Hindawi
Complexity
Volume 2021, Article ID 6642514, 12 pages
https://doi.org/10.1155/2021/6642514

Research Article

Service Discovery and Selection Based on Dynamic QoS in the Internet of Things

Naiheng Zhang

School of Computing, University of Newcastle Upon Tyne, Newcastle Upon Tyne, NE1 4FY, UK

Correspondence should be addressed to Naiheng Zhang; fcczhaoz@zzu.edu.cn

Received 5 December 2020; Revised 22 January 2021; Accepted 1 February 2021; Published 19 February 2021

Academic Editor: Wei Wang

Web services are self-describing and self-contained modular applications based on the network. With the deepening of web service applications, service consumers have gradually increased their requirements for service functions and service quality. Aiming at how to select the optimal plan from a large number of execution plans with the same function and different QoS characteristics, this paper proposes a web service selection algorithm that supports QoS global optimization and dynamic replanning. The algorithm uses position matrix coding to represent all execution paths and replanning information of the service combination. By calculating the Hamming distance of the service quality between individuals, the quality of the service portfolio is improved. By specifying the total user time limit and implementing a good solution retention strategy, the problem of the impact of algorithm running time on service quality is solved. The experimental results show that the method proposed in this paper is effectively integrated into the development trend of QoS and close to the requester's needs and can better meet user needs. This algorithm improves the user's satisfaction with the returned service to a certain extent and improves the efficiency of service invocation.

1. Introduction

With the increasingly widespread and gradual deepening of the Internet of Things applications, massive and heterogeneous Internet of Things devices are constantly pouring into people's lives [1, 2]. In order to enable IoT devices to communicate and collaborate with each other, through a service-oriented architecture, the functions of IoT devices can be encapsulated into loosely coupled IoT services, which become a resource that is shared locally or opened widely [3]. Its main purpose is to realize the interaction between physical entities through devices to build more complex and intelligent IoT applications. At the same time, due to the inherent dynamic characteristics of the Internet of Things environment, the event-driven service-oriented architecture has been used to process events in the Internet of Things environment with its asynchronous response characteristics and has been recognized by academia and industry at home and abroad [4, 5].

By preestablishing a thesaurus, the searched content is limited to the words in the thesaurus, and the searched content is also limited to the fixed thesaurus. Both service request and registration use a fixed vocabulary. There is no semantic ambiguity in the use of words in the vocabulary. Each word has its own meaning. In this case, the service search will not appear. Due to the search and registration of the words used to search for the problem of irrelevant results, the expressions of the service requester and the registrant are consistent, but this method largely limits the flexibility of service search and service registration. Through the use of the Semantic Web, the scope of keywords is expanded, thereby solving the problem of ambiguity in the word keyword and the containment relationship between keywords in the service search process. Although this keyword-based service matching method is fast, simple, and easy to implement, it often has such a problem: different words may describe the same semantic concept, and the keyword-based service discovery method only works on the same words, and words with different morphology are discarded, which results in a low recall rate; or the same word may have different semantic concepts, and the

keyword-based service discovery method cannot distinguish and identify it. As a result, false query results are mixed into the real web service discovery results, resulting in low precision. Web services are spread on the Internet, and users must find the services they need through some means. This process is similar to searching web pages on the Internet through search engines. Web service discovery is to provide users with such technical means, so that users can easily find services related to their needs [6]. Web service discovery is a key step in service-oriented computing. If users cannot find the services they need, then nothing can be said. Web service discovery is based on web service matching. The matching process refers to comparing the user's needs with the description of the web service, and the discovery process selects the service most similar to the user's needs from the matching results. When matching each service, service discovery and matching can be roughly divided into grammar-based service discovery, semantic-based service discovery, and QoS-aware service discovery according to different processing information [7–9]. Among them, the first two technologies are mainly service-oriented functions, and the third technology, service-oriented QoS, is generally based on the first two technologies [10, 11]. Related scholars have given the definition of grammar-based service discovery and pointed out that when matching grammar-based services, it is necessary to match their grammatical descriptions according to the input and output described by the service [12]. However, the accuracy of this technology is relatively low, and a large number of web services that are not related to user needs will be returned [13]. Researchers have proposed a method for service discovery based on concept map matching [14]. Because the concept map matching method is mainly user information retrieval and natural language processing, it is not accurate in function-oriented service discovery. QoS-aware service discovery is based on the technology of function-oriented service discovery, which is generally interleaved with functional discovery or as its successor stage [15]. After finding a set of services through functional service discovery, QoS-aware service discovery acts as a QoS filter to eliminate those unavailable or poor QoS services [16]. This process is also called QoS ranking. Relevant scholars have proposed a proxy-based QoS filtering scheme for the general UDDI registry that does not provide QoS support when registering services [17]. They configure a service arbitration for each service on the service provider side and count the QoS indicators of the service. Each user group is assigned an agent. The user makes a service request to the agent [18]. The agent caches the request history and recently returned services. The agent either returns the appropriate service according to the set QoS sorting rules or returns a list of services for users to choose according to their QoS preferences. Scholars pointed out that, in the process of service discovery and matching, due to the different perspectives of users and service providers, they have completely different understandings of many QoS indicators of services, such as reliability, availability, and security [19–22]. These indicators are vague, and it is difficult for users and service providers to fully unify their understanding.

Among the increasing number of web services, there will inevitably be a large number of services with the same function and different QoS. These web services can combine thousands of combinations with the same function and different QoS features. How to learn from these solutions choosing the best solution is a key problem that must be solved by web service composition. Specifically, the technical contributions of this article can be summarized as follows.

First, this paper proposes a web service selection algorithm that supports QoS global optimization and dynamic replanning. The algorithm uses the location matrix coding method to represent all the combined paths and replanning information of the service combination; it improves the quality of the service combination by calculating the Hamming distance of the service quality between individuals; it solves the problem by specifying the total time limit of the user and implementing a good solution retention strategy. The impact of algorithm's running time on service quality is discussed.

Second, this article proposes a QoS data measurement method based on user feedback and provides a method for measuring the reputation of service providers. By quantifying and measuring the QoS information fed back by end users after using web services, the dynamic characteristics of web services and the QoS requirements of different users provide a QoS data measurement method based on user feedback to ensure the credibility of QoS data.

Third, we conducted simulation experiments. The experimental results prove the feasibility and effectiveness of the algorithm. Compared with similar results, the algorithm provides a more complete and effective service composition QoS solution in a dynamic environment.

