Optimization and Solution of Cloud-fog Collaborative Computing Network Model in Heterogeneous Scenarios

Hongqing Liu*
College of Logistics Information, Hunan Modern Logistics College, Changsha 410131, China
*Email of corresponding author: liuhongqing2020@163.com

Abstract: In the context of the rapid development of the Internet of Things, cloud-fog collaborative computing has become a key technology to solve the problem of insufficient computing resources and network latency in heterogeneous scenarios. A cloud-fog collaborative computing network model is proposed to address the issues of uneven allocation of computing resources, poor performance in pure cloud or fog computing in heterogeneous scenarios, and effective resource allocation is achieved through optimized genetic algorithms. Firstly, a cloud-fog collaborative computing network model is constructed, which improves computing performance and network latency through the model's advantages. Subsequently, the control parameters of the population fitness standard deviation optimization genetic algorithm are introduced, and an improved genetic algorithm is designed to solve resource allocation problems. The results showed that the improved algorithm had an accuracy of 0.996, proving its high accuracy. The average calculation time of the improved algorithm was only 3 seconds, and the error range was controlled within 0% -1.3%, proving its high computational efficiency. In practical application scenarios, when the number of users increased from 5 to 40, the system cost only increased by 12.1%, indicating that the increase in computing resources did not have a significant impact on the system cost. The above results indicate that the proposed method has significant effectiveness in improving the performance of IoT systems, providing strong support for the development of cloud-fog collaborative computing networks.

Keywords: Internet of Things, Cloud-fog collaborative computing, Standard deviation of fitness, Genetic algorithm, Resource allocation

1. Introduction

In today's digital era, the Internet of Things (IoT), as an innovative technological concept and application model, has received widespread attention and application. The IoT connects sensors, devices, physical objects, and virtual objects to achieve information interconnection have brought tremendous convenience and change to social production and life [1-2]. However, with the expansion of the application scale of the IoT and the increase in the amount of data, the problems of scalability, delay, and energy consumption faced by traditional network models such as cloud computing (CC) and edge computing are increasingly prominent [3]. The optimization and solution of network models in the IoT also face many challenges [4]. Due to the heterogeneity and dynamism of devices and sensors in the IoT, how to reasonably allocate computing tasks and resources to achieve task balance and system stability has become a key issue. Meanwhile, due to the large scale of data and complex and diverse computing tasks in the IoT, there is an urgent need for research and improvement on how to efficiently process and analyze data to meet user needs [5-6]. For this reason, some scholars have proposed fog computing (FC). Compared to CC, FC has better storage and communication capabilities, but its computing power is significantly lower than CC. However, in the IoT, due to the heterogeneity and dynamism of devices and sensors, the allocation of computing resources is often uneven, which may lead to imbalanced task allocation and affect the stability of the entire system. Meanwhile, due to the large scale of data and complex and diverse computing tasks in the IoT, relying solely on CC or FC may make it difficult to efficiently process and analyze data to meet user needs. For example, in application scenarios such as intelligent transportation and smart cities, due to the differences in data generation and processing among different nodes, some nodes may have insufficient computing resources, while others may have excessive computing resources. In addition, if CC or FC is simply used, it may lead to delayed and inaccurate data processing due to network latency, data transmission, and other issues, thereby affecting the performance of the entire system. Therefore, to achieve optimization of the IoT in heterogeneous scenarios, a cloud-fog collaborative computing (CFCC) network model is proposed, and a multi-strategy improved genetic algorithm (IGA) is used to solve the model to improve the performance of the IoT. The contribution of the research lies in applying the IGA to solve the cloud-fog collaborative resource allocation problem in heterogeneous scenes. This method can ensure lower computational complexity and significantly reduce the average system cost. This study is divided into four parts. The first part discusses the current development status of CC and genetic algorithm (GA); the second part constructs a CFCC network model and proposes optimization solution strategies; the third part tests and analyzes the CFCC network model, and evaluates the optimization and solution effectiveness of the model; the final part summarizes the research results and explores the main directions for future research.

2. Related works

CC has strong computing power in processing and managing large amounts of user data, promoting social interaction, and building broader social networks. It has significant advantages in large-scale data processing and efficient computing tasks. Many researchers have conducted research on it. Osman et al. proposed a mixed integer linear programming model to solve network delay and resource consumption in cloud architecture and optimize the allocation of data positions generated by the fog layer in cloud architecture. It could also reduce the overall power consumption and data delay of the architecture, and a joint optimization model was established for cloud architecture power consumption and delay. The research findings indicated that the optimized model could save 92.3% of power consumption and greatly reduce data transmission delay [7]. Tripathy et al. proposed a load-balancing model of the IoT based on context awareness and CC for the smart city environment. Through the load balancer composed of the Internet, cloud, and two fog layers, the network traffic and server load in the Internet were dynamically distributed to reduce the comprehensive system load in the smart city IoT environment and improve the resource utilization and quality of service of the IoT [8]. Sharma and Park proposed a distributed CC network structure for the IoT environment of the intelligent transportation system. The integration of blockchain technology facilitated the registration and authentication processes for IoT devices deployed in intelligent transportation systems. Furthermore, it enabled the generation of user signatures in conjunction with the aggregation signature scheme and optimized the allocation of computational resources within internet systems by employing the use of intelligent deployment contracts, thus enhancing the overall security of the intelligent transportation system. The research outcomes denoted that CC networks can effectively reduce system latency, achieving 81% low latency at local nodes [9]. Liu designed a coordinated control model for a multi-agent system utilizing CFCC technology. They then employed cloud edge computing to realize the overarching coordinated planning and control of a substantial multi-agent system. Furthermore, they improved the individual agent coordination of the system by implementing active compensation for communication delay. Finally, they employed time domain learning prediction in order to obtain the potential output of the system agents. [10]. Li et al. proposed a refined 4-layer IoT structure based on CC to improve the quality of service and response speed of the Internet and enhance the processing and transmission capacity of the IoT on-edge device. The cloud-fog IoT structure was based on adaptive load balancing of links, which reasonably planned and dynamically allocated computing resources in the IoT system, effectively reducing the comprehensive operating costs of the system. The experimental findings expressed that the cloud-fog IoT structure could effectively improve data transmission rates by 12.7% and 8% in medium and heavy-load environments [11].

GA simulates natural genetic evolution for optimal solution search, with excellent global search and parallel processing capabilities, and is widely used in multiple fields. Dharma et al. proposed an inflation prediction model based on GA and regression analysis to address the inflation problem in Indonesia. This model combined historical consumption data to predict the price levels of goods and services in Indonesia, thereby achieving comprehensive prediction and evaluation of the country's inflation situation. The research results indicated that the mean square error of the inflation prediction model was 0.1099, which had a high prediction accuracy [12]. Nikbakht et al. applied GA to the optimization and improvement of neural networks and utilized the optimization advantages of GA to solve the hyperparameters of the neural network. They used GA to optimize the hidden layers and the number of neurons in each layer of the neural network to improve the prediction accuracy of the neural network [13]. Pal team introduced GA into the optimization problem of wireless sensor networks and proposed an energy-saving clustering technology for wireless sensor networks based on GA. The GA was used to optimize the load and energy allocation of wireless sensors in heterogeneous environments, and from the perspectives of sensor compactness and separation, the energy efficiency and network scalability of sensors were improved [14]. Yakubu and Murali designed a meta-heuristic method based on improved Harris-Hawks optimization to reduce the transmission delay of CFCC resource allocation, which can allocate tasks and layer resources between cloud and fog layer. The results showed that this method was significantly superior to other algorithms in terms of maximum completion time, execution cost and energy consumption [15]. To reduce the delay and power consumption of cloud networks in heterogeneous environments, Ghosh and De designed a heterogeneous cloud-fog architecture based on the weighted majority cooperative game theory, in which different cloud-fog structures can cooperate to perform tasks. Experimental results showed that the delay and power consumption of this architecture were significantly reduced compared with other networks [16].

