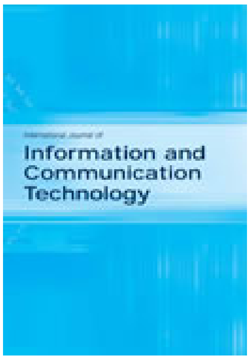


International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Optimised scheduling of network teaching resource management based on improved genetic algorithm

Jing Zhang

DOI: 10.1504/IJICT.2025.10074335

Article History:

Received:	29 July 2025
Last revised:	16 September 2025
Accepted:	17 September 2025
Published online:	13 November 2025

Optimised scheduling of network teaching resource management based on improved genetic algorithm

Jing Zhang

School of Engineering,
Changchun Normal University,
Changchun, 400056, China
Email: 8552062iris@sina.com

Abstract: Network teaching resources provide convenience for daily teaching. Intending to issues of low scheduling accuracy and long response time in management methods of network teaching resources, this paper first optimises the genetic algorithm (GA) based on adaptive neighbourhood and wolf swarm algorithm. First, through the optimisation of encoding and initial population, an adaptive crossover operation based on greedy algorithm is designed. The reversal operation and variable neighbourhood search algorithm are used to complete the mutation operation of the population. Then, a mathematical model of network teaching resource scheduling is established, and the improved GA is used to solve the mathematical model, thereby obtaining the list of network teaching resource information after optimisation scheduling. Experimental results show that the management scheduling accuracy of the proposed method is 98.45%, and the response time is 0.23 s, providing a feasible solution for the intelligent scheduling of network teaching platforms.

Keywords: network teaching resources; management optimisation scheduling; genetic algorithm; GA; adaptive neighbourhood; wolf swarm algorithm.

Reference to this paper should be made as follows: Zhang, J. (2025) ‘Optimised scheduling of network teaching resource management based on improved genetic algorithm’, *Int. J. Information and Communication Technology*, Vol. 26, No. 40, pp.70–85.

Biographical notes: Jing Zhang is a Research Assistant in the School of Engineering at Changchun Normal University, China. She received her Master’s degree from North East Normal University, China, in 2012. Her research interests are instructional technology and teaching management.

1 Introduction

As the information technology and the education sector deeply integrating, web-based instruction has grown into a significant element of modern schooling. Especially under the promotion of the Internet + education strategy, various online courses, virtual experiments, and teaching interaction platforms have shown explosive growth (Aguilar, 2020). However, the explosive growth of resources has also brought serious management challenges. How to efficiently allocate course resources to meet the dynamic needs of different student groups is challenging. Traditional resource scheduling methods are mostly based on static rules or heuristic algorithms, which are difficult to adapt to the spatiotemporal heterogeneity and uncertainty of resource demand (Zhao, 2021). During peak hours, a large number of users accessing popular courses simultaneously may cause server overload, while the long-term idleness of unpopular resources leads to storage waste (Jhonlawi, 2024). In recent years, although intelligent algorithms such as reinforcement learning have been introduced into the resource scheduling field, they rely on large-scale labelled data, have high training costs, and tend to fall into local optimal solutions in dynamic environments (Ni and Xie, 2024). Therefore, exploring a network teaching resource scheduling method with strong adaptability, outstanding global optimisation capabilities, and controllable computational complexity has become a common need in academia and industry (Niu et al., 2019).

Wu (2022) applied a fuzzy tree model to model network teaching resources, which has advantages such as high modelling accuracy and fast operation speed. Li and Li (2024) implemented a dynamic teaching resource management scheme to deploy containers in a Docker Swarm for heterogeneous node groups. The scheme considered the available resources of the node and the demand of other services for the node, to select an appropriate node for container placement. Koch et al. (2016) deployed containers on cluster nodes using runtime monitoring and garbage collection, then adopted optimal scheduling techniques to enhance the efficiency of digital learning resources and ensure balanced workloads. Yang (2024) stated that in the distributed storage system of remote education resources, the processors may be different and resource sharing leads to great randomness in the execution time of parallel tasks on each processor. Zhao (2022) analysed the characteristics of server cluster load balancing, and improved the resource utilisation for network teaching resource load balancing scheduling. Comșa et al. (2023) proposed a network teaching resource scheduling framework in light of reinforcement learning, to minimise the function value, which considered the task deadline, teaching resource constraints, and allowed partial task execution. Lu (2022) proposed a network teaching resource scheduling algorithm based on topological structure, using a resource access cost tree to make optimal dynamic scheduling decisions, to improve resource utilisation and computing efficiency. Offline teaching resource management aims to achieve optimal scheduling through advance planning, ensuring efficient allocation and utilisation of resources when all teaching resources (such as classrooms, instructors, equipment, etc.) and teaching demands (such as course schedules, student enrollment, etc.) are known. However, this process suffers from numerous shortcomings that significantly undermine the final optimal scheduling performance. Offline scheduling relies on comprehensive, accurate data collected in advance. In practice, however, obtaining all relevant data is often extremely difficult. Even if initial data is accurate, teaching resource usage and instructional demands change over time. Since offline scheduling is completed in a single pass, it cannot adjust the schedule promptly in response to these changes, leading to a disconnect between the scheduling results and actual conditions.

