Optimized Resource Allocation Algorithm for Crowd-Creation Space Computing Based on Cloud Computing Environment

Mustafa El-Taie [1], Aaras Y.Kraidi [2]

[1] Digital Charging Solutions GmbH, Germany
[2] University of Technology and Applied Science, Shinas, Oman
Email: Mustafa.iessa@gmail.com, aaras.kraidi@shct.edu.om

Abstract:

The crowd-creation space is a manifestation of the development of innovation theory to a certain stage. With the creation of the crowd-creation space, the problem of optimizing the resource allocation of the crowd-creation space has become a research hotspot. The emergence of cloud computing provides a new idea for solving the problem of resource allocation. Common cloud computing resource allocation algorithms include genetic algorithms, simulated annealing algorithms, and ant colony algorithms. These algorithms have their obvious shortcomings, which are not conducive to solving the problem of optimal resource allocation for crowd-creation space computing. Based on this, this paper proposes a In the cloud computing environment, the algorithm for optimizing resource allocation for crowd-creation space computing adopts a combination of genetic algorithm and ant colony algorithm, and optimizes it by citing some mechanisms of simulated annealing algorithm. The algorithm in this paper is improved genetic ant colony algorithm (HGAACO). In this paper, the feasibility of the algorithm is verified through experiments. The experimental results show that with 20 tasks, the ant colony algorithm task allocation time is 93ms, the genetic ant colony algorithm time is 90ms, and the improved algorithm task allocation time proposed in this paper 74ms, obviously superior. The algorithm proposed in this paper has a certain reference value for solving the creative space computing optimization resource allocation.

Keywords: Cloud Computing, Crowd Creation Space, Optimized Resource Allocation, Algorithm Optimization, Improved Genetic Ant Colony Algorithm

1. Introduction

What is the crowd-creation space? At present, there is no unified definition for all sectors of industry, academia and research. In summary, there are two definitions. One is to define the creative space from a functional perspective. Some scholars believe that Zhongchuang Space is a new type of comprehensive service platform in the Internet era, which should have four basic functions of integrating innovation resources, improving entrepreneurial efficiency, promoting creative culture, and online and offline communication. Other scholars believe that the crowd-creation space is a further expansion of the functions of traditional incubators. Compared with traditional incubators, the barriers to entry of

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

Zhongchuang Space are lower, which can better meet the demands of the general public for innovation and entrepreneurship. The essence is an innovative incubator that caters to elite innovation to mass innovation, internal closed to open collaboration, technological leadership to demand-oriented processes. The concept of crowd-creation space was proposed and soon, the resource allocation of crowd-creation space has always been the focus of research. How to reasonably and efficiently allocate user jobs to resource nodes is a problem that needs to be solved in the resource allocation of Zhongchuang. In recent years, with the widespread application of information technology in various major fields, the amount of data information on the Internet has also shown explosive growth. The Internet has entered the "big data era", and traditional computing models can no longer meet the current changing dynamics. demand. Based on the above development background, the emerging technology of "cloud computing" came into being. Cloud computing is a commercial implementation that integrates multiple computing modes such as distributed computing, parallel computing, and style computing. From a technical perspective, its essence is the virtualization of IT software and hardware resources such as computing services, storage services, servers, and application software. Cloud computing services include virtual machine technology, data storage technology, data management technology, distributed programming and computing technology in applications.

The rise of cloud computing is of great help to solve the problem of space allocation of Zhongchuang. In general, cloud computing resource allocation is an NP-hard problem, and cloud resources are mainly allocated to users through the network according to user requirements [1-2]. The objectives of resource allocation in the cloud environment are quality of service, load balancing, optimal span, security and economic principles. The allocation algorithm strategy affects the allocation of resources. The cloud computing system mainly provides service models, and the so-called management in the cloud is aimed at virtual resources. The number of virtual resources in the cloud environment is quite large and presents dynamic changes. In order to meet user requests, cloud computing needs to set a task allocation strategy in each data center, so how can it be better applied in a cloud computing environment and can better balance the resource load allocation method is the current cloud computing resource allocation Important issues to be studied [3-4]. In reality, the allocation of cloud resources to tasks may require the order of task execution. Common cloud computing resource allocation algorithms include genetic algorithms, simulated annealing algorithms, and ant colony algorithms. However, as far as current research is concerned, ant colony algorithms have initial solutions. Slow speed, prone to stagnation, inadequate data model of the system algorithm, and low algorithm efficiency when the module is large. Genetic algorithms have inherent disadvantages of the algorithm itself, such as poor local solving ability, easy to fall into local optimization, and After iterative evolution to a certain stage, the phenomenon of premature degradation is prone to affect the accuracy and convergence speed of seeking the optimal solution. The simulated annealing algorithm cannot control the most effective search direction. It has a large randomness, so the convergence speed is relatively slow. These algorithms It is a disadvantage that it is not good for solving the problem of optimizing resource allocation for Zhongchuang Space. Therefore, it is necessary to find an algorithm to solve this problem [5-6]. The cloud computing environment-based crowdfunding space computing optimized resource allocation algorithm overcomes the shortcomings of other algorithms and focuses on the advantages of each algorithm. It has certain reference value for solving the crowdsourced space optimization resource allocation problem.

Aiming at the task scheduling and resource allocation in the cloud computing environment, Rui Dong once proposed a Pareto-based fruit fly optimization algorithm (PFOA). First, Rui Dong proposed a heuristic algorithm based on the least cost to initialize the population. Secondly, Rui Dong designed a resource reallocation operator to generate non-dominated solutions. Then, Rui Dong designed a search algorithm based on the critical path, which improved the development ability of the system. In addition, when solving the TSRA problem in PFOA, Rui Dong adopted non-dominated sorting technology based on Pareto optimal concept and visual memory to deal with multi-objective problems. Finally, Rui Dong

