√	√	√		
[31]	√	√		√			
[5]	√	√	√	√			√
[18]	√	√		√			
[21]	√	√		√			
[33]		√	√				
[34]	√	√		√			
[17]	√	√	√	√			
[28]	√	√		√			
[19]	√	√		√			
[20]	√	√		√			
[22]	√	√		√			
[23]	√	√		√			
[35]		√					
[24]	√	√	√	√		√	√

type of software development is based on the on-the-fly composition of services. Due to the increasing number of cloud services, more efficient methods will be needed. In order to automate the composition process, semantic web methods will be considered. Artificial intelligence will be applied to the composition of services due to the exponential growth in the scale of services and users and the inability to accurately determine quality requirements and composite service adaptability requirements. Additionally, mobile agents will be considered because of the distributed nature of the service environment. Obviously, each of the above fields will have its own challenges and requirements. Consequently, it is necessary to conduct more specialized reviews based on the requirements of the research fields.

References

[1] S. P. Roger, R. M. Bruce, Software engineering: a practitioner's approach, *McGraw-Hill Education*, (2015).

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, *Future Generation computer systems* 25 (2009) 599-616.

[3] R. Buyya, J. Broberg, A. M. Goscinski, Cloud computing: Principles and paradigms, *John Wiley & Sons*, (2010).

[4] A. Jula, E. Sundararajan, Z. Othman, Cloud computing service composition: A systematic literature review, *Expert systems with applications* 41 (2014) 3809-3824.

[5] M. B. Karimi, A. Isazadeh, A. M. Rahmani, QoS-aware service composition in cloud computing using data mining techniques and genetic algorithm, *The Journal of Supercomputing* 73 (2017) 1387-1415.

[6] F. Liu, NIST cloud computing reference architecture, *NIST special publication* 500 (2011) 1-28.

[7] S. K. Garg, S. Versteeg, R. Buyya, A framework for ranking of cloud computing ser-

vices, *Future Generation Computer Systems* 29 (2013) 1012-1023.

[8] K. P. Joshi, Y. Yesha, T. Finin, Automating cloud services life cycle through semantic technologies, *IEEE Transactions on Services Computing* 7 (2012) 109-122.

[9] S. A. Baset, Cloud SLAs: present and future, *ACM SIGOPS Operating Systems Review* 46 (2012) 57-66.

[10] M. Teixeira, R. Ribeiro, C. Oliveira, R. Massa, A quality-driven approach for resources planning in service-oriented architectures, *Expert Systems with Applications* 42 (2015) 5366-5379.

[11] A. S. da Silva, H. Ma, M. Zhang, Genetic programming for QoS-aware web service composition and selection, *Soft Computing* 20 (2016) 3851-3867.

[12] F. Tao, D. Zhao, Y. Hu, Z. Zhou, Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system, *IEEE Transactions on industrial informatics* 4 (2008) 315-327.

[13] M. Alrifai, T. Risse, W. Nejdl, A hybrid approach for efficient Web service composition with end-to-end QoS constraints, *ACM Transactions on the Web (TWEB)* 6 (2012) 1-31.

[14] V. Hayyolalam, A. A. P. Kazem, A systematic literature review on QoS-aware service composition and selection in cloud environment, *Journal of Network and Computer Applications* 110 (2018) 52-74.

[15] Z. Ye, X. Zhou, A. Bouguettaya, Genetic algorithm based QoS-aware service compositions in cloud computing, *International Conference on Database Systems for Advanced Applications* 13 (2011) 321-334.

[16] F. Moscato, N. Mazzocca, V. Vittorini, G. D. Lorenzo, P. Mosca, M. Magaldi, Workflow pattern analysis in web services orchestration: The BPEL4WS example, *International Conference on High Performance Computing and Communications* 15 (2005) 395-400.

[17] J. Huang, Q. Duan, S. Guo, Y. Yan, S. Yu, Converged network-cloud service composition with end-to-end performance guarantee, *IEEE Transactions on Cloud Computing* 6 (2015) 545-557.

[18] Z. Z. Liu, D. H. Chu, C. Song, X. Xue, B. Y. Lu, Social learning optimization (SLO) algorithm paradigm and its application in QoS-aware cloud service composition, *Information Sciences* 326 (2016) 315-333.

[19] J. Qi, B. Xu, Y. Xue, K. Wang, Y. Sun, Knowledge based differential evolution for cloud computing service composition, *Journal of Ambient Intelligence and Humanized Computing* 9 (2018) 565-574.

