Mathematical Biosciences and Engineering

MBE, 20(9): 16886–16912.
DOI: 10.3934/mbe.2023753
Received: 28 April 2023
Revised: 30 July 2023
Accepted: 09 August 2023
Published: 24 August 2023

http://www.aimspress.com/journal/MBE

Research article

Advancing remote consultation through the integration of blockchain and ant colony algorithm

Xiang Gao[1] and Yipeng Zhang[2,*]

[1] School of Public Health and Management, Guangxi University of Chinese Medicine, Nanning 530200, China
[2] School of Basic Medical Sciences, Guangxi University of Chinese Medicine, Nanning 530200, China

* **Correspondence:** Email: zhangyp20171@gxtcmu.edu.cn.

Abstract: To guide the more reasonable and fair allocation of medical resources, to solve the problem of fee prices negotiated by various subjects in the medical and health system and patient payment, and to solve the problem of how to ensure the privacy, accuracy, consistency and traceability of data in the process of collecting patient information in each hospital, according to the operation process of a remote consultation service, a decentralized remote intelligent consultation blockchain model is proposed. The model uses the improved ant colony algorithm under a smart contract and studies the practicality of the improved ant colony algorithm on the multi-node remote consultation service simulation platform. According to the experimental analysis results, the improved ant colony algorithm can automatically execute and effectively match the target population under the smart contract.

Keywords: blockchain; remote consultation service; ant colony algorithm; smart contract; decentralized

1. Introduction

Blockchain is a novel decentralized database technology that serves as the underlying technology of Bitcoin. It is characterized by non-changeability, complete record-keeping, universal maintenance and transparency to all users, making it a promising technology for various applications. Dwivedi et al. [1] have proposed a naive blockchain and watermarking-based social media framework to control the

propagation of fake news and have postulated a new blockchain model to address existing challenges. Similarly, Srivastava et al. [2] have presented the advantages and practical challenges of blockchain-based security approaches for remote patient monitoring using Internet of Things (IoT) devices.

Remote consultation is a medical service modality that facilitates online examination, case discussion and guidance between healthcare providers and patients, leveraging computers, auxiliary equipment and network communication technologies [3]. A remote consultation platform constitutes a business system that enables remote medical diagnosis and consultation services, being a multifarious, multi-agent and multi-interactive medical enterprise system. Through a remote consultation platform, multimedia information sharing, including textual data, images, audio and video, is achievable among medical institutions. The platform further endows patients in underprivileged regions with high-quality medical services over long distances, enabling primary care providers to administer nursing and treatment, guided by medical specialists [4]. This concomitantly enhances the quality of medical services offered to patients, while driving innovation in the medical service domain.

While the remote consultation platform instituted by hospitals has positive connotations, it also engenders novel challenges. Notably, the exorbitant costs of construction, the sustainment of uninterrupted financial support and the achievement of information exchange among medical institutions to form business alliances pose significant quandaries. Furthermore, inconsistent pricing and payment modalities adopted by patients present further complications [5]. Otherwise, the remote consultation service transaction system is based on a typical centralized transaction model. The data information is in the hands of a single institution, and there are credit problems such as opacity, tamper susceptibility and repudiation of service transaction data information [6]. In addition, the remote consultation service transaction system lacks intelligent contracts with multiple transaction modes, and the transaction operation process lacks automation, intelligence and parallelism.

Moreover, the dispersion of hospital population flow and the rational and equitable allocation of medical resources have emerged as crucial aspects of promoting the intelligentization of medical services. Remote consultation constitutes a pivotal component of an intelligent medical system. The swarm intelligence ant colony algorithm, coupled with blockchain and smart contract technology, is applied to the remote consultation service platform, with the ultimate objective of allocating and matching medical resources and medical groups. Such a development is of immense significance for the application and exploration of the medical service industry and blockchain technology.

The contributions made by this study are as follows:

1) A service transaction ant colony intelligent contract model based on blockchain is proposed to ensure the integrity of the data during transmission and the non-repudiation of the sender.

2) A new improved ant colony algorithm based on bacterial foraging, the bacterial foraging ant colony algorithm, is proposed. It improves the efficiency of matching between the two sides of consultation and reduces more complicated manual operations.

3) The improved ant colony algorithm is integrated with smart contracts in remote consultation service transactions, which improves the multi-objective transaction matching and has a high success rate in building smart contracts.

The remaining sections of this paper are organized as follows. Section 2 reviews works related to remote consultation, blockchain and swarm intelligence technology. Section 3 provides a brief introduction to the construction of the remote consultation service platform under smart contracts. Section 4 presents all of the components of the proposed model. Section 5 shows the experimental results and analysis of improved ant colony algorithm. The application of improved algorithm is shown in Section 6, followed

by the conclusions of this study in Section 7.

2. Related works

The application of blockchain in the intelligent medical technology field has been researched by scholars. Fan et al. [7] proposed an efficient and secure medical data sharing scheme based on medical blockchain and presented a trusted cloud medical data sharing example. Liu et al. [8] explored a privacy-protected sharing mode of electronic medical records. Zhang [9] proposed a blockchain-based electronic medical record-sharing model that leverages the distributed and tamper-proof properties of blockchain technology. The work of Griggs et al. [10] represents an attempt to integrate smart contracts with blockchain technology to achieve remote automatic monitoring of patients, thereby contributing to the reduction of electronic medical record costs. Similarly, Shae and Tsai [11] have designed a blockchain platform aimed at facilitating clinical trials and precision medicine, while also outlining the challenges encountered in utilizing blockchain technology. Meanwhile, Christidis and Devetsikiotis [12] have constructed a blockchain and smart contract model in the context of the IoT, which represents a novel approach to exploring the integration of blockchain and big data. Moosavi et al. [13] introduced the application of blockchain in supply chain management. Alshudukhi et al. [14] have introduced blockchain security managers, based on microservice technology, for federated cloud systems in an IoT environment. Liao et al. [15] proposed a novel remote medical service model by integrating remote real-time interactive functions with electronic medical records containing the patient's pertinent examination materials. Zhou et al. [16] analyzed the communication technologies used in telemedicine systems and restructured corresponding module functions in multimedia, providing an effective solution to enhance communication efficiency and bandwidth utilization. Moreover, Chen et al. [17] expounded on the potential benefits of telemedicine in enhancing medical and healthcare service capabilities. The preservation of patient data privacy, accuracy, consistency and traceability during the collection process is crucial to ensure efficient information exchange in telemedicine [5], which subsequently impacts the quality of remote consultation. To address this, Mao and Zhang [18] designed an intelligent remote medical system tailored for households and communities through the integration of IoT platforms, big data and cloud computing technologies. In the aspect of intelligent service transactions, Bachrach [19] proposed a framework based on deep compensation learning to find the maximum win-win situation for negotiating teams. In [20] and [21], different heuristic methods are also applied to automatic negotiation in various ways to find a win-win negotiation result.

Through the above research, it is found that there are few studies on the theoretical research and application results of applying blockchain to a remote consultation service model and the establishment of doctor-patient relationships between patients and experts. How blockchain technology provides a non-central service model in the field of telemedicine, how to establish a remote consultation service platform based on blockchain, and how to apply intelligent contracts and swarm intelligence algorithm in the process of establishing doctor-patient relationships still need to be solved urgently. To fill the aforementioned gaps, this paper proposes an improved ant colony algorithm for improving the doctor-patient relationships matching module in the remote intelligent consultation platform. In the matching mode of patients and experts, the traditional cumbersome online application management and consultation management are abandoned, and the improved ant colony algorithm is used to automatically match patients with experts.

3. Remote consultation and contract model

In the current era of swift progress in the service industry, the relative proportion of this sector in overall economic development is on the rise. As a result, the quality of medical services has emerged as a crucial element of competitive advantage for major hospitals. The advent of remote consultation services has facilitated multi-objective interaction by means of information sharing, thereby transcending the spatial constraints of medical services and enabling a larger segment of the population to avail themselves of top-notch medical care. In this section, we will introduce the traditional centralized model of remote consultation service and propose a conceptual model of smart contracts for remote consultation.

3.1. Traditional centralized model of remote consultation service

The remote consultation service center realizes the informatization of the remote consultation center system through modern information technology and IoT technology and puts forward the traditional model of remote consultation. However, the traditional model of remote consultation service is based on a centralized model, and there are some problems such as credit problems and lack of automation, intelligence and parallelism in transaction operation. The traditional model of remote consultation is shown in Figure 1.

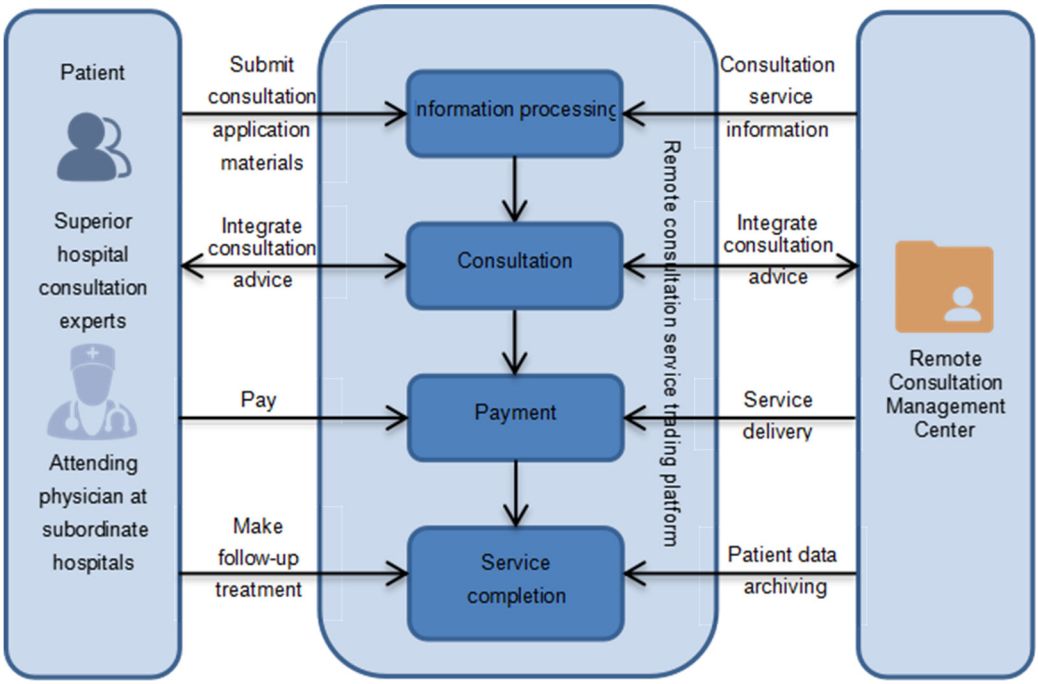

Figure 1. Traditional conceptual model of remote consultation service platform.

