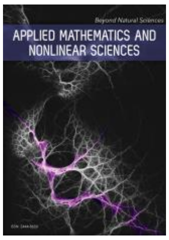
sciendo Applied Mathematics and Nonlinear Sciences, 10(1) (2025) 1-21



Applied Mathematics and Nonlinear Sciences

https://www.sciendo.com

# Distributed Database Optimization Techniques Combining Computer Network and Algorithm Design

Lihua Pan[1,†], Jin Li[2]

1. College of Information Engineering, Chenzhou Vocational Technical College, Chenzhou, Hunan, 423000, China.
2. Community Education College of Chenzhou Open University, Chenzhou, Hunan, 423000, China.

**Submission Info**

Communicated by Z. Sabir
Received October 13, 2024
Accepted February 9, 2025
Available online March 21, 2025

## Abstract

With the development of computer network technology, distributed database has become a current research hotspot. Based on the structural characteristics of distributed database systems, the article leads to the optimization of distributed database queries at the global optimization level. Then, according to the basic principle of genetic algorithms, combined with the characteristics of the biological immune system, an improved immune genetic algorithm is proposed. The improved immunogenetic algorithm is applied to the database multi-connection query optimization technology, and the distributed database multi-connection query optimization algorithm based on the improved immunogenetic algorithm is designed. In the simulation experiments, a set of optimal parameter values applicable to the system is obtained through continuous experiments, and the distributed multi-connection query is optimized with this set of parameter values, which achieves the expected optimization effect. The final experimental results show that the improved optimization algorithm has a significant improvement in terms of query cost compared to the base algorithm in dealing with distributed database query problems. Meanwhile, under the same conditions, the basic algorithm is used to test and compare the communication cost, mean and standard deviation of the optimal solutions obtained by the two algorithms, and it is concluded that the optimization algorithm in this paper can obtain better solutions and better stability.

**Keywords:** Distributed database; Immunogenetic algorithm; Genetic algorithm; Join query; Computer network.
**AMS 2010 codes:** 05C82

†Corresponding author.
Email address: czzyplh@163.com

ISSN 2444-8656
https://doi.org/10.2478/amns-2025-0611

## 1 Introduction

With the rapid development of Internet technology and the rise of the smart city concept, the amount of information data has shown unprecedented exponential growth in production, life and work [1-2]. In particular, the rapid development of social networks and video sharing platforms, as well as the widespread deployment of urban video surveillance networks, have made the shape of data far beyond the scope of simple textual information [3-4]. Traditional relational databases, structured to manage data in the form of rows and columns, predefine the structure of the table and deposit data according to the individual fields of the table [5-6]. Modern data, on the other hand, not only includes traditional text, but also covers images, videos, audios, and diverse data from various sensor networks, which are usually categorized as unstructured data, and can be transformed into more structured vector data through embedding techniques or other transformations, which can then achieve efficient similarity retrieval of images, audios, and other contents [7-9].

In today's data processing industry, the rapid growth and diversity of data puts higher requirements on distributed database systems, making them an indispensable part. In this scenario, vector indexing techniques have been emphasized for their significant role in improving the efficiency of data retrieval, especially in the areas of image search, natural language processing and recommender systems [10-13]. Integrating this technique into distributed database systems not only improves the speed of querying, but also speeds up the process of data retrieval and analysis, which is extremely crucial for handling huge datasets and meeting the demands of real-time applications. The applications of vector data in various fields such as similarity search, recommender systems, natural language understanding and computer vision demonstrate its wide impact and important role in the current data processing field [14-16]. Therefore, exploring and optimizing vector indexing in distributed databases not only has a wide range of application potentials, but also is of key significance in realizing the efficient operation of vector indexing in database systems, which provides a strong technical support to cope with the complex challenges faced in the current data processing field [17-18].

Distributed database systems can be obtained by using centralized database technology as a base and then combining it with computer networks. The difference between the data in a distributed database and a centralized database is that the data is stored in a decentralized manner in different sites of the network, and the databases in all sites have the ability to be processed independently [19-20]. And each site needs to be involved in the execution of the global application, which utilizes the results of the existing network topology for the purpose of communication and access to the data dispersed in each site [21-22]. However, due to the actual application and operation links, the distributed network will not be felt, but the operation does belong to the whole database system, so it leads to the fact that although the distributed database will be physically dispersed in each site, but logically it still belongs to the same database system's dataset, and this leads to a certain degree of complexity in terms of query processing [23-25].

Therefore, exploring and optimizing vector indexing in distributed databases not only has a wide range of application potentials, but also is of key significance in realizing the efficient operation of vector indexing in database systems, which provides a strong technical support to cope with the complex challenges faced in the current data processing field [26-27].

The article firstly gives a brief introduction to the definition of distributed database system and its structural characteristics, then analyzes the cost estimation of system query optimization on the basis of distributed data query optimization objective, and then elaborates the distributed database query optimization process under the global optimization level from the perspectives of query search space and query search strategy. Then the bio-immune property is combined with a genetic algorithm to propose a new distributed database connection query optimization algorithm based on an improved immunogenetic algorithm. In the experimental tests, the algorithm proposed in this paper is applied in a computer network system, and it is analyzed and compared in simulation experiments. In a random network topology with 30 nodes and 58 nodes, the immunogenetic algorithm and the basic algorithm are also used to find the optimal path for querying, compare the convergence curve, the final communication cost, and the mean and standard deviation of the same number of times of running, so as to verify the superiority of the algorithm proposed in this paper.

## 2 Distributed Database Query Optimization Research

### 2.1 Distributed database systems

### 2.1.1 Definition of a distributed database system

Distributed database system is usually a number of smaller and independent computer systems distributed in a computer network on a number of sites, each computer system is a relatively independent of the complete whole, which can be regarded as a centralized database, with its own independent database, database management system and the underlying hardware and software, etc., and are able to independently complete the work of the local site, the various sites of the computer system through the computer network connected to form a logical whole similar to the centralized database operation. Computer systems at each site are connected through a computer network to form a logical whole similar to the operation of a centralized database.

### 2.1.2 Distributed database system structure and features

Distributed database systems have the following characteristics:

1) Site autonomy

   Distributed database system, each site has a local data management system, CPU, memory, hard disk and other equipment, in addition to storing the corresponding data, there are independent handling of the corresponding transactions and management of its site data capabilities.

2) Data distribution

   Distributed database data distribution is mainly reflected in the various functional components and their data are scattered and stored in different sites, these nodes are connected by the computer network, the master node is responsible for the management and control. Users in the process of operating the distributed database, can not see the data specifically stored in which subsystems, how the internal data distribution, but also not clear which operating system is used at each site of the database. This physical distribution of data decentralized users cannot feel the characteristics of the data shared through the network, which makes the user experience and operation of centralized databases no different.