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

verified the effectiveness of PFOA through some test cases, comparison results and statistical analysis [7-8]. Li-Der Chou once proposed a game theory method for joint offloading and resource allocation optimization (JORAO) in mobile edge computing (MEC) systems. Li-Der Chou not only discusses offloading strategies, but also considers the allocation of cloud and wireless resources. In particular, the focus of the JOROO problem is to minimize energy consumption and currency costs from the perspective of a mobile terminal. However, the JORAO problem is a non-convex NP-hard problem. Therefore, it is defined as a JORAO game. Li-Der Chou proved the existence of Nash equilibrium (NE). In order to obtain NE, Li-Der Chou also studied the Cloud and Wireless Resource Allocation Algorithm (CWRAA), which is a sub-algorithm of the JORAO game. For CWRAA, on the one hand, the allocation of OFDM subchannels and the allocation of uplink power in the radio access network (RAN) are considered. On the other hand, the problem of computing resource allocation in MEC is studied. Li-Der Chou's simulation results show that the distributed JORAO game algorithm can make the total cost of all mobile terminals (MTS) close to low complexity. In addition, compared with existing algorithms, when the size of the data becomes larger, energy consumption and completion time are less [9-10]. In order to improve the balance of resource scheduling in cloud computing systems and maximize the benefits of resource providers, Ming Liu once proposed a cloud computing resource allocation model based on queuing theory. Based on the waiting queue length as the premise of resource allocation, the Nash equilibrium (NE) theory was used to analyze the resource allocation strategy. Ming Liu proposed an improved resource allocation (MRA) algorithm. At the same time, Ming Liu compared this algorithm with some resource allocation algorithms in the distributed computing architecture Hadoop. The experimental results of Ming Liu show that compared with the Fair algorithm, the first-in-first-out (FIFO) algorithm, and the classic random algorithm, This algorithm can not only realize the fair sharing of user resources, but also the fair distribution of resources, and also meet the needs of resource providers in improving system response time [11-12].

The innovations of this paper are as follows: First, the basic ant colony algorithm (ACO) has a long period of blind search due to the same pheromone concentration in the path. It is not a simple improvement of the parameters or formulas of the original algorithm. Instead, the basic idea of genetic algorithm (GA) is introduced, and the fast global search capability of genetic algorithm is introduced into the allocation process of initial pheromone according to the characteristics of the algorithm. The genetic algorithm (GAAA) is proposed after merging the two algorithms. The algorithm is applied to the traveling salesman problem (TSP), and the feasibility of the GAAA algorithm is proved by experimental results. The Metropolis acceptance criterion in the simulated annealing algorithm is introduced to perform path optimization and pheromone update. According to the Metropolis acceptance criterion, some " "Inferior" new solution, judging whether to accept the new solution by comparing the objective function values of the new solution and the initial solution, can theoretically prevent the ACO algorithm from falling into a local optimal problem in solving application problems, and is aimed at cloud computing In the context of resource allocation, an objective function is proposed, and the improved genetic ant colony algorithm (HGAACO) is applied to the resource allocation problem. The operation steps and processes of the algorithm are listed in detail. The algorithm development platform is carried out in Matlab. By setting the algorithm parameters, the improved genetic ant colony algorithm (HGAACO) based on the Metropolis acceptance criteria is compared with ACO and the genetic ant colony algorithm (GAAA) that has been proposed and experimental analysis.

2. Proposed Method

2.1 Genetic Algorithm

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

Genetic algorithms are named for simulating the evolutionary processes of living things in nature. Genetics is a process of continuous optimization and evolution of biological populations. All organisms in nature must follow the natural law of "survival of the fittest". The GA algorithm is to simulate the continuous evolution of such organisms. It is accepted in the process of crossover and mutation. Individuals with good fitness values and individuals with low fitness levels are eliminated at the same time. After multiple evolutionary simulations, the optimal solution of the target problem is obtained step by step [13-14].

The genetic algorithm process can be briefly described: first, determine the initial population and fitness function. The generation of the initial population is usually randomly constructed, and then select certain individuals for cross-variation and other operations according to the value of the fitness value, in order to generate new individuals. According to the fitness value, several outstanding individuals are selected to be retained and the cross mutation operation is continued.In this way, iterations are repeated to retain the highly adaptive individuals.When the algorithm termination condition is reached, the algorithm ends, and the target optimal solution is the current population searched. Most adaptive individual. The general steps of the genetic algorithm in solving combinatorial optimization problems can be described as follows:

1) Determine the population size n, and use a certain method such as random to generate n possible solutions. They will be used as the initial population and can be expressed as:

$$X_i(k)(1 \le i \le n) \quad (1)$$

2) For each individual $X_i(k)$, where k represents the evolution algebra, the initial value is set to 1, and by calculating the fitness function $f(X_i(k))$ of each individual, select the outstanding individual with high fitness for the next operation. How to define the fitness should be analyzed according to the specific problem? For example, in solving the traveling salesman problem, the fitness function can be set to the inverse of the path length, which is $f = 1/T_d$, where T_d represents the path length.

3) Calculate its survival probability $X_i(k)$ for each $p_i(k)$ body:

$$p_i(k) = f(X_i) \Big/ \sum_{i=1}^{n} f(X_i) \quad (2)$$

Then randomly generate individuals who need to enter the next generation.

4) Through 3) selected individuals, randomly select two $X_1(k)$, $X_2(k)$ to generate a new generation of individuals according to the rules of crossover and so on as $X_1(k+1)$, $X_2(k+1)$, until the end of the $k+1$ generation of n individuals.

5) Continue the iterative operations of $k+2$, $k+3\ldots\ldots$, and other generations until the algorithm termination condition is met,

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

Generally speaking, the conditions for algorithm termination are that the algorithm has obtained a satisfactory solution or is limited by time or number of iterations. In general, the fitness value is constantly increasing during the iterations, and it will converge to a certain maximum value at a certain stage. This is the goal of GA operations. GA operation provides a general framework for solving the objective optimization problem. The individual is encoded according to the specific problem, and the encoded individual is selected for genetic operation. After a limited number of iterations, the optimal solution of the objective is obtained.

In the process of solving the genetic algorithm, the algorithm starts the entire iterative process through the individual population, so that the algorithm has good parallelism characteristics, and it is easy to achieve scale expansion and algorithm fusion [15]. In addition, the algorithm is also based on the probabilistic heuristic operation, and the random characteristics are obvious. But at the same time, the GA algorithm also has inherent shortcomings of the algorithm itself, such as poor local solving ability, easy to fall into local optimization, and after the algorithm iteratively evolves to a certain stage, premature degradation is prone to affect the accuracy and convergence speed of seeking the optimal solution.