The rest of this article is organized as follows. Section 2 discusses the key technologies of QoS-aware web service intelligent acquisition. Section 3 designs a web service selection algorithm that supports QoS global optimization and dynamic replanning. Section 4 carries out experimental verification and analysis. Section 5 summarizes the full text.

2. QoS-Aware Key Technologies for Intelligent Acquisition of Web Services

2.1. IoT QoS Architecture. Figure 1 shows the scalable service QoS management architecture. The management layer provides related application support for the aggregation layer and application layer. The aggregation layer mainly implements the classification and clustering of the management layer QoS data to provide the application layer roughness. With the support of granular QoS information, the application layer performs QoS-aware service selection and other related applications based on QoS information of different granularities provided by the aggregation layer and management layer. In the SQMF system, the management layer includes QoS model customization, QoS information collection, QoS information storage, and QoS information analysis and calculation; the aggregation layer mainly completes the hierarchical mapping of QoS classification information.

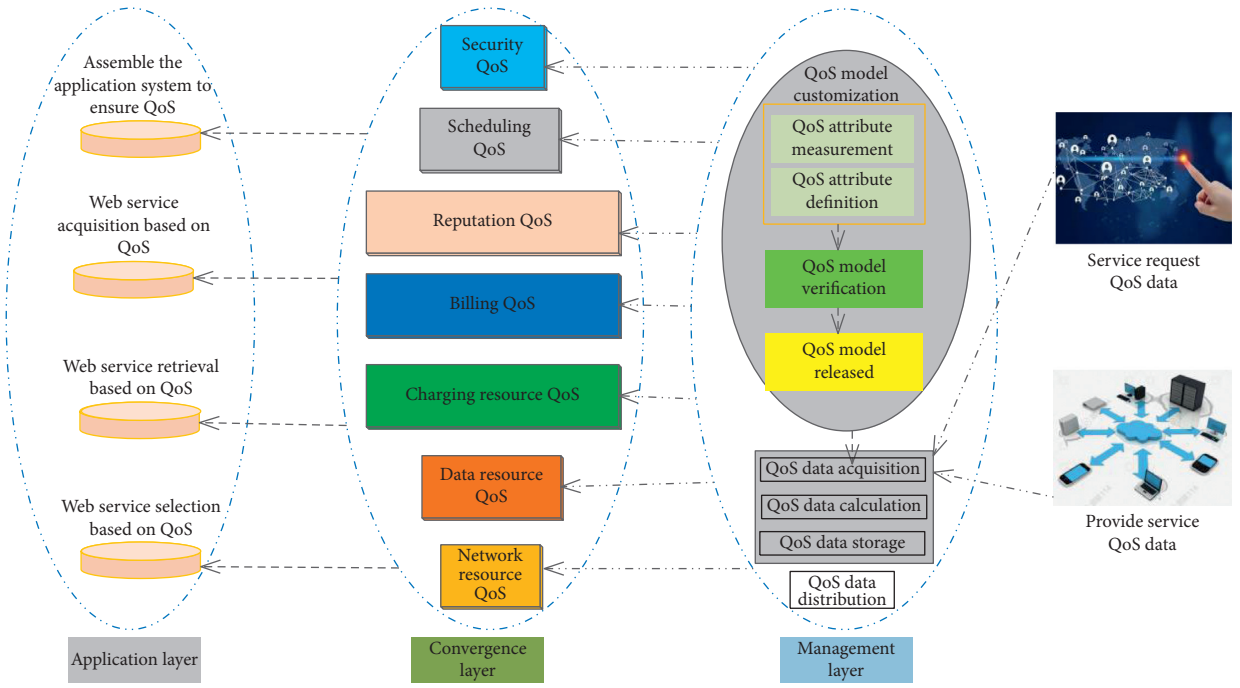

Figure 1: Web service QoS management architecture.

In the network environment, QoS parameters are described in different forms, and the QoS parameters of different layers have different characteristics. In order to summarize the QoS parameters with general characteristics, it is necessary to classify the network QoS parameters. The application layer QoS is often coarse-grained. From this layer, the user proposes an abstract description of the QoS of the entire application. It is difficult to describe the QoS parameters in detail. The QoS description of the underlying service resources is more detailed, and its granularity is finer, specific to the QoS attributes of physical service resources or logical service resources. In order to not only make the underlying service resource structure opaque to users but also enable service selection based on QoS constraints to be dynamically implemented, the QoS parameters are classified at the aggregation layer to provide flexibility and scalability of service QoS management.

2.2. QoS Perception Model in the Internet of Things. The QoS description of web services is not defined in the current UDDI specification, so UDDI itself does not support service discovery based on QoS constraints. Therefore, based on the web service architecture, a QoS-aware web service model (Q-WSFM) is proposed. Its purpose is to expand the description capabilities of UDDI without changing the UDDI specification and implementation, increase the description and discrimination capabilities of web services QoS, and further improve the efficiency of service discovery based on QoS constraints. Based on the existing SOA guidance framework, the Q-WSFM model adds a QoS perception center role and four interactions between the four roles, namely, quantification, negotiation, feedback, and ranking, as shown in Figure 2.

Application software modules or other web services that service consumers are called web services. They follow the "quantify-find-negotiate-bind-invoke" model. After the QoS perception center quantifies and judges the QoS requirements, it searches for services in UDDI according to the constraint conditions and then binds the agreement after it is successfully matched with the target service provider.

The service registry provides service registration and discovery service QoS functions. In order to realize service discovery based on QoS, the service registry must have the ability to support the description of QoS for web services. These functions are not implemented in the current UDDI specification. Here, the QoS properties of web services are described by a set of classification models.

The service provider performs this operation to publish the web service description information to the UDDI registry. The service description information should include service QoS attribute information and provide support for service discovery based on QoS constraints. At the same time, the QoS attribute information provided can also be used as the main indicator of differentiated services.

Service consumers call this operation to find and judge services that meet QoS requirements. The QoS classification information can be included in the query request to find services that meet the QoS constraints. The service discovered by this operation is only to locate the QoS of this type of service.