In summary, CC, as an emerging computing model, has strong capabilities in processing and managing a large amount of user data, promoting social interaction, and building a wider social network. However, currently, most studies still find it difficult to further improve their computational and solving abilities, which limits their performance in practical applications. In order to solve this problem, this study proposes a multi-strategy GA optimization solution strategy, which has the advantages of improving the computing power of CFCC network models in heterogeneous environments, better-utilizing resources, improving computing efficiency and service quality. Moreover, it can be widely applied in practical applications to provide technical support for the development of the IoT.

3. CFCC network model and GA optimization algorithm design

The IoT has promoted social development, and how to further improve the development speed of the IoT is a problem that many research institutions need to consider. This study proposes a CFCC network model to address the high latency defects of CC in the IoT. Firstly, a CFCC network model is constructed in heterogeneous scenarios. Secondly, an optimization plan is proposed for resource management and allocation. Finally, an optimization solution strategy based on an IGA is constructed to further optimize the CFCC network model.

3.1. Construction of CFCC network model

In the IoT system, CC provides powerful computing power for devices connected to the network, enabling them to transfer computing tasks to the cloud for solving, greatly reducing the computational workload of devices and improving their operational efficiency. However, CC has a significant drawback, which is its high computing latency and transmission energy consumption, which cannot be applied in some special scenarios, such as smart transportation. FC is developed on the basis of CC and can effectively solve the high latency and privacy security issues of CC [17-18]. However, as an extension of CC, FC cannot completely replace CC because its data processing capabilities are still not ideal when facing a large number of IoT devices. From the current research status, using CC and FC alone cannot be applied to data processing in multiple scenarios [19-20]. Therefore, a CFCC network resource allocation model in heterogeneous scenarios is proposed to enable CC and FC to work together, compensate for each other's shortcomings, and improve the performance of IoT systems. Specifically, based on the CC and FC architecture of the IoT, this article constructs an optimization problem model that minimizes system costs by optimizing offloading decisions, computing resource allocation, and uplink transmission power allocation. Through comparative simulation, the superiority of CFCC is proved. The resource allocation model for the CFCC network constructed in heterogeneous scenarios is shown in Figure 1.

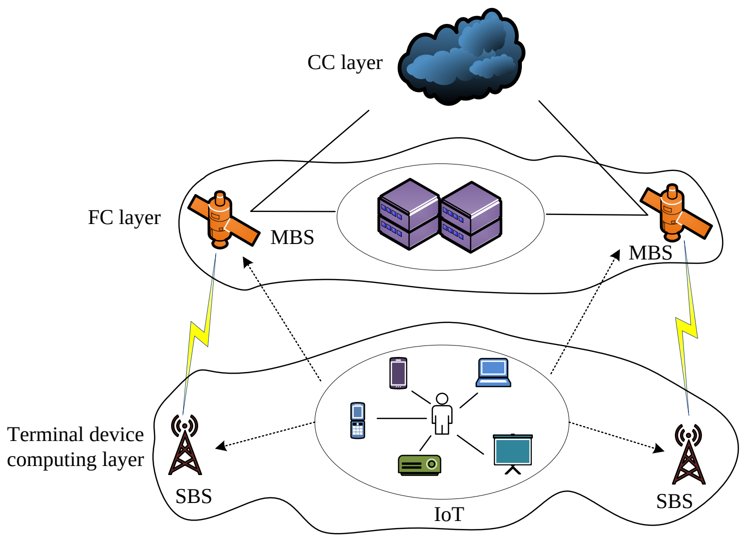


Figure 1. CFCC network resource allocation model

In Figure 1, a collaborative computing model combining CC, FC, and the IoT is constructed. CC is mainly used for calculation, providing computing resources for FC, and in-depth analysis and planning of the results of FC is conducted. It utilizes fog nodes to reasonably allocate tasks to various devices.

With the surge of multi-source heterogeneous data in the IoT, not only is data collection and uploading facing a huge load, but data processing and analysis systems also face great pressure. In traditional IoT systems, clouds are often important data processing and analysis centers. At present, the time required for cloud processing operations is increasing exponentially, and the current computing power is no longer able to support the calculation and analysis of large amounts of data in a short period of time. Therefore, there is an urgent need to optimize cloud capabilities [21-22]. In response to the above issues, research summarized three directions for optimization. Firstly, it is necessary to have better computing power than batch processing. Secondly, it is necessary to shorten data transmission time. Finally, it needs to ensure that it can be connected to the microcontroller and has an accurate and efficient information exchangeability. The optimized CC and FC for this study is shown in Figure 2.

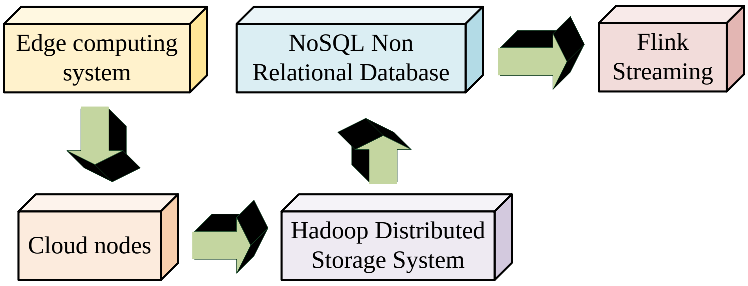


Figure 2. Flow chart of CC and FC

In Figure 2, the work of improving CC and FC mainly includes the platform construction of Hadoop and the data connection between Flink and Hadoop. Currently, the widely used data processing methods include Flink, Storm, and SparkStreaming. However, in the testing of the study, Flink and Storm have significantly higher processing speeds than SparkStreaming. Although the speed difference between Flink and Storm data processing is not significant, the latency of Storm data processing is much higher than Flink, so it often consumes more time when implementing operations outside of data processing. In addition, Flink has higher data throughput and a better fault tolerance mechanism. Therefore, Flink can ensure that the results after the operator operation are only stored once, without causing resource waste or a situation where incorrect results overwrite correct ones. Therefore, to meet the requirements of lower latency and data security, this study chose Flink as the computing method for CC. Flink processing requires building Hadoop as data storage. The Hadoop platform built in this study includes a Master node, a special sub node, two general sub nodes, and two Zookeeper nodes. Due to excessive code and similar basic configurations, there is no introduction to building Hadoop.

3.2. Resource management and allocation design of CFCC network model

The CFCC network resource allocation model constructed by the research is a three-layer architecture, mainly including CC, FC, and the IoT. In this model, IoT devices can transfer their own computing tasks to the cloud or fog through heterogeneous networks. In this heterogeneous CFCC network system, there are a total of n users, which can be represented as a user set $n=\{1,2,\ldots,N\}$. In this user set, each user undertakes a computing task, as shown in Equation (1).

$$W_n=\{D_n,C_n,t_n^{\max}\},n\in(1\sim N) \tag{1}$$

In Equation (1), W_n represents the calculation task; D_n denotes the input data size for the task; C_n means the computing resources required after completing the task. In Figure 1, there is a macro base station (MBS) and M small base station (SBS). Therefore, the base stations (BS) set can be represented as Equation (2).

$$D_M = \{1, 2, \ldots, M\} \tag{2}$$

In Equation (2), D_M represents the set of base stations, and M represents the M -th base station. The channel bandwidth of each base station can be expressed as ω_M , $\omega_1 = \omega_2 = \ldots = \omega_{M-1} \neq \omega$. In this model, there are three calculation methods for any task of the user, namely, there are two ways for the user to migrate the calculation task. The first is to directly unload the task to MBS. The second method is to unload the task to SBS and then transfer it from SBS to MBS. The operating frequency bands of MBS and SBS are different. The spectrum is divided into K subchannels, indicated as $K = \{1, 2, \ldots, K\}$, and each subchannel can and can only be used by one user. It sets all BS channel bandwidth to local processing, offloading to cloud processing, and offloading to fog processing. The unloading decision of a task can be expressed as Equation (3).