2.2 Simulated Annealing Algorithm

The simulated annealing algorithm is a global optimization method based on the metal annealing mechanism.It can find the global minimum point of the objective function in a probabilistic sense with the help of random search technology.It has a strong local search ability and can prevent the search process from getting stuck. The local optimal solution is different from the general local search algorithm in that it selects a state with a large cost value in the neighborhood with a certain probability [16-17]. In the simulated annealing algorithm, the state points between every two temperatures are independent.

The specific steps of the simulated annealing algorithm (SA) can be divided into the following three parts:

1) Generation of the new solution X' : Generated in the solution space by a generation function.The most commonly used method is generally obtained by random simple transformation from the current solution X . The method has the function of randomly replacing some elements in the current solution X . In order to facilitate later calculations and reduce conversion time.

2) Determine whether the new solution X' is accepted: Whether the new solution is accepted is determined by two aspects, one is randomness, and the other is the objective function difference Δf , usually the objective function between the original solution X and the new solution X' the difference is calculated by increments, and the Metropolis acceptance criterion is currently the most commonly used acceptance principle.

3) Dealing with the new solution X' : the situation when X' is accepted and the situation when X' is not accepted When X' is accepted after being judged, X' is used to replace the current solution X . For the replacement of step 1), the inverse replacement needs to be performed and the objective function value is modified accordingly. The next experiment is based on the current solution as X' Just go. At this time, for the current solution X , one iteration has been completed; when X' is not accepted, then it can continue to use X as the current solution for the next iteration. In this case, this change means that this change affects

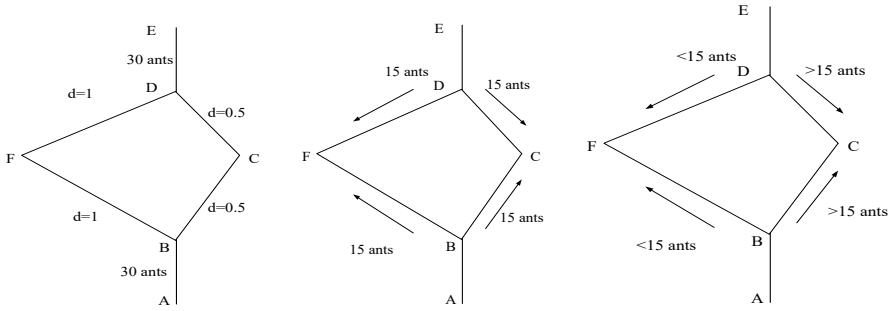
DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

the current solution. The results have no effect. The algorithm ends until the algorithm termination conditions are met.

2.3 Ant Colony Algorithm

(1) Basic ideas of ant colony algorithm

The proposed ant colony algorithm was inspired by observing ants in nature, and found that the ants looking for food turned out to have the ability to always find the shortest path from the nest of the ant colony to the food source [18-19]. Through research and detection, it was found that ants in nature can release a chemical called pheromone, and then the ants can determine the size of the concentration on the path by sensing this chemical, and determine the next step to be taken based on the concentration. Passing path ants will still release pheromone, but the pheromone in nature is not static, it will volatilize with time, that is, the path pheromone that will not be "patronized" for a long time will become less [20-21] . In this way, the ACO algorithm obviously has the characteristic of positive feedback: the higher the pheromone concentration on the path selected by more ants, the higher the concentration path will attract more ants to choose. There are two main issues involved in the whole process: one is the choice of ant path, and the other is the dynamic update of pheromone.



(a)30 each ready to leave B and D (b)15 for each C and F (c)Choose more C only

Figure 1. Examples of ant colony algorithm ideas

As can be seen from Figure 1, suppose A is an ant cave and E is a food source. Now the ant needs to start from point B to find food. Set the departure time t = 0, and the number of ants from point A is 30. Food ants need to go home, and there are also 30 ants starting from point D. The path length is represented by d, which shows the possibility and length of the selectable path: DFB with a distance of 2 and DCB with a distance of 1, as shown in Figure 1 (a). At the initial moment, the pheromone concentration on the two paths is the same.When the ants start, they choose the path with the same probability, that is, half of the ants at point B choose to go to point F, and the other half go to point C. Similarly, at point D The same is true for ants, but at time t = 1, the ant who chose to take the DCB path has reached point D or point B, and the one who chose the DFB path is still at point F. At this point, the pheromone concentration on the path DFB is It is lower than the concentration on the DCB path, so at time t = 2, when another 30 ants need to start from point B to find food, according to the perception that the concentration on the DCB road is higher than the concentration on the DFB, the ant who chooses DCB will be better than the one who chose There are many ants, and so are the ants preparing to go home at point D. In this way, after many search loop feedbacks, most ants know that the DCB path is relatively close, so the ant can find the shortest path between A and E. This is the process in which ants seek paths in reality. The ant colony algorithm ACO

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

draws on this mechanism of releasing and sensing pheromone selection paths to complete positive feedback iterative search [23-24].

(2) Application of ant colony algorithm to classical TSP problem

A businessman goes to n cities to sell goods. All the cities go over and back to the starting point, so that the distance traveled is the shortest, that is, given a certain set of cities, the city is traversed to make the total distance the shortest [25]. The application of ant colony algorithm to TSP problem can be described as follows:

Let $b_i(t)$ indicate how many ants are in the city i at the time of t, $\tau_{ij}(t)$ indicates the pheromone concentration between the city i and the city j at the time of t, and there are m ants, then there are

$$m = \sum_{i=1}^{n} b_i(t) \quad (3)$$

In the formula, n represents the size of the TSP problem, and stipulates that the pheromone concentration on each urban path is equal and constant at the initial moment, that is, $\tau_{ij}(0) = const$. It is on each path that each ant $k(k = 1,2,\ldots,m)$ affects its transfer path during its movement Pheromone concentration. The taboo table $tabu_k = (k = 1,2,\ldots,m)$ is used to indicate the cities that the k has traveled. The ant table will dynamically update the taboo table after choosing the next path. For the ants k, the set of cities that can be selected in the next step is $allowed_k$. When determining which city to go to next, there are two influencing factors: the pheromone concentration in the path and the heuristic factor. These two points can help the ant calculate the state transition probability of the city. The transition probability is as follows:

$$p_{ij}^{k} = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} * [\eta_{ij}(ij)]^{\beta}}{\sum_{s \subset allowed_k} [\tau_{ij}(t)]^{\alpha} * [\eta_{ij}(ij)]^{\beta}}, if j = allowed_k \\ 0, \text{otherwise} \end{cases} \quad (4)$$