The conventional mode of remote consultation typically comprises four distinct stages:

1) Information processing-The remote consultation center disseminates the patient application information and expert database information. The platform allows patients and consultation experts to conduct bidirectional queries based on the information posted.

2) Consultation stage-The management center receives the consultation request and consultation data sent by the patient, preliminarily screens the eligible patients, synchronizes the information with the experts of the higher hospital and coordinates the consultation time. During the consensus period, experts consult patients remotely through multimedia to determine the treatment plan.

3) Payment stage-The final treatment plan is determined by the patient and attending doctor in conjunction with the treatment opinions of the consultation experts, and the patient receives treatment after paying the cost.

4) Service completion and feedback stage-The management center records the entire treatment process and data of the patients and classifies them to provide a reference for other similar cases.

3.2. Conceptual model of smart contract for remote consultation

Drawing on an analysis of the demand for remote intelligent consultation services, this study proposes a remote consultation service platform underpinned by a smart contract, wherein the functions of the remote consultation management center are decentralized, and the information processing and negotiation stage are incorporated into the smart contract. The conceptual model of remote consultation service smart contract is shown in Figure 2.

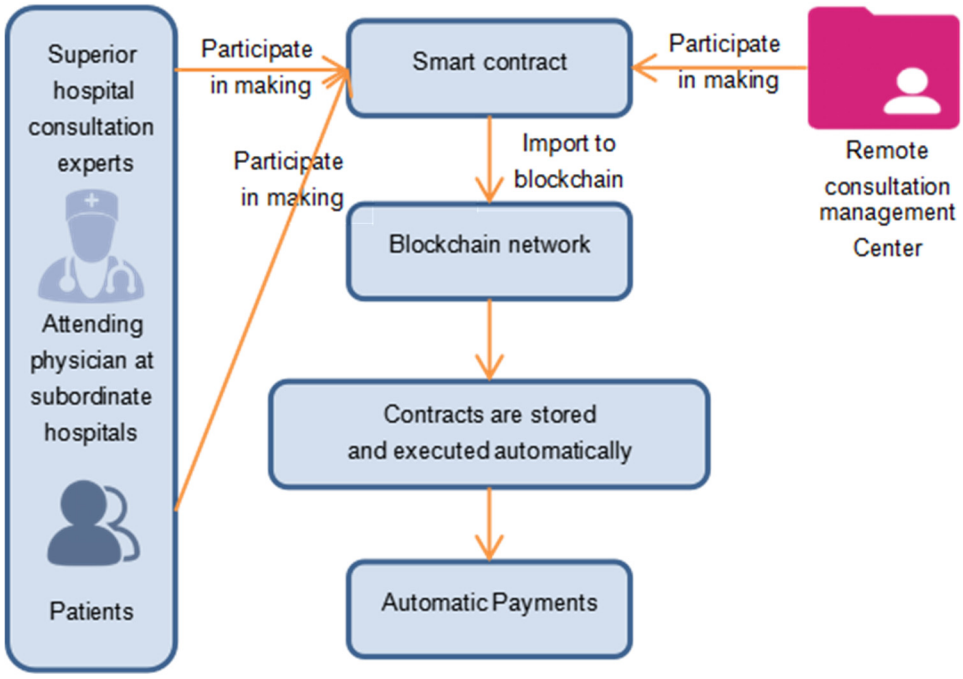

Figure 2. Conceptual model of remote consultation service smart contract.

The execution steps of the proposed model are as follows:

1) Participants in remote consultation formulate smart contracts according to relevant standards and requirements, including patients and doctors, both consultation experts in higher hospitals and attending doctors in lower hospitals.

2) The blockchain network receives the information generated after the execution of smart contracts and synchronizes the information to each block after the execution.

3) Upon completion of the consultation service process, the patient pays the fee as per the fee standard established by the contract. The remote consultation management center no longer serves as the coordination center in the entire process and only automatically records the logs of each link.

4. Remote consultation and algorithm improvement based on smart contract

4.1. Encryption and verification algorithm for remote consultation service

In the context of remote intelligent consultation services, the blockchain network adopts an effective big data outlier detection algorithm based on distributed density for data encryption and digital signature to ensure the consistency, accuracy and non-repudiation of data transmission. The detailed steps involved are illustrated in Figure 3 [22].

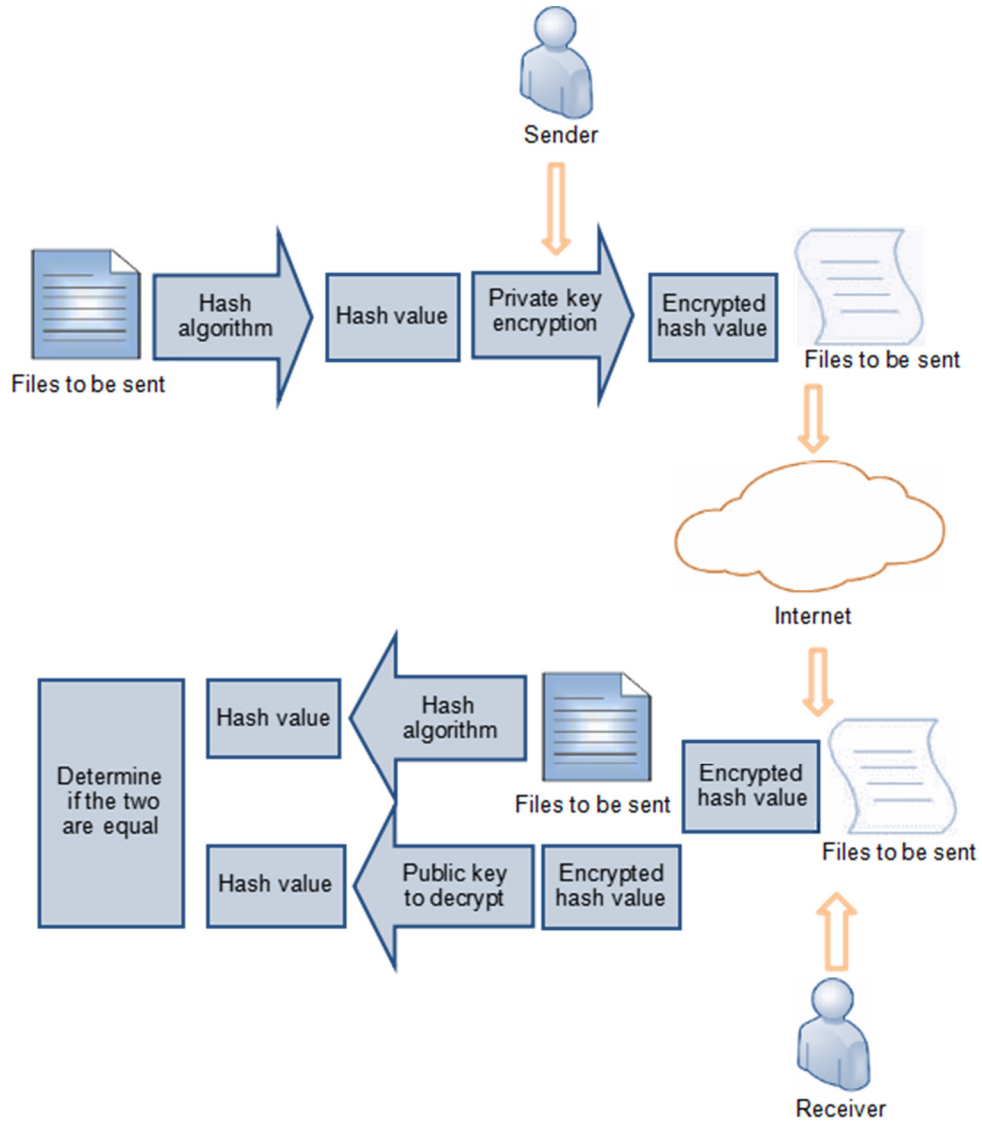

Figure 3. Digital signature flow chart.

4.2. Overview of smart contracts under blockchain

In contrast to traditional contract methods, modern smart contract technology can independently complete the transaction process without requiring the supervision or intervention of a third party. Once formulated, the smart contract cannot be revoked or modified and is automatically executed upon being placed into the blockchain. The immutability of the blockchain and the openness and transparency of the information facilitates multi-party transaction subjects to perform transactions without the need for a centralized trust mechanism. A smart contract is a digital form of computer protocol code that can run within a computer system and is triggered to execute automatically when specific conditions are met [23]. Its operating principle shown in Figure 4 can be summarized as follows:

1) Establishment of contracts: Multiple parties involved in the contract reach a mutual agreement based on their respective requirements, and the contract is digitized, expressed in computer code to convey its complete meaning and executed automatically based on the predetermined conditions. The involved parties sign the digital signature using their private keys to validate the contract.