3) Logical integrity of data

   According to user needs and developers' practical considerations, the data in distributed database are scattered and stored in different network sites, in which each site is similar to the subsystem of centralized distributed database, which has its own local data management system, CPU, memory, hard disk and other equipments, and it is a relatively independent database system, which is not only able to carry out the self-management and maintenance of subsystems, but also able to It is a database system with relative independence, not only capable of self-management and maintenance of sub-systems, but also capable of handling data transmission between sites. In order to ensure the consistency and integrity of data, the database management system carries out unified management of subsystems distributed on each site through network communication [28]. The distributed database system network structure can be seen in Fig. 1.

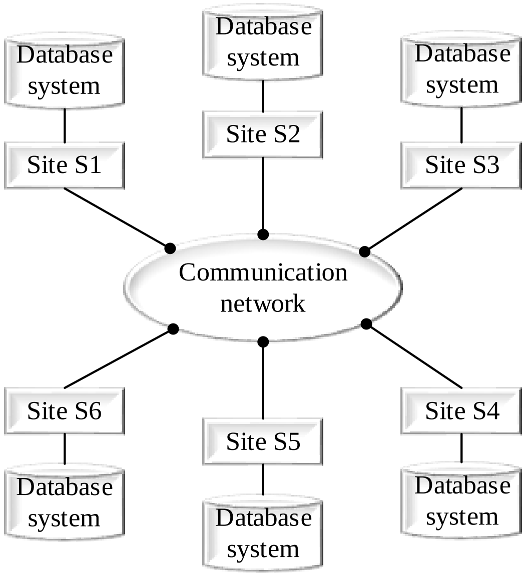


**Figure 1.** Network structure of the distributed database system

4) Data redundancy

Data redundancy is unique to distributed databases, centralized databases do not have, because distributed databases are based on the requirements of the data stored in different sites in the network, and through the computer network connected to each site to achieve unified control, so the redundancy of the data is critical to distributed databases need to focus on consideration. Distributed database data redundancy enhances the reliability of the distributed database to a certain extent, when a network site has a problem that causes the system can not run normally, the distributed database system through the network communication can be accessed from other sites to obtain the relevant data to ensure the normal operation of the entire system. When the user accesses the data, the distributed database system can effectively select data copies according to the query strategy, improving data access efficiency. The network architecture of the distributed database system is shown in Figure 2.

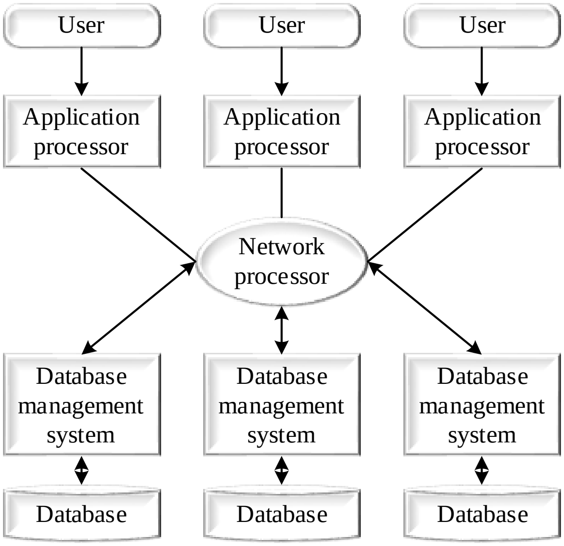


**Figure 2.** Network architecture of the distributed database system

## 2.2 Distributed Database Query Optimization Analysis

### 2.2.1 Distributed Query Optimization Objectives

In a distributed database system, since the databases on each network site store different data and a query operation generally involves data transfer across sites, the efficiency of a distributed database query optimization can generally be measured from two aspects for distributed databases. On the one hand, it can be measured from the query cost generated by the query execution plan, i.e., the communication cost and site I/O cost and CPU cost generated by the inter-site data transfer of the query execution strategy can be considered. On the other hand, it can be considered from the length of the response time of the user query request, that is, the user issued a query request to the completion of the query plan to return the query results generated by the time, in general, the shorter the query response time, the higher the query efficiency, this kind of query evaluation criteria has a very important practical needs of the significance of the user is generally also extremely concerned about.

### 2.2.2 Distributed Query Optimization Cost Estimation

In distributed databases, the communication transmission cost between sites is generally high, and to minimize the total query cost and minimize the total response time, it is necessary to focus on the communication cost.

In distributed database query operations, the query cost that affects the query response time is usually:

$$C_{\mathrm{Total}} = C_{CPU} + C_{I/0} + C_{\mathrm{Message}} \tag{1}$$

The communication cost of inter-site data transfer is usually:

$$C_{\mathrm{Message}} = C_{\mathrm{Inception}} + R * X \tag{2}$$

where $C_{Total}$ is the query cost and value that affects the timeliness of the distributed database query response. $C_{I/O}$ is the cost incurred by the input and output of the device at the site: $C_{CPU}$ is the surrogate value incurred by the database memory at the site. $C_{\text{Message}}$ is the communication cost incurred by inter-site data transfer in a computer network. $R$ is the inter-site data transfer rate, i.e., the time taken to transfer one unit of data in seconds per bit, generally related to the site hardware and communication network [29]. $X$ is the amount of data transferred for site communication in bits. $C_{\text{Inception}}$ is the cost of time required to initialize the site data communication once, the value of which is generally related to the site hardware equipment in seconds.

### 2.2.3 Distributed Query Optimization Hierarchy

In the distributed database data query process, a query optimization process often includes four parts: data query decomposition, data localization, data global optimization and data local optimization. The distributed database query optimization structure is shown in Figure 3.

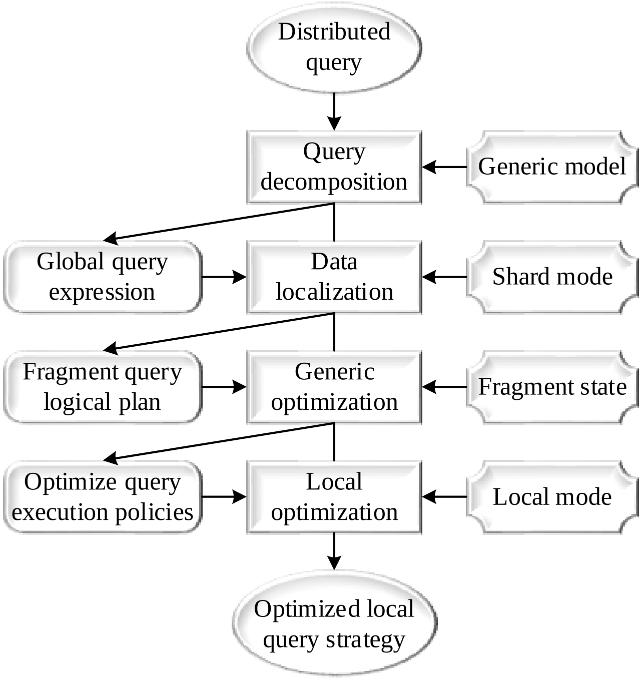


**Figure 3.** Distributed database query optimization layer structure

### 2.2.4 Distributed Query Optimization Process

The cost model for data query optimization is set according to user requirements and network conditions. With the set cost model, quantitative query computation can be carried out on all solution sets in the search space as a way to provide data basis for selecting a better query execution plan. The search strategy of the query execution process determines the final order of the query execution plan based on the search space, and specifies the data processing strategy, so as to achieve the purpose of reducing the cost of query execution and shortening the query response time. The distributed database query optimization process is shown in Figure 4.

