

Research Article

Research on Optimal Matching Scheme of Public Resource Management Based on the Computational Intelligence Model

Linna Li¹ and Renjun Liu ^{2,3}

¹*School of Design, The University of Melbourne, Melbourne 3052, Australia*

²*Wuhan Zhenghua Architectural Design Co., Ltd., Wuhan 430010, China*

³*Graduate School of Architecture, Planning, and Preservation, Columbia University, New York, NY 10027, USA*

Correspondence should be addressed to Renjun Liu; rl2783@columbia.edu

Received 11 August 2021; Revised 18 September 2021; Accepted 29 September 2021; Published 18 November 2021

Academic Editor: Bai Yuan Ding

Copyright © 2021 Linna Li and Renjun Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The management of public resources means that people's governments at all levels and other public administrative subjects should use certain means and methods, follow certain principles, rationally allocate and utilize public resources, and maximize their functions and benefits. Under the background of limited human resources, this study adheres to the principle of maximizing the benefits of human resources and rationally allocates the use of human resources. In this study, this kind of resource allocation problem is regarded as a linear programming problem by specifying the benefit function, and then, genetic algorithm, ant colony algorithm, and hybrid genetic-ant colony algorithm are used to solve the problem; the cost and time consumption of different algorithms under different scales are evaluated. Finally, it is found that genetic algorithm is superior to ant colony algorithm when the task scale is small and the effect of genetic algorithm is lower than ant colony algorithm with the expansion of task scale, whereas the improved hybrid genetic-ant colony algorithm is better than ordinary algorithm in general.

1. Introduction

With the development of the times and the progress of society, the current situation of unreasonable resource allocation has become more and more serious. The distribution of public resources has also become an issue that cannot be ignored in today's society. From the perspective of computational intelligence, resource allocation is, in the final analysis, a linear programming problem, which generally refers to the allocation of a limited amount of resources to all types of activities. Wu et al. [1] used an adaptive ant colony algorithm to solve the optimization problem of multi-resource allocation. After a series of verification experiments, the final results show that the ant colony algorithm has the ability to solve large-scale and multiresource problems, and the comparison results show that the adaptive ant colony algorithm has more advantages than the genetic algorithm. Xu et al. [2] aimed at specific medical resource allocation problems, combined with computational

intelligence to establish a customized medical resource allocation model, and combined the model with genetic algorithm to solve the model. The final experimental results showed that the model has excellent practical value. In order to solve the UAV assignment problem [3], it is transformed into an optimization problem under multiple constraints, and then, genetic algorithm and several other computational intelligence algorithms are used. Because the actual size of UAVs is large, it is a big problem in assigning tasks. The simulation experiment on the scale of the assigned task in this document shows that the genetic algorithm has better performance in smaller-scale problems, but when the problem scale becomes larger, its average calculation time far surpasses other algorithms. Nezhad and Shabbazian [4] converted the orthogonal frequency division multiple access technology (OFDMA) resource allocation problem into a function optimization problem and obtained the optimal solution to the objective function after mixing the traditional genetic algorithm and the particle swarm algorithm. The

research results showed that the hybrid algorithm has an effective improvement in system fairness and throughput compared with ordinary algorithms. Zhang and Chen [5] resourced the logistics personnel in the medical system for allocating various tasks in the hospital, in order to obtain the highest operating efficiency of the hospital and used genetic algorithms to build an automatic task allocation system. Among them, the selection operator adopts the championship mode, and some parent genes with low fitness are also involved in the parent selection to make the offspring more diversified while ensuring the speed of its operation. According to the comparison with the simulated annealing algorithm, its performance is better than that of the simulated annealing algorithm, but it performs poorly in terms of overall computing speed. Some researchers [6, 7] studied the ant colony algorithm to solve the multiobjective resource allocation problem. The literature optimized the pheromone update system and the probability selection form of the ant colony algorithm and finally compared it with the genetic algorithm in solving the employee assignment problem. The results show that the optimized ant colony algorithm is better than the genetic algorithm. Bublitz et al. [8] compared the four important branches of neural network, evolutionary computing (genetic algorithm), swarm intelligence (particle swarm algorithm), and artificial immune system through the research of the current branch algorithms of computational intelligence. Although the genetic algorithm can achieve the global optimal effect, it cannot effectively use the feed forward information, which results in the low efficiency of the algorithm. The particle swarm algorithm is also easy to fall into the local optimum. Li et al. [9] solved the problem of TSP. When optimizing the problem, the genetic algorithm is compared with the ant colony algorithm. The results show that the genetic algorithm is better than the ant colony algorithm when the problem is small, but as the scale increases, the effect is lower than the ant colony algorithm. Forootani et al. [10] studied the use of optimal strategies to solve the problem of resource allocation, combined with the company's dynamic allocation of project funds and processing resources under actual conditions, using the principle of optimization and multistage decision making while combining the chart method to maximize the company's profits. The study shows that the planning problem has practical feasibility in dynamic resource allocation. Tan et al. [11] cited genetic algorithm in the financial field and transformed the problem of project fund allocation into a planning problem. The article mentioned that the income function can be obtained by fitting actual data by neural network. Finally, combined with examples, it proves that genetic algorithm is solving the problem. When solving the problem optimally, it is possible to achieve the global optimum. Ikeda et al. [12] proposed a genetic algorithm based on fuzzy logic in order to optimize the multimedia communication problem. The simulation results show that the proposed framework has good performance and is a promising decision-making method. Lin and Lin [13] uses genetic algorithm for multiobjective optimization and solved some sample optimization problems involving two and three objective functions. It is suitable for both the