In the formula, $p_{ij}^{k}(t)$ indicates the state transition probability of the ant k from city i to city j at the time of t. $allowed_k$ indicates the set of cities that the ant k has not passed and is allowed to transfer in the next step; the parameter α is the pheromone heuristic factor, that is, the path the weight of the pheromone concentration on the path is the importance of the path selection. The larger the value, the greater the impact. β is the expected heuristic factor, which indicates the size of the impact of the path selection after the length of the path is known. When the value is large, it is similar to the greedy rule, that is, the ant will choose the path with the short path.The expression for:

$$\eta_{ij}(ij) = \frac{1}{d_{ij}} \quad (5)$$

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

After calculating the transition probability of each of the selectable paths, in the process of selecting the next path, the ant will bias to the path with the higher transition probability. This has certain disadvantages, that is, all the ants will have the same choice when they go to this place, which will make the algorithm converge early and fall into the local optimal solution.

2.4 Improved Genetic Ant Colony Algorithm

(1) Improved algorithmic ideas

This article does not set a fixed value to terminate the genetic algorithm, but instead determines when the efficiency of the GA algorithm tends to decrease.The specific method uses two algebras, min and max, and a minimum evolution rate to determine how efficient. When the number of iterations of the algorithm is between min and max, the evolution rate is judged. If the evolution rate of successive generations is less than the minimum evolution rate, this number is also user-defined. If the transition conditions are met, it means that the GA algorithm optimization efficiency is slow at this time. Go to the ACO algorithm. The results obtained by the GA algorithm are converted into the initial pheromone information of the ACO algorithm. After the GA algorithm is stopped, an optimized solution for several populations will be obtained. A certain number of optimized solutions will be selected with a certain probability, and all of these individual optimized solutions will be converted into the number of ants on each node. The number is converted into the pheromone path of the node. The initial pheromone is defined as:

$$\tau_{ij}^{S} = \tau_0 + \tau_{ij}^{GA}, \tau_{ij}^{GA} = \theta^*\left(X_i + X_j\right) \quad (6)$$

(2) User task and resource node matching factors

The definition of matching factor is introduced to indicate the matching degree of each task and each resource node. The larger the difference between the expected value of the task and the resource value of the node, the more mismatched the task is assigned to the resource, and the smaller the more suitable it is.The definition of the matching factor for the i task and j resource node is as follows:

$$Match_{ij} = \Big/ \sqrt{\sum_{m=1}^{5}\left(Tm_1 - Vm_j\right)^2} \quad (7)$$

In the formula, the larger the $Match_{ij}$, the more the task i hopes to execute on the resource j, that is, the resource node can provide better service to the task i.

(3) Load balance and computing power of resource nodes

Let X be a mapping sequence of tasks and resources, then the load balancing degree is defined as:

$$Load(X) = \sqrt{\sum_{i=1}^{n}\left(1 - C_1\frac{Y_i}{Lv_i}\right)^2} \quad (8)$$

Among them, C_1 is a normalized parameter, which satisfies $0 < C_1\frac{Y_i}{Lv_i} < 1$.

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

(4) Definition of fitness function

The fitness function indicates the superiority of the individuals in the population. The genetic algorithm requires the fitness value to be the maximum value, and the cloud computing resource allocation problem is to find the minimum total cost. Define the fitness function as follows:

$$f(i)=\left[\frac{\sum_{k=1}^{M}\left(t(k)-t_{\min}+1\right)}{M*\left(t(i)\right)-t_{\min}+1}\right]^{2},i=1,2,\ldots,M \tag{9}$$

In order to prevent the fitness values between individuals from getting too close, the formula performs a square operation on the fitness function, so as to enhance the individual's significance.

(5) Definition of objective function

In the cloud environment, task allocation refers to the process of allocating tasks submitted by users to the corresponding resources to achieve the most scheduled process. The execution time of the assigned tasks needs to be minimized as much as possible, and the load of resources is as balanced as possible. Therefore, this article takes the time span and load balance as the evaluation criteria.The objective function is composed of:

$$F(X)=Load(X)*Makespan(X) \tag{10}$$

(6) Algorithm design steps

Through the introduction of the improved ideas above, the improved ant colony optimization algorithm in this paper to solve the resource allocation problem in cloud computing is described as follows:

1) Parameter definition and initialization: The parameters involved in the improved algorithm mainly include the size of the population, cross probability, mutation probability, evolution algebra, pheromone volatility factor, pheromone conversion factor, various importance weights, annealing coefficients, and end conditions;

2) Generate an initial population randomly, that is, randomly allocate resources, encode the allocation, and set the initial evolutionary generation to 0;

3) According to the size of the fitness function, the coding of the allocation strategy is performed by selecting heredity, and iterative evolution algebra plus 1.

4) Determine whether the evolution rate between the minimum iterations and the maximum iterations is less than the minimum evolution rate for a certain generation.If the successive generations are less than the minimum evolution rate, the GA algorithm has a slow convergence effect. You can end the GA algorithm and go to step 5.), Otherwise continue iteratively to step 3).