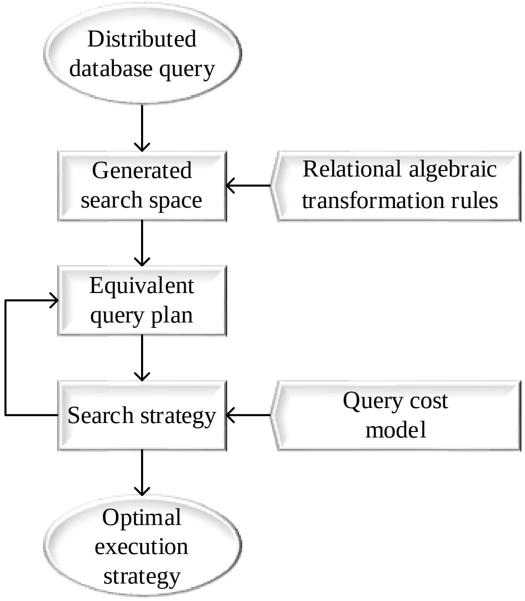

Distributed database query
Generated search space
Relational algebraic transformation rules
Equivalent query plan
Search strategy
Query cost model
Optimal execution strategy

**Figure 4.** Query optimization process of distributed database

## 3 Distributed database connection query optimization techniques

### 3.1 Immunogenetic algorithms

### 3.1.1 Genetic algorithms

Genetic Algorithm is an adaptive global probabilistic search algorithm, which uses probabilistic optimization method to find the best, and can automatically acquire and optimize the search space, adaptively adjust the search direction according to the environment, which is a bottom-up optimization method. Genetic algorithms represent complex problems with simple coding, process the coding of a set of variables, and guide learning and determine the direction of the search through genetic manipulation of a set of coded representations and the natural selection of the best and the worst.

### 3.1.2 Immunogenetic algorithms

Immunogenetic algorithm is a kind of optimization algorithm that combines the advantages of the basic genetic algorithm with the characteristics of biological immunity theory, and it is a kind of optimization algorithm that combines and penetrates the knowledge of multiple disciplines and fields. Immunogenetic algorithm process is:

1) Set initialization parameters and randomly generate a certain number of individuals to form the initial parent population $A_1$.

2) Select a certain number of individuals with high adaptation from the initial population as vaccine extraction based on prior knowledge.

3) Judge whether the current population has contained the best individuals, if so, the algorithm ends. Otherwise, enter the following steps.

4) Operate on the individuals according to the concentration and fitness of the individuals in the population to replicate the high-quality individuals into the next generation.

5) Perform crossover operations on individuals in the current $K$ nd generation parent population $A_K$ according to the set crossover probability to obtain population $B_K$.

6) Perform mutation operation on individuals in population $B_K$ according to the set mutation probability to obtain population $C_K$, and return to step (3).

## 3.2 Immune Genetic Algorithm for Query Optimization of Multi-Join Expressions

### 3.2.1 Encoding of multi-connected expressions

The ordered string encoding (real number encoding) method operates by first numbering the n relations contained in the query starting from 1, and then forming the chromosome in the numbering order of the bottom-up leaf nodes of the left linear tree [30]. The antibody coding diagram is shown in Fig. 5. The left linear tree as shown in the figure is genetically encoded as (R2, R3, R1, R4).

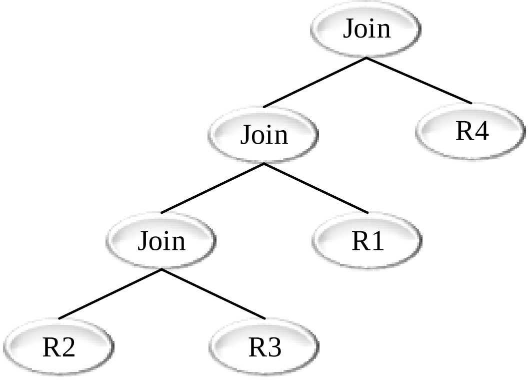


**Figure 5.** Antibody coding diagram

### 3.2.2 Adaptation function

Setting the antibody evaluation criteria focuses on the fitness of the solution to the problem. When using immunogenetic algorithms, it is very important to select suitable antibodies as well as to determine a realistic antibody expectation, and here we use the sum of records of the operation results to determine the cost of the query execution plan. For this problem, the following fitness function is used.

$$F(x)=\frac{1}{\sum_{n=1}^{n-1}|s_i|} \tag{3}$$

Where $x$ is the antibody and $S_i(i=1,2,\cdots,n-1)$ to represent the intermediate relationship.

When the immune system is invaded by an antigen, it will secrete the corresponding antibody through B cells to defend against the invading antigen and eventually destroy the antigen, and this stress response behavior of the immune system is based on the concentration of antibodies. Accordingly we correspond antigen, B cells and antibodies to the problem to be solved, a feasible solution to the problem to be solved $x_i$, and the fitness function of the solution $f(x_i)$. In a non-empty immune system $X$ set consisting of $N$ antibodies, the vectorial distance of antibody $(x_i)$ from another antibody is:

$$\rho(x_i) = \sum_{j=1}^{N} | f(x_i) - f(x_i) | \tag{4}$$

From the relationship between concentration and vector distance, the concentration of antibody $f(x_i)$ can be expressed as:

$$Density(x_i) = \frac{1}{\rho(x_i)} = \frac{1}{\sum_{j=1}^{N} | f(x_i) - f(x_i) |} \tag{5}$$

The selection probability based on antibody concentration can be derived from Eq:

$$P(x_i) = \frac{\rho(x_i)}{\sum_{i=1}^{N} \rho(x_i)} = \frac{\sum_{j=1}^{N} | f(x_i) - f(x_i) |}{\sum_{i=1}^{N} \sum_{j=1}^{N} | f(x_i) - f(x_i) |} \tag{6}$$

### 3.2.3 Operational Arithmetic Design

Genetic operators mimic the process of biological inheritance and natural evolution to achieve the superiority of individuals in a population. There are mainly 3 kinds of genetic operators: selection, crossover and mutation. The following are the arithmetic rules of the 3 types of operators designed for multi-connected query trees.

1) Selection operator

It is shown that the probability that the genetic algorithm with the strategy of preserving the best individual converges to the optimal solution is 1. Meter with the optimal preservation selection strategy, the least costly connected tree in the current population does not participate in the crossover and mutation operations, and the costly individuals are eliminated [31]. This makes the current optimal individual pattern will not be destroyed by the crossover and mutation operations, which ensures the convergence of the immunogenetic algorithm.

2) Crossover operator

The crossover operator is to recombine the genes of both parents selected in the selection operation. A new generation of individuals can be obtained through the crossover operation, and the new individuals combine the characteristics of their parent individuals. Crossover embodies the idea of information exchange. Crossover operation generates new individuals by exchanging part of the genes of two parent individuals under certain probability.

