A Novel Hybrid Bio-Inspiration Technique for Service Composition

MAHMOUD A. SALAM[1], M.M.EL-GAYAR[1]
[1] Information Technology Department, Faculty of Computers and Information, Mansoura University, Egypt
Emails: {< Mahmoud20; mostafa_elgayar> @Mans.edu.eg}

Abstract

There are many challenges facing the service composition process. These challenges include, how to integrate services to satisfy global user requirements, missing or changeable values of QoS, and how to reduce the large solution space of candidate services. In this paper, we proposed a framework to address these challenges. The proposed framework consists of three phases. The Normalizer phase gives a certain range for all QoS attributes and historical user orders. During the Clustering phase, the search space is reduced. Finally, the composition process is done, and a list of candidate composite services is generated through the Service composer phase. We present the hybrid bio-inspiration technique to implement the proposed framework and prove its applicability. In addition, we introduced the MR-FPSO algorithm to implement this phase by merging PSO and FOA optimization algorithms over the MapReduce framework to handle the large scale of data in the cloud environment. Our technique is compared to different techniques, including MR-GA, MR-IDPSO, and MRPSO. The simulation results proved that our technique outperforms the other techniques.

Keywords: MapReduce, FOA, PSO, large scale QoS, parallel composition, Fuzzy clustering

I. Introduction

Service-oriented architecture (SOA) helps in dividing the whole system into a set of loosely coupled services that perform a specific function. Many firms and organizations use online services to outsource their primary operations to the internet. The kernel of service-oriented computing in the environment of developing software is to choose a web service that fulfils different needs of users. In many previous studies, satisfying user requirements within an acceptable interval of time is considered a challenging research area within many previous studies. A web service is defined by the things it does and the things it does not do. Services share the same functional attributes while differing in nonfunctional ones. The main challenge is how to select a service from many candidate services. These services are equivalent in functionality and belong to

DOI: 10.5281/zenodo.5515275

the same service class in order to obtain the best combination of many services to fulfill the end-to-end global user's constraints as possible. While the selection process aims to select the most appropriate service among a large number of candidate services, which have the same functionality but with different QoS. Finally, the selected services are composed in order to find the optimal composite service. A huge number of available services are in fulfilning for this procedure, thus we need to narrow the search space to discover the dependable services with shorter execution time that will fit the composite service. Current techniques for cloud composability solve the issues they are meant to (such as bad programming) but do not address the core challenge of scaling code well. For this reason, this paper addresses the problem of large scale data in the cloud environment. A framework is proposed to address the whole process of service composition. In addition, a hybrid Bio-Inspiration technique has been introduced to implement the proposed framework to prove its applicability through the technique of implementing the introduced phases in the framework. Many existing algorithms for service selection can be used [2-3]. The proposed technique uses both particle swarm optimization and fruit fly optimization based on the skyline results in the service composer phase. The proposed technique provides a composition process at an acceptable time cost with more accuracy, while satisfying the needs of users as much as possible.

The contributions of this paper can be briefed as:

a. Propose a framework to handle the whole process of web service composition.

b. Use statistical concepts in order to remove unstable services to enhance the selection process of services.

c. Introduce the MR-FPSO algorithm which is a hybrid inspiration composition technique. The MR-FPSO algorithm merges between PSO and FOA optimization algorithms and integrated with the MapReduce framework to handle the large scale of data in the cloud environment.

d. Experiments are conducted to reflect the impact of the proposed method over large scale number of concrete services. The proposed technique is compared to MR-GA [4], MR-IDPSO [5] and MRPSO [6] algorithms.