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

5) Several optimized solutions obtained by genetic algorithm are transformed into initial pheromone. Among them, the failure rate and the price of the node are negatively correlated with the initial pheromone amount. According to the above description, it is defined as:

$$\tau_0 = a\tau_{ip}(0) + b\tau_{ir}(0) + c\tau_{ib}(0) + d\tau_{if}(0) + e\tau_{ip}(0) \quad (11)$$

6) Put m ants on n vertices, that is, put m tasks on n resource nodes;

7) Use ant colony algorithm to reassign tasks.The specific method is to equip each task with an ant.These ants select the resource node they want to assign according to their corresponding task requirements and the size of the resource matching factor and pheromone concentration. The probability that task i selects the resource node j is:

$$p_{ij}^{k}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha} * \left[Match_{ij}(t)\right]^{\beta}}{\sum_{k \subset allowed} \left[\tau_{ij}(t)\right]^{\alpha} * \left[Match_{ij}(t)\right]^{\beta}} \quad (12)$$

8) Calculate the load balance degree and objective function value after selecting the resource node, and record the optimal solution X of this generation of ant colony.

9) Utilize the Metropolis acceptance mechanism in the SA algorithm idea to perform path optimization and pheromone update. After the end, the optimal solution will be generated after multiple iterations, and the pheromone will be released for update according to the optimal solution:

$$\Delta\tau_{ij}^{k}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad (13)$$

10) The loop iteration judges whether the number of iterations set by the ACO algorithm is reached. If the number of iterations is not reached, step 6) is performed, otherwise the process ends.

3. Experiments

3.1 Experimental Environment

This paper based on the cloud computing environment, the creative space calculation algorithm optimization algorithm experiment was implemented on MATLAB7.0, the performance test was performed on a computer with Pentium 2.59GHz, 2.00GB memory, Win7 operating system, using Matlab7.0 programming. The specific improved algorithm flowchart is shown in Figure 2.

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

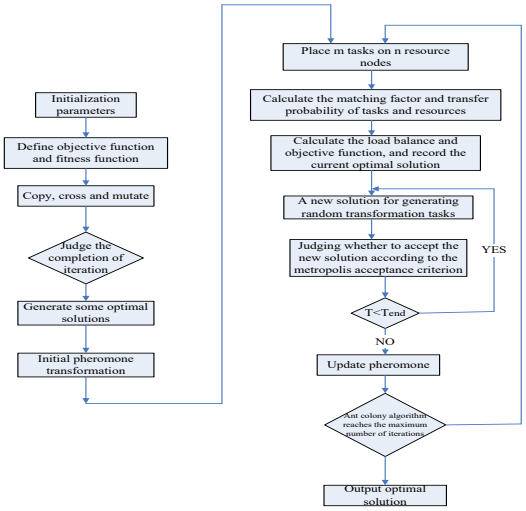


Figure 2. Flow chart of the improved algorithm

As can be seen from Figure 2, the algorithm improves the process. Considering the specific requirements in the cloud computing environment, the improved fusion algorithm is applied to the problem of cloud computing resource allocation, and different pheromone update mechanisms are introduced in the later stage to avoid stagnation. According to the characteristics of existing algorithms and the special resource allocation environment of cloud computing, the advantages of the three algorithms are combined while their disadvantages are avoided.

3.2 Parameter Setting

In order to be able to observe the experimental results intuitively, this article decided to compare the improved genetic ant colony algorithm with the basic ant colony algorithm and the genetic algorithm based ant colony algorithm. The two aspects of the objective function composition are time span and load balance Make comparisons to verify the quality of the algorithm, and set multiple tasks to perform multiple experiments to eliminate the chance of a single experiment.

This article sets the resource point n = 8, and the number of tasks m has two settings, one is increased from 20 to 100, and the other is increased from 50 to 350.This is done to show the performance comparison when the task amount is large and the task amount is small. Randomly generate the execution time of each task on the resource point, and set the processing time range from 1 to 6, that is, randomly generate an m × n matrix. The six-tuple of resource nodes and task expectation of resource capabilities is determined by static information according to specific problems Obtained and produce a matrix of size n × 6. The processing capacity of the resource node ranges from 100MIPS to 400MIPS, the memory capacity ranges from 512M to 2G, the bandwidth ranges from 1M / s to 4M / s, and the failure rate ranges from 0.1 to 0.3. then:

(1) Parameters related to genetic algorithm: population number 30, crossover probability 0.6, mutation probability 0.05, genetic algorithm iterations 500.

(2) About the parameters of the fusion point: the minimum number of iterations is 15, the maximum number of iterations is 100, the minimum evolution rate is 3%, the number of consecutive declines allowed to judge the algorithm termination is 5, and the pheromone conversion factor is 0.6.

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

(3) It is related to the parameters in the ant colony algorithm: the settings of a, b, c, d, and e are set according to the daily concept. Generally, the execution of tasks is greatly affected by the capacity of the processor compared to other ones. The values of, b, c, d, e are set to 0.4,0.15,0.15,0.15,0.15, respectively. The pheromone concentration heuristic factor is set to 3, the importance of the matching factor is set to 4, the pheromone volatility coefficient is 0.5, and the pheromone intensity Q is 5000.

(4) Parameters related to the annealing simulation algorithm: the initial temperature is set to 100,000, the end temperature is set to 10000, and the annealing coefficient is set to 0.95.

The distribution of the improved algorithm (HGAACO) is shown in Table 1.

Table 1. Resource allocation of HGAACO algorithm

Task Sequence number	Select the resource sequence number	Task Sequence number	Selection of resources
1	6	11	4
2	6	12	7
3	2	13	6
4	3	14	1
5	1	15	7
6	1	16	2
7	3	17	7
8	4	18	8
9	8	19	6
10	6	20	5

4. Discussion

4.1 Performance Analysis of Genetic Ant Colony Algorithm

In order to verify the performance advantage of genetic ant colony algorithm (GAAA), the performance of genetic ant colony algorithm (GAACO) and basic ant colony algorithm (ACO) was tested. Experiments were performed using a typical 51-city TSP problem with a population size of 100, a cross probability of 0.9, a mutation probability of 0.1, a minimum number of iterations of 15, a maximum number of iterations of 5000, an ant colony iteration of 500, a pheromone volatility factor of 0.4, and a pheromone conversion factor of 0.5 The initial pheromone value is 0.00001, and the pheromone intensity is 200, which is 1, which is 5. The combined algorithm, basic ACO algorithm and basic GA algorithm are tested for performance, and the running time and convergence path length are output. The test results are shown in Table 2 and Figure 3.