In this paper, two-point crossover is used, and when two parent individuals crossover, the individual is generated by selecting a part of parent 1 and preserving the relative order of the table in parent 2.

For example, there are two parent individuals P1 and P2 encoded using the ordered string coding scheme, and two crossover points "|" are randomly selected.

$$\begin{aligned} &P1:(R1R2R3R41R5R6R71R8R9) \\ &P2:(R4R8R9R11R6R7R31R2R5) \end{aligned} \tag{7}$$

First, the segment before the first intersection is kept constant to obtain:

$$\begin{aligned} &\Delta F1:(R1R2R3R4 \mid XXX \mid XX) \\ &\Delta F2:(R4R8R9R1 \mid XXX \mid XX) \end{aligned} \tag{8}$$

Then, parent individual 1 starts encoding backward from the first bit after the first intersection, and when it reaches the second intersection, it continues to encode backward from the first bit of the second intersection until the end of the table, so that the coding arrangement of the linkage table encoded by parent individual 1 from the first bit after the first intersection can be obtained as R5-R6-R7-R8-R9-R1-R2-R3-R4. For the parent individual 2, there are already connection table codes R4, R8, R9, R1, remove them from the connection table code arrangement of the parent individual 1 to get the arrangement R5-R6-R7-R2-R3 and then copy this arrangement to the parent individual 1, the starting point of copying is also from the first bit after the first intersection point, so as to determine the unknown code X in the corresponding position of the child individual 1, which is then born as individual 1:

$$\Delta F1:(R4R8R9R11R5R6R71R2R3) \tag{9}$$

Similarly, subindividual 2 can be generated as:

$$\Delta F2:(R1R2R3R41R6R7R51R8R9) \tag{10}$$

Duplication and crossover cannot produce new genes although they can produce new strings of genes, and if all gene strings are identical at a given position, the characteristics characterized by that gene do not change.

The probability formula for crossover is:

$$p_{c=}\begin{cases} \dfrac{a(f_{\max}-f')}{f_{\max}-\bar{f}} & f \geq \bar{f} \\ b & 0<a,b<1 \end{cases} \tag{11}$$

where $f_{\max}$ is the maximum adaptation value in the population. $f$ is the average adaptation value of individuals in the population in each generation. $f'$ is the fitness value of the individual with the larger fitness value among the two individuals to be crossed.

3) Mutation operator: the introduction of mutation operator makes up for the shortcomings of selection and crossover operation that can only produce new gene strings but not new genes, and ensures the continued evolution of biological populations. Mutation operations are mainly

in the following ways: uniform mutation, single-point random mutation, and multiple pairs of locus transposition. In this paper, uniform mutation is adopted. The mutation probability function is:

$$p_m = \exp(-\rho(x_i) * \exp(-a * 1 / \rho(x_i)))(0 < a < 1, \ \textit{Conditional variation}) \tag{12}$$

In this paper, we use single point gene transposition for gene mutation. The single point gene transposition operation is to take two positive integers $i, j(l < i, j < n, i \neq j)$ at random and exchange the position of a pair of gene loci $R_i, R_j$ in antibody $R$ with a certain probability for antibody R=(R1, R2, R3, R4,...Rn) (n is the number of linkage tables).

### 3.2.4 Algorithm flow

The basic idea of vaccination is to immunize and immunoselect the antibody population on the basis of reasonable extraction of vaccine, to overcome the blindness of crossover and mutation operations in the genetic algorithm, to improve the antibody adaptation value and to inhibit the degradation of the population. In order to suppress the phenomenon of population degradation in the genetic process and speed up the convergence of the algorithm, the immunogenetic algorithm in this study is based on the vaccine mechanism of IGA, with the core of the three steps of extracting vaccine, vaccinating vaccine, and immune selection operation.

## 4 Experimental analysis

### 4.1 Simulation experiment and performance analysis

### 4.1.1 Experimental environment

The execution efficiency of the experimental system varies in different operating environments. The experimental environment used in this paper consists of the mainframe computer running the system (Server1) and two database servers (Server2 and Server3), and the three machines are interconnected to become a small local area network (LAN).

### 4.1.2 Optimization of system parameters

Immunogenetic algorithm, like other non-exhaustive search strategies, different parameters have a great impact on the performance of the algorithm, so the optimization strategy based on genetic algorithm needs to carry out the optimization of the algorithm's own parameters first. So we first explore the relationship between the average values of the optimal solutions in the last 10 generations obtained when the genetic algorithm evolves to the corresponding number of generations under different parameters.

First, let's investigate the performance of the genetic algorithm under different crossover rates. When we set the crossover rate Pc of the immunogenetic algorithm to be 0.4, 0.6, 0.8 and the number of connections to be 8, the performance of the immunogenetic algorithm under different crossover rates is shown in Fig. 6. In the results shown, we can see that when the crossover rate Pc = 0.4, the convergence of the genetic algorithm is better. When Pc =0.8, the average value of the optimal solution for the last 200 generations is worse than the other two cases and the genetic algorithm converges slower. When Pc =0.6, the average value of the optimal solution and the convergence speed are better than the other two cases. The crossover rate Pc=0.6 is the best choice when considering both convergence speed and the average value of the optimal solution.

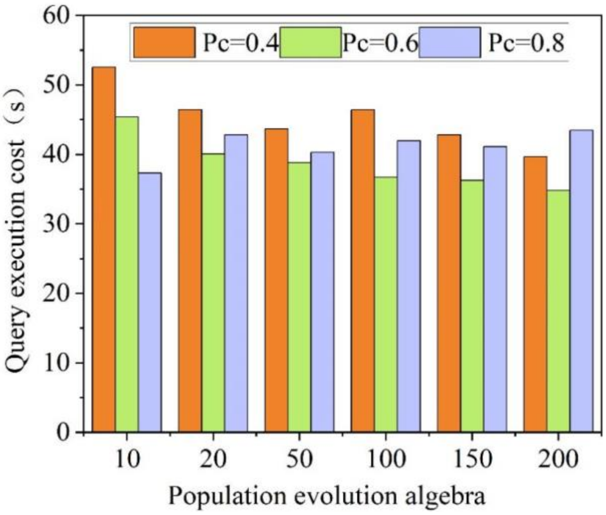


**Figure 6.** The performance of the immune genetic algorithm in different cross-rates

Next, we examine the performance of the genetic algorithm with the initial population size, pop_size, varying in size. In general, if the initial population is small, it can improve the efficiency of the immunogenetic algorithm operation, but due to the reduction of the diversity of the population, it is possible that the algorithm will be premature. And when the initial population is larger, although it reduces the efficiency of the immunogenetic algorithm, but it is conducive to the algorithm to solve in a larger space. In this paper, when the number of connections is set to 8, the initial population size of the immunogenetic algorithm is 30, 50 and 70 respectively, and the execution cost of the algorithm is shown in the figure below. The performance of the immunogenetic algorithm with different initial population sizes is shown in Fig. 7. The performance of the algorithm with an initial population size size of 30 is better than the population size of 50 and 70 cases, and the performance of the algorithm with a population size of 50 is intermediate between the population size of 30 and 70. So, the initial population size size taking the value of 30 is a better choice.