2) Storage of contracts: The compiled electronic contract is transmitted to the blockchain, and each node of the blockchain synchronizes the contract information to verify the status of the contract and store it.

3) Implementation of the contract: Before the execution of the contract, the status of the predetermined trigger condition and the response rule is verified, and the transactions that meet the response condition are queued for verification. After the digital signature validation, the blockchain reaches a consensus, and the contract is executed.

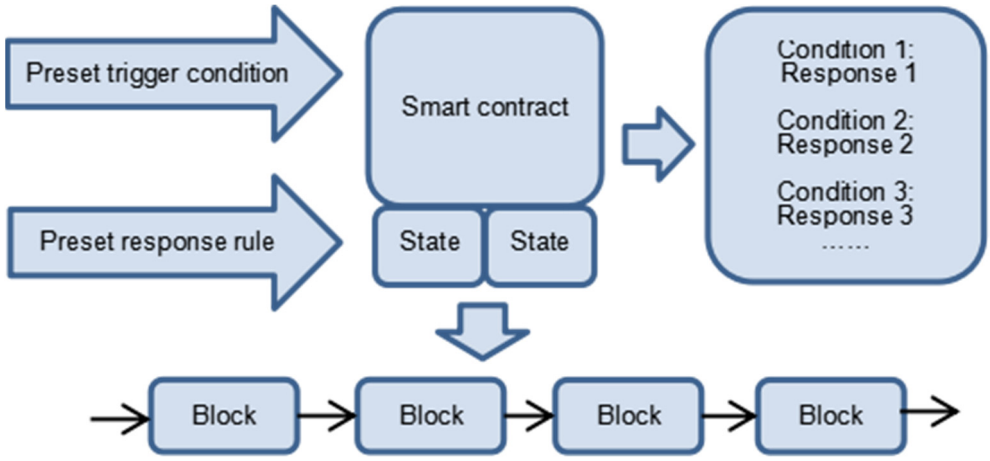

Figure 4. Smart contract model under blockchain.

4.3. Remote intelligent consultation transaction model under blockchain

Based on the conventional remote consultation service and in conjunction with the conceptual model of smart contracts for a remote consultation service, the integration of swarm intelligence algorithm and blockchain technology into the central link of the remote consultation service leads to the construction of a centerless intelligent trading platform. Consequently, a decentralized remote consultation service contract model under blockchain is proposed as shown in Figure 5.

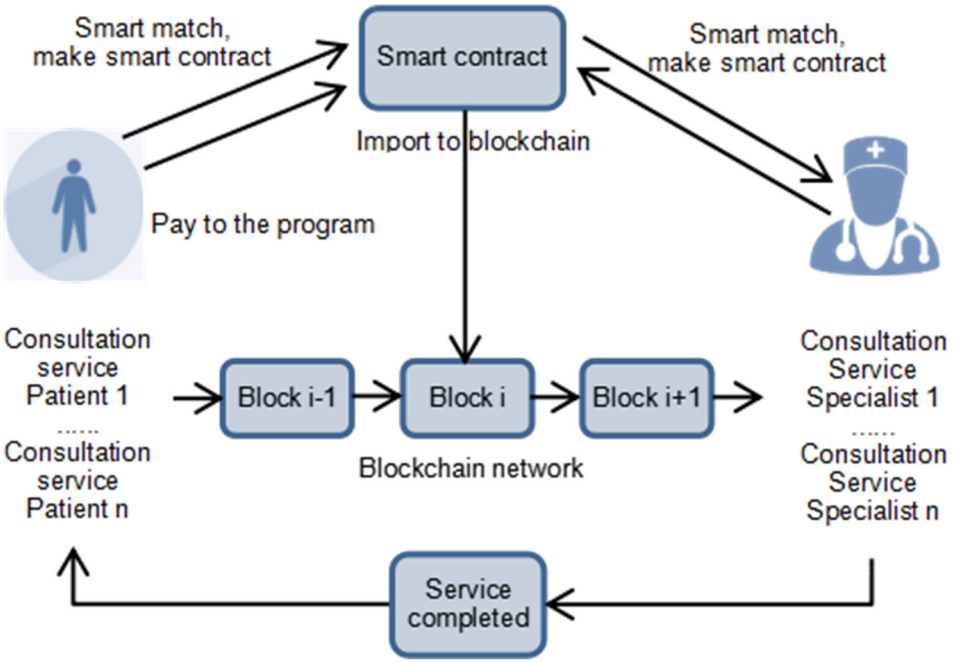

Figure 5. Blockchain decentralized remote consultation contract model.

In the blockchain network, the public and private keys obtained by the superior and subordinate hospitals, remote consultation management centers and patients upon registration of accounts on the blockchain are employed for encryption and decryption, and the public key is utilized to address customers on the blockchain. The superior and subordinate hospitals as well as the remote consultation management center participate in the development of a contract. The multi-party contract is signed with the key to render the contract effective, thereby improving the contract's anti-interference capability during execution. The contract is compiled into a computer-recognized language and imported into the blockchain through a peer-to-peer network. Each block implements a consensus mechanism and stores the contract. Additionally, the patient customers and superior hospital experts are matched via the group intelligent algorithm integrated into the contract. All transaction processes are devoid of central institutions, and the parties involved in the transaction do not require mutual trust. The entire process is irreversible to ensure information consistency and transparency. The blockchain structure used for contract execution is as follows:

In the data layer of the blockchain network, all blocks are encapsulated with their hash values, the hash value of the previous block and a consensus time record. The contract layer encapsulates each authenticated contract in a decentralized remote intelligent consultation model, which enables both parties to engage in flexible two-way selection and negotiate to formulate contracts. Preference value expresses the fit degree of time matching between patients and experts. In the context of remote consultation services, n represents the number of free times of patients, m represents the number of free times of experts in higher hospitals, $I = \{i|i = 1,2,..., n\}$ represents the set of patients, $J = \{j|j = 1,2,..., m\}$ represents the set of superior hospital experts, $S_k = \{(i,j)|i \in I, j \in J\}$ represents a successful match between patients and experts, and $S = \{s_1,s_2,...,s_k\}$ represents the set of all successful matches between patients and experts. $X = (x_1,x_2,...,x_n)$ represents the idle time vector of each patient, $Y = (y_1,y_2,...y_n)$ represents the idle time vector of experts in the superior hospital, x_i represents a certain idle time of the patient, y_i represents the idle time of a certain expert, X_{min} represents the minimum idle time of a certain patient, X_{max} represents the maximum idle time of a certain patient, Y_{min} represents the

minimum idle time of the expert, and the idle time format is yyyyy-mm-dd hh: mm: ss. w_i and w_j represent the weight values of the degree of mutual demand between patients and superior hospital experts, respectively. c_{ij} represents the number of patients matched with experts in a consultation, and T_{ij} represents the time of consultation. A mathematical model is constructed based on these variables:

$$U_i = \frac{X_{\min} - y_i}{X_{\max} - Y_{\min}}, j \in J \tag{1}$$

$$U_j = \frac{x_i - Y_{\min}}{X_{\max} - Y_{\min}}, i \in I \tag{2}$$

$$z_i = U_i \cdot w_i \tag{3}$$

$$z_j = U_j \cdot w_j \tag{4}$$

$$z_{(i.j)} = \sum_{i=1}^{m} z_i + \sum_{j=1}^{n} z_j \tag{5}$$

The constraints of the proposed model are as follows:

$$w_i, w_j \in [0,1] \tag{6}$$

$$\sum_{i=1}^{m} w_i = 1 \tag{7}$$

$$\sum_{j=1}^{n} w_j = 1 \tag{8}$$

$$U_i, U_j \in [0,1] \tag{9}$$

The preceding equations denote various aspects of the mathematical model, where Eq (1) and Eq (2) represent the time preference functions of the patient and the expert from the superior hospital, respectively. Equations (3) and (4) represent the satisfaction degree of the patient following the matching process with the expert from the superior hospital. Equation (5) denotes the overall preference value of the patient post-matching with the expert from the superior hospital. Equation (6) limits the value range of the weight values, w_i and w_j. Equations (7) and (8) indicate that the sum of the time weights of the patient and the expert from the superior hospital is equal to 1. Equation (9) stipulates the value range of the preference values, U_i and U_j.

4.4. Classical ant colony algorithm

Ant colony optimization (ACO) is a bionic stochastic search algorithm proposed to find an optimal path, based on the summary and analysis of the foraging behavior of ants in nature, with the

characteristics of positive feedback, self-organization, parallel search, strong robustness and easy combination with other algorithms. In the process of searching for food, ants will always release a kind of pheromone on the path they pass through and can sense the pheromones released by other ants. As more ants pass through a certain path, the concentration of pheromone released by ants on the path will increase. In a unit of time, as more and more ants leave pheromones on shorter paths, and ants are more inclined to move to the path with the high pheromone concentration, the chance that later ants will choose that path increases, and eventually the colony will find the shortest path. The ant colony algorithm solves combinatorial optimization problems through search mechanism, updating mechanism and coordination mechanism.

1) Search mechanism

Suppose there are n nodes and m ants, $\tau_{ij}(t)$ is the pheromone concentration between node *i* and node *j* at time *t*, and the initial pheromone concentration $\tau_{ij}(t)$ of each path is the same. The ant decides to visit the node according to the pheromone concentration, and the probability of ant *k* transferring from node *i* to node *j* is

$$p_k(i,j)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum\limits_{s\in J_k(i)}[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}, & if\ j\in J_k(i)\\ 0, & others\end{cases} \tag{10}$$

where $\eta_{ij}=\dfrac{1}{\mathrm{d}(i,j)}$ is the heuristic information. $\mathrm{d}(i,j)$ means the distance between nodes. $J_k(i)$ denotes the set of all superior hospital experts as the tabu list. α and β indicate the weight of pheromone concentration and heuristic information. The larger α and β are, the faster the algorithm converges, but the randomness of the search is weakened, and it is easy to fall into a local optimum.