operation of existing factories and the design of new factories. A set of nondominated Pareto solutions is obtained for the research problem. Marler and Arora [14] introduced the current investigation of continuous, nonlinear, multiobjective, optimization concepts and methods. It integrated and correlated with seemingly different terms and methods. The characteristics of these methods are summarized. The conclusion reached depends on the type of information provided in the question, user preferences, solution requirements, and software availability. Vrugt et al. [15] introduced a hydrological model. Through the practical experience of research and calibration, an efficient Markov chain Monte Carlo sampler called a multiobjective, shuffled, complex, evolution urban algorithm was proposed, which can solve the problem of hydrological models. Hai [16] mentioned that the current cloud computing used in various fields also requires optimization of resource scheduling and allocation. The ant colony algorithm is introduced to apply to the resource scheduling problem of cloud computing. In order to make the ant colony algorithm better, it is mentioned that constraints can be added. The conditions also give a series of starting points for changing and adding constraints, and it also mentions the use of other algorithms combined with ant colony algorithm to achieve complementarity. Bai et al. [17] analyzed the current research status of cloud computing resource scheduling algorithm, genetic algorithm, and ant colony algorithm. Genetic algorithm has high search efficiency in the early stage, and it is easy to produce local optimal solutions in the later stage. However, the search efficiency of ant colony algorithm is low in the early stage and the later stage. Because of its positive feedback and other characteristics, it is easy to obtain the optimal solution. Through experiments, the improved genetic algorithm and the improved ant colony algorithm are compared, and the effectiveness of the algorithm in the cloud computing resource scheduling process is proved. Through the research on the management of public resources in China, it shows that there are still a series of major problems in the management of public resources [18], including the problem of resource allocation, and the unreasonable resource allocation directly leads to the excess of resources and the lack of resources, thus making society inefficient.

This study first uses genetic algorithm to solve the problem in the second part and uses the ant colony algorithm to solve the problem in the third part. After that, the two algorithms are merged, combining the advantages and disadvantages of the two algorithms to solve the problem again. The final result shows that the genetic-ant colony hybrid algorithm has the ability to solve large-scale and multiresource problems, and the comparison result shows that the adaptive ant colony algorithm has more advantages than the genetic algorithm.

2. Overview of the Problem

The solution of resource allocation problem mainly includes two aspects, namely, problem modeling and solving.

2.1. Modeling of Resource Allocation Problems. From the perspective of the problem itself, as one of the common types of linear programming problems, resource allocation problems are widely used in real life. Therefore, a large number of computational intelligence methods are also widely used in the analysis and solution of such problems, such as genetic algorithm, ant colony optimization algorithm [19], particle optimization algorithm [20], dynamic

programming [21], branch and bound, and decision tree methods.

In real-life public resource management problems, there are many types of resource allocation problems, and the constraints are also different, but the ideas for solving the problems are roughly the same. For the allocation of multiple public resources to different activities, under the premise of limited resources, seek the optimal allocation plan with the greatest benefit. Given decision variables,

$$\begin{aligned}
 X_{ij} & \begin{cases} = 1, & \text{if } j \text{ quantity of resources is allocated to project } i \\ = 0, & \text{else,} \end{cases} \\
 \max \sum_{i=1}^N \sum_{j=0}^M e_{ij} X_{ij}, & \left\{ \begin{array}{l} \min \sum_{i=1}^N \sum_{j=0}^M c_{ij} X_{ij} \text{ s.t. } \sum_{i=1}^N \sum_{j=0}^M X_{ij}, \\ j \leq M, \sum_{j=1}^M X_{ij} = 1, \quad \forall i, X_{ij} = 0 \text{ or } 1 \forall i, j, \end{array} \right. \quad (1)
 \end{aligned}$$

where $i (i \in \{1, 2, 3, \dots, n\})$ represents the task index, $j (j \in \{1, 2, 3, \dots, n\})$ represents the number of resources, n represents the total number of tasks, m represents the total number of resources, c_{ij} represents the cost of resource j occupied by task i , and e_{ij} represents the benefit generated by resource j occupied by task i . The objective function (2) and the objective function (3) indicate that the maximum benefit should be generated, and the least resources should be consumed. Constraint $\sum_{i=1}^N \sum_{j=0}^M X_{ij}, j \leq M$, ensures that the total number of resources does not overflow; constraint $\sum_{j=1}^M X_{ij} = 1$ ensures that each task i can only get one resource allocation. Pareto optimal solution is usually the solution of multiobjective programming problem. e_{ij} represents the cost required to allocate resource j to task i , whereas c_{ij} represents the benefit generated by allocating resource j to task i . The purpose of this constraint condition is to maximize the benefit under the minimum cost and conditions. Although there are some changes in e_{ij} and c_{ij} under different problems, their ideas are still practical.

2.2. Solving the Problem of Resource Allocation. In the current background of rapid development of artificial intelligence [22], more and more intelligent algorithms are used to solve such problems. For example, the common dynamic programming, branch and bound, and decision tree methods are used in the study of this article. However, when such algorithms deal with np-hard problems, such as resource allocation, as the scale of the problem becomes larger, the computational difficulty and computational cost also increase exponentially.

On this basis, this article calculates some control methods in the process of artificial intelligence immune system, such as genetic algorithm, ant colony control algorithm, artificial intelligence immune control algorithm, and so on. Such algorithms do not have the characteristics of polynomial time when solving resource allocation problems, so they are often widely used in the process of solving such problems. But for specific problems, it is necessary to

separately describe and model the problem of resource allocation.

3. Method

3.1. Genetic Algorithm. In the genetic algorithm, a population of size n is initialized, and the parental fitness value of each individual in the population is adjusted. Larger individuals are more likely to choose fitness values as individuals with parents. Through crossbreeding, n parents will produce n offspring, the offspring of n parents will mutate with a certain probability, and they will all survive and form a new next generation. In the genetic algorithm, the crossover operator is considered to be a major genetic operator, and the individual is mutated with a low probability.

Adaptive genetic algorithm [23] can be used as a kind of genetic algorithm, and the general way of realization is similar to that of genetic algorithm. First, analyze various variables for initial and evolution and then determine the scale of occurrence of each biological population, the probability of crossover, the probability of occurrence of mutation, and a series of parameters. Then, generate the initial population, take the initial population as the parent, and take the individuals with higher fitness to perform cross-mutation and other operations. After the offspring are produced, the qualified offspring are screened out, and the accompanying individuals with higher fitness are used to replace the fitness of the offspring. Then, it is judged whether there is an individual in the offspring that meets the termination condition of the algorithm. If it cannot be terminated, the offspring will be used as the new parent to repeat the above steps until the termination condition is met, and the optimal solution is found.