When the service is called, the service consumer starts the monitoring process, measures the service quality QoS of this service call, and feeds back the actual measurement result to the QoS perception center. At the same time, the

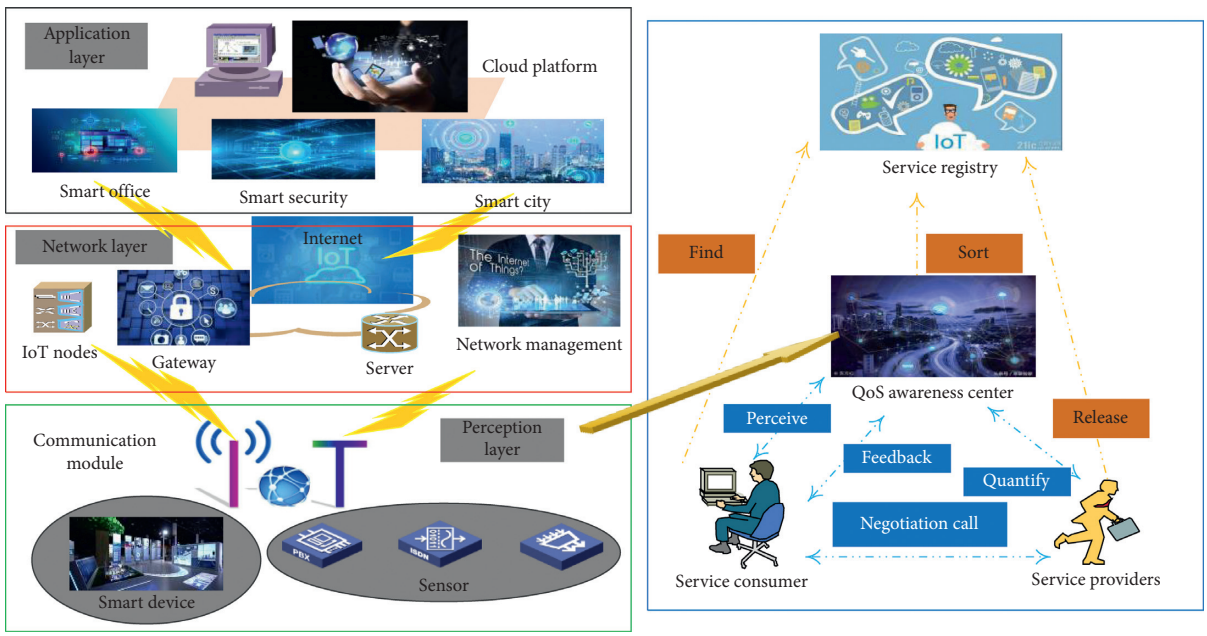

Figure 2: Q-WSFM model in IoT environment.

QoS perception center senses changes in the QoS of the service computing environment and compares and calculates it with the feedback QoS.

2.3. Intelligent Acquisition of Services Perceived by QoS. The key technologies that support intelligent acquisition of services based on QoS perception are composed of the following parts: service discovery and selection based on SVM machine learning, dynamic service selection based on QoS perception, service selection under the fuzzy or uncertain QoS of users, and service failure after service failure and other technologies. These key technologies can make the obtained web services have better adaptability, robustness, and service quality perception characteristics, can better provide users with the high-quality services they need, and ensure high service availability and high service quality.

2.3.1. Service Selection Technology Based on SVM Classification Mechanism. The service QoS requirements put forward by users are often coarse-grained, while the service QoS attribute description information provided by the underlying service is relatively fine. The service QoS management system architecture (SQMF) layered system can solve the different service QoS requirements. The granularity is dynamically adapted to different applications. Through the service QoS registration database and service selection history database for feature extraction and using support vector machines to train these characteristic historical data samples, a classification function, that is, the service selection decision function, can be obtained. The decision function is based on the classification of the service QoS. The service QoS requirements of the service requester are also characterized and normalized, and the service selection decision function obtained from the training is used for service selection.

2.3.2. Service QoS Uncertainty Description and Service Matching Technology. Web service selection based on QoS constraints generally uses multiple QoS parameters as evaluation indicators to calculate the overall service quality. On the one hand, how to measure the specific parameters of web service quality attributes has not formed a unified method and standard, and some attributes can be directly given. The specific quantified numerical form is expressed, and other attributes are usually expressed by quantified evaluation grades. The QoS requirements put forward by users are sometimes not clear enough, and there are certain ambiguities. Service quality prediction and machine learning service selection research provide new ideas. Figure 3 shows the framework of web service discovery and composition based on QoS.

For services with the same or similar functions, there are no effective strategies and criteria to judge the quality of services. Service QoS is an important measure to further distinguish service quality. Based on the scalable service QoS model and decision-making strategy, the service selection problem is transformed into a multiattribute decision-making problem, and the service is selected through the subjective weighting model, the objective weighting model, and the subjective and objective ideal point weighting model, which solves the resource reservation and reflects the service personality.

2.3.3. Service Failure Detection and Recovery Technology Based on Reflection Middleware Technology. Web service discovery based on QoS perception requires the solution of service selection when the service dynamically adapts to changes in the environment and resources. It can judge the status of the environment and resources through the detection mechanism and adjust the specific links

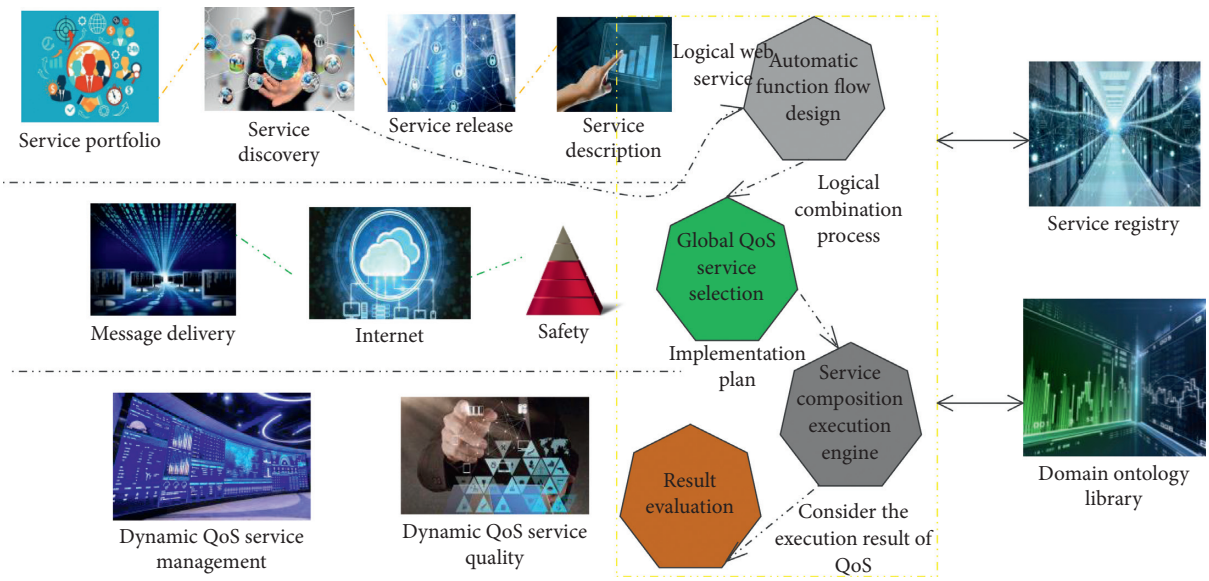

Figure 3: Web service discovery and composition framework based on QoS.

implemented within the service to improve the availability of the service.