2) Pheromone update

Due to the positive feedback effect of the ant colony algorithm, after all ants complete a cycle, in order to prevent the pheromone content on the path from being too high, the pheromone concentration on the path needs to be updated. The pheromone concentration updating mechanism is as follows:

$$\tau(i,j)=(1-\rho)\cdot\tau(i,j)+\rho\Delta\tau(i,j) \tag{11}$$

$$\Delta\tau(i,j)=\begin{cases}\sum\limits_{k=1}^{m}\Delta\tau_k(i,j), & (i,j)\in NC\\ 0, & others\end{cases} \tag{12}$$

$$\Delta\tau_k(i,j)=\begin{cases}\dfrac{Q}{L_k}, & \text{if } (i,j)\in NC\\ 0, & others\end{cases} \tag{13}$$

where ρ represents the global volatilization factor of the pheromone, $\rho\in(0,1)$. The greater ρ is, the stronger the positive feedback effect of pheromone and the faster the convergence speed, but the randomness of the algorithm search is weakened, and it is easy to fall into a local optimum. $\Delta\tau_k(i,j)$ represents the pheromone concentration released by the *k*th ant on the path between node *i* and node *j*. Q is the total pheromone released by ants in a cycle, and the size of Q has an influence on the

convergence speed of the algorithm. The larger Q is, the stronger the positive feedback mechanism is and the faster the convergence speed of the algorithm is, but the randomness of the search is weakened, and it is easy to fall into a local optimum. L_k is the path length of the kth ant.

4.5. Bacterial foraging ant colony optimization (BFACO)

In this paper, to solve the problem that the algorithm can easily fall into a local optimum, and the search speed is always slow, we improved the classical ACO algorithm.

1) Improvement of the search mechanism

The heuristic information of the search mechanism is dynamically introduced to improve the efficiency of the ant colony algorithm in effectively matching patients and experts. A patient is selected randomly as the initial point for the search process aimed at identifying a suitable match. Supposing that ant k selects patient i as the starting point for its search, the heuristic information that impacts patients and experts is represented by a new $\eta\ (i,j)$:

$$\eta(i,j) = \frac{1}{2} \cdot \left(z_i + z_j\right) \tag{14}$$

z_i and z_j represent the satisfaction degree of the patient following the matching process with the expert from the superior hospital from Eqs (3) and (4).

2) Improvement of pheromone update

Another pheromone increment is introduced into the pheromone updating formula of classical ant colony algorithm, which avoid the problem that ant colony algorithm falls into local optimization prematurely. The diffusion rate of pheromone is controlled by the coefficient of pheromone change, denoted as ρ, where $\rho \in (0,1)$. The change values of pheromone released by ants k during the process of searching pairing on the successful pairing (i,j) are represented by $\Delta\tau\ (i,j)$. The number of current successful pairings is denoted as c, while C$\in$ [0,n] represents the pheromone left by a successful pairing (i,j), and $\Delta\ \tau^*\ (i,j)$ is the pheromone increment introduced and decide by Eq (5). when a matching pheromone is too large, the next matching pheromone in the current position is fine-tuned, to avoid falling into local optimum. which changes according to the following formula:

$$\tau(i,j) = (1-\rho)\tau(i,j) + \rho\Delta\tau(i,j) + \rho\Delta\tau^*(i,j) \tag{15}$$

$$\Delta\tau^*(i,j) = \begin{cases} z_{(i.j)}, & f(i,j) \in S_k \\ 0, & others \end{cases} \tag{16}$$

3) Combination of improved ACO and bacterial foraging optimization (BFO)

In an ant colony algorithm, the search for the optimal value is achieved through pheromone feedback, resulting in strong parallel searchability and robustness. However, the convergence speed and the ability to search for the global optimal solution require improvement, as reported in [24]. BFO, on the other hand, is a bionic algorithm that emulates the behavior of E. coli parasites in the intestine, which search for food using chemotaxis, reproduction and elimination and dispersal of bacteria [25]. BFO aims to find the optimal solution using these principles.

To address the issues of slow convergence speed and the risk of falling into local optima inherent in the ant colony algorithm, this paper proposes a novel approach that combines the strengths of two

distinct algorithms. Specifically, the reproduction and chemotaxis operations of the bacterial foraging algorithm are integrated into the basic ant colony algorithm to enhance its performance.

Integration of chemotaxis of bacteria-Let θ_i *(j,r,l)* denote the current location of bacterial individual *i,* where *j* represents the number of times that bacteria perform chemotaxis, *r* denotes the number of times that bacteria replicate, and *l* signifies the number of times that bacteria migrate. Moreover, *C (i)* represents the step size of each displacement of bacteria, while Δ *(i)* indicates the direction of chemotaxis. The update of the location of bacteria *i* is governed by the following formula:

$$\theta^i(j+1,r,l) = \theta^i(j,r,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{17}$$

Integration of bacterial reproduction.

$$J^i_{health} = \sum_{i=1}^{NC+1} J(i,\ j,\ r,\ l) \tag{18}$$

In the above formula, J^i_{health} represents a robust function of bacteria, while *J (i,j,r,l)* represents the fitness of bacteria. Here, *r* denotes the number of reproductions performed by bacteria, *l* denotes the number of elimination and dispersals undertaken by bacteria, and *j* signifies the number of chemotaxis operations carried out by bacteria. NC indicates the number of chemotaxis operations performed by bacteria. Furthermore, the bacterial population is pruned by eliminating half of the individuals with low robustness, while the remaining half of the individuals undergo reproduction operations to ensure that the total number of individuals remains unchanged.

After the ant colony traverses all the nodes, each ant corresponds to a crawling trajectory, which is then mapped to the position of each bacterium. The robustness of the bacteria is inversely proportional to the sum of the trajectory length traversed by each ant. By discarding the ants with longer trajectories (corresponding to bacteria with lower robustness), the individuals with shorter trajectories are selected and replicated to expedite the convergence process.

The following steps describe the execution of the improved algorithm and pseudo-code is presented in Algorithm 1:

Step 1 Initialize the pheromone value τ *(i,j)* to 0 and the iteration number NC to 0, and randomly distribute *m* ants on *n* nodes. The ants calculate the probability of matching node *j* with the starting point *i* based on Eq (10) and select the endpoint accordingly. The trajectory is recorded in the tabu list J_k *(i)*, and the process is repeated until all initial nodes are traversed. An ant individual corresponds to a bacterial individual.

Step 2 By the proposed methodology, it is imperative to designate eligible patients and experts as S_{ki} and S_{kj}, respectively. Additionally, the path length of each ant is assigned a label of L_k, signifying the starting point of each bacterium.

Step 3 Execute bacterial reproduction.

Step 4 Subsequently, the bacterial chemotaxis action is carried out, and a comparison of the L_k value before and after the execution is conducted. If the L_k value decreases, the location is updated, and the resultant data is stored.

Step 5 The subsequent step involves computing the pheromone concentration remaining on the trajectory as per Eq (15) to Eq (16). Save the current matching initial node *i* and the endpoint *j*, and

remove the matching node j in $J_k\ (i)$. Subsequently, the cycle is resumed by returning to Step 2 until all nodes have been matched.

Step 6 In the final step of the proposed methodology, if the number of iterations does not exceed the predetermined upper limit, the process proceeds to Step 3. However, if the maximum limit is reached, the process is halted, and the best matching node is outputted.

The flowchart of the BFACO algorithm is presented in Figure 6.

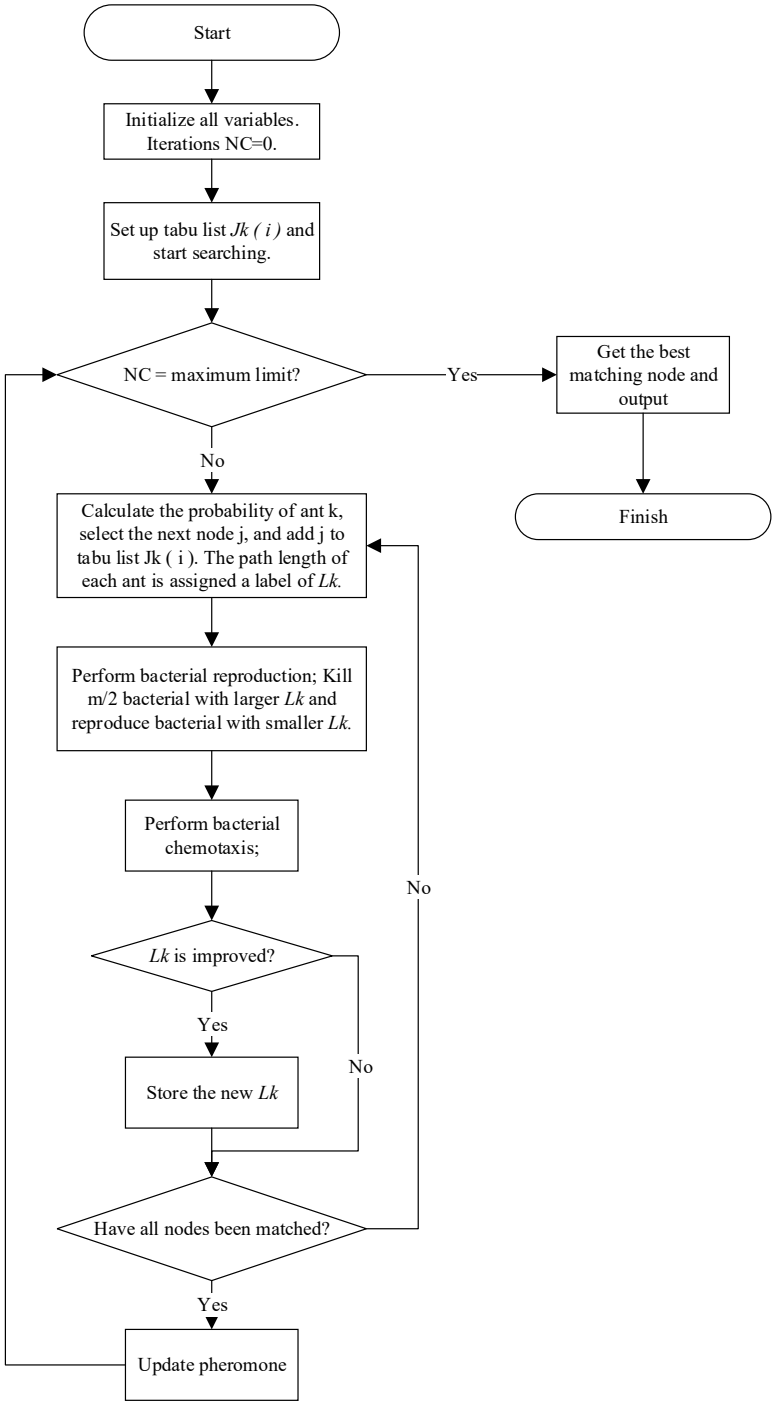

Figure 6. Flow chart of BFACO algorithm.