Table 2. Algorithm performance comparison

Algorithm Name	Time spent (s)	Iterations	Optimal result
GA	135	1546	449.204

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

ACO	76	173	445.613
GAAA	47	119	445.268

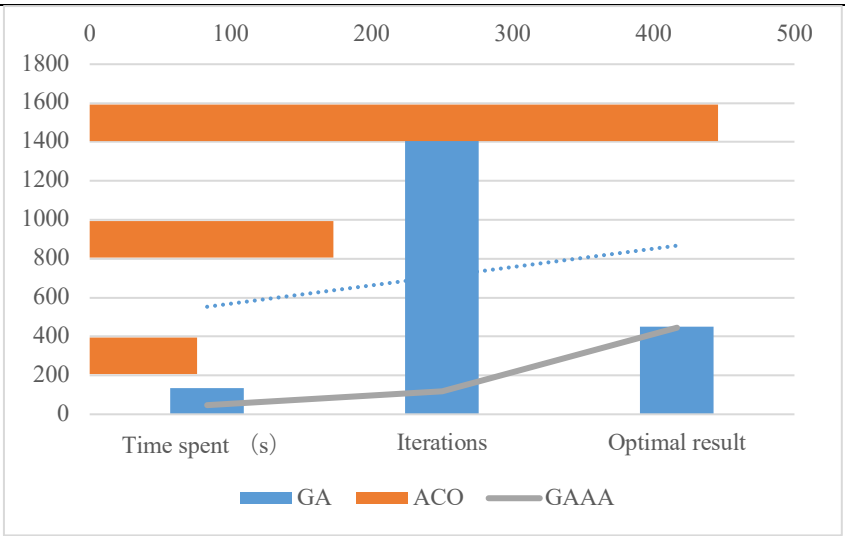


Figure 3. Analysis of algorithm performance comparison results

Ant colony algorithm is an intelligent bionic algorithm, which is based on observing the foraging activities of ant colonies in nature. Through observation, it is found that ants can always find the nearest path between the anthill and the food source. The main principle is that the ants in the ant colony You can identify and determine the path that needs to be transferred in the next step by releasing and sensing the concentration of pheromone. After finding a food source, a certain concentration of pheromone will be released according to the path length and other information.The path with a higher pheromone concentration is selected. Bigger. Ant colony algorithm has been widely studied and improved since it was proposed because of its positive feedback mechanism and robustness. It has shown great advantages in application, especially for combinatorial optimization problems.

From Figure 3, it can be seen from the performance analysis on the TSP problem that the genetic algorithm-based ant colony algorithm (GAAA) is better than the general ant colony algorithm, but it can be seen that although the improved algorithm has greatly improved in time and number of iterations However, the optimal result, that is, the shortest path is similar to the basic ant colony algorithm (ACO), which provides a solution for the application of the improved genetic algorithm-based ant colony algorithm (GAACO) to the problem of cloud computing resource allocation.

4.2 Experimental Results and Analysis

(1) Analysis of algorithm execution time and task time span

For experiments with 20 tasks, the generated random matrix is shown in the figure below. The random matrix is used as the initial data to improve the algorithm compared with the experiments of ant colony algorithm and genetic ant colony algorithm. For the example with 20 tasks, The initial data is set to be the same, and the parameter values of the three algorithms are set to be the same.Compared with the basic ant colony algorithm (ACO) and genetic-based ant colony algorithm (GAACO) using the improved algorithm,

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

the three algorithms have different algorithm execution time and task time. The span is shown in Figure 4 below.

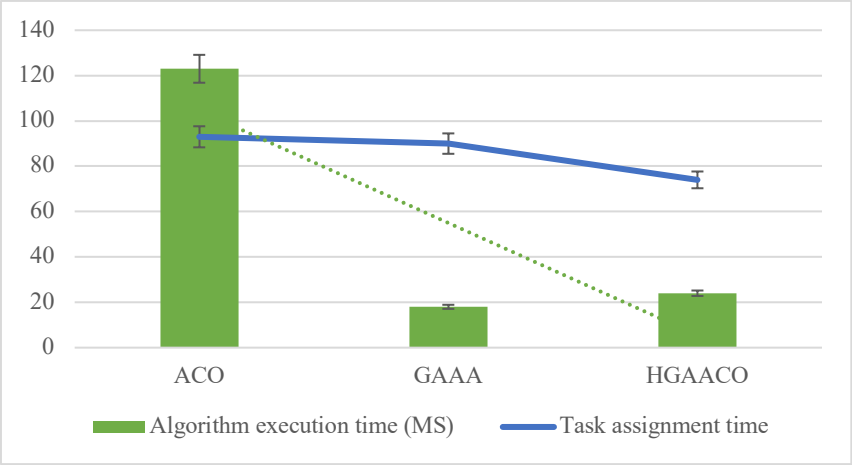
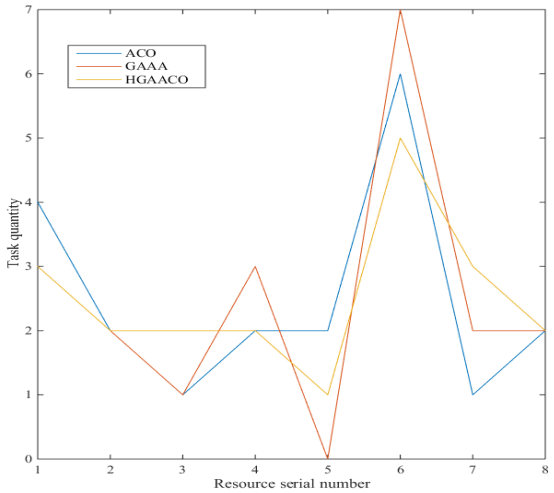


Figure 4. Analysis of comparison results between algorithm execution time and task allocation time

It can be seen from FIG. 4 that when the number of tasks is 20, the task allocation time of the ant colony algorithm is 93ms, and the time of the genetic ant colony algorithm is 90ms. The task allocation time of the improved algorithm proposed in this paper is 74ms, which is obviously superior. Due to the combination of genetic algorithm and ant colony algorithm, the blind search of the early stage of the ant colony algorithm is avoided, and the shortcomings of the late performance of the genetic algorithm are avoided. The fusion genetic ant colony algorithm (GAAA) is significantly faster than the algorithm execution time the general ant colony algorithm, and the improved algorithm introduces the Metropolis acceptance criterion in the later stage of the GAAA algorithm to increase the possibility of finding a better allocation method.It has been limited in the algorithm execution time, but it has made some progress in the scheduling time. .