2. Related work:

There have been multiple studies carried out in the context of reliable service, service selection, and service composition. We examine similar studies in this section. The authors of the two referenced papers [8], [9] considered a multi-phase procedure and many criterion decision making in their effort to uncover the value of QoS. The researchers also utilised the weight assignments for each QoS. Although the time cost was reasonable, the global end-to-end limitations set by the user were not satisfied. Jin et al. [10] discovered that service composition was achievable via genetic algorithm and correlation-aware model on a cloud platform. We had previously put out a method called MRNSGA-II [7] to help with handling the composition problem. This approach used the NSGA-II optimization, running over a MapReduce structure with the help of a clustering methodology. Because it can't manage large-scale of candidate services efficiently, beside it is confined to search for similar services for similar previous user requests, MRNSGA-II drawbacks include that it has difficulty finding comparable services for a similar search request. Seghir and Khababa devised a way to combine two evolutionary algorithms, and they designed it to work with the different classes of problem defined by Khababa et al. [11]. The genetic algorithm and the fruit fly optimization algorithm are paired in order to exploit the strengths of each of the two algorithms to search for the composite service. Chen et al. [12] described a method for service composition that uses partial selection to choose services

according to QoS awareness. They confirmed its success in theory, and by doing several simulations. The Hadoop environment is commonly used for cloud computing and services composition. Zhang and his colleagues investigated the possibility of using the Hadoop infrastructure with the PSO algorithm [5]. Hossain et al. [13] attempted to use a combination of two algorithms: parallel particle swarm optimization with K-means clustering to manage the enormous volume of available data on the Hadoop distributed platform. However, their approach was only successful in retrieving the data. The two-step method for service composition presented by Yong et al [14], first lowering the number of candidate services with k-means clustering, then selecting the best appropriate service with mixed integer programming.

3. The Proposed framework

The proposed framework consists of three phases as shown in figure 3. First, the normalization phase is responsible for giving a certain range for all QoS attributes and historical user orders. Normalization helps in performing further phases in an efficient way. Second, the clustering phase is responsible for clustering data such as QoS-based services, service users, and historical user orders. The Clustering Composer phase consists mainly of a sub-phase. This subphase is service clustering. The goal of the user clustering sub-phase is to reduce the solution space of candidate services in order to reduce execution time. Finally, the service composer phase is responsible for the composition process. It generates a list of composite services. Several algorithms can be implemented in the Service Composer phase. MRNSGA-II and MR-FPSO algorithms are practical examples of implementing this phase. MRNSGA-II was used to implement this phase in our previous work [7]. The Service composer phase is also responsible for the execution and monitoring of the service composition. In case of the failure of any composite service, it selects an appropriate alternative composite service from the generated list.

3.1. Hybrid Bio-Inspiration technique:

The proposed Hybrid Inspiration technique as shown in figure 3 proves the applicability of the proposed framework. It implements and following phases from the proposed framework: Normalizer phase, Cluster phase, and Service composer phase. Cluster composer phase represents a fuzzy clustering phase that uses the user clustering sub-phase to cluster the service users into set of clusters to fuzzy cluster.

Finally, the Service composer phase, which is responsible for service composition and evaluation, concludes. The Service composer phase is implemented through the proposed MR-FPSO algorithm to perform the composition process. Next, more details about each phase will be discussed.