Algorithm 1 Bacterial Foraging-ACO

Input: ant number m**,** iteration number maxlimit, idle time vector of each patient **X**, idle time vector of experts in the superior hospital **Y**

Output: the best matching node

1: Initialize the parameters of the algorithm; // iteration **NC**; pheromone value τ *(i,j)*; tabu list J_k *(i)*

2: Calculate the satisfaction degree of the patient z_i by Eq (3), the satisfaction degree of the experts z_j by Eq (4), overall preference value $z_{(i.j)}$ by Eq (5);

3: Randomly distribute m ants on n nodes, an ant individual corresponds to a bacterial individual;

4: WHILE (**NC** < maxlimit)

5: WHILE (all nodes not matched)

6: FOR (every ant)

7: Calculate the heuristic information $\eta(i,j)$ by Eq (14);

8: Calculate the matching probability of ant k by Eq (10), select the end node j;

9**:** Add ***j*** to J_k *(i)*;

10: Calculate the path length L_k from *i* to *j*;

11**:** END

12: Perform bacterial reproduction: Sort bacteria (ant) in descending order of L_k, kill m/2 bacterial (ant) with Larger L_k and reproduce bacterial (ant) with smaller L_k.

13: Perform bacterial chemotaxis;

14: If the path length is improved

15: Update L_k;

16: Pheromone update by Eq (12), Eq (13), Eq (15) and Eq (16);

17: Update J_k *(i)* using current matching node *j*;

18: END

19: **NC**+1;

20: END

21: Return J_k *(i)* with the best matching node;

4.6. Process of smart contract algorithm for remote consultation based on improved ant colony algorithm

Based on the decentralized remote consultation contract model of blockchain and the swarm intelligence algorithm, this paper proposes a multi-party smart contract algorithm for remote consultation. The algorithm can be executed under the condition that patients and experts conform to the terms and obligations stipulated in the consultation service. Figure 7 illustrates the implementation process of the algorithm.

1) Information dissemination: The Management Center engages in the dissemination of information by publishing expert information that has been subjected to an audit on the remote consultation platform.

2) Consultation application: The proposed consultation application entails the submission of pertinent medical information by patients, which subsequently awaits platform review. To elaborate, patients will provide relevant health data via the application interface, after which the platform will conduct a thorough assessment of the submitted information.

3) Consultation matching: The process of consultation matching involves the utilization of an improved ant colony algorithm to match patients and experts who have passed the information audit on the platform. Specifically, the algorithm is applied to match the relevant medical expertise of the available experts with the specific health needs of the patients, based on the information provided by both parties.

4) Contract establishment: Upon the successful matching of a patient with an expert, a contract is automatically established. As part of this process, the patient is required to prepay the consultation fee to the designated contract address. Subsequently, the consultation expert is obliged to fulfill the contractual requirements by delivering the service as stipulated. Finally, the two parties are expected to utilize the digital signature key to ensure the authenticity and security of the transaction.

5) Implementation and preservation of the contract: The implementation and preservation of the contract entail the completion of consensus and testing of each block, which automatically triggers the execution of the contract. Upon completion of the consultation service, the cost of the service is cleared as per the terms of the contract.

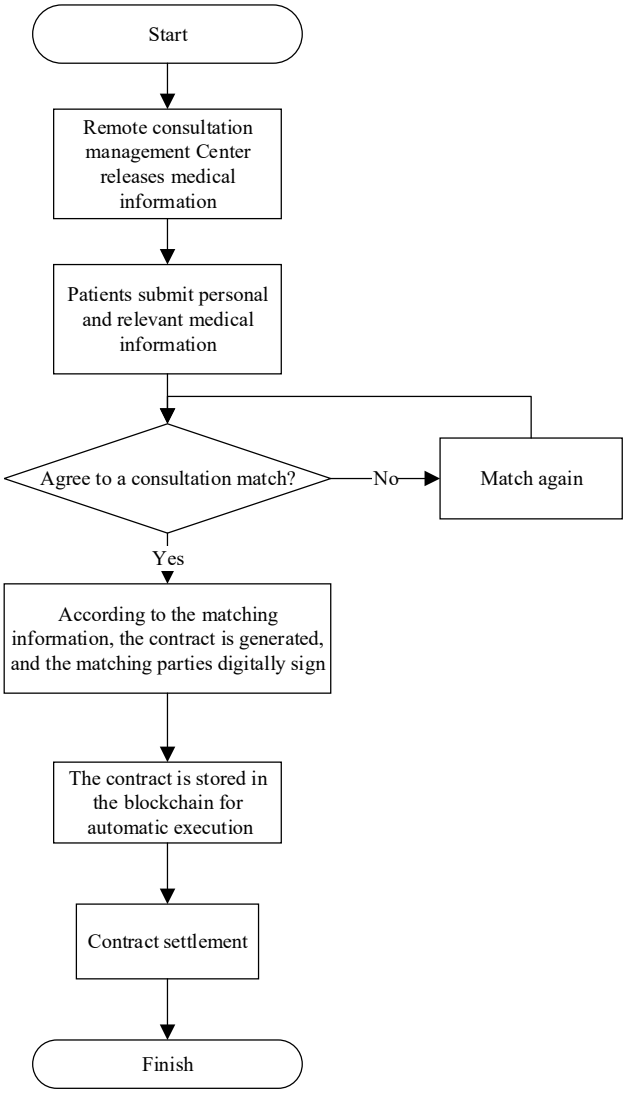

Figure 7. Flow chart of remote consultation smart contract algorithm based on improved ant colony algorithm.

5. Experiments and analysis of improved ant colony algorithm

This paper aims to verify the performance and effectiveness of the BFACO through an optimization simulation that examines two aspects of the problem: the Traveling Salesman Problem (TSP) and function optimization.

5.1. Solution and result analysis of TSP

The TSP is a widely recognized and classical combinatorial optimization problem [26]. The problem involves determining the shortest path for a traveler who visits n cities and returns to the origin, subject to the condition that each city can only be visited once. The model of the TSP problem can be simply described: Given an undirected graph $G = (V,E)$, in which each edge is $e \in E$, and there is a non-negative weight $d(i,j)$ representing the distance between nodes, find the loop that traverses all nodes in the graph G and has the minimum total weight. In this study, to solve the discrete problem, the position P of each bacterial (ant) individual corresponds to a path sequence. Path coding is employed to encode the problem [27], that is, each bacterial (ant) individual represents a path sequence, the dimensions are consistent with the number of cities, and the numerical value of each dimension represents the city number of the position. Let $N = \{1,2,...,n\}$ represent the set of all site numbers, and the code of each bacterial (ant) individual is a random arrangement of elements in set N, $P = (p_1,p_2,... p_n)$.

To evaluate the efficacy of the bacterial foraging ant colony algorithm in addressing the TSP, this study employed the TSPLIB standard library to simulate seven classical datasets (Eli51, Eli76, Rat99, Berlin52, St70, D657 and Nrw1379) using Matlab 2016b. The optimization outcomes obtained through Chaos Ant Colony Optimization (CACO) [28], Simulated Annealing Ant Colony Optimization (SAACO) [29] and Levy flight Ant Colony Optimization (Levy-ACO) [30] were compared and analyzed alongside those obtained through the ACO and BFACO.

During the testing phase, we established the maximum iteration number NC_{max} as 200, while setting the ant colony size m as $1.5n$. The algorithm parameters were selected based on the guidance provided in [31]. Specifically, we set the pheromone importance factor α to 1, the heuristic function importance factor β to 5, the global pheromone evaporation factor ρ to 0.5 and the total pheromone release Q to 100.

Table 1. Experimental setup.

Parameter	Value
Maximum number of iterations, NC_{max}	200
Scale, m	1.5n
Pheromone importance factor, α	1
Heuristic function importance factor, β	5
Pheromone global volatilization factor, ρ	0.5
Total pheromone release, Q	100

1) Algorithm optimization ability

The algorithm's optimization ability was evaluated based on two metrics: the test optimal value and the average value. To conduct this evaluation, we solved the TSP for seven classical datasets,

including Eli51, Eli76, Berlin52, St70, Rat99, D657 and Nrw1379, using the ACO, CACO, SAACO, Levy-ACO and BFACO algorithms. This process was repeated for 20 consecutive runs. The optimal solution is known and was obtained from the TSPLIB standard database. Meanwhile, the test optimal solution is the shortest path length out of the 20 obtained solutions, while the average value is the average path length across all 20 solutions. The results are presented in Tables 2 and 3.