$$S = \{-M-1, -M, \ldots, M\} \tag{3}$$

In Equation (3), S represents the unloading decision of the character. By developing different offloading strategies for computing tasks for users, tasks can be offloaded to either a MBS or a SBS and then transferred to another base station for further processing. This strategy can minimize system costs while maintaining high performance. The user's offloading strategy can be expressed as Equation (4).

$$A = \{a_n = i \mid i \in S, n \in N\} \tag{4}$$

In Equation (4), when $a_n = 0$, it indicates that the user n is processed locally. When $a_n = m, m \in M$ occurs, it denotes that the user n offloads the calculation task to SBS, transfers it to MBS, and then offloads it to the fog end for calculation. When $a_n = -m, m \in M$, it means that the user n uninstalls the computing task to SBS, then transfers it to MBS, and then uninstalls it to the cloud for computing. When $a_n = M$, it expresses that user n directly unloads tasks to the fog end through MBS. When $a_n = -M-1$, it refers to that user n directly uninstalls tasks to the cloud through MBS. In local computing, the delay t_n^0 and energy consumption P_n^0 of task W_n are calculated as shown in Equation (5).

$$\begin{cases} t_q^0 = \dfrac{C_s}{f_q^0} \\ P_q^0 = \xi\left(f_q^0\right)^2 C_q \end{cases} \tag{5}$$

In Equation (5), f_q^0 denotes the computing power of the device q; C_q means the resource required for device q solving tasks; ξ expresses the cyclic energy consumption coefficient of the CPU. If the data D_q uploaded by the device q is unloaded to the fog or cloud, the uplink transmission rate is calculated using Equation (6).

$$r_q^m = n_m \omega_m \log_2 \left(1 + \frac{p_{q,m}^k g_q^m}{N_0 + I\left(0 < \left|s_q\right| < M\right) \sum_{q' \neq q}^N \sum_{m' \neq m}^N p_{q',m'}^k g_{q'}^{m'}} \right) \tag{6}$$

In Equation (6), N_0 means the noise power. g_n^m refers to the channel gain between the device q and the base station m. $\sum_{n' \neq n}^N \sum_{m' \neq m}^N p_{n',m'}^k g_{n'}^{m'}$ stands for the interference caused by other base stations on the same subchannel to the user n on the base station m. $p_{n,m}^k$ indicates the data transmission power from the subchannel k of the device q to BS. $I(x)$ expresses the Indicator function, wherein, when x is true, it is equal to 1, otherwise it is equal to 0. n_m means the number of orthogonal subchannels of the device q. The FC delay is shown in Equation (7).

$$t_q^f = I\left(1 \le a_n \le M\right) t_{q,m}^f + I\left(a_n = M\right) t_{q,M}^f \tag{7}$$

In Equation (7), $t_{q,m}^f$ expresses the delay required for device q to transfer tasks from SBS to MBS and then unload to the fog end. The energy consumption for FC is shown in Equation (8).

$$e_q^f = I\left(1 \le a_n \le M\right) e_{q,m}^f + I\left(a_n = M\right) e_{q,M}^f \tag{8}$$

In Equation (8), $e_{q,m}^f$ means the energy consumption required for the device q to transfer tasks from SBS to MBS and then unload to the fog end. The total cost P_q^f in the FC model is indicated in Equation (9).

$$P_q^f = \beta_q e_q^f + \alpha_q M_q^f \tag{9}$$

In Equation (9), β_q means the influencing factor of energy consumption; α_q denotes the influencing factor of monetary cost; M_q^f denotes the monetary cost for FC. Similar to the FC model, the total cost of the CC model can be expressed as Equation (10).

$$P_q^f = \beta_q e_q^c + \alpha_q M_q^c \tag{10}$$

In Equation (10), e_q^c expresses the energy consumption of the CC model; M_q^c indicates the monetary cost of the CC model. Based on FC and CC models, a CFCC network resource allocation model can be obtained. Thus, the optimization of the resource allocation model for a heterogeneous CFCC network is completed, and the specific allocation process is shown in Figure 3.

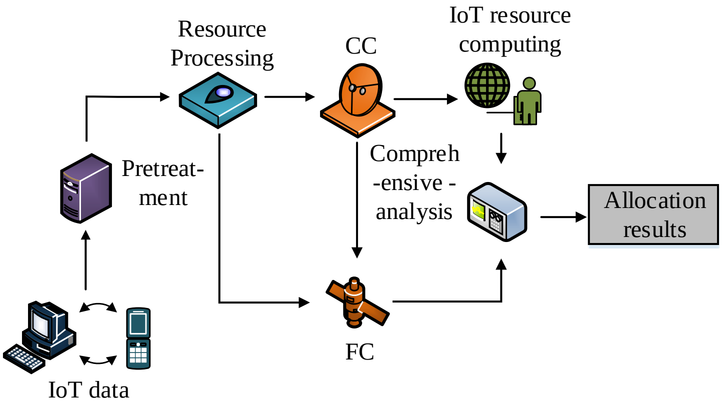


Figure 3. CFCC network resource allocation

3.3. GA optimization algorithm design

Firstly, a CFCC model is proposed, and an optimized allocation scheme for the model is proposed to optimize the resource allocation model of heterogeneous CFCC networks. In addition, GA is used as the core algorithm of the model to reduce the computation while maintaining the performance. GA is a macro biomimetic optimization algorithm that simulates biological evolution, which can retain superior individuals and evolve during the iteration, eliminate inferior individuals, and ultimately obtain the optimal solution [23-24]. In response to the problem that traditional GA cannot solve complex constraint functions, the study introduces a penalty function to reflect the degree of deviation from the constraint conditions, thereby eliminating infeasible solutions. At the same time, non-uniform mutation operators are used to improve the convergence speed of the algorithm. A parameter adaptive improvement method is proposed to address the issues of premature convergence and poor convergence performance of GAs, to improve the convergence performance of GAs. In view of this, further research is conducted to allocate offloading decisions, computing resource allocation, and uplink transmission power of devices in the model, to minimize the system cost of the model. The objective function of the optimization problem can be expressed as Equation (11).

$$\begin{cases} \min\limits_{a_n, f, p} \sum\limits_{m=-M-1}^{M+1} \sum\limits_{q=1}^{Q} I(a_n = m) P_q \\ s.t.\, C1: f_q^0 \geq 0, q \in Q \\ C2: 0 \leq p_{q,m}^t \leq p_q^{\max}, q \in Q \\ C3: f_{q,m}^c \geq 0, q \in Q, m \in M \\ C4: \sum\limits_{m=1}^{M+1} \sum\limits_{q=1}^{Q} f_{q,m}^f \leq F, q \in N, m \in M \\ C5: \sum\limits_{m=-M-1}^{M+1} I(a_n = m) = 1, n \in N, m \in S \\ C6: I(a_n = m)\left(t_q^f + t_q^0 + t_q^c\right) \leq t_q^{\max}, q \in N, m \in S \end{cases} \tag{11}$$

In Equation (11), $C1$ represents the non-negative local computing resources of IoT devices in the IoT system. $p_q^{\max}$ represents the maximum value of the transmission power. $C2$ is the transmission power range of the uplink. $C3$ means the non-negative nature of the computing resources required by the device q to transfer tasks to the cloud. $C4$ stands for the non-negativity of the computing resources required for device q to transfer tasks to the dance fog end. F represents the maximum computing power at the fog end. $C5$ constrains device q to only choose one path to solve tasks. $C6$ constrains that any device in the IoT system must complete within the maximum latency. t_q^0 represents the maximum delay for the device q to complete the calculation task locally. t_q^c represents the maximum delay for device q to migrate computing tasks to the cloud. $t_q^{\max}$ represents the maximum delay for device q to complete the calculation task. By solving Equation (11), it can minimize computational complexity while maintaining the performance of the IoT system.