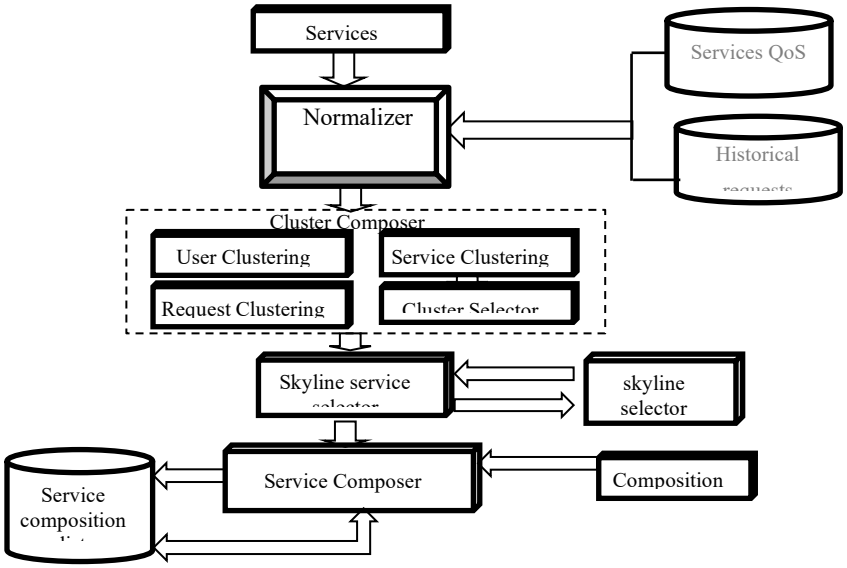

Figure 1 Hierarchal design of the proposed framework

1. Normalizer phase:

There is bias since both QoS qualities of services and historical requests have distinct scales, which leads to low values being treated differently than high values. Equations (1) and (2) are used to normalise all attributes, by setting each value between 0 and 1 [19]. Normalization phase uses the available maximum and minimum values for QoS parameters.

$$q_{nrm} = \begin{cases} \frac{q_{max} - q}{q_{max} - q_{min}} & q_{max} - q_{min} \neq 0 \\ 1 & q_{max} - q_{min} = 0 \end{cases} \quad (1)$$

$$q_{nrm} = \begin{cases} \frac{q - q_{min}}{q_{max} - q_{min}} & q_{max} - q_{min} \neq 0 \\ 1 & q_{max} - q_{min} = 0 \end{cases} \quad (2)$$

The minimum and maximum values for the specified QoS attribute will be referred to as q_{min} and q_{max}, respectively. The normalised value is known as q_{nrm}.
The following actions are done once the candidates' QoS levels have been normalized:

2. Clustering phase:

To implement the clustering phase, the user clustering sub-phase is used to cluster the service users into a set of clusters. This gives several advantages. First, it reduces the search space.

DOI: 10.5281/zenodo.5515275

2.1 FUZZY C-MEANS CLUSTERING (FCM):

The fuzzy c-means clustering algorithm implements the user clustering sub-phase, which is a component of the clustering phase in the framework (FCM). FCM is used to address the overlap of values because QoS attributes can't be sharply grouped in any case because they are not good enough. The fuzzy clustering algorithm, as suggested in this research, makes an excellent clustering approach for the MR-FPSO.

The input and output of the fuzzy C-means algorithm can be established by making the assumption that there are T customers for service $\{us_1, us_2, us_3, \dots, us_T\}$ where us_r refers to the r^{th} user's service. Also, there are W many other services that are in the same category $\{se_1, se_2, \dots, se_W\}$ that se_n refer to the n^{th} web service. Every time a web service is used, a vector of QoS attributes describing that web service (i.e., us_j referring to q_{ji}) exists as a rating. The overall vector defining the QoS of all services triggered by $H_j = \{h_{j1},\ h_{j2}, \dots, h_{jW}\}$.

To determine which service user belongs to which cluster, provide the parameters of the clustering procedure and the service users. A user is described as a vector with QoS properties, which show the history of services the user has used. The fuzzy C-Means algorithm's result is a set of clusters' centres and each object's membership to those clusters. The Euclidean distance is the measuring stick in the classic fuzzy C-means method. It gives things enough spacing, it matches together comparable objects, and it helps to centre the whole group. The fuzzy C-Means technique uses Pearson Correlation Coefficient (PCC) to increase the accuracy of computations since it replaces Euclidean distance with PCC. Since PCC informs you how strongly service users are linked to one another, it has superior accuracy. Quality values like those discovered by the user in the real world are best handled by PCC.

Let's suppose there are two service users, one called "a" and the other "b." The PCC equation (below) is used to calculate the correlation and similarity of their numbers:

$$\text{similarity}'(a,b) = \frac{\sum_{i\epsilon I(a)\cap I(b)}(q_{ai}-\overline{q_a}).(q_{bi}-\overline{q_b})}{\sqrt{\sum_{i\epsilon I(a)}(q_{ai}-\overline{q_a})^2}\cdot\sqrt{\sum_{i\epsilon I(b)}(q_{bi}-\overline{q_b})^2}} \quad (3)$$

Where $\overline{q_a}$ and $\overline{q_b}$ represents the average QoS values that were seen by both users A and B. $I(A)$ and $I(B)$ consider the services that User A and B called on.