The results of the optimization simulation are presented in Tables 2 and 3, which demonstrate the algorithm's optimization ability in terms of test optimal value and average value. The tables show the outcomes of solving Eli51, Eli76, Berlin52, St70, Rat99, D657 and Nrw1379 with ACO, CACO, SAACO, Levy-ACO and BFACO over 20 consecutive runs. The known optimal value provided in the TSPLIB standard database serves as the benchmark for the optimal solution. The test optimal solution represents the shortest path length among 20 solutions, while the average value represents the average path length of the 20 solutions.

Tables 2 and 3 reveal that under the same basic parameter settings, BFACO consistently outperforms ACO and the three improved ant colony algorithms in terms of optimal solution, worst solution and average value. For instance, in Eli51, BFACO yields a test optimal solution and average value that are 4.9 and 5.8 less than ACO, respectively. Similarly, in Eli76, BFACO produces a test optimal solution and average values that are 1.9 and 4.1 less than ACO, respectively. In Rat99, BFACO yields a test optimal solution and average value that are 8.687 and 15.4 less than ACO, respectively. For St70, BFACO achieves a test optimal solution and average value that are 10.4 and 16.7 less than ACO, respectively. For Berlin52, BFACO achieves a test optimal solution and average value that are 117.6 and 104.3 less than ACO, respectively. However, for large-scale problems such as D657 and Nrw1379, the gap between the test optimal solution and the known optimal solution is considerable. Specifically, in D657, the test optimal solution and average value obtained by BFACO are 1023.9 and 1114.5 less than ACO, respectively. In Nrw1379, BFACO yields a test optimal solution and average value that are 2019.5 and 2405.4 less than ACO, respectively. This finding suggests that the bacterial foraging ant colony algorithm exhibits a weaker optimization ability for large-scale problems.

In summary, the bacterial foraging ant colony algorithm surpasses ACO and three improved ant colony algorithms in terms of optimization ability, as evidenced by the better optimal solution, worst solution and average value. Additionally, the algorithm can escape local optima, making it more effective. However, for large-scale problems, the optimization performance of the bacterial foraging ant colony algorithm is relatively poor.

Table 2. Experimental results of TSP (optimal solution).

TSP	Optimal solution	Test the optimal solution				
		ACO	CACO	SAACO	Levy-ACO	BFACO
Eli51	426	446.156	443.268	442.512	442.532	441.253
Eli76	538	568.639	567.325	566.993	566.854	566.775
St70	675	699.565	694.315	691.346	690.733	689.179
Berlin52	7542	7663.585	7596.356	7559.431	7553.221	7548.993
Rat99	1211	1300.896	1296.489	1294.378	1293.344	1292.299
D657	48912	56627.013	56007.121	55987.911	55704.633	55603.046
Nrw1379	56638	66960.642	65589.369	65121.461	650345,764	64941.148

Table 3. Experimental results of TSP (mean value).

TSP	Optimal solution	Mean value				
		ACO	CACO	SAACO	Levy-ACO	BFACO
Eli51	426	449.711	445.272	445.052	444.076	443.867
Eli76	538	571.933	569.311	568.521	568.342	567.869
St70	675	712.321	702.698	697.883	695.833	695.633
Berlin52	7542	7707.768	7619.458	7597.981	7587.822	7603.512
Rat99	1211	1316.707	1308.357	1305.731	1303.662	1301.334
D657	48912	57142.719	56592.082	56397.364	56239.348	56028.209
Nrw1379	56638	68093.165	66162.813	66231.551	65778.672	65687.724

2) Algorithm reliability and convergence speed

In terms of evaluating the reliability and convergence speed of the algorithm, success rate and mean convergence generations are used as indicators. The success rate represents the ratio of the number of times the algorithm reaches the critical value to the total number of runs based on a preset critical value. Tables 4 and 5 present the comparison results of the reliability and convergence rate obtained by ACO, CACO, SAACO, Levy-ACO and BFACO for 20 consecutive solutions to Eli51, Eli76, Rat99, Berlin52, St70, D657 and Nrw1379. As shown in Table 4, BFACO demonstrates superior reliability, with a 100% success rate for solving the seven test data sets. This performance is significantly better than that of ACO and also superior to those of CACO, SAACO and Levy-ACO. Moreover, as shown in Table 5, the mean convergence generations of the BFACO algorithm are smaller than those of ACO and also smaller than those of CACO, SAACO and Levy-ACO in terms of convergence rate. Therefore, it can be concluded that the bacterial foraging ant colony algorithm outperforms the basic ant colony algorithm, chaotic ant colony algorithm, simulated annealing ant colony algorithm and Levy flight ant colony algorithm in terms of reliability and convergence speed.

Table 4. Comparison results of reliability and convergence rate (success rate).

TSP	Success rate / %				
	ACO	CACO	SAACO	Levy-ACO	BFACO
Eli51	40	75	80	85	100
Eli76	35	55	70	90	100
St70	10	55	70	85	100
Berlin52	45	65	75	90	100
Rat99	20	60	70	85	100
D657	35	70	85	5	100
Nrw1379	10	75	80	10	100

Table 5. Comparison results of reliability and convergence rate (mean convergence generations).

TSP	Mean convergence generations				
	ACO	CACO	SAACO	Levy-ACO	BFACO
Eli51	117.1	75.3	69.5	64.9	64
Eli76	109.2	61.5	55.3	42.0	45.9
St70	107.5	85.2	86.5	84.2	83.8
Berlin52	90.1	60.4	58.0	51.0	54.0
Rat99	107.3	86.3	78.1	75.9	74.8
D657	120.5	84.2	85.3	82.7	82.5
Nrw1379	136.2	102.4	100.5	101.9	97.3

3) Algorithm stability

The stability of an algorithm is typically evaluated by its standard deviation, which measures the degree of variation of each solution. To assess the stability of ACO, CACO, SAACO, Levy-ACO and BFACO, Table 6 presents the standard deviation comparison results obtained from solving Eli51, Eli76, Rat99, Berlin52, St70, D657 and Nrw1379 after 20 consecutive runs.

Table 6 presents the standard deviation comparison results calculated by ACO, CACO, SAACO, Levy-ACO and BFACO after 20 consecutive solutions to Eli51, Eli76, Rat99, Berlin52, St70, D657 and Nrw1379. The analysis reveals that the standard deviation of BFACO for solving the seven test data sets is smaller than that of ACO, and it is also superior to those of CACO, SAACO and Levy-ACO. These results demonstrate that the bacterial foraging ant colony algorithm exhibits superior stability compared to the basic ant colony algorithm, chaotic ant colony algorithm, simulated annealing ant colony algorithm and Levy flight ant colony algorithm.

Table 6: Comparison of stabilities.

Algorithm	Standard deviation						
	Eli51	Eli76	Rat99	St70	Berlin52	D657	Nrw1379
ACO	3.85	1.90	7.81	4.31	51.49	310.83	586.65
CACO	3.12	1.15	6.81	3.48	31.23	159.62	329.36
SAACO	2.96	1.02	6.79	3.51	27.61	163.46	338.25
Levy-ACO	2.87	0.97	6.72	3.32	25.90	157.33	327.24
BFACO	2.83	0.95	6.72	3.14	25.54	141.85	306.94

5.2. Test and result analysis of function optimization problem

To evaluate the effectiveness of the bacterial foraging ant colony algorithm in solving function optimization problems, two commonly used test functions, the Rastrigin function and the Schaffer function, were selected and solved [32]. The expressions of these test functions are presented below.

$$minf(x_i) = \sum_{i=1}^{2} [x_i^2 - 10\cos(2\pi x_i) + 10] \tag{19}$$

$$minf(x_1, x_2) = 0.5 + \frac{(\sin\sqrt{x_1^2 + x_2^2})^2 - 0.5}{[1 + 0.01(x_1^2 + x_2^2)]^2} \tag{20}$$

Equation (19) represents the Rastrigin function, where xi is bounded within the range of [−5.12,5.12]. Equation (20) represents the Schaffer function, where x1 and x2 are bounded within the range of [−10,10].

The algorithm parameters are configured as follows: NC_{max}, the maximum iteration number, is set to 100; m, the ant colony scale, is set to 100; α, the pheromone importance factor, is set to 1; β, the heuristic function importance factor, is set to 5; ρ, the global pheromone volatilization factor, is set to 0.5; and Q, the total pheromone release, is set to 100.

To optimize the Rastrigin function and Schaffer function, the basic ant colony algorithm and bacterial foraging ant colony algorithm were utilized for 20 iterations. The outcomes of these iterations are presented in Table 7.

Table 7. Test results of function optimization.

Test function	Optimal solution		Average solution		Standard deviation	
	ACO	BFACO	ACO	BFACO	ACO	BFACO
Rastrigin	0.00689	0.00000	0.09505	0.00216	0.14800	0.00261
Schaffer	0.00016	0.00000	0.00412	0.00015	0.04660	0.00018

From Table 7, it is apparent that the optimal solution and average solution values obtained by the bacterial foraging ant colony algorithm for both the Rastrigin function and the Schaffer function are less than those obtained by the basic ant colony algorithm. Furthermore, the standard deviation computed by the bacterial foraging ant colony algorithm is smaller than that of the basic ant colony algorithm. These results indicate that the bacterial foraging ant colony algorithm exhibits better optimization ability and stability than the basic ant colony algorithm for function optimization problems.

Figure 8 illustrates the comparison of optimization convergence between the basic ant colony algorithm and bacterial foraging ant colony algorithm for the Rastrigin function, while Figure 9 presents the optimization convergence comparison between the basic ant colony algorithm and the bacterial foraging ant colony algorithm for the Schaffer function.

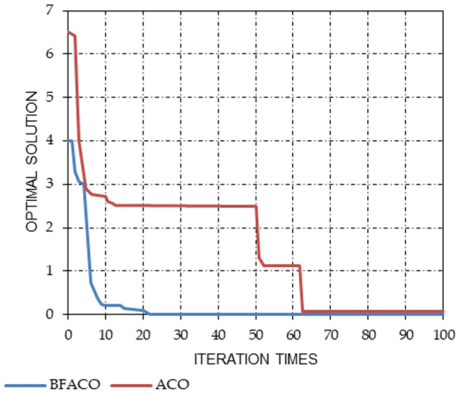

Figure 8. Comparison of optimal convergence process (Rastrigin).