3.2. Implementation Steps of Genetic Algorithm. The optimization model that defines its two-dimensional resource allocation problem is as follows:

$$\max b = F(y, x) \begin{cases} \sum_{i=0}^n y_i = P_1, & \sum_{i=0}^n x_i = P_2, \\ y_i \geq 0, & x_i \geq 0. \end{cases} \quad (2)$$

The resources A and B are allocated to N activities, and the allocation schemes are denoted by y_i and x_i , respectively. The total resource of A is P_1 , and the total resource of B is P_2 . Then, stipulate the benefit function $F(y, x)$ according to the actual situation. For the multidimensional resource allocation problem, it can also be solved in the above-mentioned way, with only a little improvement. The meaning expressed in formula (2) is to ensure that the number of allocated resources does not exceed the total number of resources, and x_{ij} is a decision variable. If resource j is allocated to task i , it means that the value of resource j in formula (2) is 1 and the sum of allocated resources cannot be greater than the total number M . The two resource allocations are independent and the total amount is fixed.

Step 1: generate the initial population

Use binary encoding to generate a set of chromosomes G_k ($k = 1, 2, \dots, N$), where N is the size of the group. Because this article is mainly based on the two-dimensional resource allocation problem as an example, the length of the variable code string is $2N$, so the length value of one of the chromosomes is $2N * l$, and the code string length for the variable is 1 digits, and each 1 digit. The length of the code string is equivalent to a decimal integer b_i ($i = 1, 2, 3, n$). What is different is that G_k is the first-generation population.

Step 2: feasibility process

The chromosome code is mapped to the position to meet the feasibility of the problem. Normalize b_i to b'_i , let $x_i = ab'_i$, at this time all the vectors Z_p ($p = 1, 2, 3 \dots, n$) composed of x_i can satisfy the constraints. That is a vector corresponding to the same chromosome produces a feasible solution.

Step 3: calculate fitness

Take a new objective fitness function as an optimal fitness function. For each chromosome in different population pedigrees of the same generation of chromosome G_k , combine all the corresponding optimal feasible solutions Z_k that we require above substituting an objective fitness function, you can directly and accurately find their optimal fitness function value.

$$f_x = \sum_{i=1}^n g_i(x_i). \quad (3)$$

The higher the fitness value, which shows that their corresponding optimal feasible solutions are closer to the optimal solutions.

Step 4: chromosome selection

Copy the chromosomes with the highest fitness in each generation population to the next generation population. For the cumulative population up to the next generation, first, the system will automatically

generate a random probability number between 0 and f'_k , and count each random value, starting from the first chromosome G_1 for random accumulation. When the value of accumulation and random adaptability is greater than the random probability number generated by the second-generation accumulation, the system will automatically suspend the accumulation and finally add the accumulated chromosomes that have been replaced. This may also be each chromosome that needs to be randomly selected as described in this article. Finally, the random probability of each cumulative chromosome being selected and the cumulative fitness value are kept in direct proportion, and then, the unselected chromosomes are maintained. Each chromosome is eliminated.

Step 5: chromosome hybridization

After the new population reproduces, select "individual pairs" according to the probabilities P_C of various crosses and perform a single-point cross (a total of $n/2$ times). In this study, P_C is 0.7. Single-point cross hybridization is adopted, and the method of cross-exchange points is randomly selected, and the feasibility optimization of all the obtained chromosome progeny data can effectively ensure that the chromosomes after cross hybridization in the population correspond to each other and the feasibility optimization. Calculate the fitness value that the offspring needs and then enter the new population by replacing other fathers and sons.

Step 6: chromosome variation

In order to effectively maintain the evolutionary diversity of various groups on biological chromosomes and maintain the "evolution" of populations on biological chromosomes, this article randomly selected individual populations on some biological chromosomes for adaptive mutation, considering that these are some of the biological variant chromosomes, i.e., the basic feature, the probability of mutation is not very large.

$$P_m = \frac{N_m}{N}. \quad (4)$$

This article first uses random $P_m = 0.05$ and randomly selects some sites that adapt to mutation as $d(1 \leq d \leq 1 * n)$. The adaptive mutation and infeasibility of these mutated biological chromosome populations can be effectively done to ensure that the solutions corresponding to these mutated chromosomes are feasible solutions and then sequentially calculate the values, and values that are adapted on these chromosomes may enter the varieties.

Step 7: judging the conditions for stopping evolution

If during the evolution process from $m - s$ generation to $m + s$ generation, the highest fitness value in the chromosomes of each generation remains unchanged, you can stop its evolution; otherwise, return to Step 4.

3.3. Adaptive Ant Colony Algorithm. The resource allocation problem can be regarded as a boxing problem. The public resource v to be allocated is regarded as an item, and each activity p is regarded as a box. This article will be analogous to the idea of solving the boxing problem to deal with the optimization and matching problem of public resource management. Ant colony algorithm has the characteristics of distributed calculation and positive feedback of information and heuristic search, so it has become one of the most classic algorithms for solving box packing problems. Based on the above characteristics, compared with other algorithms, it can more effectively converge to the optimal solution when solving the packing problem. Therefore, this article will select the ant colony algorithm to deal with the optimization matching problem of public resource management. This algorithm is completed by simulating the entire process of ant foraging. First, the ant-bee randomly selects a feasible path. When the ant-bee reaches the expected planned path, it stops and starts to calculate the adaptability of this path. The ant-bee establishes a corresponding information on the path according to its own adaptability. Finally, in order to concentrate the entire ant clustering on a higher fitness approach and find the best solution as quickly as possible, it is necessary to update the pheromone and select actions. For example, the use of honeycomb swarm algorithm for task scheduling. First, the initialized pheromone and heuristic information and other related parameters. Second, each ant receives the assigned task, and when the assigned task is successfully assigned to each ant's resource, it will be recorded in its own taboo list. Third, for the following tasks that have not been accessed, repeat the above steps until the task is no longer added to table, and all tasks are considered to be completely scheduled. The goal of adaptive ant colony algorithm is to find the optimal solution of optimization problem. Ants often choose a path with a large amount of information in the process of traveling. However, when many ants choose the same path, the amount of information in the path will suddenly increase, which makes many ants concentrate on a certain path, resulting in a blockage and stagnation phenomenon, which is easy to lead to prematurity and local convergence when using ant colony algorithm to solve problems. This is the root cause of the shortcomings of ant colony algorithm. Therefore, we modify the selection strategy, adopt the combination of deterministic selection and random selection, and dynamically adjust the probability of deterministic selection in the search process. When the evolution direction has been basically determined after a certain algebra, the information amount on the path is dynamically adjusted. The shortcoming of the basic ant colony algorithm can be effectively overcome by narrowing the information gap between the best path and the worst path and appropriately increasing the probability of random selection so as to search more completely less than one pair of solution spaces.