They use reflection middleware technology to configure reflection middleware in web services. Through the Q-WSFM model with QoS awareness center, the QoS awareness center can realize the intelligent discovery of web services and dynamically negotiate and adjust web services for QoS operations. The parameter configuration or changing the service behavior adapts to changes in the service environment. Through the learning mechanism of hierarchical hybrid expert neural network (HME), it can effectively identify and classify the types of service effectiveness and environmental factors and the service QoS of service providers and service users, as well as the QoS affected by the environment and computing resources. Various indicators are detected and quantified, and the HME of the reflection middleware is used to implement service failure handling under server effectiveness, binding failures, and resource constraints, so that the QoS external environment and service internals of web services can be effectively guaranteed.

2.3.4. Service Dynamic Binding Technology under QoS Constraints. There are a large number of similar services in the open network. In these services and the distributed service registry, the required services can be found and selected, and similar alternative services can be obtained through similarity calculations, effectively handling service hits and service binding failures. According to the independent distribution of the service network library, the optimal service can be found through the distributed network library search algorithm based on the QoS constraint; according to the short path and high aggregation characteristics of the small-world network, the peers that provide web services can be peered. The node is constructed as a network with small-world attributes, the ant colony algorithm is improved, and the small-world network theory is used to solve the problem of web service selection based on QoS perception.

3. Web Service Selection That Supports QoS Global Optimization and Dynamic Replanning

3.1. Genetic Algorithm Optimization. The evolution of organisms is based on the group, and the corresponding genetic algorithm is a collection of M individuals, called a group. Similar to the natural evolution process of organisms from generation to generation, the calculation process of genetic algorithm is also an iterative process. The t-th generation population is recorded as $P(t)$, and the $t+1$ generation population is obtained after one generation of inheritance and evolution, which is recorded as $P(t+1)$. This group continues to undergo genetic and evolutionary operations, and each time, according to the rule of survival of the fittest, more individuals with higher fitness are inherited to the next generation, and finally an excellent individual can be obtained, which is at or close to the optimal problem.

The biological evolution process is mainly accomplished through the crossover and mutation of chromosomes. In the genetic algorithm, the generation of the next-generation $P(t+1)$ from the parent $P(t)$ is mainly achieved through selection, crossover, and mutation operations. The selection operation determines the individuals inherited to the next generation according to the fitness of the individual; the crossover operation randomly matches the individuals in the group to exchange some of their chromosomes with a certain probability to obtain a new individual; the mutation operation is for each individual in the group. A certain probability changes the value of the gene at the locus to other alleles, thereby generating a new individual.

The genetic algorithm first initializes the population $P(0)$, randomly generates M individuals, sets the evolution

algebra counter t to zero, and sets the evolution termination condition; it then performs individual evaluation, calculates the fitness of the individuals in $P(t)$, applies the selection operation to the population $P(t)$, and gets the output population. Then the next-generation population $P(t+1)$ is obtained; finally, the termination condition is determined. When the termination condition is reached, the evolution is stopped, and the optimal solution is output for the individual with the maximum fitness. If the termination condition is not reached, the genetic operation is used on the new population and repeated until the termination condition is met. Figure 4 shows the construction process of genetic algorithm.

The genetic algorithm deals with individuals with similar coding structure templates. As the concrete representation of some similar templates, the individual search process is actually the search process of these similar templates. After introducing the concept of pattern, the essence of genetic algorithm is the operation of pattern. That is, the excellent patterns in the current population are inherited to the next-generation population through the selection operator, the recombination of the patterns is carried out through the crossover operator, and the mutation of the patterns is carried out through the mutation operator.

3.2. Service Quality of Web Service Composition. The basic logical unit that constitutes a service composition only contains function description and interface information and does not point to a specific web service. It is a node of a functional flowchart or a logical web service. In order to effectively explain the problems in the service composition, the initial virtual task and the termination virtual task (two tasks that exist logically but not physically) are introduced to mark the logical beginning and end of a service composition.

The QoS of web services is a necessary element of web services, and it is also an important condition in various transactions between enterprises and enterprises and between enterprises and consumers. The QoS problem of service composition based on web services also occupies an extremely important position.

The quality of service of the web service composition describes the nonfunctional attributes of the execution plan. The QoS problem of service composition is particularly important in the service selection stage. The core is how to select a suitable one from multiple services with different QoS attributes to form an execution plan to meet the overall QoS needs or constraints of users.

The QoS of the web service composition also contains multidimensional QoS properties, and the description of the properties is similar to the previous single web service QoS properties. It can be seen that, in terms of the QoS model of service composition, the research has just started. The existing research work is relatively simple to deal with the problem, and it has not been able to give a general QoS model that supports service composition. Here we still choose more general execution time, cost, credibility, reliability, availability, and other attributes as the research object of QoS for service composition.

The QoS properties of the web service composition studied in this paper include execution time T (time), cost C (cost), reputation Rep (reputation), and reliability R (reliability). Considering that both availability and reliability are attributes that describe service assurance capabilities, this paper selects the reliability attributes that are more commonly used in service composition for research. Let ep be an execution plan composed of multiple services, and let si be a single service that composes the execution plan. The QoS attribute calculation method of the basic structure of the service composition is as follows:

$$\begin{aligned} T_{ep} &= \operatorname{Max}\{T_1, T_2, T_3, \ldots, T_{n-1}, T_n\}, \\ C_{ep} &= \prod_{i=0}^{n-1} (C_i \cdot C_{i+1}), \\ \operatorname{Re}p_{ep} &= \frac{\prod_{i=0}^{n-1} (\operatorname{Re}p_i \cdot \operatorname{Re}p_{i+1})}{n}, \\ R_{ep} &= \operatorname{Min}\{R_1, R_2, R_3, \ldots, T_{n-1}, T_n\}. \end{aligned} \tag{1}$$