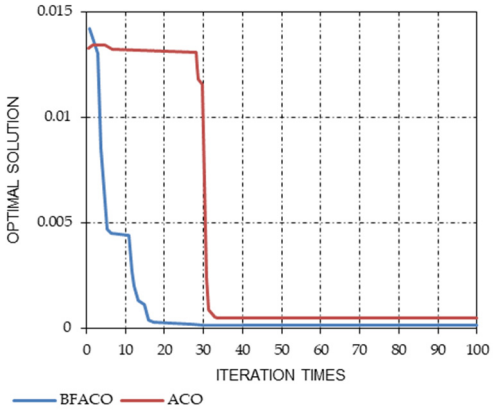

Figure 9. Comparison of optimal convergence process (Schaffer).

Based on Figures 8 and 9, it is apparent that the Bacterial Foraging Ant Colony Algorithm exhibits superior optimization ability and convergence efficiency compared to the basic ant colony algorithm when addressing function optimization problems.

6. Application of improved ant colony algorithm

In this study, the feasibility of the enhanced ant colony algorithm is investigated using a multi-node remote consultation service simulation platform. The Solidity programming language is employed to develop the computer protocol code, and the resulting smart contracts are tested on the Ethereum platform to assess their efficacy.

In the context of the Ethernet virtual machine, the value assigned to the number of ants N_m is 20, with a density parameter of ρ = 0.8 and an iterative upper limit of 200. Tables 9 and 10 present statistical information concerning 10 patients seeking remote consultation services $(Y_1,Y_2,...,Y_{10})$ and 10 expert consultants affiliated with superior hospitals $(X_1,X_2,...,X_{10})$. The relevant indicator information includes the blockchain-based addresses of patients and consultation experts in the higher hospital, the cost of consultation *(Bp/eth)*, the duration of consultation *(Bt/h)* and the number of consultations *(Bq/person)*. *Bp* and *Bt* are parameters of particular interest to patients, and their corresponding weights have been assigned as w_i = (0.50,0.50). On the other hand, *Bp*, *Bt* and *Bq* are three indicators of concern to consultation experts, with respective weights of w_j = (0.6,0.15,0.25). The simulation parameters table and the statistical results about the willingness of patients to be matched with consultation experts are presented in the following Table 8:

Table 8. Simulation parameters.

Parameter	Value
Number of ants, N_m	20
Pheromone global volatilization factor, ρ	0.8
Iteration limit	200
Patient's demand weight, w_i	(0.50,0.50)
Expert's demand weight, w_j	(0.6,0.15,0.25)

Table 9. Statistical table of patients' matching intention tendency.

B_a		B_p/eth	B_t/h	B_q/person
Y_1	0xb005804a49e73acb17dle7645dfd0a33dde6eb0e	0.62580–0.81355	3	10
Y_2	0x7d577a597b2742b498cb5cf0c26cdcd726d39e6e	1.37677–1.62709	4	13
Y_3	0xc4b1bd9a0bb5a7ddd3e8eeb6d50b411804de70c7	2.06520–2.37810	3	18
Y_4	0xc8bd569f5bea0ad73d6c72ed21c84aa16a55be8c8	1.75225–1.87741	3	15
Y_5	0xbd54e8da7ad5b83b8a3b1a865bd3e8bbd5e4dc92	2.50320–2.81610	2	15
Y_6	0x2cce1a0c40e3505abb67d4e96db3d78c9eb7e52c	2.87870–3.12900	4	20
Y_7	0x5b1d0d0bc4e520c0e720ee102b1bd9bbe25ad445	1.56451–1.87741	2	12
Y_8	0x9de3ccb9d8bb7dd07ca8ec083e9a3e0c4a2b838f	0.87613–1.06387	3	8
Y_9	0x0db8eca9ba2e90ce305a8c6a33cd00d4eaa634a9	1.12645–1.37677	3	5
Y_{10}	0xa436d480ea4ebe9e71eae7b3703ac3c3a3dd338a	1.31419–1.81483	4	10

Table 10. Statistical table of the matching willingness of consultation experts.

B_a		B_p/eth	B_t/h	B_q/ person
X_1	0x62b1746767522b36f6421e630fa0198151d72964	0.71967–0.93871	2–3	8–12
X_2	0x115ced3f8b7ea92d324902e3a3a421a07540eb2b	1.50193–1.87741	2–4	10–15
X_3	0x6a248b2dd2ecdabbc25ccc507558ddebde7adcda	2.56580–2.94130	2–3	17–25
X_4	0x65c26319c9b34418aeb6ee13a8b8ea40eed82ec4	1.56451–1.87741	2–3	13–18
X_5	0xdadcc96e78b9bec5891dba90aab5aa7510ee200d	1.25161–1.62709	2–4	10–17
X_6	0x82a978b3f5962a5b0957d9ee9eef472ee55b42f1	1.00129–1.43935	2–4	8–13
X_7	0x4b1334c2b9daba17341d3e1bba0d7dc062b3abed	1.75225–2.25290	2–5	9–15
X_8	0xe0c026aeb8c8e10d11ac93a21cc3b0ecd07cc376	2.75350–3.25420	2–3	20–27
X_9	0xb0cac9c060aa6a5de1b633415a50b9b8dd8a1ead	2.12770–2.44060	2–3	17–24
X_{10}	0x6dce02b24cdc2e7e6c4dc1acda5c0b6d82d4dc65	1.50193–1.75225	2–6	10–18

The utilization of the enhanced ant colony algorithm enabled matching on the virtual machine, yielding the results and overall preference values presented in Table 11. The data indicate a high level of satisfaction among both patients and experts upon completion of the matching process. Following this outcome, the algorithm initiates contract execution automatically.

Table 11. Matching results and overall preference values.

Matching results	Overall preference value
Y_1–X_1	0.103
Y_1–X_1	0.263
Y_1–X_1	0.164
Y_1–X_1	0.220
Y_1–X_1	0.428
Y_1–X_1	0.176
Y_1–X_1	0.219
Y_1–X_1	0.158
Y_1–X_1	0.182
Y_1–X_1	0.339

Through iterative tests on the simulation platform, the performances of the classical ant colony algorithm, improved ant colony algorithm, basic particle swarm optimization (PSO) [33] and parallel dynamic weigh niches artificial fish swarm (PDN-AFS) [34] algorithm are compared and analyzed, without altering the initial parameters.

Based on the comparison results presented in Figure 10, under the same parameters, the curve of the improved ant colony algorithm fluctuates smoothly with the increasing number of iterations, and it tends to be stable earlier than the other three algorithms. Therefore, the improved ant colony algorithm has better performance than the other three algorithms and is more stable.

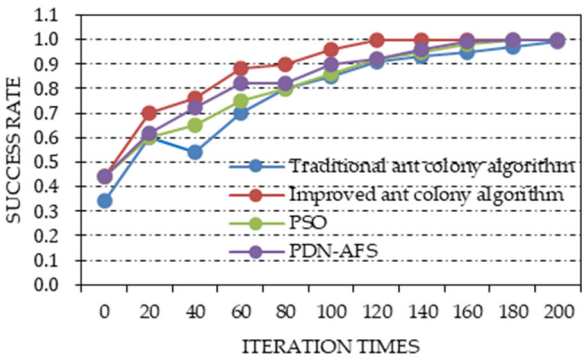

Figure 10. Performance comparison chart of four algorithms.

Figure 11 shows the change curves of the global optimal mean of three objective functions, Eqs (3)–(5), under the comparative experiments of improved ant colony algorithm, classical ACO, PSO and PDN-AFS algorithm. As can be seen from Figure 11, the global optimal mean of the three objective functions of the improved ant colony algorithm has a decreasing trend compared with the calculation results of the classical ant colony algorithm. The classical ant colony algorithm and the PSO have unstable fluctuations compared with the improved ant colony algorithm in the process of finding the optimal solution from the change curve, and the iterative effect of the PDN-AFS algorithm in the early stage is not very ideal. Therefore, the improved ant colony algorithm obtains the better global optimal average value of three objective functions, and the iterative process is more stable, which can prove that the improved ant colony algorithm keeps finding effective global optimal solutions in the iterative process.

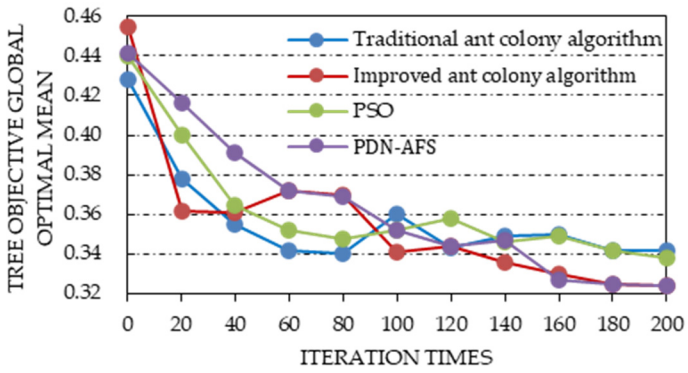

Figure 11. Variation curves of the global optimal mean of the three objectives.

The smart contract algorithm introduced in this study was implemented in a peer-to-peer network, and the execution of contracts within a seven-day timeframe was analyzed. The results in Table 12 showed a contract scrap proportion of only 0.96% and a matching success rate of 99.02% in the contracts that were established. Furthermore, the efficiency of the service execution was substantially enhanced.

Table 12. Smart contract signing status.