In solving Equation (11), it is essential to find the optimal solution for the three variables p, a_n, and f, so that the heterogeneous CFCC network resource allocation model can achieve high performance while minimizing system costs. First, it initializes the population, that is, randomly generates a set of solutions. All chromosome individuals in the population are a group of solutions of the objective function shown in Equation (11), and its structure is shown in Figure 4.

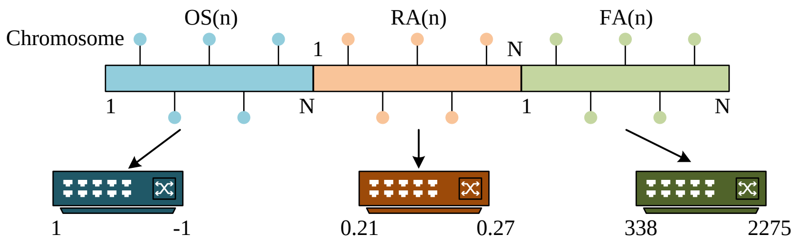


Figure 4. Chromosome structure

In Figure 4, OS (n), RA (n), and FA (n) are obtained by dividing chromosomes into three parts in the population, representing p, a_n, and f, respectively. The above three parts are all composed of N gene sequences, each corresponding to a user. Among them, OS (n) means the task offloading method of user n. When $OS(n)=0$, it indicates that the user n's strategy for processing computing tasks is local computing; when $OS(n)=m, m\in M$ is used, the strategy for user n to handle computing tasks is: SBS transfers the tasks to MBS and then offloads them to the fog end for computation; when $OS(n)=-m, m\in M$, the strategy for processing computing tasks by user n is: SBS transfers the tasks to MBS and then uninstalls them to the cloud for computing; when $OS(n)=M+1$ is used, the strategy for user n to handle computing tasks is to directly unload the task to the fog end through MBS for calculation; when in $OS(n)=-M-1$, the strategy for user n to process computing tasks is to directly offload the tasks to the cloud through MBS for computation. RA (n) and FA (n) respectively represent the uplink transmission power of the user and the computing resources allocated by the system, and their different values have similar meanings to OS (n). In GA, the criterion for determining whether an individual is retained until the next iteration is the fitness value. The study uses the objective function shown in Equation (11) as the fitness function, so the smaller the fitness value, the closer the chromosome individual is to the optimal solution. For a certain chromosome g in the population, its fitness function is shown in Equation (12).

$$Fit(g)=\begin{cases}\varepsilon_{ob} & g=1\\ \varepsilon_{ob}+\gamma\sum_{n=1}^{N}penaly(n,g) & g=-1\end{cases} \tag{12}$$

In Equation (12), ε_{ob} denotes the objective function of the problem to be solved; γ means a penalty factor mainly used to increase the fitness function value of infeasible solutions, thereby better removing infeasible solution individuals from the population; $g=1$ indicates that chromosome individuals are feasible solutions; $g=-1$ expresses that chromosome individuals are infeasible solutions. Through cross-operation, genes from different chromosomes are exchanged to generate new chromosomes and expand population size. The cross-operation is shown in Figure 5.

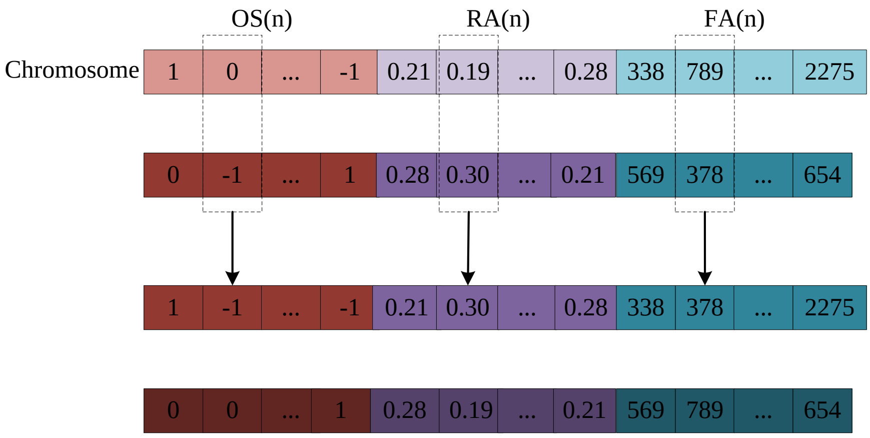


Figure 5. Cross operation

The rate of convergence of the GA is slow, and it is prone to premature, which leads to the unsatisfactory performance of the GA, and the solution result of the objective function may not be the optimal value. Therefore, the study proposes strategies to improve the GA. There are three main parameters that affect the performance of the GA, namely population size s, chromosome crossover probability Pc, and mutation probability Pm. The parameters of traditional GAs are determined manually and do not change during the iteration process. Therefore, they cannot adapt to the dynamic changes of the algorithm, making the performance of GAs unable to achieve optimal results. A parameter adaptive improvement strategy is proposed to dynamically change the parameters of the GA, thereby ensuring its performance. The adaptive parameter improvement strategy is shown in Equation (13).

$$P=\begin{cases}(P_{\max}-P_{\min})\ln\left(A\cdot f_i(g)+B\right) & f_i(g)\le f_{average}\\ P_{\max} & f_i(g)>f_{average}\end{cases} \tag{13}$$

In Equation (13), f expresses the fitness value; P refers to the probability of crossover or mutation; A and B are two parameters that change with changes in population fitness values. In the GA, due to the limited population size, after multiple iterations, most excellent individuals are retained in the population, resulting in a too-single population and affecting the algorithm's optimization performance. In response to this deficiency, a mutation strategy based on the standard deviation of individual fitness values is proposed, which enables individuals in the GA population to change their mutation probability based on the population fitness value, thereby increasing the diversity of the population. The standard deviation (σ) of individual fitness values is shown in Equation (14).

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left| f_i(g) - f_{average}(g)\right|^2} \tag{14}$$

When σ is less than the set threshold, it indicates that the population is relatively single. Therefore, increasing the mutation probability Pm makes it 2 Pm, thereby improving population diversity and avoiding local optimization. Using the IGA algorithm to solve the problem of cloud-fog collaborative resource allocation in heterogeneous environments, when parameters change, adaptive changes in crossover probability and mutation probability can effectively retain excellent individuals, accelerate the elimination of inferior individuals, and thus optimize resource allocation.

4. CFCC network model testing

A CFCC network model for heterogeneous environments was proposed, and to improve the resource allocation ability of the computing network, an IGA was introduced to enhance the classification ability of the model. To verify the effectiveness and feasibility of the network model, this study first evaluated the relevant performance of the CFCC network model and then analyzed the application effect of the model in heterogeneous data processing.

4.1. Performance evaluation of the CFCC network model

The study was conducted on the Inter Core i7 7700 central processing unit, NVIDIA GeForce RTX 3080 Ti graphics card with 32GB of running memory. Simulation experiments were conducted using MATLAB 2022b in the Windows 10 system environment. The crossover probability of traditional GA was set to 0.8 and the mutation probability was set to 0.2. The crossover probability range of IGA was [0.25, 0.8], the mutation probability range was [0, 0.2], and the standard deviation range of individual fitness values in the population was [0, 1]. It assumed that there was a macro base station and a small base station in the system, and the service area of the small base station was covered by the service area of the macro station, and there were multiple IoT devices randomly deployed in the area. The initial population size of IGA was set to s, the number of IoT devices was set to N, the number of iterations was set to I, and the length of a chromosome was set to 3×N. The simulation parameters of the CFCC model are shown in Table 1. Meanwhile, parameters with fixed values in the CFCC model are shown in Table 1.