3.3. GODRP Algorithm Idea. Global Optimal and Dynamic Replanning (GODRP), in view of the limitations of current coding methods in genetic algorithms, adopts the idea of adjacency matrix in graph theory and designs a representation tool that supports service composition-location matrix. This can describe not only the static positional relationship between the chromosome structure and tasks but also the dynamic situation of combined paths, service execution, and service replanning. It not only improves the expressive power of coding methods but also improves the computing power of service selection. GODRP is based on the Hamming distance of service quality between individuals, introduces a population diversity control mechanism, and uses the combination of the ranking value of the objective function and the density value of the individual in the group to calculate the individual fitness value. The selection mechanism can obtain the optimal service combination of QoS. When GODRP processes dynamic replanning, it considers the impact of the execution time of the algorithm itself on the service composition. The service composition undergoes multiple replanning during the execution process, and the total execution time will gradually become larger as the algorithm runs. The execution time of the service composition is less than the limit time given by the user. GODRP introduces the auxiliary population to retain the good solution individuals with a large current fitness value and meets the user's time requirements, so that the web service composition controlled by QoS is closer to the actual situation and the algorithm is more applicable. In short, GODRP is based on genetic algorithm to better solve the problem of supporting QoS global optimization and dynamic replanning in service composition.

In this paper, tasks t are defined as a locus, the number of which is equal to the number of tasks in the service portfolio structure diagram. All loci constitute the basic structure of

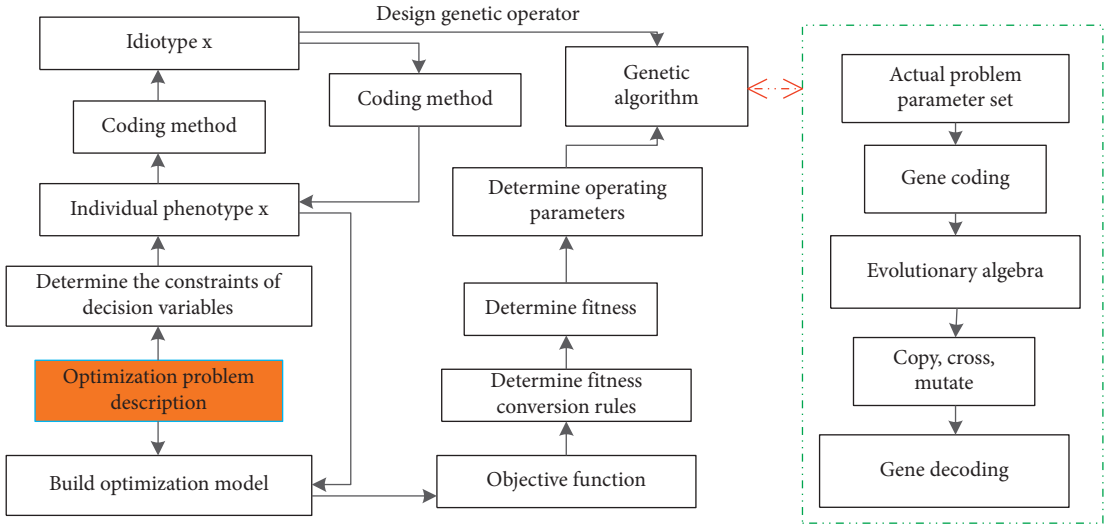

Figure 4: The construction process of genetic algorithm.

the chromosome, and each locus contains a set of candidate services matching the corresponding task. Each service in the service set then points to an array representing the QoS attribute. Since the lengths of all chromosomes are the same, the integer fixed-length encoding method is adopted. The first (last) gene in the chromosome is always the starting (end) point of the service combination, and each gene in the middle of the chromosome corresponds to a specific service in the candidate.

3.4. GODRP Algorithm Implementation Process. The GODRP algorithm is divided into two stages. In the first stage, from the perspective of QoS global optimization, we search for a set of noninferior solutions that meet the constraints in all execution plans. The main process is encoding the execution plan into chromosomes and generating new chromosomes with higher fitness through recombination operations such as crossover and mutation between chromosomes. This process is continuously carried out to realize a parallel global search in the solution space. When the search stops, a set of chromosomes is obtained, that is, a set of optimized or approximately optimized execution plans which meet the constraints. In the second stage, the dynamic replanning of the service portfolio is completed when the execution plan is running. The main process is to record the execution status of the service composition through the location matrix. When the execution plan fails or the QoS value of an unexecuted web service in the chromosome changes significantly, first we calculate the service composition and the running time that the current execution plan has taken and then estimate the time it will take to run the GODRP algorithm again. When the running time of the algorithm is within the acceptable range of the user, we execute the GODRP algorithm again. On the contrary, we select the execution plan that can meet the user's requirements from the set of alternative execution plans to complete the service portfolio.

Fitness evaluation is the basis of genetic operation, in which the design of the objective function directly affects the performance of genetic algorithm. According to the QoS model, we get the calculation formula of the objective function $H(x)$:

$$H(x) = \frac{\theta_c C_{ep} + \theta_T T_{ep}}{\theta_R R_{ep} - \theta_{\mathrm{Rep}} \mathrm{Re}p_{ep}}. \tag{2}$$

In the previous equation, Tep, C_{ep}, $\mathrm{Re}p_{ep}$, and R_{ep} are all normalized QoS attribute values, and θT, θC, θRep, and θR are the corresponding weights, which indicate the degree of user attention to the QoS attribute.

Therefore, the objective function $f(x)$ needs to be adjusted and mapped to the objective function $F(x)$ to obtain the maximum value. Normally, in order to transform a minimization problem into a maximization problem, it is only necessary to simply the function, but, in genetic algorithm, it is necessary to sort according to the objective function value and calculate the probability of being selected on this basis, so the value of the function should be positive. To this end, the following methods are used for conversion:

$$K(x) = \begin{cases} 0, & A \le H(x), \\ A - H(x), & A > H(x). \end{cases} \tag{3}$$

A is an adaptive constant. Its function is to inspire according to the distribution of individual objective function values during the operation of the algorithm, make adaptive adjustments, change the distribution of individual objective function values of a generation, and make the algorithm approach the global optimal solution. Usually, the initial value of A is a large positive number.

During the operation of the algorithm, if the objective function D exceeds the above range, then A will be adjusted:

$$A = \begin{cases} A + \dfrac{K_{\max} + K_{\min}}{2\ln(t+1)}, & D \geq D_{\max}, \\ A - \dfrac{K_{\min}\ln(t-1)}{\ln(t+1)}, & D < D_{\min}. \end{cases} \quad (4)$$

In the early stage of evolution, the function value of each individual is quite different, and the performance result is that D is larger. By adjusting the value of A (increase), the difference in the function value of each individual is reduced, so the probability of each individual being selected is not too different, which can avoid falling into the local optimum and effectively prevent the occurrence of premature problems. In the later stage of evolution, individuals in the group are generally close to each local optimum, and the difference in function value is relatively small; that is to say, D is relatively small. Through the adjustment (decrease) of A, the degree of discrimination between individual function values is increased, so that better individuals have a higher probability of being selected, thus speeding up the search near the optimal neighborhood and improving the convergence speed.