3.4. Pheromone Function Definition. The whole process of simulating a path in which ants form a path is to allocate resources to each task reasonably. Therefore, the selection of

each task and the update of pheromone in the heuristic information and algorithms are greatly interconnected. $t(T_i, R_j)$, which means that a task is allocated to the information element required by a certain resource. In the initialization phase, the pheromone is calculated by the following formula:

$$t_0 = \frac{1}{[n * (p'(S_0) + w(s_0))]}, \quad (5)$$

where n represents the number of activities, the solution S_0 obtained using the descending first adaptation algorithm to solve the boxing problem is marked as S_0 , and $W(S_0)$ the amount of waste of resources generated by this solution of S_0 . $p'(S_0)$ is the result of normalizing the energy consumption of this solution of S_0 . It can be calculated by the following formula:

$$p'(S_0) = \sum_{j=1}^m \left(\frac{p_j}{p_j^{\max}} \right), \quad (6)$$

where p_j^{\max} represents the amount of resources consumed by task j .

3.5. Transition Probability of Behavioral Choice. The following methods are used to select its behavior so that we can obtain the optimal solution: A hypothesis is that in the process of the k -th replacement, the resource set marked by the deadline and budget constraints required by this task will be able to be met into $g_k(T_i, R_j)$. Therefore, in the k -th iteration, the resource R_j that task T_i needs to select is the probability of completing the scheduling work, which is given by the following formula:

$$P_k(T_i, R_j) = \left\{ \frac{[t(T_i, R_j)]^\alpha \cdot [\eta(T_i, R_j)]^\beta}{\sum_{h \in g_k(T_i, R_j)} [t(T_i, R_h)]^\alpha \cdot [\eta(T_i, R_h)]^\beta} \right\} \quad (7)$$

In equation (7), $t(T_i, R_j)$ indicates that the task T_i is allocated to a pheromone of the resource R_j in a certain path and $\eta(T_i, R_j)$ indicates the heuristic information corresponding to the path. It can be set as the reciprocal of the starting time of a task assigned to the resource. The parameters α and β are, respectively, regarded as the weighting factors occupied by heuristic information and other pheromones. They both represent the relative importance of heuristic information and other pheromones, and they play an important regulatory role.

3.6. Fitness Function. For example, when an ant passes through all active tasks, a path has been formed, and this path is a feasible solution to the problem. In order to effectively avoid falling into a local optimal state, to ensure the accuracy and quality of the obtained solution, and to ensure that the obtained solution is the global optimal solution as much as possible and in order to accurately evaluate the accuracy and pros and cons of the obtained solution, we need to choose a function of fitness. This article mainly adopts the research method of defining the fitness function

based on the theoretical basis of the optimization model problem. Taking the scheduling optimization model established in the article as the theoretical basis, two scheduling goals are analyzed and designed, and the cost is reduced to a minimum. Therefore, the fitness function of the following formula is the evaluation value.

$$\text{Fit}(x) = \gamma e^{-F(x)} + \delta e^{-B(x)}, \quad (8)$$

where γ and δ in represent the two weighting factors of operating cost and resource waste, respectively. The two conditional formulas are $\gamma < \delta$, γ and $\delta \in (0, 1)$, and $F(x)$ and $B(x)$ are the objective functions used to represent operating cost and resource waste, respectively. It can be concluded from the above summary that the lower the value of the fitness function, the less waste caused by operating costs and human resources.

3.7. Pheromone Update. Ants cruise the path according to the pheromone concentration on the path. If a path is highly adaptable, the pheromone concentration of this path should be strengthened so that more ants can find this path. Therefore, it is necessary to update the pheromone at each point on the path.

The update rule is as follows:

$$t(T_i, R_j) = (1 - \rho)\Delta t(T_i, R_j) + \Delta t(T_i, R_j). \quad (9)$$

In formula (9), ρ represents the speed and degree of pheromone disappearing in the air, that is, the evaporation factor, and $t(T_i, R_j)$ represents the number and increment of pheromone in the air. The higher the adaptability of the path, the more the corresponding increase in diffusion, which is a positive feedback adjustment mechanism.

$$\Delta t(T_i, R_j) = Q \cdot (\gamma e^{-F(x)} + \delta e^{-B(x)}), t(T_i, R_j) \in \text{path}. \quad (10)$$

Q in equation (10) is a constant, and its initial value is 10. Among them, the lower the value of $f(x)$ and $b(x)$, the higher the value increment of the pheromone. Based on the positive feedback mechanism, the better solution in the path may gradually increase with the further update of the pheromone, whereas the worse solution may also gradually weaken with the further update of the pheromone and finally be discarded. After several consecutive years of iteration, more and more ants will move towards an optimal path. In order to prevent the solution, we can only be optimized in one part; the evaporative factor of pheromone can play an important regulatory role.