It is evident that minimal distances exist when there is a high level of similarity between users. In FCM, the following objective function was created by exploiting this information:

$$J = \sum_{j=1}^{N}\sum_{i=1}^{c} u_{ij}^{m}\left[\frac{1}{similarity(Q_j,C_i)}\right]^2 \quad (4)$$

FCM iterates until a termination condition is reached where:

$$|\text{new degree of membership} - \text{current degree of membership}| \leq \epsilon$$

3. Service Composer phase:

The composition process is performed in such a way to obtain accurate near-optimal solutions. To implement the service composer phase, a new hybrid bio-inspiration method, MR-FPSO, is proposed. The proposed algorithm merges between the PSO and FOA optimization algorithms and is integrated with the MapReduce framework as follows.

The standard PSO algorithm's downfall is that it is trapped in local optima, which causes poor results far from the ideal ones. To improve on the features of PSO, a new method with altered PSO is provided to solve the issue. On order to manage the enormous data that is in the cloud, the novel approach MR-FPSO has been implemented using MapReduce.

The velocity mechanism has been modified, and that together with the fruit fly method and freshly updated PSO algorithm have been employed to enhance the existing PSO algorithm. This makes the particle, solution, trend toward the new solution site. To keep up with new scents, the position of the smell operator can be updated as needed. This information can be obtained quickly and has simplicity on its side.

3.1 COMPOSITION MECHANISM FOR SOLUTION ENCODING

We've decided to go with an approach that uses an integer array for its task representation. A group of concrete services serves as a single proxy for each abstract service. The candidate service identifier is symbolised by each array item. Figure 4 is given below.

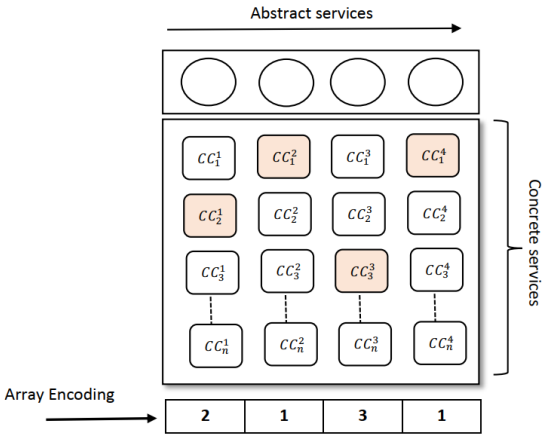

Figure 4. Service encoding scheme

3.3 PROPOSED SERVICE COMPOSITION TECHNIQUE (MR-FPSO):

The suggested technique's major purpose is to efficiently handle massive data sets of services and construct a composite service with higher accuracy and speed. The technique being presented introduces parallelism into the mix. The MapReduce framework relies on two primary components: Map and Reduce (). The method includes two major stages: preprocessing and C-Cloud. The details of each stage are outlined in the subsections below.

A. PREPROCESSING STEP

The first stage is to generate a population of N particles, each of which represents a potential solution. To begin, each particle in the population gets a random location, with the variable x_i set to be that initial value. This can be accomplished by using the fitness function, which will assign a number to each particle to assess them. The optimal position is produced and saved to begin with. The <key, value> pairs are generated for each particle in the original population and they will be used in the MapReduce implementation. Information

for every particle is described by the key, which is an integer, and the value, which is an integer: $v_i, x_i, Pbest, Gbest\ and\ Fitness$ as shown in figure 5 where $x_i = \{x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,n}\}$,$v_i$ $\{v_{i,1}, v_{i,2}, v_{i,3}, \dots, v_{i,n}\}$, $F(*)$ is in reference to the evaluation of the fitness function The particles are then kept in files, where they serve as the foundation for MapReduce.

Key	Value
ID	$id; x_i; v_i; Pbest;\ Gbest; F(x_i);\ F(Pbest); F(Gbest)$

Figure 5. Particle Key-Value structure

B. C-CLOUD STEP:

Disjoint subsets of data blocks are created according to the partitioning criterion. Data blocks are sent to separate Map nodes so they can be processed in simultaneously. The aim of dividing data into blocks is to provide the partitions with a good fit in memory. Other advantages include being able to spread out the workload among multiple servers to improve performance. Each division is given its own individual part of a query to process. Finally, the overall results are put together. Many partitioning methods utilise data partitioning (such as angular partitioning [22], grid-portioning [21], and…). Angular partitioning is far more successful than other methods.