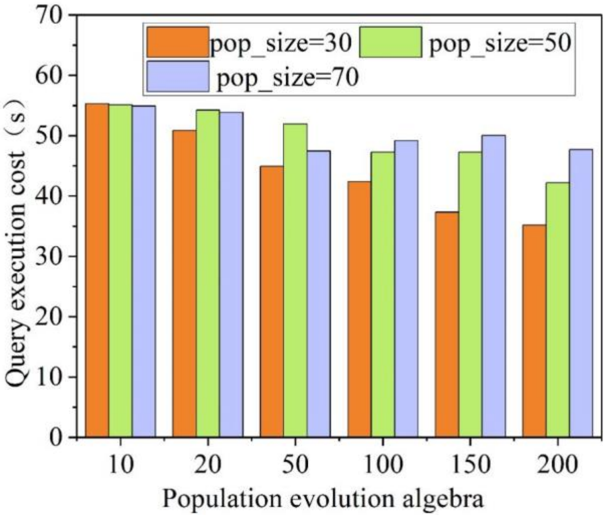


**Figure 7.** The immune genetic algorithm performs in different initial population sizes

### 4.1.3 System performance comparison

In this paper, we compare the performance of dynamic programming algorithms and immunogenetic algorithms by conducting comparative simulation experiments. The control parameters are set according to the initial population pop_size=30, crossover rate 0.6cP=, and mutation rate 0.08mP= obtained from the previous experiment. Input the parameters into the genetic algorithm program, each query is executed 10 times, and the query execution time is taken as the average of the 10 executions, when the number of relations we want to optimize is 4, the final genetic algorithm and the dynamic programming algorithm get an average execution time of 5 and 6 seconds, which is not a big difference. However, when the number of relations starts from 12, there is a significant difference in the query execution cost, and the comparison of the optimization performance of the immune genetic algorithm and the dynamic programming algorithm is shown in Figure 8. According to the analysis of the results in the figure, when the number of relations connected by the query is less than 10, the cost of the two algorithms is basically the same. However, when the number of relations connected to the query is greater than 10, the immunogenetic algorithm query optimization is significantly better than the dynamic programming algorithm, in the number of relations connected to the query optimization of the immunogenetic algorithm are 16 and 20, respectively, the cost of the execution of the query optimization is better than the cost of the dynamic programming algorithm execution of the cost of 22% and 37.5%, respectively.

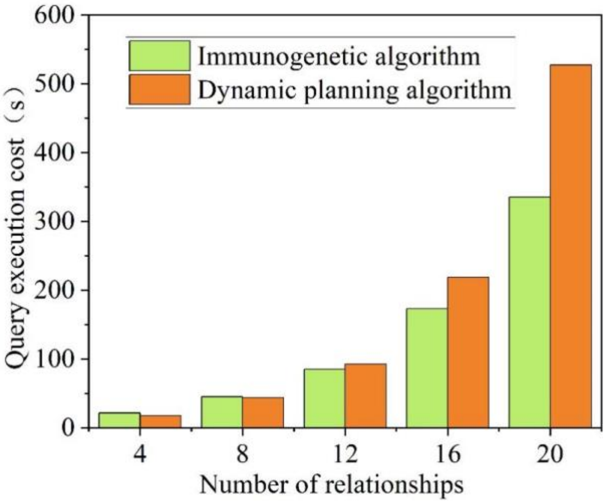


**Figure 8.** The optimization performance comparison

## 4.2 Performance Experiments on Query Optimization with Immune Genetic Algorithm

### 4.2.1 Experimental environment

Although the experiments in this thesis mainly focus on the query optimization problem for distributed type databases, due to the limitations of the experimental environment, the simulation experiments were ultimately conducted on only one computer.

### 4.2.2 Experimental results and comparative analysis

In the course of this simulation experiment the focus is on checking the network cost characteristics of this algorithm and the stability of the algorithm on the distributed database query optimization problem. The tests are carried out in a number of 30 as well as 58 network nodes respectively. It is assumed that the network node issuing the query command is 3, i.e., the source node S, and all the query data required to complete the query command is distributed in 5 nodes, 4, 6, 8, 10, and 12, i.e., the target node E. How to connect from the source node S to the target nodes through a certain path with the lowest possible communication cost is the focus of this experiment. The performance is also to be compared. So in this simulation, in the same environment and the same network topology relationship, the basic ant colony algorithm is used and also simulation is done so that the optimization algorithm can be compared. The results of the simulation experiment are as follows:

1) The network topology relationship graph (30 nodes) is shown in Fig. 9.

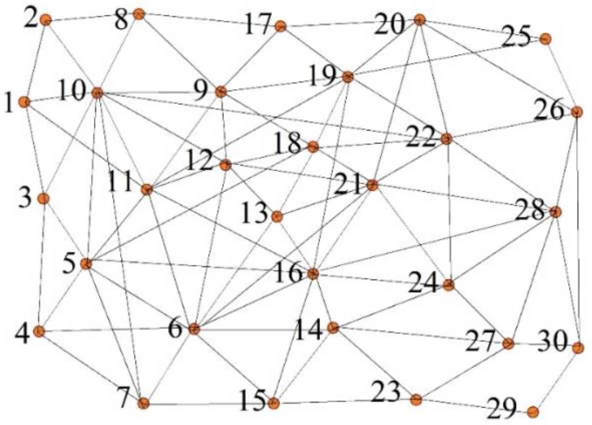


**Figure 9.** Network topology diagram

The optimal crawling path (30 nodes) for the immunogenetic algorithm is shown in Figure 10. As can be seen from the figure, the solid dots indicate the target nodes, and the thicker lines connected together represent one of the best paths searched by the algorithm.

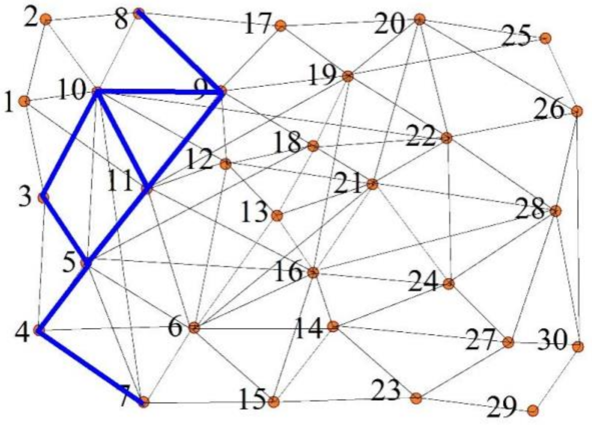


**Figure 10.** The optimal crawling path of immune genetic algorithm

In order to prove the advantage of the algorithm, simulation experiments with the basic ant colony algorithm were also done in the same environment used for comparison. The basic immunogenetic algorithm optimal solution crawling path is shown in Fig. 11.