(2) Analysis of resource load

Each resource load is shown in Figure 5.



DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

Figure 5. Analysis of the quantity of each resource load

It can be seen from Figure 5 that when the number of tasks is 20, the improved algorithm HGAACO is superior to the GAAA algorithm in resource load balancing, but it may be caused by randomness. Therefore, the number of tasks is increased from 20 to 100 for experiments. In reality, the allocation of cloud resources to tasks may require the order of task execution. Assuming that n tasks have no interdependence, tasks are assigned to m resource nodes, i.e. task processing units, usually m <n, cloud computing resources. The essence of the allocation problem is to solve the problem of large-scale, multi-task allocation and scheduling.

(3) Task time span analysis

The task time span of the three algorithms is shown in Figure 6.

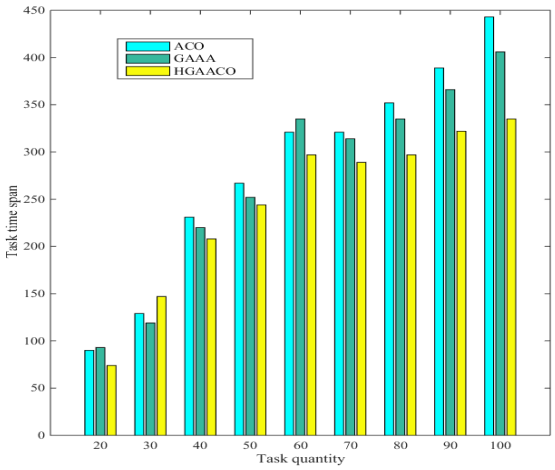


Figure 6. Comparison of task time spans

From Figure 6, it can be seen that when the number of tasks is small, the optimization effect of the task time span is not good, or even reduced to a certain extent. This is due to the randomness in the algorithm process, but the number of tasks is increasing. In this case, the randomness tends to be balanced, and the improved algorithm reflects the advantage, that is, the performance of the HGAACO proposed in this paper has a certain optimization effect on the task time span compared with the ACO and GAAA algorithms. Because the improvement of this paper in the later period of the algorithm is due to the introduction of the Metropolis acceptance criteria to search for some possible solutions.Although the solution selected by the path selection probability during the GAAA algorithm is the optimal solution of the algorithm mechanism, it may also appear to be "inferior" due to a certain randomness The effect of the solution will be better. For example, if the number of tasks is 20, we know that task number 3 searches for the resource node V2 through the Metropolis acceptance criteria, calculates the objective function in the case of resource nodes V6 and V2, and according to the acceptance criteria Determine whether to accept or not, and find that V2 meets the acceptance conditions, then accept resource node V2 as the assignment of task 3. This seemingly "inferior" solution will actually be able to obtain a global optimal solution in the final solution process, and it can be seen from Figure 5 that the results of the algorithm have not been greatly

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

optimized with a small number of tasks, but experiments It is found that the load balance of the algorithm shows instability when the number of tasks is small.

(4) Analysis of the load balance of the algorithm

The experimental results show that when the number of tasks is small, the load balance of the algorithm shows instability, so the second experiment is performed, and the number of tasks is increased from 50 to 350. The load balance of the algorithm is shown in Figure 7.

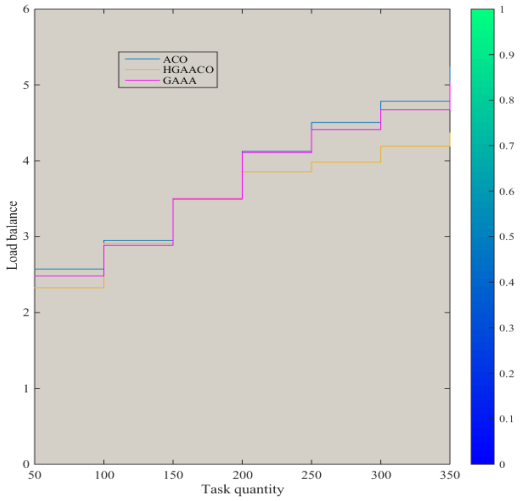


Figure 7. Load balance comparison chart

As can be seen from Figure 7, when the number of tasks is relatively large, compared with the ACO and GAAA algorithms, the HGAACO algorithm proposed in this paper is better in load balancing performance. It is fully reflected that the size of the load balance is similar to the ACO and GAAA algorithms, so this algorithm has obvious optimization effects when processing large-scale tasks. However, the introduction of the Metropolis acceptance criterion makes the execution of the algorithm not as fast as the GAAA algorithm. However, if users do not have high requirements for the execution speed of the algorithm, the HGACO algorithm still has certain advantages.

5. Conclusion

The concept of crowd-creation space is to comply with the trends of user innovation, open innovation, collaborative innovation, and mass innovation in the era of innovation 2.0. Entrepreneurial characteristics and needs, a low-cost, convenient, full-factor, open new entrepreneurial public service platform constructed through market-oriented mechanisms, specialized services, and capitalization approaches, and with the creation of the concept of crowd-creation space, its resource allocation The problem has become a research hotspot. This article studies the optimization of the allocation of space resources.

This paper mainly studies the problem of cloud computing resource allocation. Based on the existing algorithms, the intelligent algorithm is fused. Based on the blind search in the early stage of the basic ant colony algorithm and the shortcomings of the local optimal solution in the later stage, the pheromone initial allocation of genetic algorithm is introduced To solve the blind search problem, the Metropolis acceptance

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

criterion of the simulated annealing algorithm was introduced to avoid the problem of local optimal solutions in the later stage, and the improved algorithm was applied to the specific problem of cloud computing resource allocation, and experimental calculations and analysis were performed. Both the span performance and the resource load have better results, which proves the applicability of the algorithm.