According to evolutionary theory, it is always hoped to select chromosomes with higher quality to participate in subsequent genetic operations. This article uses two steps to calculate the individual fitness value. Firstly, the objective function value of each individual is obtained based on the objective function, and then the probability of each individual being selected is calculated and sorted according to the size of the probability to obtain the ranking value. Then, through the individual diversity maintenance strategy to modify the individual ranking results, you finally get the fitness value of each individual in the group.

In a dynamic environment, considering the recurrence and randomness of replanning and the time it takes, users often have certain requirements for the total execution time. For example, if the total execution time is required to not exceed T0, once the service execution process is required for combined replanning, the execution time T^{ga} occupied by the replanning GODRP algorithm itself needs to be considered. If a total of k times of rescheduling are carried out from the start of the service composition to the completion of the service execution, the execution time of the service composition can be expressed as

$$T = T_{\text{end}} - \prod_{i=1}^{k} \left[T_i^{ga} + T_i\right]. \quad (5)$$

In the previous equation, T_i represents the time difference from the completion of the i-1th replanning of the combined service to the time when the i-th replanning occurs, T_1 is the time difference from the start of the combined service to the first replanning, and T_{end} is the time of the end of service execution.

4. Experimental Verification and Analysis

4.1. Data Set and Experimental Environment. In order to verify the accuracy of QoS and the extent to which it satisfies users' QoS needs, this article only considers the measurement of QoS data under the same service function. The simulation experiment environment is as follows: an IBM server, which is configured with Intel (R) 2.33 GHz CPU and 4 GB of memory, and the operating system is Windows Server 2003; three PCs are used as user test machines, configured as Pentium (R) D 3.00 GHz CPU, 8 GB memory, Windows 7 operating system; the software development environment is Visual Studio.Net 2010, and the development language is C#.

Three weather forecast services are configured on the IBM server. In addition, the QoS proxy is also configured on the server and embodied as a web service. The three test machines, respectively, make call requests to the three services on the server, and the QoS agent records the demand information, feedback information, and feedback time of the service requester. The experiment simulates a system with 1,000 service requesters and simulates the requester to make arbitrary calls to three services on the IBM server and collect feedback QoS information. The service requester gives its QoS demand value when calling the required service and gives the QoS feedback value after the service call is completed, and the QoS agent records this information.

4.2. Analysis of the Impact of Dynamic QoS Calculation Accuracy on User Satisfaction. In order to evaluate the accuracy of the QoS measurement method used in this paper to calculate QoS data and the web service ranking method based on user weight, we use the satisfaction of the service requester as the evaluation index.

We use the .Net delay function to adjust the actual response time of each service. The response time of service 1 (WS1) starts at 30 s and then continues to change at a rate of 1 s each time; that is, its response time continues to fluctuate; service 2 (the response time of WS2) has always been in the middle area; the response time of service 3 (WS3) started at 8.5 s and then continued to fluctuate and finally was 8 s. The simulation result of the change trend of response time is shown in Figure 5. The simulation result of the change trend of service price is shown in Figure 6.

When the time is 30 s, the simulated requester takes the response time as a parameter to request the service. Since the calculation of QoS based on user feedback information of similar needs is affected by the feedback information of users who use the service earlier, it cannot be based on the dynamic characteristics of web services. The QoS information is reflected in real time and accurately. Therefore, we calculate QoS's satisfaction with current service requesters based on recent user feedback. The experimental results are shown in Figures 7 and 8, respectively.

It can be seen from Figure 7 that when the forgetting factor $\lambda < 0.6$, the requester's satisfaction with the service WS3 with the largest response time decreases as λ continues to increase; on the contrary, the satisfaction with the service WS1 with the shortest response time increases. The smaller the λ, the greater the influence of the feedback that is farther from the current time on the QoS value of the service; when $\lambda = 1$, all the QoS values of the feedback have the same

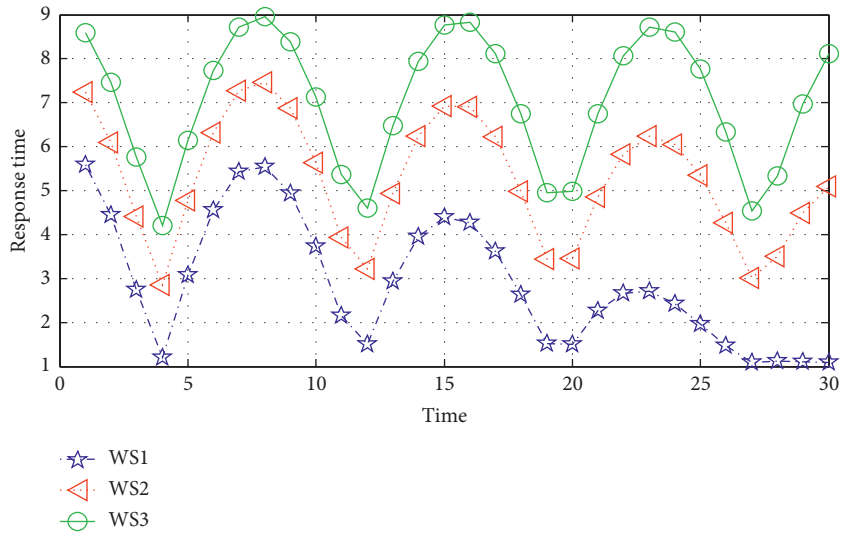

Figure 5: Trend of response time.