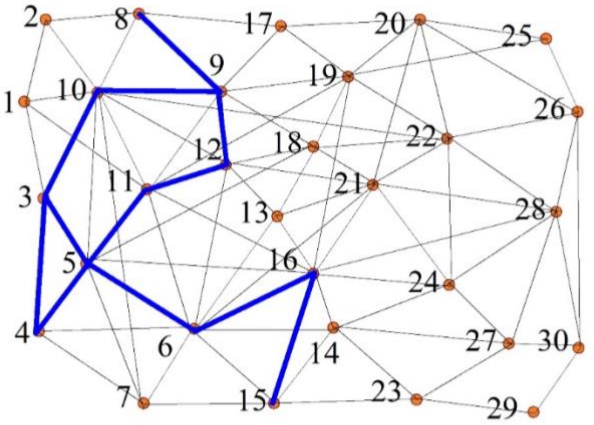


**Figure 11.** The basic immune genetic algorithm is optimal

Separately, both algorithms have been utilized to find the target node in the same environment, so how do they perform in terms of communication cost respectively? In this simulation experiment, since it is operated on one computer experimentally, the I/O cost and CPU cost are the same for both algorithms, so we only compare the communication cost. The communication cost here refers to the communication time, including the time within the node and the time on the link. When the network topology is generated in advance, the communication time on all the paths is set using a randomized method, and at the same time, the communication time on each node is also effectively set in milliseconds. In Immunogenetic Algorithm, the division is carried out while 50 effective iterations are performed. The immunogenetic algorithm query convergence curve (30 nodes) is shown in Fig. 12. The optimal communication cost obtained during the optimization of the immunogenetic algorithm is: ans=359.0032. The basic immunogenetic algorithm query convergence curve (30 nodes) is shown in Fig. 13. The optimal communication cost obtained during optimization as per the basic algorithm is: ans=475.1463.From the figure, it is clear that the optimal solution found by the immunogenetic algorithm is less in terms of communication cost than the solution found by the basic immunogenetic algorithm. The basic ant colony algorithm starts out with a communication cost of about 560, which decreases relatively quickly in the early stages of the algorithm, but when four iterations have been made, the rate of decrease in communication cost slows down significantly, and by about generation 5, it is close to convergence and falls into a local optimum, and the communication cost no longer decreases, at about 472. The immune genetic algorithm in the beginning of the communication cost reduction rate is not as fast as the basic ant colony, in the 3rd generation there is a short stagnation of the algorithm, with the operation of the smoothing mechanism and learning mechanism, the algorithm jumps out of the stagnation state, continue to search, to about 10 generations before it begins to converge, at this time the communication cost has been reduced to 361, in this aspect of performance compared to the basic algorithms is relatively good.

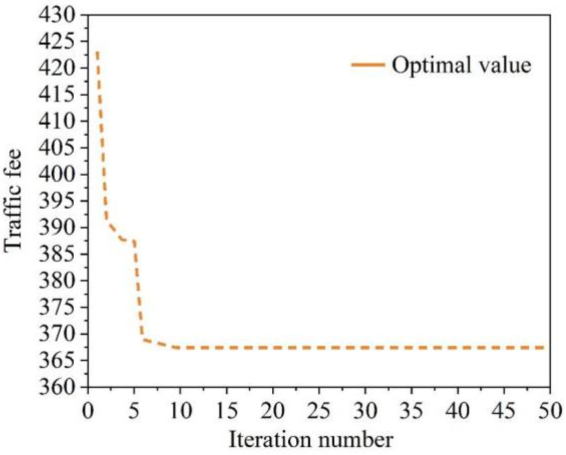


**Figure 12.** Immune genetic algorithm query convergence curve

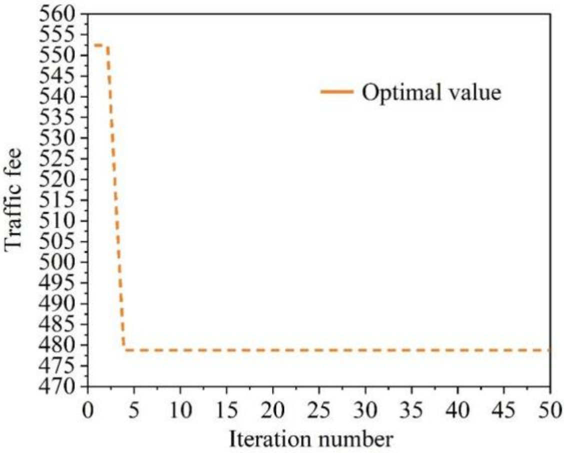


**Figure 13.** Basic algorithm query convergence curve

2) Network topology of 60 nodes

Using the same algorithm and the same parameters, the same simulation was conducted on the simulated network topology of 58 nodes, and the results are as follows. The network topology relation diagram (58 nodes) is shown in Fig. 14.

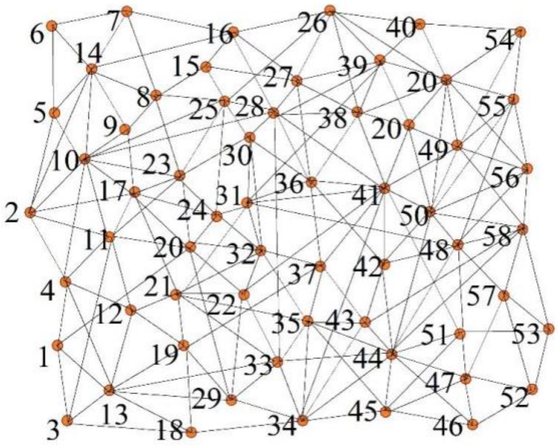


**Figure 14.** Network topology diagram

The final solution found using the multiple ant colony genetic algorithm is listed in the figure below. The optimal solution crawling path (58 nodes) for the immune genetic algorithm is shown in Figure 15.

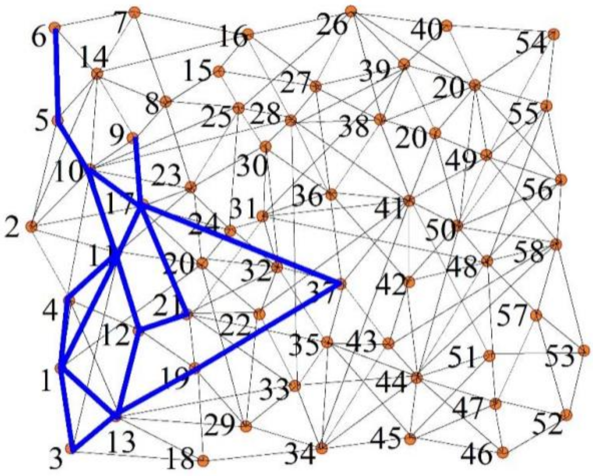


**Figure 15.** The immune genetic algorithm is optimal

Similarly, the basic genetic algorithm has been used for the related simulations. The basic genetic algorithm optimal solution crawling path is shown in Fig. 16.