An iterative step that will be done as a job is performed in the MapReduce. A fresh swarm of updated particles is conducted at the end of each MapReduce iteration. The prior solution swarm has been replaced by the newer swarm. Each swarm's changes depend on the functions used in Map and Reduce.

The PSO's original group is separated into subpopulations that go to the Mapper for each input. Each MapReduce step relies on the results of the previous one.

Map() function:

- The particle's position is set to that of the better of the two in this case, which is what one should do unless the particle has the superior position by a wide margin. Comparing the fitness value of x i with the others can do this. In setBest, all the particles with better fitness than x_i are selected. A random number of T is used to select a guiding particle from the ***setBest*** to be used in the process of choosing the next update for x_i:

$$v_i^d(t+1) = wv_i^d(t) + c_1 r_1 \left(pbest_i^d - x_i^d(t)\right) + c_2 r_2 (x_j^d(t) - x_i^d(t)) \tag{11}$$

$$w = w_{max} - \frac{w_{max} - w_{min}}{itmax} iter \tag{12}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{13}$$

- Next scenario for when the particle stands for the ideal location. The population will remain more diverse if there is a blend of random and selected particles. The following equations demonstrate how convex combination and opposition learning can be carried out:

$$\widehat{x_i} = (1-\lambda)x_i(t) + \lambda x_k(t) \tag{14}$$

$$x_i^{new} = x_{min} + x_{max} - \widehat{x_i} \tag{15}$$

The position of the particle is updated in both cases. Then, FOA's distance and scent operators are used to assist little movements to find a better answer faster. The resulting position is updated using the following equations:

$$Dist_i(t+1) = 2/x_i^d(t+1) \tag{16}$$

$$smell_i(t+1) = 2/Dist_i(t+1) \tag{17}$$

To see if **Pbest** should be replaced, the comparison is made with an updated value of x_i. A representative placement is used to reflect the global best placement for a subpopulation in each Map job.

Reduce() function:

At the end of every iteration, a global best particle is created that represents the optimal solution. To make this determination, all of the local best particles are merged to find the best global particle. Once the Reduce task is finished, the old set of particles is expelled, and it is utilised as an input to the next MapReduce cycle.

5. Simulations and results

5.1 Requirements configuration

To test our hypothesis, we use Hadoop 2.8.5 on a Hadoop cluster to apply the concept of parallelism. There are five nodes; one of them is the master node and the rest handle data. The MapReduce method is applied with the help of four functions on a single map. The latest Intel hardware (2.50 GHz Core i7 with 8GB RAM) is employed for running Windows 8 on our previously used hardware. Simulations were done to assess and interpret the proposed technique. In comparing approaches such as MR-GA and MRPSO for handling huge data scale, which is often referred to as having large data scale, their efficiency and ease of use are contrasted. WSDream data is used for MR-FPSO simulations. It consists of around 339 customers, each using two different service levels: reaction time and throughput. There are 4 different services to handle the massive data. There are around 10000 concrete services for each one. For MR-GA, both mutation and crossover are set to 0.68 and 0.203, respectively.

5.2 Results summary

As seen in Figure 6, the number of iterations correlates to an appropriate increase in average fitness. This shows the success of MR-FPSO in locating the best possible solutions. Because they use skyline operator and methods like entropy and correlation fluctuation to drastically decrease the solution space, the services will be more trustworthy. Moreover, due to the enhancements in how MapReduce jobs handle particles, Figure 7 reveals that the appropriate MR-FPSO time will suffice for multiple iterations.