Table 1 Simulation parameters of cloud-fog collaborative computing model

Simulation parameters	Value
Node computing power (GHz)	4000
Delay in transmission time of unit data (s/KB)	0.0015
Maximum processing latency	5
Input data size (KB)	CN: (1000, 50)
The computing resources required for the task (KB)	CN: (1000, 50)
The computing power of the device (KB)	CN: (1000, 50)
Noise	CN: (5, 1)

This study introduced Schaffer, Sphere, Griewangk, Rastigrin, and Ackerly functions to evaluate the performance of the IGA. The mobile edge computing system consisted of M base stations and n users. All nodes had only one antenna. The distance between adjacent base stations was 200 m. Each base station served a circular area with a radius of 150 m. Firstly, the change in loss value was analyzed. The results are shown in Figure 6.

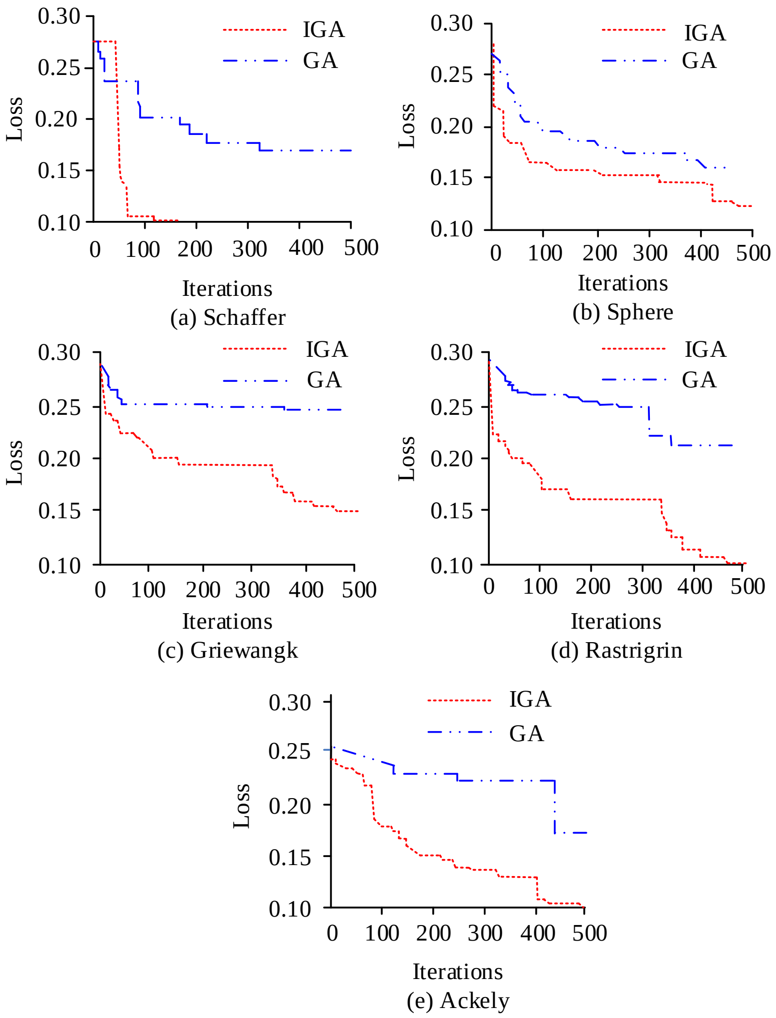


Figure 6. Algorithm loss value change

From Figure 6, the traditional GA had fewer iterations than the IGA in the Sphere, Griewangk, Rastigrin, and Ackerly functions tests, but its convergence accuracy was significantly lower than the IGA. Figures 6(b) and 6(c) show the iteration diagrams of the algorithm on the Sphere and Griewangk functions, respectively. From these two figures, after the objective function stabilized, neither algorithm converged to the optimal value and was in the exploratory stage. Figures 6(a), 6(d), and 6(e) show the iteration diagrams of the algorithm on the Schaffer, Rastigrin, and Ackerly functions, respectively. In these three figures, the traditional GA did not obtain the optimal solution after the iteration, while the IGA found the optimal solution. The IGA had higher convergence accuracy and better optimization effects compared to simple GAs. Overall, IGA showed better performance in the testing of these functions, with higher convergence accuracy and optimization effects. Secondly, using heterogeneous data as the test object, it evaluated the complexity results of traditional and improved algorithms, as shown in Figure 7.

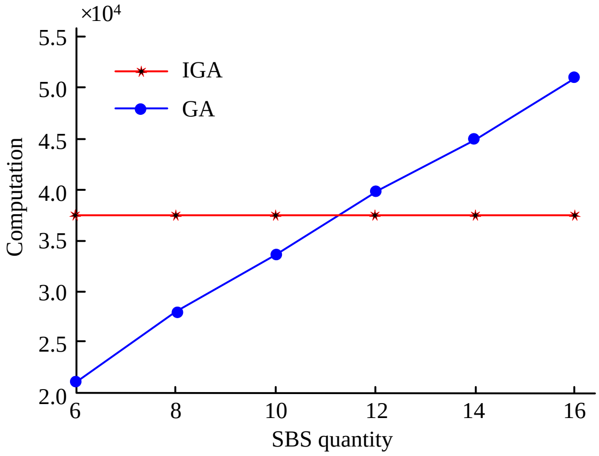


Figure 7. Algorithm complexity of two algorithms

From Figure 7, as the amount of small base stations continued to increase, the complexity of traditional GAs continued to increase, ultimately breaking through 5.0 $\times 10^4$, while the complexity of the IGA remained stable for a long time and continued to remain stable at 3.7 $\times 10^4$ around. Therefore, compared to traditional GAs, the IGA had a higher degree of complexity in the calculation and had higher computational efficiency. This means that when dealing with large-scale problems, IGA can perform calculations more efficiently and provide better computational results while maintaining stable complexity. Secondly, the study compared and analyzed the average levels of traditional GAs, IGAs, and current CFCC models, as shown in Figure 8.

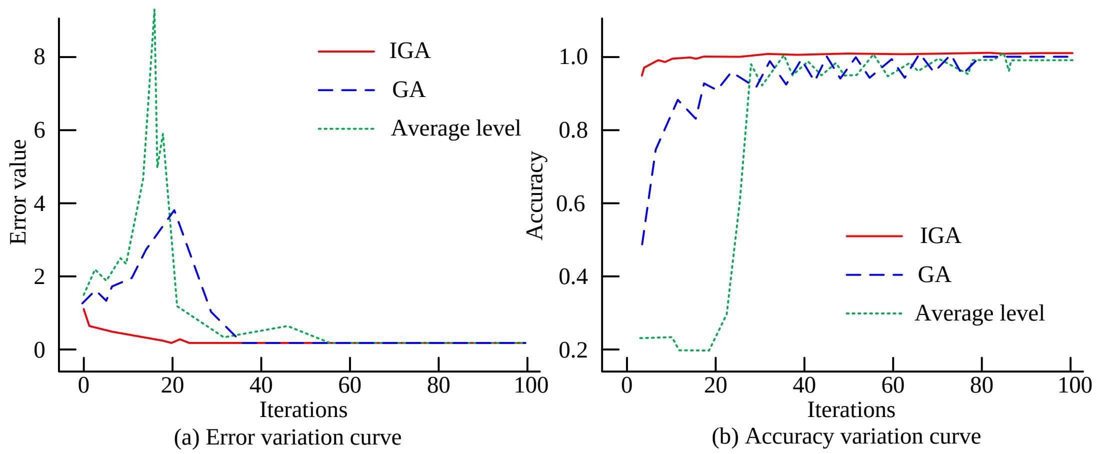


Figure 8. Loss and accuracy results of different models

Figure 8(a) shows the loss results of different models. The IGA model could reach the target loss value after 22 iterations, while the GA model needed 38 iterations to begin convergence. Figure 8(b) shows the comparison of accuracy results between different models. Accuracy is an indicator used to evaluate classification models, which is the proportion of the total number of correct predictions made by the model. The IGA model had the highest classification accuracy of

0.996, while the GA model had a stable accuracy value of 0.991, and it only began to stabilize after 80 iterations. At the same time, from the above research results, the IGA loss value and accuracy proposed in the study were significantly higher than the average level of the current CFCC model. These results indicated that the IGA model achieved the target loss value with fewer iterations and had higher classification accuracy. Therefore, the IGA model had higher efficiency and accuracy in solving problems and could be widely applied to improve and optimize the CFCC model. In addition, to further validate the performance of the algorithm proposed in the study, multiple algorithm comparison experiments were conducted to evaluate the performance of the IGA by comparing it with other algorithms. Firstly, the calculation time and error rate of each algorithm were analyzed. The results are shown in Figure 9.