3.8. Ant Colony Algorithm Process. The flow chart of the ant colony algorithm is shown in Figure 1.

- (1) Parameter initialization: the main functions include parameter initialization of application pheromone types, initialization of heuristic application information, and optimization of parameter initialization value information of various factors such as

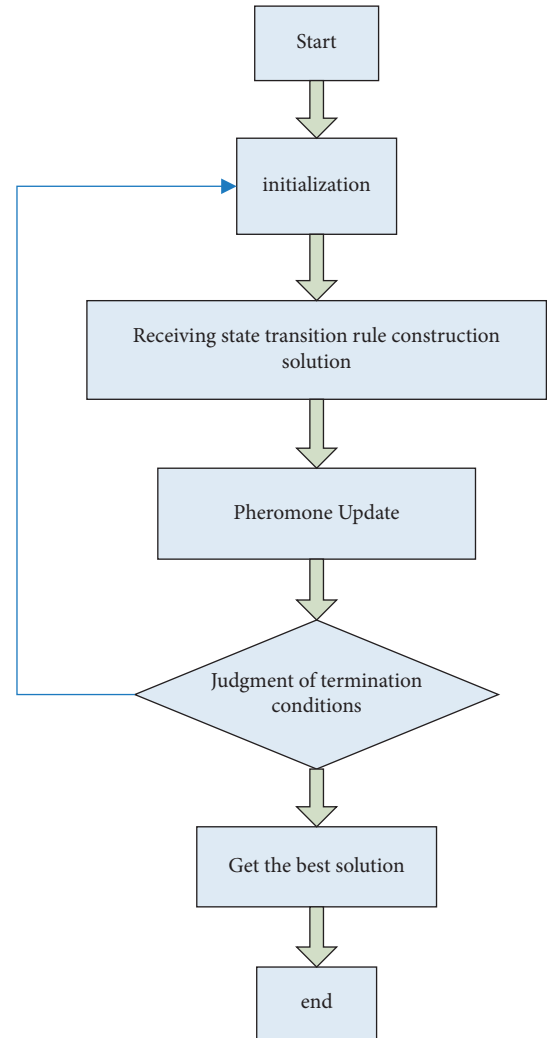


FIGURE 1: Ant colony algorithm flow chart.

information population distribution and pheromone biological volatilization rate.

- (2) The ant starts the first cycle: $k = k + 1$.
- (3) Randomly place several ants at different starting points and establish a search space.
- (4) The probability of each ant moving to the next node is calculated separately. The ant can move on its own according to the results of these numerical calculations and finally reach its corresponding new node.
- (5) During this period, when an ant moves to the path to a new node, the information elements in all the paths it passes are updated, and the content in the taboo table will be adjusted accordingly.
- (6) Repeat steps 3 to 5, until each individual in the entire ant colony has found a feasible path.
- (7) For an output result, output and optimize it according to requirements, including related indicators, such as running time, convergence, and the number of iterations.

4. Improved Genetic-Ant Colony Algorithm Fusion

At present, most scholars [24] optimize the ant colony algorithm mainly in two aspects: one is to add constraints to the ant colony algorithm to overcome the shortcomings of the original algorithm; secondly, to integrate other algorithms with the ant colony algorithm. Complementary, use the advantages of other algorithms to improve the shortcomings of the ant colony algorithm. In order to effectively avoid the shortcomings of the two algorithms, this study adopts the second method of algorithm optimization, which is to combine the two algorithms to form complementary advantages. Hybrid algorithm not only synthesizes the advantages of the two algorithms but also inherits the shortcomings of the two algorithms to a certain extent, so we still need to adapt to local conditions and choose the appropriate algorithm in the actual production process. When the amount of data is small, a single genetic algorithm can be directly used to solve it.

4.1. Improved Dynamic Fusion Strategy of Genetic-Ant Colony Algorithm. In this study, the optimal solution obtained by the genetic algorithm is transformed into the initial value allocation strategy of the ant colony algorithm, and the specific operations are as follows:

- (1) First, use the characteristics of genetic algorithm to generate a series of optimized solutions for resource scheduling problems. In this evolution process, the evolution speed of each generation of population in the genetic algorithm is calculated. Within the range of satisfying the maximum number of runs, if the evolution speed of the population for n consecutive generations is less than the preset minimum value, the genetic algorithm is terminated.
- (2) Enter the ant colony algorithm after meeting the fusion conditions. First, the first 10% of the optimized solution of genetic algorithm is converted into the pheromone value of the initial path of the ant colony algorithm, and then, the optimal scheduling plan is calculated according to the characteristics of the ant colony algorithm. According to the above operation process, it can be seen that the main difficulty of fusion is to determine the conditions for terminating the genetic algorithm. After reading a large number of documents, the evolution speed of the two algorithms over time is summarized. The evolution speed of genetic-ant colony algorithm changes with time as shown in Figure 2.

It can be seen from the figure that the evolution rate of the genetic algorithm in the initial stage, that is, from the beginning to the t_b stage, is very high, but after the time point t_b , its evolution rate decreases with the increase in time, until the time point t_c drops to a certain level degree, which is relatively stable. The evolution rate of the ant colony algorithm from the beginning to the t_b stage is very low, but after the time point t_b , its evolution rate increases with the

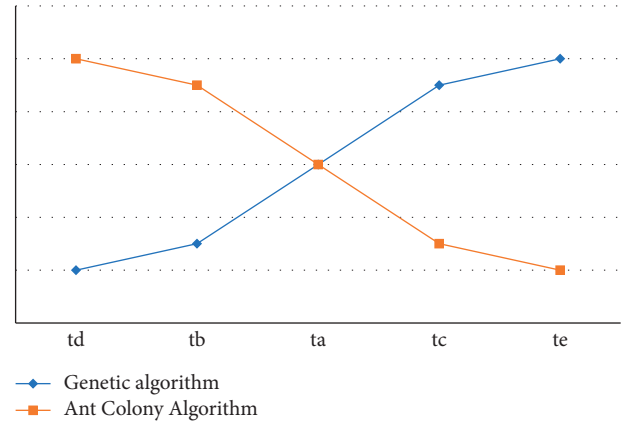


FIGURE 2: Evolutionary speed of genetic-ant colony algorithm over time.

increase in time, until the time point t_c , where the evolution rate has risen to a certain degree and is relatively stable. In the evolution rate-time curve, the most special time point is t_a . Before this time point, the evolution rate of genetic algorithm is higher than that of ant colony algorithm. At this time, the evolution rate of the two algorithms is the same. After that, the evolution rate of ant colony algorithm is higher than that of genetic algorithm. Therefore, the dynamic fusion of genetic algorithm and ant colony algorithm at time t_a is the most ideal, that is, the best results can be obtained in all aspects of time and efficiency. But in the execution process, it is difficult to accurately determine the time point t_a . Therefore, when we are improving the algorithm, as long as we determine that the evolution rate is switched during the $t_b \sim t_c$ time period, the algorithm has been improved to a large extent.