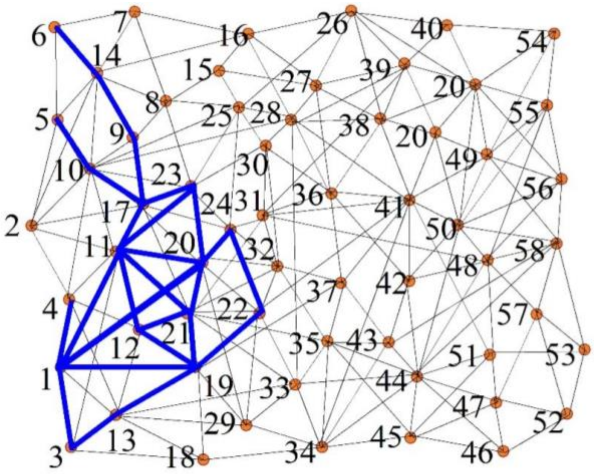


**Figure 16.** The basic algorithm is optimal

How does the convergence performance of the two algorithms compare for the case of a network topology with 58 nodes? The convergence curves of the immunogenetic algorithm and basic genetic algorithm queries (58 nodes) are shown in Figures 17 and 18. The optimal communication cost obtained by the immunogenetic algorithm optimization is: ans=455.5255, and the optimal communication cost obtained by the basic genetic algorithm optimization is: ans=463.2485. It can be seen from the figure that the two algorithms begin to converge at the beginning of the basic algorithm performs better, but this is the basic algorithm falls into the local optimal performance too early. The immune genetic algorithm obtains the optimal solution in the 23rd generation after a short period of stagnation in the 5th, 10th, and 15th generations. In terms of the communication cost of the final optimal path, the immunogenetic algorithm performs better, but the advantage does not seem to be particularly significant.

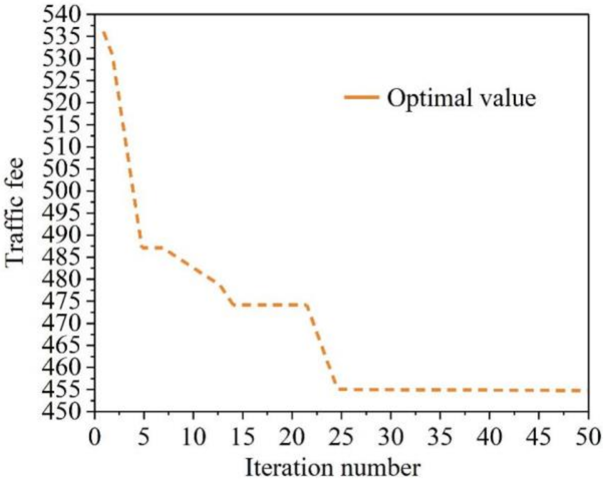


**Figure 17.** Immune genetic algorithm query convergence curve

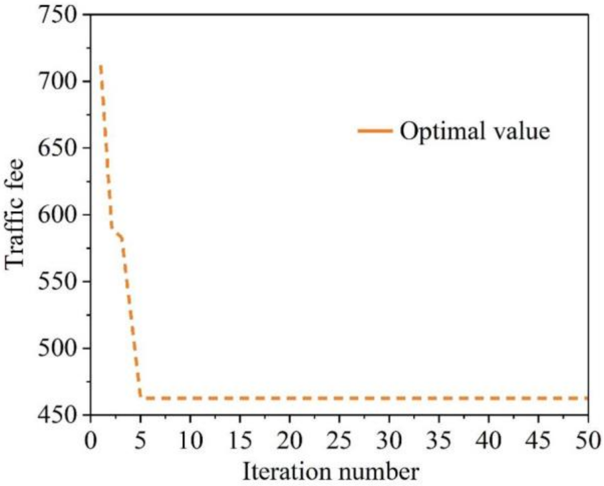


**Figure 18.** Basic algorithm query convergence curve

The stability of the immunogenetic algorithm and the basic algorithm was again tested and compared in the same environment of 60 nodes, both algorithms were run 10 times, since both the immunogenetic algorithm and the basic algorithm randomly select routes with some probability when finding routes, the optimal paths found could be different even though they were run 10 times in the same environment. The stability of the two algorithms is compared using mean and standard deviation. Comparison of the standard deviation and mean value of the running results of the immune genetic algorithm and the basic ant colony algorithm are shown in Fig. 19 and Fig. 20. The algorithm's stability increases as the standard deviation decreases, indicating a smaller fluctuation of the objective function value obtained by the algorithm each time. The smaller the mean value, the better the algorithm's optimization performance. From the figure, it can be clearly seen that the immunogenetic algorithm is superior to the basic algorithm in terms of stability, and the optimization seeking performance is also slightly better. So although the generation of this network topology is random and the performance of the two algorithms will have some changes in each run, overall, the immunogenetic algorithm outperforms the basic genetic algorithm in terms of communication cost and is more stable.

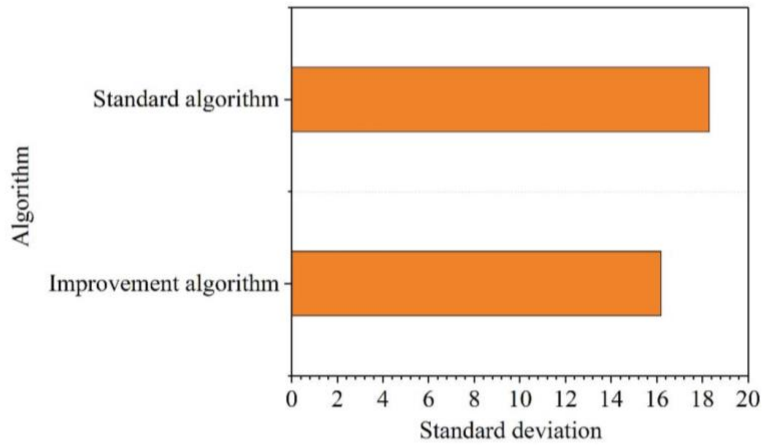


**Figure 19.** The algorithm runs the comparison of the standard deviation

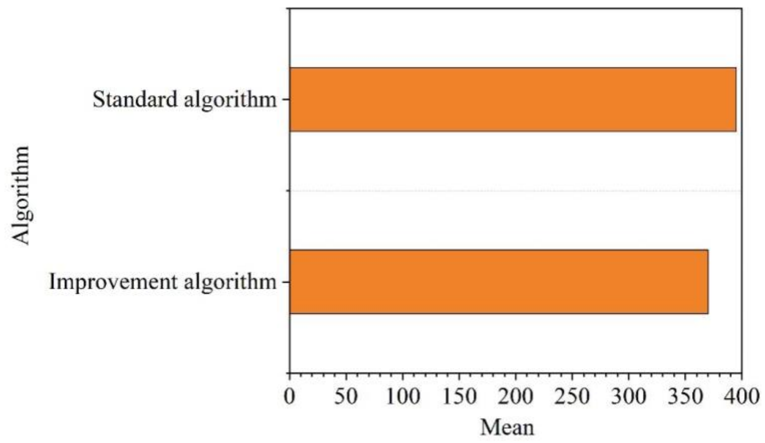


**Figure 20.** The algorithm runs the mean comparison

## 5 Conclusion

In this paper, the application of immunogenetic algorithm in the multi-connection query optimization problem is studied in depth, and finally it is concluded that the algorithm in this paper has certain advantages over other basic algorithms through simulation experiments and performance analysis.