[20] C. Jatoth, G. Gangadharan, U. Fiore, Optimal fitness aware cloud service composition using modified invasive weed optimization, *Swarm and evolutionary computation* 44 (2019) 1073-1091.

[21] C. Jatoth, G. Gangadharan, R. Buyya, Optimal fitness aware cloud service composition using an adaptive genotypes evolution based genetic algorithm, *Future Generation Computer Systems* 94 (2019) 185-198.

[22] S. K. Gavvala, C. Jatoth, G. Gangadharan, R. Buyya, QoS-aware cloud service composition using eagle strategy, *Future Generation Computer Systems* 90 (2019) 273-290.

[23] F. Dahan, An effective multi-agent ant colony optimization algorithm for QoS-aware cloud service composition, *IEEE Access* 9 (2021) 17196-17207.

[24] H. Tarawneh, I. Alhadid, S. Khwaldeh, S. Afaneh, An Intelligent Cloud Service Composition Optimization Using Spider Monkey and Multistage Forward Search Algorithms, *Symmetry* 14 (2022) 82-98.

[25] F. Tao, Y. LaiLi, L. Xu, L. Zhang, FC-PACO-RM: a parallel method for service composition optimal-selection in cloud manufacturing system, *IEEE Transactions on Industrial Informatics* 9 (2012) 2023-2033.

[26] D. Wang, Y. Yang, Z. Mi, A genetic-based approach to web service composition in geo-distributed cloud environment, *Computers & Electrical Engineering* 43 (2015) 129-141.

[27] H. Kurdi, A. Al-Anazi, C. Campbell, A. Al Faries, A combinatorial optimization algorithm for multiple cloud service composition, *Computers & Electrical Engineering* 42 (2015) 107-113.

[28] A. Jula, Z. Othman, E. Sundararajan, Imperialist competitive algorithm with PROCLUS classifier for service time optimization in cloud computing service composition, *Expert Systems with applications* 42 (2015) 135-145.

[29] Q. Yu, L. Chen, B. Li, Ant colony optimization applied to web service compositions in cloud computing, *Computers & Electrical Engineering* 41 (2015) 18-27.

[30] F. Chen, R. Dou, M. Li, H. Wu, A flexible QoS-aware Web service composition method by multi-objective optimization in cloud manufacturing, *Computers & Industrial Engineering* 99 (2016) 423-431.

[31] F. Seghir, A. Khababa, A hybrid approach using genetic and fruit fly optimization algorithms for QoS-aware cloud service composition, *Journal of Intelligent Manufacturing* 29 (2018) 1773-1792.

[32] M. Ghobaei-Arani, A. A. Rahmanian, M. S. Aslanpour, S. E. Dashti, CSA-WSC: cuckoo search algorithm for web service composition in cloud environments, *Soft Computing* 22 (2018) 8353-8378.

[33] L. Liu, S. Gu, D. Fu, M. Zhang, R. Buyya, A new multi-objective evolutionary algorithm for inter-cloud service composition, *KSII Transactions on Internet and Information Systems (TIIS)* 12 (2018) 1-20.

[34] J. Zhou, X. Yao, Y. Lin, F. T. Chan, Y. Li, An adaptive multi-population differential artificial bee colony algorithm for many-objective service composition in cloud manufacturing, *Information Sciences* 456 (2018) 50-82.

[35] G. J. Ibrahim, T. A. Rashid, M. O. Akinsolu, An energy efficient service composition mechanism using a hybrid meta-heuristic algorithm in a mobile cloud environment, *Journal of parallel and distributed computing* 143 (2020) 77-87.

[36] E. Al-Masri, Q. H. Mahmoud, Investigating web services on the world wide web, *Proceedings of the 17th international conference on World Wide Web* 20 (2008) 795-804.

[37] OWLS-Xplan Service Composition Planner.

[38] A. G. M Klusch, OWLS-Xplan Service Composition Planner, (2006).

[39] Z. Zheng, Y. Zhang, M. R. Lyu, Distributed qos evaluation for real-world web services, *2010 IEEE International Conference on Web Services* 21 (2010) 83-90.

Mohammad Bagher Karimi is Ph.D of Software Engineering. He is member of Faculty in Tabriz Branch, Islamic Azad University, Tabriz, Iran. He is Interested in cloud computing, software Engineering, service oriented architecture, blockchain and distributed systems.