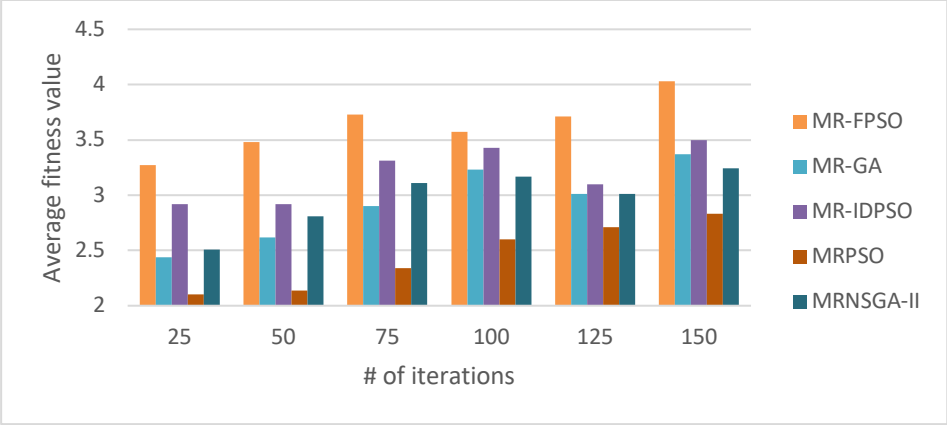

Figure 6. Feasability evaluation with number of iterations

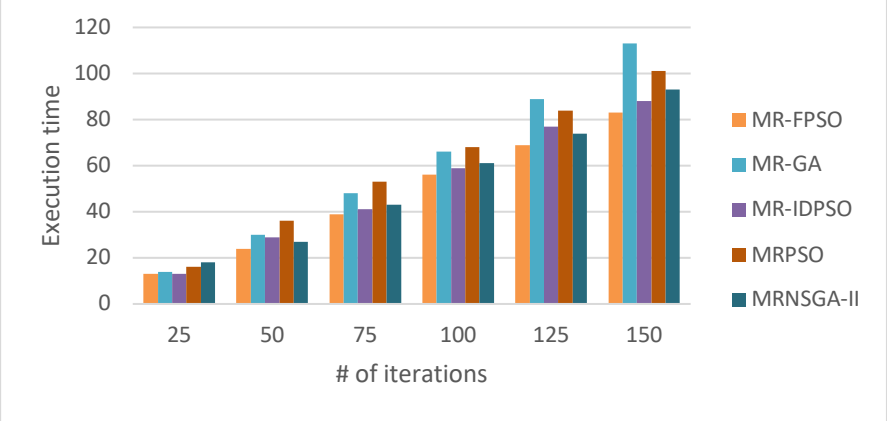

Figure 7. Execution time evaluation with number of iterations

6. Conclusion and future work

With the rapid proliferation of web services over the cloud environment, the problems of the service composition process need to be faced. In this paper, a framework to address service composition is introduced. A Hybrid Bio-Inspiration technique is introduced to implement the proposed framework. Moreover, the MR-FPSO algorithm was introduced to improve the global end-to-end solution. It efficiently provides parallel service composition and employs subjective trimming operators to provide diversity for future generations of solutions. Simulations showed that the proposed technique efficiently addresses the challenges of service composition over the cloud computing environment rather than compared techniques. In the future, further enhancements to the efficiency of the proposed technique will be conducted. More efficient optimization for service composition will be introduced considering the dynamic environment of service composition in large scale data.

References:

[1] X. Guo, S. Chen, Y. Zhang, and W. Li, "Service Composition Optimization Method Based on Parallel Particle Swarm Algorithm on Spark," Secur. Commun. Networks, vol. 2017, no. 1, pp. 1–8, 2017.

[2] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient QoS-aware service composition," in Proceedings of the 18th international conference on World wide web - WWW '09, 2009, p. 881.

[3] G. Canfora, M. Di Penta, R. Esposito, and M. L. Villani, "An approach for QoS-aware service composition based on genetic algorithms," in Proceedings of the 2005 conference on Genetic and evolutionary computation - GECCO '05, 2005, p. 1069.

[4] N. E. A. Khalid, A. F. A. Fadzil, and M. Manaf, "Adapting MapReduce framework for genetic algorithm with large population," in 2013 IEEE Conference on Systems, Process & Control (ICSPC), 2013, pp. 36–41.

[5] Y. Zhang, Z. Jing, and Y. Zhang, "MR-IDPSO: A novel algorithm for large-scale dynamic service composition," Tsinghua Sci. Technol., vol. 20, no. 6, pp. 602–612, 2015.