4.2. Improved Genetic-Ant Colony Algorithm Fusion Parameter Setting. It can be seen from the above that the evolutionary speed of genetic algorithm and ant colony algorithm change over time completely opposite. Therefore, the combination of these two algorithms does not jump to the ant colony algorithm after the genetic algorithm is fully executed but enters the ant colony algorithm when the combination conditions are met. In order to switch the evolution rate of the algorithm within $t_b \sim t_c$, the parameters of the two algorithms need to be set as follows:

Step 1: suppose that the maximum number of iterations of the genetic algorithm is g_{max} , the current number of iterations is g_{num} , the minimum number of iterations is g_{min} , the minimum evolution rate is g_{rate} , and the population of offspring evolves rate is g_{pop} .

Step 2: if g_{num} is less than g_{max} , then compare g_{pop} and g_{rate} ; if the evolution rate of successive generations and the evolution rate of the progeny population g_{pop} is less than g_{rate} , the genetic algorithm is ended, and the ant colony algorithm is run.

Step 3: as a key parameter of algorithm fusion, the main significance of evolution rate is to obtain the current

evolution degree of genetic algorithm. Therefore, evolution rate is the comparison between the fitness value of the latest individual and the previous evolution value, as shown in the following equation:

$$RI = \left(\frac{(f(1) - f(i-1))}{(f(i-1))} \right). \quad (11)$$

The optimal solution of the genetic algorithm is transformed into the path pheromone value t_G of the ant colony algorithm, so that the pheromone constant is t_C . In this article, the constant t_G is represented by the path value in MMAS as t_{\min} , and t_G is the path value passed plus 2, then the initial pheromone value of the ant colony algorithm is given as

$$t_s = t_c + t_G. \quad (12)$$

Step 4: the method of conversion between algorithms is as follows: First, each individual in the top 10% of the genetic algorithm can be represented as a task T_i executed on the resource node P_j , which is equivalent to selecting P_j for the task T_i of the ant and then changing the resource P_j . The pheromone value can save 70% of the time wasted by the ant colony algorithm at the beginning due to the lack of pheromone.

4.3. Implementation Steps of the Improved Genetic-Ant Colony Algorithm. Genetic algorithm stage:

Step 1: corresponding parameter settings are minimum number of iterations g_{\min} , maximum number of iterations g_{\max} , minimum evolution rate g_{gate} , progeny population evolution rate g_{pop} , and population iteration number g_{num} . The initial population size is the initial population G of Popsiz.

Step 2: at the beginning of the algorithm, the chromosome is encoded with decimal real numbers.

Step 3: the fitness function is defined as the average completion time of the task and the average load weighted sum function $f(j)$ of the virtual machine, and the genetic operation is started.

Step 4: use the fitness value ratio selection method (the selection strategy of the selection operator $p(i)$ to filter out the high fitness value.

The parent individual waits to perform the crossover operation.

Step 5: calculate the adaptive crossover probability P_c , select the parental individuals for single-point crosses to produce a given number of new individuals.

Step 6: calculate the adaptive mutation probability P_c and use the method of randomly generating mutation positions to perform mutation operations on the new individuals obtained. The group $G(g)$ is genetically operated to obtain the next generation group $G(g+1)$.

Step 7: compare the new generation with the parents and select outstanding individuals as the final offspring.

Step 8:

$$g_{\text{num}} = g_{\text{num}} + 1 \text{ if (calculate the evolution rate } R(I)). \quad (13)$$

Step 9: determine whether the fusion condition is met: If $g_{\text{num}} > g_{\max}$, or the evolution rate of consecutive g_{die} generations is less than g_{rate} , then end the genetic algorithm, run the ant colony algorithm, and go to Step 10. Otherwise, go to Step 3.

Ant colony algorithm stage:

Step 10: convert the top 10% individuals with the best optimization solution in the genetic algorithm into the path value t_G of the ant colony algorithm, get the value of t_s , set the ant colony algorithm parameters, and set the number of ants to the number of tasks m and the current iteration number $NC=0$, the maximum number of iterations NC_{\max} , and the taboo table tabu_k is empty.

Step 11: m ants are randomly placed in n resource nodes. For each ant, use the probability $P_{ij}^k(t)$ to select the resource node to perform the next task.

Step 12: when an ant completes this task, add the node to tabu_k , m minus 1, and record the current score.

Configure the strategy and update the pheromone value of the resource node locally.

Step 13: when all the ants complete the task, the number of iterations NC increases by 1, the optimal allocation strategy is screened out, and the pheromone value of the resource node is updated globally.

Step 14: $(NC > NC_{\max})\{\text{break the cycle}\}$ else go to Step 11.

5. Comparative Analysis of Simulation Experiments

After understanding the specific implementation steps of the three algorithms, this article applies the three algorithms to the actual problem solving and uses experiments to verify the efficiency of the three algorithms in solving the resource allocation problem.

5.1. Problem Description. This example is used to deal with the problem of hospital medical resource allocation. For example, what kind of medical activities should be allocated, what kind of medical resources, and what kind of deployment methods can maximize the efficiency of medical activities. The medical resources here mainly refer to medical personnel. After instantiating the problem, this article assumes that there are ten medical personnel and four medical activities at the same time. Aiming at these four medical activities, this article needs to obtain the optimal medical personnel deployment plan.