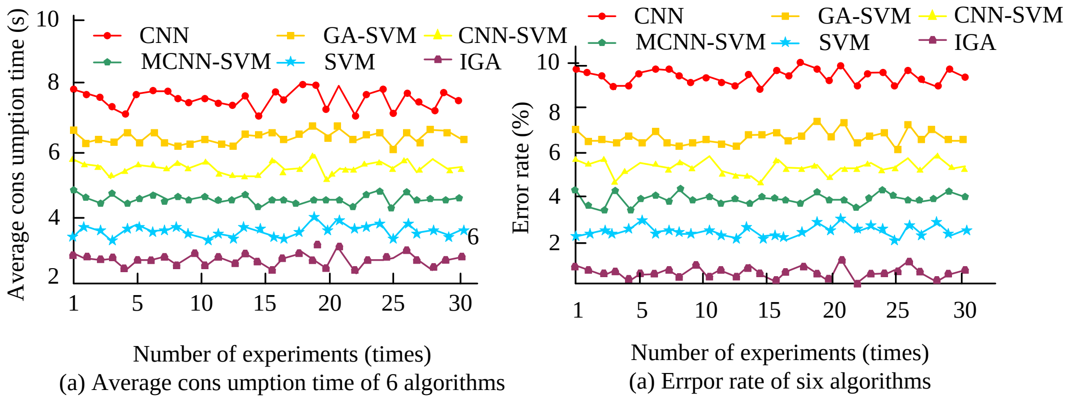


Figure 9. Comparison of calculation time and error rate of different algorithm models

Figures 9(a) and 9(b) show the comparison results of the calculation time and the recognition error rates of each algorithm, respectively. The IGA took less computational time compared to other algorithms and could stay within 3 seconds for a long time, while the maximum computational time of the other algorithms reached 8.1 seconds. In the comparison of error rates among various algorithms, it was found that the proposed IGA had a minimum error rate of 0% and a maximum value of only 1.3%, which was significantly lower than other algorithms. In summary, the IGA has a shorter computational time and a lower error rate in a shorter period of time.

4.2. Application effect of CFCC network model

After completing the performance analysis of the CFCC network model, it was necessary to conduct application testing on the network model to verify the feasibility of this study. The experiment calculated the network model and the improved network model architectures in this study, and the time required to complete 200 data processing was from reading the data to completing the data processing. The experimental results are indicated in Figure 10.

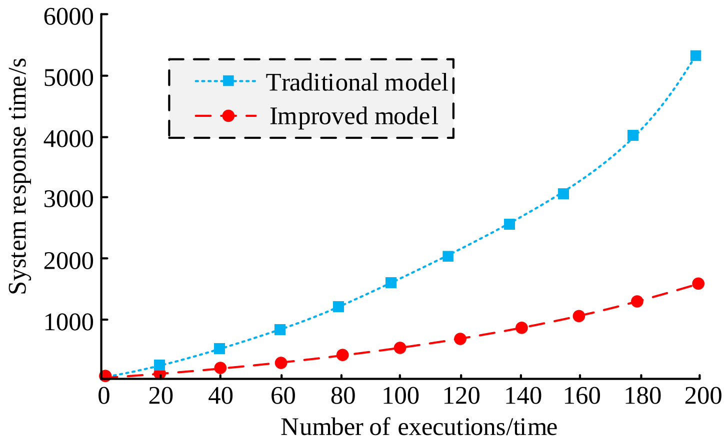


Figure 10. Comparison of the time required for data processing

From Figure 10, there was a significant difference in the time required for processing data between the traditional and the improved network models in this study. Under the two sets of network model architectures, the system response time showed a gradual upward trend as the number of executions increased. However, the traditional network model architecture had a greater increase in system response time, while the network model architecture proposed in this study had a smaller increase in system response time. When the number of executions was 20, the system response time of the traditional network model architecture was 141 seconds. The improved network model architecture in this study had a system response time of 92 seconds, which reduced 34.75%; when the number of executions reached 200, the system response time of the traditional network model architecture was 5874 seconds, while the system response time of the improved network model architecture in this study was 1634 seconds, reducing by 72.18%. It can be observed that the system response time of the improved IoT architecture was significantly lower than that of the traditional IoT architecture, indicating that its operating speed was faster. Meanwhile, as the amount of data processing increased, the processing time gradually slowed down, which may be due to the large amount of data occupying too much system memory, thereby reducing processing speed. However, the proposed improved IoT architecture was less affected, further proving its good performance. It calculated the system response time for two sets of architectures with different execution times, as shown in Table 2.

Table 2. System response times of different number of executions for two sets of architectures

The number of execution	System response time		
	Traditional IoT architecture (s)	Improved IoT architecture (s)	D-value (s)
20	141	92	49
40	561	123	438
60	924	242	682
80	1217	462	755
100	1721	512	1209
120	2012	721	1291
140	2484	877	1607

160	2912	1023	1889
180	4011	1348	2663
200	5874	1634	4240

As shown in Table 2, the system response time of the network model proposed by the research was lower than that of the traditional network model architecture between 20 and 200 executions, which verified the effectiveness of this study. Secondly, it evaluated the difference in calculation standard deviation between two collaborative computing network models in different heterogeneous environments, as shown in Figure 11.

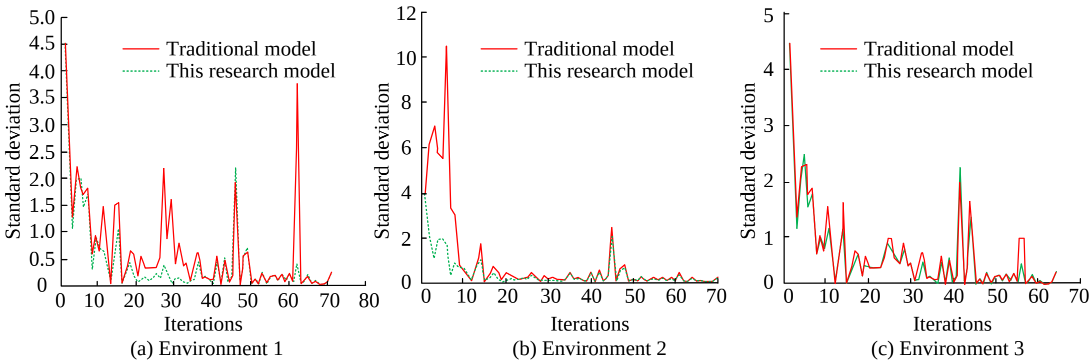
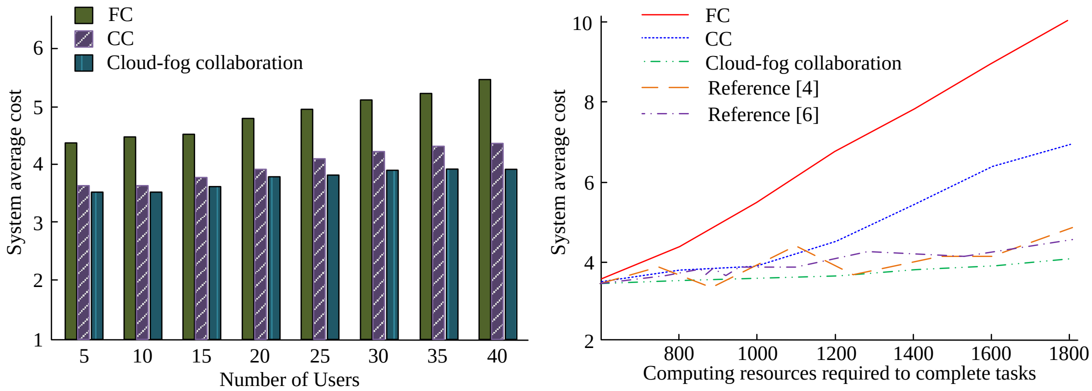