Date	Patients	Experts	Number of successful match contract signings	Number of unsuccessful matches	Number of expired contracts
2023-04-03	10	10	10	0	0
2023-04-04	100	150	94	6	4
2023-04-05	302	208	306	2	6
2023-04-06	727	716	716	11	5
2023-04-07	529	458	529	0	8
2023-04-08	635	648	630	5	2
2023-04-09	859	782	846	13	5
Total	3162	2972	3131	37	30

7. Conclusions

With the maturity of telemedicine service, more and more transactions will be carried out through the intelligent platform of network information. The integration of blockchain and artificial intelligence provides a new development direction for telemedicine services. In this paper, based on blockchain and swarm intelligence technology, an improved ACO algorithm based on bacterial foraging is used to combine the conditional attributes of the patient and the consultation experts in the superior hospital, so as to obtain a high degree of fit between the two parties and realize the automatic management of consultation services. According to the experimental results, the ACO algorithm based on bacterial foraging is effective in dealing with multi-objective matching problems.

However, the proposed model was limited by the fields of smart contracts and blockchain, and it is still in the initial stage of exploration. The performance and efficiency of the transaction matching algorithm can be further improved by integrating it with other heuristic algorithms. In addition, this paper only solves the common problems such as decentralization and intelligence of remote consultation, but in the distributed, dynamic and open big data environment with increasing transaction data of remote consultation, how to solve the problem of massive storage for blockchain is the focus of future research.

Use of AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

Acknowledgments

The study was supported by the 2022 Guangxi University Young and Middle-aged Teachers' Basic Scientific Research Ability Improvement Project (No. 2022KY0293).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. A. D. Dwivedi, R. Singh, S. Dhall, G. Srivastava, S. K. Pal, Tracing the source of fake news using a scalable blockchain distributed network, in *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, (2020), 38–43. https://doi.org/10.1109/MASS50613.2020.00015
2. G. Srivastava, J. Crichigno, S. Dhar, A light and secure healthcare blockchain for IoT medical devices, in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, 1–5. https://doi.org/10.1109/CCECE.2019.8861593
3. Z. Chen, Z. Jin, J. Liu, Research and application of remote consultation support system based on the information exchange platform, *China Digital Med.*, **9** (2014), 88–90. https://doi.org/10.3969/j.issn.1673-7571.2014.03.026
4. L. Wang, Thoughts on how telemedicine can promote regional medical informatization construction, *Chongqing Med.*, **40** (2011), 3574–3575. https://doi.org/10.3969/j.issn.1671-8348.2011.35.018
5. A. I. Hernández, F. Mora, M. Villegas, G. Passariello, G. Carrault, Real-time ECG transmission via the Internet for nonclinical applications, *IEEE Tran. Inf. Technol. Biomed.*, **5** (2001), 253–257. https://doi.org/10.1109/4233.945297
6. J. Guo, Y. Bao, M. Jing, Building of multipath remote consultation platform based on medical alliance, *J. Med. Intell.*, **39** (2018), 22–25. https://doi.org/10.3969/j.issn.1673-6036.2018.01.005
7. K. Fan, S. Wang, Y. Ren, H. Li, Y. Yang, MedBlock: efficient and secure medical data sharing via blockchain, *J. Med. Syst.*, **42** (2018), 1–11. https://doi.org/10.1007/s10916-018-0993-7
8. J. Liu, X. Li, L. Ye, H. Zhang, X. Du, M. Guizani, BPDS: a blockchain-based privacy-preserving data sharing for electronic medical records, in *2018 IEEE Global Communications Conference (GLOBECOM)*, (2018), 1–6. https://doi.org/10.1109/glocom.2018.8647713
9. J. Zhang, Q. Xia, Y. Zhao, Research on electronic medical record data storage system based on blockchain technology, *China Med. Devices*, **36** (2021), 106–109. https://doi.org/10.3969/j.issn.1674-1633.2021.07.024
10. K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, T. Hayajneh. Healthcare blockchain system using smart contracts for secure automated remote patient monitoring, *J. Med. Syst.*, **42** (2018), 1–7. https://doi.org/10.1007/s10916-018-0982-x
11. Z. Shae, J. J. P. Tsai, On the design of a blockchain platform for clinical trial and precision medicine, in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, (2017), 1972–1980. https://doi.org/10.1109/ICDCS.2017.61
12. K. Christidis, M. M. Devetsikiotis, Blockchains and smart contracts for the internet of things, *IEEE Access*, **4** (2016), 2292–2303. https://doi.org/10.1109/access.2016.2566339
13. J. Moosavi, L.M. Naeni, A. M. Fathollahi-Fard, U. Fiore, Blockchain in supply chain management: A review, bibliometric, and network analysis, *Environ. Sci. Pollut. Res.*, (2021), 1–15. https://doi.org/10.1007/s11356-021-13094-3

14. K. S. Alshudukhi, M. A. Khemakhem, F. E. Eassa, K. M. Jambi, An interoperable blockchain security frameworks based on microservices and smart contract in IoT environment, *Electronics*, **12** (2023), 776. https://doi.org/10.3390/electronics12030776
15. J. Liao, J. Yang, W. Zhu, L. Cai, L. Han, H. Wang, Design and application of telemedicine consultation system, *Pract. J. Clin. Med.*, **8** (2011), 206–208. https://doi.org/10.3969/j.issn.1672-6170.2011.06.078
16. Y. Zhou, H. Lin, Q. Geng, Y. Zhang, Constitution and new development of remote medical system, *China Digital Med.*, **4** (2009), 21–23. https://doi.org/10.3969/j.issn.1673-7571.2009.09.005
17. Y. Chen, X. Zhang, X. Zou, Q. Ye, L. Nie, Y. Chen, et al., Effect of telemedicine on improving medical service capacity in the county, *Chin. J. Hosp. Adm.*, **30** (2014), 408–410. https://doi.org/10.3760/cma.j.issn.1000-6672.2014.06.003
18. Y. Mao, L. Zhang, Optimization of the medical service consultation system based on the artificial intelligence of the internet of things, *IEEE Access*, **9** (2021), 98261–98274. https://doi.org/10.1109/access.2021.3096188
19. Y. Bachrach, R. Everett, E. Hughes, A. Lazaridou, J. Z. Leibo, M. Lanctot, et al., Negotiating team formation using deep reinforcement learning, *Artif. Intell.*, **288** (2020), 103356. https://doi.org/10.1016/j.artint.2020.103356
20. P. Bharti, R. Ranjan, B. Prasad, Broker-based optimization of SLA negotiations in cloud computing, *Multiagent Grid Syst.*, **17** (2021), 179–195. https://doi.org/10.3233/MGS-210349
21. C. Roy, S. Misra, J. Maiti, U. Chakravarty, Safe-serv: Energy-efficient decision delivery for provisioning safety-as-a-service, *IEEE Trans. Serv. Comput.*, **15** (2020), 1954–1966. https://doi.org/10.1109/TSC.2020.3026135
22. M. Bai, X. Wang, J. Xin, G. Wang, An efficient algorithm for distributed density-based outlier detection on big data, *Neurocomputing*, **181** (2016), 19–28. https://doi.org/10.1016/j.neucom.2015.05.135
23. N. Atzei, M. Bartoletti, T. Cimoli, A survey of attacks on Ethereum smart contracts (sok), in *International Conference on Principles of Security and Trust*, **10204** (2017), 164–186. https://doi.org/10.1007/978-3-662-54455-6_8
24. L. Wang, M. Li, Z Liu, Application of an ant colony optimization based on attractive field in TSP, *J. Jiangsu Univ. Nat. Sci. Ed.*, **36** (2015), 573–577. https://doi.org/10.3969/j.issn.1671-7775.2015.05.014
25. Z. Zhou, J. Yang, L. Ma, A hybrid bacteria foraging algorithm for solving the traveling salesman problem, *Math. Pract. Theory*, **45** (2015), 159–165.
26. K. Zhou, X. Qiang, X. Tong, J. Xu, Algorithm of TSP, *Comput. Eng. Appl.*, **43** (2007), 43–47. https://doi.org/10.3321/j.issn:1002-8331.2007.29.013
27. M. Mojtahedi, A. M. Fathollahi-Fard, R. Tavakkoli-Moghaddam, S. Newton, Sustainable vehicle routing problem for coordinated solid waste management. *J. Ind. Inf. Integr.*, **23** (2021), 100220. https://doi.org/10.1016/j.jii.2021.100220
28. L. Zhang, T. Fei, T. Liu, J. Zhang, Application in medical device emergency logistics distribution routing optimization based on chaos ant colony optimization, *J. Civ. Aviat. Univ. China*, **29** (2011), 61–64. https://doi.org/10.3969/j.issn.1001-5590.2011.03.015
29. L. Zhang, Y. Wang, T. Fei, X. Zhou, Research on low carbon logistics routing optimization based on chaotic-simulated annealing ant colony algorithm, *Comput. Eng. Appl.*, **53** (2017), 63–68. https://doi.org/10.3778/j.issn.1002-8331.1503-0167

30. Y. Liu, B. Cao, A novel ant colony optimization algorithm with Levy flight, *IEEE Access*, **8** (2020) 67205–67213. https://doi.org/10.1109/ACCESS.2020.2985498
31. Z. Ye, Z. Zheng, Configuration of parameters α, β, ρ in ant algorithm, *Geomatics Inf. Sci. Wuhan Univ.*, (2004), 597–601.
32. T. Fei, L. Zhang, Y. Bai, L. Chen, Improved artificial fish swarm algorithm based on DNA, *J. Tianjin Univ. Sci. Technol.*, **49** (2016), 581–588.
33. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95-International Conference on Neural Networks*, (1995), 1942–1948.
34. S. Li, W. Li, H. Shun, Z. Lin, Artificial fish swarm parallel algorithm based on multi-core cluster, *J. Comput. Appl.*, **33**, (2013), 3380–3384.

AIMS Press