A large number of experimental research results show that different parameter values have a great impact on the accuracy of the algorithm. In this study, a large number of experimental results are used to select the more accurate parameter value configuration as follows:

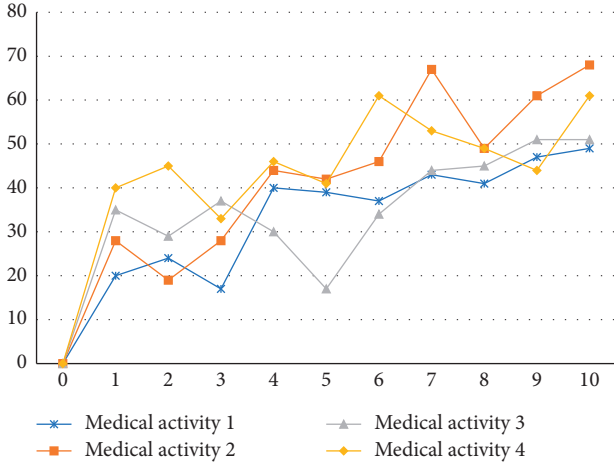


FIGURE 3: Working cost of medical staff.

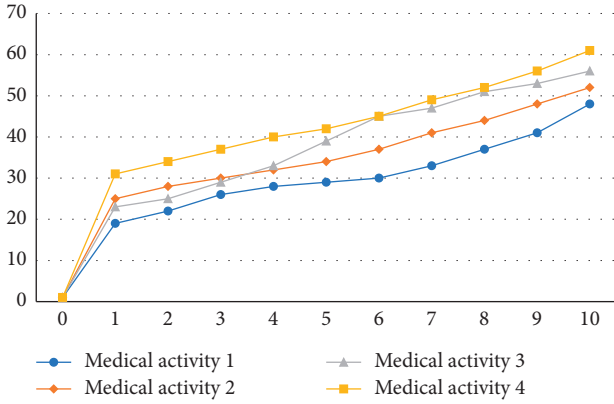


FIGURE 4: Work efficiency of medical staff.

TABLE 1: Nondominated solutions of the genetic algorithm.

Solution	1	2	3	4	Cost	Efficiency
1	3	2	1	4	101	115
2	0	2	6	2	99	106
3	3	2	5	0	87	92
4	3	1	6	0	83	88
5	1	1	6	2	107	121
6	0	1	6	3	96	105

Genetic algorithm: $P_C = 0.7, P_m = 0.05$.

Ant colony algorithm: $\alpha = 0.5, \rho = 0.3, t_0 = 0.01$

Hybrid algorithm: $g_{num} > 1, g_{num} > g_{max}, NC > NC_{max}$.

Figures 3 and 4 show the work cost and work efficiency of four different medical activities corresponding to different numbers of medical personnel. The cost and efficiency here are an estimate.

5.2. Comparison of Experimental Results. Tables 1–3 show the solutions of the three algorithms to solve the problem, respectively. Figures 5 and 6 show the comparison of the cost and time consumed by the three algorithms to solve the problem when the computational scale increases.

TABLE 2: Nondominated solution of ant colony algorithm.

Solution	1	2	3	4	Cost	Efficiency
1	3	2	1	3	88	112
2	1	2	6	0	77	91
3	3	2	0	0	76	53
4	1	1	6	1	102	118

TABLE 3: Nondominated solutions of the hybrid algorithm.

Solution	1	2	3	4	Cost	Efficiency
1	1	3	2	3	74	98
2	1	6	0	2	63	77
3	3	0	0	2	62	39
4	1	6	1	1	88	104

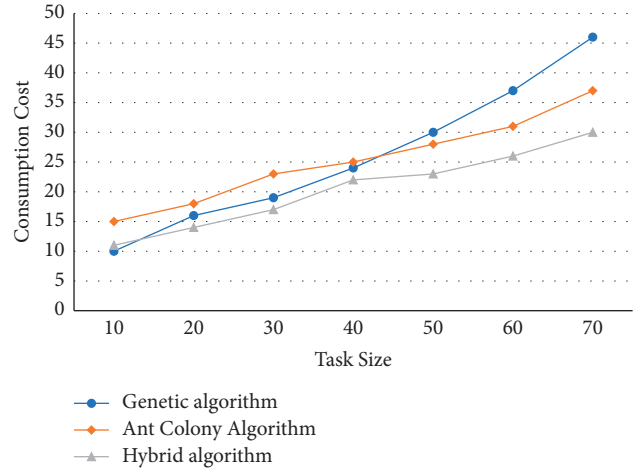


FIGURE 5: Diagram of the cost consumption of various algorithms.

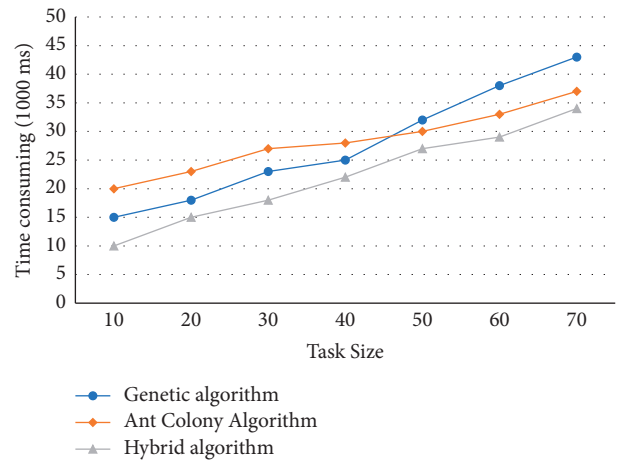


FIGURE 6: Diagram of the time consumption of various algorithms.

6. Conclusion

Both genetic algorithm and ant colony algorithm are more efficient optimization search algorithms. In solving resource

allocation problems, they have their own strengths and their own shortcomings. In this deployment problem, when the task size is below 45, the genetic algorithm can have better computing power; when the task size gradually becomes larger, the genetic algorithm appears to be a little bit powerless. At this time, the ant colony algorithm needs to be used to solve the problem. As the task size increases, the genetic algorithm is more affected than the ant colony algorithm. The hybrid algorithm can take into account the advantages of both and achieve better results overall.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding this work.