In the environment of simulation experiments, a set of optimal parameter values applicable to computer network systems are obtained through continuous experiments, for example, when the crossover rate Pc=0.6 is the optimal choice from the consideration of two factors, namely, the convergence speed and the average value of optimal solutions. By applying the experimentally derived parameter values to optimize the distributed multi-connection query, the expected optimization effect is achieved.

Under the same conditions, the basic ant colony algorithm is tested and it is found that the optimal communication cost obtained during the optimization of the immunogenetic algorithm is 359.0032.The optimal communication cost obtained during the optimization of the basic genetic algorithm is 475.1463.It can be concluded that the algorithm designed in this paper has a more optimal solution and better stability.

## References

[1] Stieglitz, S., Mirbabaie, M., Ross, B., & Neuberger, C. (2018). Social media analytics–Challenges in topic discovery, data collection, and data preparation. International journal of information management, 39, 156-168.

[2] Zhao, Y., & Zhang, J. (2017). Consumer health information seeking in social media: a literature review. Health Information & Libraries Journal, 34(4), 268-283.

[3] Chen, E., Lerman, K., & Ferrara, E. (2020). Tracking social media discourse about the covid-19 pandemic: Development of a public coronavirus twitter data set. JMIR public health and surveillance, 6(2), e19273.

[4] Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. Journal of economic perspectives, 31(2), 211-236.

[5] Freitag, M., Bandle, M., Schmidt, T., Kemper, A., & Neumann, T. (2020). Adopting worst-case optimal joins in relational database systems. Proceedings of the VLDB Endowment, 13(12), 1891-1904.

[6] Dimitriu, C. (2023). The difference between relational and non-relational databases in programming. In Conferinţa tehnico-ştiinţifică a studenţilor, masteranzilor şi doctoranzilor (Vol. 4, pp. 332-336).

[7] Gaulton, A., Hersey, A., Nowotka, M., Bento, A. P., Chambers, J., Mendez, D., ... & Leach, A. R. (2017). The ChEMBL database in 2017. Nucleic acids research, 45(D1), D945-D954.

[8] Stoffel, M. A., Nakagawa, S., & Schielzeth, H. (2017). rptR: Repeatability estimation and variance decomposition by generalized linear mixed-effects models. Methods in ecology and evolution, 8(11), 1639-1644.

[9] Kuo, T. T., Kim, H. E., & Ohno-Machado, L. (2017). Blockchain distributed ledger technologies for biomedical and health care applications. Journal of the American Medical Informatics Association, 24(6), 1211-1220.

[10] Dorri, A., Steger, M., Kanhere, S. S., & Jurdak, R. (2017). Blockchain: A distributed solution to automotive security and privacy. IEEE communications magazine, 55(12), 119-125.

[11] Coronel, C., & Morris, S. (2019). Database systems: design, implementation and management. Cengage learning.

[12] Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., Depristo, M., ... & Lakshminarayanan, B. (2019). Likelihood ratios for out-of-distribution detection. Advances in neural information processing systems, 32.

[13] Li, Q., Diao, Y., Chen, Q., & He, B. (2022, May). Federated learning on non-iid data silos: An experimental study. In 2022 IEEE 38th international conference on data engineering (ICDE) (pp. 965-978). IEEE.

[14] Francisco, K., & Swanson, D. (2018). The supply chain has no clothes: Technology adoption of blockchain for supply chain transparency. Logistics, 2(1), 2.

[15] Schober, P., Boer, C., & Schwarte, L. A. (2018). Correlation coefficients: appropriate use and interpretation. Anesthesia & analgesia, 126(5), 1763-1768.

[16] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., ... & Yellick, J. (2018, April). Hyperledger fabric: a distributed operating system for permissioned blockchains. In Proceedings of the thirteenth EuroSys conference (pp. 1-15).

[17] Liu, W., Wang, X., Owens, J., & Li, Y. (2020). Energy-based out-of-distribution detection. Advances in neural information processing systems, 33, 21464-21475.

[18] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems, 2, 429-450.

[19] Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., & Yu, S. X. (2019). Large-scale long-tailed recognition in an open world. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2537-2546).

[20] Lee, K., Lam, M., Pedarsani, R., Papailiopoulos, D., & Ramchandran, K. (2017). Speeding up distributed machine learning using codes. IEEE Transactions on Information Theory, 64(3), 1514-1529.

[21] Verbraeken, J., Wolting, M., Katzy, J., Kloppenburg, J., Verbelen, T., & Rellermeyer, J. S. (2020). A survey on distributed machine learning. Acm computing surveys (csur), 53(2), 1-33.

[22] Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., & Tan, K. L. (2017, May). Blockbench: A framework for analyzing private blockchains. In Proceedings of the 2017 ACM international conference on management of data (pp. 1085-1100).

[23] Tanenbaum, A. S., & Van Steen, M. (2017). Distributed systems (pp. 298-303). CreateSpace Independent Publishing Platform.

[24] Dinh, T. T. A., Liu, R., Zhang, M., Chen, G., Ooi, B. C., & Wang, J. (2018). Untangling blockchain: A data processing view of blockchain systems. IEEE transactions on knowledge and data engineering, 30(7), 1366-1385.

[25] Tian, F. (2017, June). A supply chain traceability system for food safety based on HACCP, blockchain & Internet of things. In 2017 International conference on service systems and service management (pp. 1-6). IEEE.

[26] Lu, Y., Huang, X., Dai, Y., Maharjan, S., & Zhang, Y. (2019). Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. IEEE Transactions on Industrial Informatics, 16(6), 4177-4186.

[27] Tanwar, S., Parekh, K., & Evans, R. (2020). Blockchain-based electronic healthcare record system for healthcare 4.0 applications. Journal of Information Security and Applications, 50, 102407.

[28] Jiaming He,Qinliang Tan & Hanyu Lv. (2025). Data-driven climate resilience assessment for distributed energy systems using diffusion transformer and polynomial expansions. Applied Energy124957-124957.

[29] Jun Jin,Chenyan Hao & Yewen Chen. (2024). Composite quantile regression for a distributed system with non-randomly distributed data. Statistical Papers(1),1-1.

[30] Ge Yong Feng,Wang Hua,Cao Jinli,Zhang Yanchun & Jiang Xiaohong. (2024). Privacy-preserving data publishing: an information-driven distributed genetic algorithm. World Wide Web(1).

[31] Kamal Maryam,Amin Shahzad,Ferooz Faria,Awan Mazhar Javed,Mohammed Mazin Abed,Al-Boridi Omar & Abdulkareem Karrar Hameed. (2022). Privacy-aware genetic algorithm based data security framework for distributed cloud storage. Microprocessors and Microsystems.