Figure 11. Analysis of model application in different heterogeneous environments

From Figure 11, for different heterogeneous environments, the standard deviation of the network model proposed by the research was significantly lower than that of traditional network models. In multiple heterogeneous environments, the minimum standard deviation of the research model could be reduced to 0, and the maximum standard deviation was only 4.4. To avoid the shortcomings of high latency and transmission energy consumption in CC, as well as relatively weak computing and data storage capabilities in FC, a heterogeneous CFCC network resource allocation model was studied and constructed, enabling CC and FC to work together and compensate for each other's shortcomings, to improve the performance of the IoT system. To verify the application effect of the model, the study first set up a system with one MBS and one SBS, as well as multiple IoT devices deployed according to random principles, and compared the system costs of CC, FC and the proposed heterogeneous CFCC network resource allocation models. Simultaneously, it calculated the computational resources required for completing tasks using CC, FC, the designed CFCC network resource allocation model, reference [4] model, and reference [6] model, as shown in Figure 12.



(a)Number of Users (b)Computing resources required to complete tasks

Figure 12. System cost of heterogeneous CFCC network resource allocation

In Figure 12(a), the higher the number of users in the IoT system, the higher the system cost required to complete computing tasks. However, the system cost improvement of the heterogeneous CFCC network resource allocation model proposed in the study was smaller. When the number of users was 5, the system cost of the heterogeneous CFCC network resource allocation model was 3.3; the system costs of the FC model and the CC model were 4.6 and 3.8, respectively, which were 1.3 and 0.2 higher than those of the heterogeneous CFCC network resource allocation model. When the number of users in the IoT system reached 40, the system cost of the heterogeneous CFCC network resource allocation model was 3.7; the system cost of the FC model was 5.3, which was 1.6 higher than the heterogeneous CFCC network resource allocation model; the system cost of the CC model was 4.2, which was 0.5 higher than the heterogeneous CFCC network resource allocation model. In Figure 12(b), the more computing resources required to complete computing tasks in the IoT system, the higher the system cost. However, the curve of the resource allocation model for heterogeneous CFCC networks was very flat, indicating that the increase in computing resources required for computing tasks had a relatively small impact on their system to verify the effectiveness of IGA in improving the performance of IoT systems. The superiority of IGA was validated using the average system cost as the fitness value. Different population sizes were taken and compared with traditional GA algorithms. The results are shown in Figure 13.

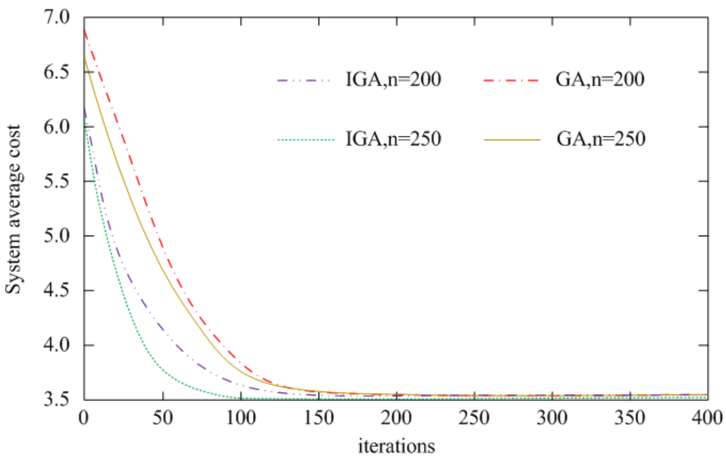


Figure 13. Convergence performance of IGA and GA

From Figure 13, when the population size was 200, the system cost corresponding to the stabilization of IGA was about 3.5, and the system cost corresponding to the stabilization of GA was about 3.7. When the population size was 250, the system cost when the convergence curves of IGA and GA reach stability was consistent with that when the population size was 200. It can be observed that as the population size increased, both GA and IGA had more opportunities to find excellent solutions, so their convergence speed accelerated and system costs correspondingly decreased. In addition, when the population size reached a certain value, the impact of further increasing the population size on convergence speed and system cost gradually decreased. Therefore, in practical applications, selecting an appropriate population size was the key to reducing system costs and improving algorithm efficiency. Meanwhile, the convergence speed of IGA was significantly lower than that of GA, indicating its feasibility in improving the performance of IoT systems.

5. Discussion

With the explosive growth of information generated by IoT devices, CC has always been favored in processing massive amounts of data due to its powerful computing power and on-demand charging characteristics. However, the high latency brought by CC and the inability to provide real-time and mobile support are not desirable for networks in mobile scenarios such as smart transportation. Therefore, FC was introduced into computationally intensive applications at the edge of the network. Therefore, a solution method for heterogeneous network cloud-fog collaborative resource allocation problem based on IGA was proposed in the study. The experimental results showed that the designed CFCC model only increased the system cost by 12.1% when the number of users increased from 5 to 40, indicating that the resource allocation strategy designed in the study has high cost-effectiveness and scalability. Compared with the research results of J Li et al. [11], the results of this article are significantly better . This is due to J Li's cloud-fog IoT architecture only improved data transmission speed, without addressing the cost impact of expansion. Meanwhile, the accuracy of the design method in this article reached 0.996, with an average calculation time of only 3 seconds, proving its high computational efficiency and accuracy. N I. Osman et al. obtained similar results [12]. This may be because this article is related to N I. Osman et al. have optimized GA, thereby improving the accuracy and computational speed of the algorithm. In summary, the proposed method has significantly improved the performance of IoT systems and provided strong support for the development of CFCC networks.

6. Conclusion

In the current information age, traditional CC and FC can no longer meet the rapidly growing data processing needs and faces problems such as high latency, bandwidth bottlenecks, and data security. To solve the problem of resource allocation and its solution in the IoT, a CFCC network resource allocation model for heterogeneous environments was studied and constructed, and a parameter adaptive improvement strategy was introduced to optimize the traditional GA. The IGA was used to solve the resource allocation problem. The results showed that in the calculation of time and error, the minimum error of IGA was 0%, the maximum error was 1.3%, and the average calculation time of IGA was about 3 seconds, which was significantly lower than other algorithms, indicating an increase in its accuracy and calculation speed. In the calculation of system response time, when the execution volume was 20 times, the response time of the traditional model was 141 seconds, while the designed model response time was 92 seconds, reducing 34.75%. When the execution volume was 200 times, the response time of the traditional model was 5874 seconds, and the designed model response time was 1634 seconds, reducing 72.18% compared to the traditional model, indicating that its running speed was relatively high. In the standard deviation calculation, the minimum standard deviation of the designed model was 0, and the maximum was only 4.4, proving that the designed model had high accuracy. The above results demonstrate that the designed CFCC network model has good performance and lower system costs for resource allocation. However, the research did not test the actual IoT environment in the application analysis. Therefore, in future research, the experimental environment will be expanded to provide theoretical support for the implementation and use of the proposed network model.

Declarations

Availability of data and material

The data will be made available on request.