The improved algorithm proposed in this paper has certain performance optimization, but there are certain disadvantages. The algorithm becomes slightly more complicated and has a great test on the speed of execution. Moreover, in actual application, the actual situation should be fully considered, such as more diversity. And dynamic user requirements and the complexity of resources need to consider the needs and interests of users and providers.

References

[1] Samaresh Bera, Sudip Misra, Joel J.P.C. Rodrigues. Cloud Computing Applications for Smart Grid: A Survey[J]. Parallel & Distributed Systems IEEE Transactions on, 2015, 26(5):1477-1494.
[2] Yahya Al-Dhuraibi, Fawaz Paraiso, Nabil Djarallah. Elasticity in Cloud Computing: State of the Art and Research Challenges[J]. IEEE Transactions on Services Computing, 2017, PP(99):1-1.
[3] Yumin Wang, Jiangbo Li, Harry Haoxiang Wang. Cluster and cloud computing framework for scientific metrology in flow control[J]. Cluster Computing, 2019, 22(1):1-10.
[4] Zhifeng Zhong, Kun Chen, Xiaojun Zhai. Virtual machine-based task scheduling algorithm in a cloud computing environment[J]. Tsinghua Science & Technology, 2016, 21(6):660-667.
[5] Tarandeep Kaur, Inderveer Chana. Energy Efficiency Techniques in Cloud Computing- A Survey and Taxonomy[J]. Acm Computing Surveys, 2015, 48(2):1-46.
[6] Jian Shen, Member, IEEE. Anonymous and Traceable Group Data Sharing in Cloud Computing[J]. IEEE Transactions on Information Forensics & Security, 2018, 13(4):912-925.
[7] Rui Dong, Changyang She, Wibowo Hardjawana. Deep Learning for Hybrid 5G Services in Mobile Edge Computing Systems: Learn from a Digital Twin[J]. IEEE Transactions on Wireless Communications, 2019, PP(99):1-1.
[8] Fuhong Lin, Yutong Zhou, Giovanni Pau. Optimization-Oriented Resource Allocation Management for Vehicular Fog Computing[J]. IEEE Access, 2018, PP(99):1-1.
[9] Li-Der Chou, Hui-Fan Chen, Fan-Hsun Tseng. DPRA: Dynamic Power-Saving Resource Allocation for Cloud Data Center Using Particle Swarm Optimization[J]. IEEE Systems Journal, 2018, 12(2):1554-1565.
[10] Pan Zhao, Lei Feng, Peng Yu. A fairness resource allocation algorithm for coverage and capacity optimization in wireless self-organized network[J]. China Communications, 2018, 15(11):10-24.
[11] Ming Liu, Yuming Mao, Supeng Leng. Full-Duplex Aided User Virtualization for Mobile Edge Computing in 5G Networks[J]. IEEE Access, 2017, PP(99):1-1.
[12] Zijian Cao, Jin Lin, Can Wan. Optimal Cloud Computing Resource Allocation for Demand Side Management in Smart Grid[J]. IEEE Transactions on Smart Grid, 2017, 8(4):1943-1955.
[13] Y. Xu, J. Zhi. Optimal PMU configuration based on improved adaptive genetic algorithm[J]. Power System Protection & Control, 2015, 43(2):55-62.
[14] Mohammad Shokouhifar, Ali Jalali. An evolutionary-based methodology for symbolic simplification of analog circuits using genetic algorithm and simulated annealing[J]. Expert Systems with Applications, 2015, 42(3):1189-1201.
[15] L. Ye, Z. Chen, Y. Zhao. Photovoltaic power forecasting model based on genetic algorithm and fuzzy radial basis function neural network[J]. Dianli Xitong Zidonghua/automation of Electric Power Systems, 2015, 39(16):16-22.
[16] C. Y. Wang, Min Lin, Y. W. Zhong. Solving travelling salesman problem using multiagent simulated annealing algorithm with instance-based sampling[J]. International Journal of Computing Science & Mathematics, 2015, 6(4):336-353.
[17] SUN Shiping, ZHANG Weihong. Frequency optimization of composite laminates using an improved simulated annealing algorithm[J]. Acta Materiae Compositae Sinica, 2015, 32(3):902-910.
[18] Fernando Esteban Barril Otero, Alex A. Freitas. Improving the Interpretability of Classification Rules Discovered by an Ant Colony Algorithm: Extended Results[J]. Evolutionary Computation, 2016, 24(3):385-409.
[19] X. Wu, Z. Xie, D. Song. Forward kinematics of 3-PPR parallel mechanism based on improved ant colony

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021

algorithm[J]. Transactions of the Chinese Society for Agricultural Machinery, 2015, 46(7):339-344.
[20] Zhengping Liang, Jiangtao Sun, Qiuzhen Lin. A Novel Multiple Rule Sets Data Classification Algorithm Based on Ant Colony Algorithm[J]. Applied Soft Computing, 2015, 38(C):1000-1011.
[21] Chong Shen, Ding Wang, Shuming Tang. Hybrid image noise reduction algorithm based on genetic ant colony and PCNN[J]. Visual Computer, 2016, 33(1):1-12.
[22] Zhenhua Han, Shugui Liu, Guoxiong Zhang. A 3D measuring path planning strategy for intelligent CMMs based on an improved ant colony algorithm[J]. International Journal of Advanced Manufacturing Technology, 2017, 93(9):1-11.
[23] Tang, Yong, Wang, Wei-Zhen, Dong, Shu-Na. An Improved Ant Colony Algorithm for Routing in Software Defined Networking[J]. Journal of Computational & Theoretical Nanoscience, 2016, 13(1):438-442.
[24] Z. Xie, H. Liang, D. Song. Forward kinematics of 3-RPS parallel mechanism based on a continuous ant colony algorithm[J]. China Mechanical Engineering, 2015, 26(6):799-803.
[25] WAN Xiaofeng, HU Wei, ZHENG Bojia. Robot path planning method based on improved ant colony algorithm and Morphin algorithm[J]. Science & Technology Review, 2015, 33(3):84-89.

DOI: 10.5281/zenodo.5203089
Received: January 21, 2021 Accepted: may 15, 2021