[6] A. W. McNabb, C. K. Monson, and K. D. Seppi, "MRPSO," in Proceedings of the 9th annual conference on Genetic and evolutionary computation - GECCO '07, 2007, p. 177.

[7] M. A. Salam, W. M. Bahgat, E. El-daydamony, and A. Atwan, "A Novel Framework for Web Service Composition," pp. 1–11, 2019.

[8] Z. U. Rehman, O. K. Hussain, and F. K. Hussain, "Parallel cloud service selection and ranking based on QoS history," Int. J. Parallel Program., vol. 42, no. 5, pp. 820–852, 2014.

[9] M. S. Hossain, M. M. Hassan, M. Al Qurishi, and A. Alghamdi, "Resource Allocation for Service Composition in Cloud-based Video Surveillance Platform," in 2012 IEEE International Conference on Multimedia and Expo Workshops, 2012, pp. 408–412.

[10] H. Jin, X. Yao, and Y. Chen, "Correlation-aware QoS modeling and manufacturing cloud service composition," J. Intell. Manuf., vol. 28, no. 8, pp. 1947–1960, 2017.

[11] F. Seghir and A. Khababa, "A hybrid approach using genetic and fruit fly optimization algorithms for QoS-aware cloud service composition," J. Intell. Manuf., vol. 29, no. 8, pp. 1773–1792, Dec. 2018.

[12] Y. Chen, J. Huang, C. Lin, and J. Hu, "A Partial Selection Methodology for Efficient QoS-Aware Service Composition," IEEE Trans. Serv. Comput., vol. 8, no. 3, pp. 384–397, May 2015.

[13] M. S. Hossain, M. Moniruzzaman, G. Muhammad, A. Ghoneim, and A. Alamri, "Big Data-Driven Service Composition Using Parallel Clustered Particle Swarm Optimization in Mobile Environment," IEEE Trans. Serv. Comput., vol. 9, no. 5, pp. 806–817, Sep. 2016.

[14] Z. Yong, L. Wei, L. Junzhou, and Z. Xiao, "A novel two-phase approach for QoS-aware service composition based on history records," Proc. - 2012 5th IEEE Int. Conf. Serv. Comput. Appl. SOCA 2012, 2012.

[15] Q. Wu, Q. Zhu, and M. Zhou, "A correlation-driven optimal service selection approach for virtual enterprise establishment," J. Intell. Manuf., vol. 25, no. 6, pp. 1441–1453, Dec. 2014.

[16] H. Jin, X. Yao, and Y. Chen, "Correlation-aware QoS modeling and manufacturing cloud service composition," J. Intell. Manuf., vol. 28, no. 8, pp. 1947–1960, Dec. 2017.

[17] Y. Ma, S. Wang, P. C. K. Hung, C. H. Hsu, Q. Sun, and F. Yang, "A highly accurate prediction algorithm for unknown web service QoS values," IEEE Trans. Serv. Comput., vol. 9, no. 4, pp. 511–523, 2016.

[18] H. Al-Helal and R. Gamble, "Introducing Replaceability into Web Service Composition," IEEE Trans. Serv. Comput., vol. 7, no. 2, pp. 198–209, Apr. 2014.

[19] Z. Ye, X. Zhou, and A. Bouguettaya, "Genetic Algorithm Based QoS-Aware Service Compositions in Cloud Computing," Springer, Berlin, Heidelberg, 2011, pp. 321–334.

[20] S. Borzsonyil and K. Stocker, "The Skyline Operator *," pp. 421–430, 2001.

[21] K. Deng, X. Zhou, and H. Tao, "Multi-source Skyline Query Processing in Road Networks," in 2007 IEEE 23rd International Conference on Data Engineering, 2007, pp. 796–805.

[22] A. Vlachou, C. Doulkeridis, and Y. Kotidis, "Angle-based space partitioning for efficient parallel skyline computation," in Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD '08, 2008, p. 227.

[23] C. Wang and W. Song, "A modified particle swarm optimization algorithm based on velocity updating mechanism," Ain Shams Eng. J., Mar. 2019.