Competing interests

There is no conflict of interest in this paper.

Funding

The research is supported by: A Study on the Teaching Reform of Vocational Colleges in Hunan Province, Research on the training mode of innovative and entrepreneurial talents of computer application technology in the era of "Internet +" (No. ZJGB2016110).

Authors' contributions

This paper was completed from the first draft to the final draft by Hongqing Liu.

Acknowledgment

A Study on the Teaching Reform of Vocational Colleges in Hunan Province, Research on the training mode of innovative and entrepreneurial talents of computer application technology in the era of "Internet +" (No. ZJGB2016110).

Reference

[1] D. S. K. D. Kumar, "Optimal workflow scheduling in cloud computing based on hybrid bacterial evolutionary and bees mating optimization algorithm," Turk. J. Comput. Math. Educ. (TURCOMAT), vol. 12, no. 3, pp. 4762-4775, Apr. 2021, 10.17762/turcomat.v12i3.1939.

[2] P. Neelima, and A. R. M. Reddy, "An efficient load balancing system using adaptive dragonfly algorithm in cloud computing," Cluster Comput., vol. 23, no. 4, pp. 2891-2899, Dec. 2020, 10.1007/s10586-020-03054-w.

[3] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues," IEEE Commun. Surv. Tutorials, vol. 22, no. 2, pp. 1191-1221, Second quarter. 2020, 10.1109/COMST.2019.2962586.

[4] J. Bisht, and V. S. Vampugani, "Load and cost-aware min-min workflow scheduling algorithm for heterogeneous resources in fog, cloud, and edge scenarios," Int. J. Cloud App. Comput., vol. 12, no. 1, pp. 1-20, Sep. 2022, 10.4018/IJCAC.2022010105.

[5] J. Okwuibe, J. Haavisto, I. Kovacevic, E. Harjula, I. Ahmad, J. Islam, and M. Ylianttila, "SDN-enabled resource orchestration for industrial IoT in collaborative edge-cloud networks," IEEE Access, vol. 9, pp. 115839-115854, Aug. 2021, 10.1109/ACCESS.2021.3105944.

[6] S. R. Hassan, I. Ahmad, J. Nebhen, A. U. Rehman, M. Shafiq, and J. G. Choi, "Design of

latency-aware IoT modules in heterogeneous fog-cloud computing networks," Comput., Mater. Contin., vol. 70, no. 3, pp. 6057-6072, Oct. 2022, 10.32604/cmc.2022.020428.

[7] N. I. Osman, M. A. S. Eisa, and S. M. Imam, "Power-delay trade-offfor optimum data storage in a cloud-fog-mist architecture," ECTI-CIT Trans., vol. 17, no. 2, pp. 153-167, Apr. 2023, 10.37936/ecti-cit.2023172.247297.

[8] S. S. Tripathy, D. S. Roy, and R. K. Barik, "M2FBalancer: a mist-assisted fog computing-based load balancing strategy for smart cities," J. Ambient Intell. Smart Environ., vol. 13, no. 3, pp. 219-233, May. 2021, 10.3233/AIS-210598.

[9] P. K. Sharma, and J. H. Park, "Blockchain-based secure mist computing network architecture for intelligent transportation systems," IEEE Trans. Intell. Transp. Syst., vol. 22, no. 8, pp. 5168-5177, Aug. 2021, 10.1109/TITS.2020.3040989.

[10] G. P. Liu, "Coordinated control of networked nonlinear multiagent systems using variable horizon learning predictors via cloud edge computing," IEEE Trans. Contr. Netw. Syst., vol. 9, no. 4, pp. 1975-7986, Jun. 2022, 10.1109/TCNS.2022.3181549.

[11] J. Li, X. Li, J. Yuan, and G. Li, "Load balanced data transmission strategy based on cloud-edge-end collaboration in the internet of things," Sustainability, vol. 14, no. 15, pp. 9602-9623, Aug. 2022, 10.3390/su14159602.

[12] F. Dharma, S. Shabrina, A. Noviana, M. Tahir, N. Hendrastuty, and W. Wahyono, "Prediction of indonesian inflation rate using regression model based on genetic algorithms," J. Online Informatika, vol. 5, no. 1, pp. 45-52, Jul. 2020, 10.15575/join.v5i1.532.

[13] S. Nikbakht, C. Anitescu, and T. Rabczuk, "Optimizing the neural network hyperparameters utilizing genetic algorithm," J. Zhejiang Univ.-Science A., vol. 22, no. 6, pp. 407-426, Jun. 2021, 10.1631/jzus.A2000384.

[14] R. Pal, S. Yadav, R. Karnwal, and Aarti, "EEWC: energy-efficient weighted clustering method based on genetic algorithm for HWSNs," Complex Intell. Syst., vol. 6, no. 2, pp. 391-400, Jul. 2020, 10.1007/s40747-020-00137-4.

[15] I. Z. Yakubu, M. Murali. An efficient meta-heuristic resource allocation with load balancing in IoT-Fog-cloud computing environment. J Amb Intel Hum Comp, vol. 14, no. 3, pp. 2981-2992, Feb. 2023, DOI: 10.1007/s12652-023-04544-6.

[16] S. Ghosh, D. De. TARA: weighted majority cooperative game theory-based task assignment and resource allocation in 5G heterogeneous fog network for IoT. J Supercomput, vol. 79, no. 13, pp. 14633-14683, Jul, 2023, DOI: 10.1007/s11227-023-05228-w.

[17] Y. Guo, Z. Mustafaoglu, and D. Koundal, "Spam detection using bidirectional transformers and machine learning classifier algorithms," J. Comput. Cognitive Eng., vol. 2, no. 1, pp. 5-9, Apr. 2022, 10.47852/bonviewJCCE2202192.

[18] W. Li, Q. Li, L. Chen, F. Wu, and J. Ren, "A storage resource collaboration model among edge nodes in edge federation service," IEEE Trans. Veh. Technol., vol. 71, no. 9, pp. 9212-9224, Sept. 2022, 10.1109/TVT.2022.3179363.

[19] U. Arora, and N. Singh, "IoT application modules placement in heterogeneous fog-cloud infrastructure," Int. J. Inf. Tech. Decis., vol. 13, no. 5, pp. 1975-1982, May. 2021, 10.1007/s41870-021-00672-4.

[20] K. J. Naik, "A cloud-fog computing system for classification and scheduling the information-centric IoT applications," Int. J. Commun. Netw. Distrib. Syst., vol. 27, no. 4, pp. 388-423, Nov. 2021, 10.1504/IJCNDS.2021.119208.

[21] W. Z. Zhang, I. A. Elgendy, M. Hammad, A. M. Iliyasu, X. Du, M. Guizani, A. A. Abd El-Latif, "Secure and optimized load balancing for multitier IoT and edge-cloud computing systems," IEEE Internet Things, vol. 8, no. 10, pp. 8119-8132, Dec. 2020, 10.1109/JIOT.2020.3042433.

[22] Chen X, Cheng L, Liu C, Q. Liu, J. Liu, Y. Mao, and J. Murphy, "A WOA-based optimization approach for task scheduling in cloud computing systems," IEEE Syst. J., vol. 14, no. 3, pp. 3117-3128, Jan. 2020, 10.1109/JSYST.2019.2960088.

[23] B. Sadhana, R. K. Tata, P. K. Chandrika, M. S. Mekala, N. Srinivasu, and G. P. S. Varma, "Resource integrity-aware flexible resource scaling approach over sensor-cloud," Int. J. Powertrains, vol. 10, no. 2, pp. 175-187, Aug. 2021, 10.1504/IJPT.2021.117463.

[24] J. Zan, "Research on robot path perception and optimization technology based on whale optimization algorithm," J. Comput. Cogn. Eng., vol. 1, no. 4, pp. 201-208, Mar. 2022, 10.47852/bonviewJCCE597820205514.