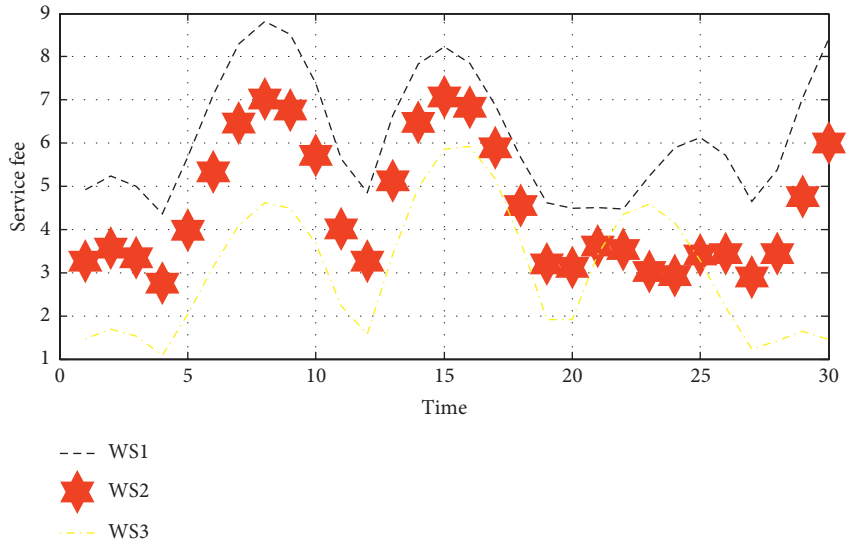

Figure 6: Trends in service prices.

importance. When the value of λ continues to increase, the satisfaction gap of the requester with the three services gradually narrows. This is mainly because as the value of λ increases, the calculated QoS value is affected by the feedback of users who use the service earlier. It can be seen that, by setting the λ value reasonably and calculating according to recent user feedback, real-time QoS information can be better reflected. It can also be seen from Figure 7 that when $\lambda = 0.9$, user satisfaction is the highest. Therefore, when $\lambda = 0.9$, user feedback based on similar needs is used to calculate the service's satisfaction with the current requester.

It can be seen from Figure 8 that when the value of α is smaller, the calculated QoS value is closer to the QoS calculated by the feedback average value, and the satisfaction of the three services to the requester is not much different; as the value of α increases, if the QoS calculated based on similar needs based on the recently obtained user feedback information is closer to the requester's demand information, the requester's satisfaction is higher. In this example, when λ is 0.9 and α is 0.6, the satisfaction is the highest. It can be seen that using the method in this article to calculate QoS can effectively integrate the development trend of QoS and the degree of closeness to the requester's needs, so as to better meet user needs. In addition, when calculating QoS data, the forgetting factor λ and similarity α can be set according to the development trend of QoS and the degree of similarity to the requester's needs and the number of feedbacks.

4.3. Dynamic QoS Weighted Validity Verification. The user needs to give a set of weight values when requesting services to indicate the importance of each QoS attribute. Table 1 lists the QoS values of 10 weather forecast services calculated by the above method, where Qc (Si) represents the service price

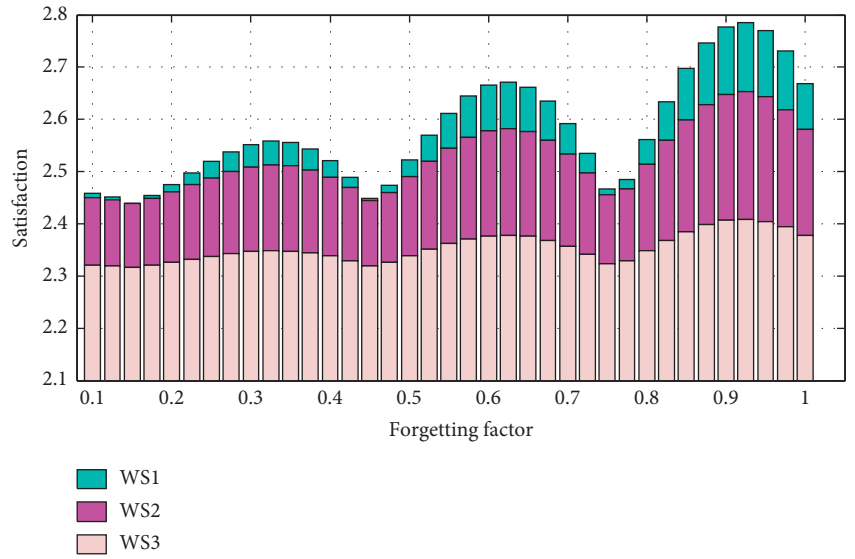

Figure 7: Calculating satisfaction based on forgetting factor.

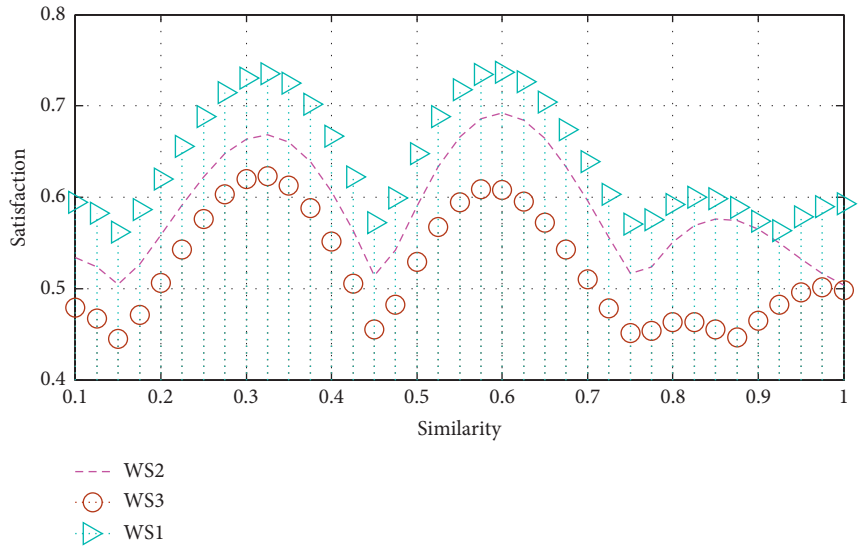

Figure 8: Calculating satisfaction based on similarity.

Table 1: Web service QoS information.

	Qc (Si)	Qt (Si)	Qa (Si)	Qr (Si)
S1	1.3	169	79	87
S2	1.8	148	97	94
S3	0.9	157	80	91
S4	1.4	151	94	83
S5	1.7	164	96	79
S6	1.9	175	87	91
S7	0.9	192	91	83
S8	0	154	95	89
S9	0.9	149	92	82
S10	4.7	143	87	94

of the service Si, which refers to the fee that must be paid to call the service, which is determined by the web service provider. This is a certain value and does not change due to the dynamics of the environment; Qt (Si) represents the response time of the service Si, which refers to the time interval from the call request from the client to the server to

the response. The waiting time, execution time, and communication time of the service are measured by embedding a timer code in a third-party QoS certification center.

Qa (Si) represents the availability of the service Si and refers to the probability of the service running normally. Since the web service is called on a time-by-time basis, after each call, the requester reports whether the call is successful or not. The ratio of the number of successful calls to the total number of calls can be used to describe its availability.