Acknowledgments

This work was financially supported by the China Scholarship Council (CSC) fund.

References

- [1] Z. Wu, L. Zhang, Y. Wang, and K. Wang, "Optimization for multi-resource allocation and leveling based on a self-adaptive ant colony algorithm," in *Proceedings of the International Conference on Computational Intelligence and Security*, IEEE Computer Society, Suzhou, China, December 2008.
- [2] Y. Xu, S. Liu, and B. Wang, "Research on computational intelligence in medical resource allocation based on mass customization," *Journal of Universal Computer Science*, vol. 24, no. 6, pp. 753–774, 2018.
- [3] H. A. Le Thi, D. M. Nguyen, and T. Pham Dinh, "Globally solving a nonlinear UAV task assignment problem by stochastic and deterministic optimization approaches," *Optimization Letters*, vol. 6, no. 2, pp. 315–329, 2012.
- [4] Y. M. Nezhad and M. Shahbazian, "Hybrid predictive control design based on particle swarm optimization and genetic algorithm," in *Proceedings of the 2011 3rd International Conference on Computer Research and Development*, pp. 129–134, Shanghai, China, March 2011.
- [5] X. Zhang and X. Chen, "UAV task allocation based on clone selection algorithm," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 5518927, 9 pages, 2021.
- [6] M. Vaibhav, S. Lakshminarayanan, R. Gade Pandu, and L. Titus, "A novel optimal experiment design technique based on multi-objective optimization and its application for toxin kinetics model of hemodialysis patients," *Computer Aided Chemical Engineering*, vol. 30, pp. 1362–1366, 2012.
- [7] S. K. Chaharsooghi and A. H. Meimand Kermani, "An effective ant colony optimization algorithm (ACO) for multi-objective resource allocation problem (MORAP)," *Applied Mathematics and Computation*, vol. 200, no. 1, pp. 167–177, 2008.
- [8] F. M. Bublitz, A. Oetomo, K. S. Sahu et al., "Disruptive technologies for environment and health research: an overview of artificial intelligence, blockchain, and internet of things," *International Journal of Environmental Research and Public Health*, vol. 16, no. 20, pp. 1–24, 2019.
- [9] K. Li, L. Kang, W. Zhang, and B. Li, "Comparative analysis of genetic algorithm and ant colony algorithm on solving traveling salesman problem," in *Proceedings of the IEEE International Workshop on Semantic Computing and Systems*, pp. 72–75, IEEE, Huangshan, China, July 2008.
- [10] A. Forootani, R. Iervolino, M. Tipaldi, and J. Neilson, "Approximate dynamic programming for stochastic resource allocation problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 4, pp. 975–990, 2020.
- [11] B. Tan, H. Ma, and Y. Mei, "A group genetic algorithm for resource allocation in container-based clouds," in *Evolutionary Computation in Combinatorial Optimization. EvoCOP*, L. Paquete and C. Zarges, Eds., Springer, New York, NY, USA, 2020.
- [12] M. Ikeda, L. Barolli, A. Koyama, A. Durrresi, G. De Marco, and J. Iwashige, "Performance evaluation of an intelligent CAC and routing framework for multimedia applications in broadband networks," *Journal of Computer and System Sciences*, vol. 72, no. 7, pp. 1183–1200, 2006.
- [13] C.-H. Lin and P.-L. Lin, "Improving the non-dominated sorting genetic algorithm using a gene-therapy method for multi-objective optimization," *Journal of Computational Science*, vol. 5, no. 2, pp. 170–183, 2014.
- [14] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, 2004.
- [15] J. A. Vrugt, H. V. Gupta, L. A. Bastidas, W. Bouten, and S. Sorooshian, "Effective and efficient algorithm for multi-objective optimization of hydrologic models," *Water Resources Research*, vol. 39, no. 8, 2003.
- [16] Y. Hai, "Improved ant colony algorithm based on PSO and its application on cloud computing resource scheduling," *Advanced Materials Research*, vol. 989–994, pp. 2192–2195, 2014.
- [17] X. Bai, H. W. Kang, Q. Y. Chen, X. P. Sun, Y. Shen, and H. Zhou, "Research of the emergence of virtual organizations based on cloud computing environment," *Applied Mechanics and Materials*, vol. 644–650, pp. 2290–2294, 2014.
- [18] W. Wang, J. Zhao, and J. Huang, "Improved ant colony genetic algorithm for solving traveling salesman problem," *Journal of Physics: Conference Series*, vol. 1693, no. 1, pp. 1–7, 2020.
- [19] M. Vafaie, A. Khademzadeh, and M. A. Pourmina, "A new QoS adaptive multi-path routing for video streaming in urban VANETs integrating ant colony optimization algorithm and fuzzy logic," *Wireless Personal Communications*, vol. 118, no. 6, pp. 1–34, 2021.
- [20] G. Chen, L. Wang, M. Alam, and M. Elhoseny, "Intelligent group prediction algorithm of GPS trajectory based on vehicle communication," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 3987–3996, 2020.
- [21] D. A. Zhuang, C. L. Xiaoyue, C. Xi, and M. Xiaolei, "Joint optimization of scheduling and capacity for mixed traffic with autonomous and human-driven buses: a dynamic programming approach," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 598–619, 2020.
- [22] J. Li and J.-S. Huang, "Dimensions of artificial intelligence anxiety based on the integrated fear acquisition theory," *Technology in Society*, vol. 63, Article ID 101410, 2020.
- [23] H. Chen, S. Chen, M. Li, and J. Chen, "Optimal operation of integrated energy system based on exergy analysis and adaptive genetic algorithm," *IEEE Access*, vol. 8, p. 1, 2020.
- [24] G. Chen and S. Li, "Network on chip for enterprise information management and integration in intelligent physical systems," *Enterprise Information Systems*, vol. 15, no. 7, pp. 935–950, 2021.