Qr (Si) represents the reputation of the service Si and reflects the credibility of the service. The larger the value, the higher the credibility; that is, the closer the actual execution result of the service is to the user's expectation, the higher the credibility is, and the credibility of the service is calculated based on past statistical data.

5. Conclusion

With their unique advantages, web services enable people to see their wide range of application prospects, and the effective solution of the web service portfolio that supports QoS will play a powerful role in promoting the popularization of Web services. The genetic-algorithm-based QoS global optimization and dynamic replanning web service selection algorithm (GODRP) proposed in this paper uses a position matrix to encode genes, so that the encoding method can represent multiple types of service combinations and all combination paths. The initial state and update status of dynamic replanning effectively improve the efficiency of algorithm search. GODRP uses a diversity maintenance strategy to make the final selected execution plan have a larger objective function value, which can maximize the QoS requirements of users. In addition, the research in this paper pays attention to the impact of the execution time of the GODRP algorithm itself on the quality of the service portfolio and solves this problem by specifying a total execution time limit and a good solution retention strategy. The experimental results show that, using the method of measuring QoS data in this paper, when calculating QoS data, the forgetting factor λ and similarity α can be set according to the development trend of QoS and the degree of similarity to its demand and the amount of feedback. This can be effectively integrated into the development trend of QoS and close to the requester's needs and can better meet user needs. In the future, in terms of supporting QoS service discovery, we will further study the dynamic expression of web service QoS notice and QoS demand and authenticate the service provider QoS notice. We need to study a more reasonable quantization interval and a better compromise between description accuracy and matching efficiency. We also need to study the optimized representation method and operation method of the quality matrix when the number of web services is large, so as to reduce the time complexity of the algorithm operation. These improvements will make the discovery of web services that support QoS closer to reality.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no known conflicts of interest or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] B. Pourghebleh, V. Hayyolalam, and A. Aghaei Anvigh, "Service discovery in the Internet of Things: review of current trends and research challenges," *Wireless Networks*, vol. 26, no. 7, pp. 5371–5391, 2020.

[2] N. Kashyap, A. C. Kumari, and R. Chhikara, "Service discovery and selection in internet of things-a review," *Recent Patents on Engineering*, vol. 14, no. 1, pp. 4–11, 2020.

[3] W. Osamy, A. M. Khedr, and A. Salim, "ADSDA: adaptive distributed service discovery algorithm for internet of things based mobile wireless sensor networks," *IEEE Sensors Journal*, vol. 19, no. 22, pp. 10869–10880, 2019.

[4] R. K. Chahal, N. Kumar, and S. Batra, "Trust management in social Internet of Things: a taxonomy, open issues, and challenges," *Computer Communications*, vol. 150, pp. 13–46, 2020.

[5] M. J. Aslam, S. Din, J. J. P. C. Rodrigues, A. Ahmad, and G. S. Choi, "Defining service-oriented trust assessment for social internet of things," *IEEE Access*, vol. 8, pp. 206459–206473, 2020.

[6] B. Pourghebleh, K. Wakil, and N. J. Navimipour, "A comprehensive study on the trust management techniques in the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9326–9337, 2019.

[7] Y. He, J. Chen, and P. Lu, "Testing dynamic composition of semantic internet of things services based on QoS," *IEEE Access*, vol. 7, pp. 113103–113113, 2019.

[8] A. AlZubi, A. Alarifi, M. Al-Maitah, and O. A. Albasheer, "Location assisted delay-less service discovery method for IoT environments," *Computer Communications*, vol. 150, pp. 405–412, 2020.

[9] R. Kurte, Z. Salcic, I. Kevin et al., "A distributed service framework for the internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4166–4176, 2019.

[10] F. Marino, C. Moiso, and M. Petracca, "Automatic contract negotiation, service discovery and mutual authentication solutions: a survey on the enabling technologies of the forthcoming IoT ecosystems," *Computer Networks*, vol. 148, pp. 176–195, 2019.

[11] H. Li, J. Tong, S. Weng, X. Dong, and T. He, "Detecting a business anomaly based on QoS benchmarks of resource-service chains for collaborative tasks in the IoT," *IEEE Access*, vol. 7, pp. 165509–165519, 2019.

[12] M. S. Roopa, S. Pattar, R. Buyya et al., "Social internet of things (SIoT): foundations, thrust areas, systematic review and future directions," *Computer Communications*, vol. 139, pp. 32–57, 2019.

[13] I. Aoudia, S. Benharzallah, L. Kahloul et al., "Service composition approaches for internet of things: a review," *International Journal of Communication Networks and Distributed Systems*, vol. 23, no. 2, pp. 194–230, 2019.

[14] T. Ojha, S. Misra, N. S. Raghuwanshi, and H. Poddar, "DVSP: dynamic virtual sensor provisioning in sensor-cloud-based internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5265–5272, 2019.

[15] X. Ding and J. Wu, "Study on energy consumption optimization scheduling for internet of things," *IEEE Access*, vol. 7, pp. 70574–70583, 2019.

[16] D. Ntalasha and T. Chisanga, "Context aware self-optimisation scheduling in internet of things," *International Journal of Intelligent Internet of Things Computing*, vol. 1, no. 2, pp. 129–144, 2020.

[17] B. Nour, K. Sharif, F. Li et al., "Security and privacy challenges in information-centric wireless internet of things networks," *IEEE Security & Privacy*, vol. 18, no. 2, pp. 35–45, 2019.

[18] D. G. Roy, P. Das, D. De et al., "QoS-aware secure transaction framework for internet of things using blockchain mechanism," *Journal of Network and Computer Applications*, vol. 144, pp. 59–78, 2019.

[19] Z. Yang, Y. Ding, K. Hao, and X. Cai, "An adaptive immune algorithm for service-oriented agricultural Internet of Things," *Neurocomputing*, vol. 344, pp. 3–12, 2019.

[20] Y. A. Qadri, A. Nauman, Y. B. Zikria, A. V. Vasilakos, and S. W. Kim, "The future of healthcare internet of things: a survey of emerging technologies," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1121–1167, 2020.

[21] Y. Liu, X. Sun, W. Wei, and W. Jing, "Enhancing energy-efficient and QoS dynamic virtual machine consolidation method in cloud environment," *IEEE Access*, vol. 6, pp. 31224–31235, 2018.

[22] J. Han, N. Lin, J. Ruan, X. Wang, W. Wei, and H. Lu, "A model for joint planning of production and distribution of fresh produce in agricultural internet of things," *IEEE Internet of Things Journal*, p. 1, 2020.