In the optimisation scheduling of network teaching resources, heuristic optimisation approaches for example GA, particle swarm optimisation (PSO), simulated annealing (SA), and ant colony optimisation (ACO) have shown significant advantages due to their unique search mechanisms and adaptability. Ge and Chen (2022) first constructed groups for each network teaching resource, and then used ACO to solve each sub-problem, which greatly reduced the difficulty of solving the problem Chen et al. (2021) proposed a network teaching resource management optimisation scheduling method based on GA, considering optimisation objectives such as network communication overhead, load balancing, and resource utilisation, which significantly improved the scheduling accuracy. Cui (2023) proposed a method to optimise the deployment of educational resources and improve workload management using the PSO algorithm. This implementation used PSO to avoid local minima, improving performance by 20%. A scheduling algorithm based on ACO was proposed, which adjusted algorithm parameters by means of neighbourhood division, while considering response time and load balancing. The results showed that the algorithm performance improved by 20–25%. Wei (2020) introduced a multi-objective ACO-based approach for managing online teaching resources, enhancing their utilisation efficiency. Karthick and Gomathi (2022) proposed an improved teaching resource management optimisation method in light of the whale optimisation algorithm (WOA), which has significant advantages. Wang (2023) proposed a hybrid scheduling technology that combines ACO with PSO for network-based teaching resource allocation enhancement, seeking to minimise instructional resource expenditures through optimised workload distribution. Kareem Awad et al. (2025) proposed an adaptive hybrid differential evolution algorithm and modelled the network teaching resource management optimisation scheduling problem, improving resource utilisation.

From the current research on network teaching resource management optimisation scheduling, it can be seen that although existing research has improved resource utilisation to some extent, there are still problems such as low scheduling accuracy and long response time. To this end, this paper proposes a network teaching resource management optimisation scheduling approach in light of an improved genetic algorithm (GA). First, the GA is optimised based on adaptive neighbourhood and wolf pack algorithm. First, the population is initialised using the near-neighbour matrix and adaptive neighbourhood method (ANM), and the selection is carried out using an improved roulette selection method to improve the quality of the initial population. Introduce the adaptive greedy crossover method and parallel crossover method into the crossover operator, and introduce the wolf pack algorithm's fierce wolf charging operation to provide direction for the GA's crossover operation, improving the optimisation efficiency of the algorithm. Introduce the reverse operation and variable neighbourhood search algorithm into the mutation operator to reduce the uncertainty of the GA's mutation operation, and further expand the search range of the mutation operation, improving the algorithm's convergence. Then, a mathematical model of network teaching resource scheduling is established, and the improved GA is used to solve the mathematical model, thereby obtaining the list of optimised network teaching resource information. The experimental outcome implies that the scheduling accuracy rate of the suggested appraoch is 98.45%, and the response time is 0.23s, which can effectively reduce the system response time and improve the scheduling accuracy rate.

2 Relevant technologies

2.1 Basic concepts of scheduling

Scheduling is a decision-making process that plays a key role in information processing in manufacturing and service industries. The subject of scheduling research is to allocate scarce resources to different tasks within a certain time period (Grigorkevich et al., 2022). In the current competitive environment, effective sorting and scheduling have become a necessary condition for survival in the market. For example, companies must meet the delivery dates they have committed to customers, otherwise it will cause significant reputation losses. They must also plan activities in a way that effectively utilises available resources.

Real-time scheduling algorithms are divided into non-preemptive scheduling algorithms and preemptive scheduling algorithms (Xu and Yu, 2024). Once a non-preemptive scheduling algorithm grants a task control of resources, the task must execute until completion or actively release the resources, and it will not be preempted by other tasks during this period. The advantage of this algorithm is that it does not require maintaining a complex task state switching mechanism, reducing the computational burden on the scheduler itself. Preemptive scheduling algorithms allow high-priority tasks or tasks that have exhausted their time slices to interrupt the current task, forcing a control switch, and achieving more granular resource allocation. This algorithm uses time-slice rotation or dynamic priority adjustment to avoid a single task from occupying resources for a long time.

The core principle of non-preemptive scheduling is that once a task begins execution, it continues running until completion or voluntarily relinquishes the CPU (e.g., while waiting for I/O). In non-preemptive scheduling, once a task starts executing, it runs until completion without interruption from higher-priority tasks. If a low-priority task has already begun execution and is running for an extended period, and a high-priority urgent task arrives, the high-priority task must wait until the low-priority task completes before it can be processed. This leads to increased response times for high-priority tasks, failing to meet the rapid response requirements for urgent tasks in real-time systems. The core principle of preemptive scheduling is: the scheduler can interrupt the currently running task at any time and allocate the CPU to another more important task based on priority policies. Preemptive scheduling algorithms must consider additional factors such as task priority, deadlines, and resource utilisation to achieve optimal task scheduling and resource allocation. Compared to non-preemptive scheduling, preemptive algorithms involve more complex design, requiring comprehensive consideration of the interactions between various factors to ensure system real-time performance and reliability.

2.2 Genetic algorithm

Modelled after biological evolution, GA computationally simulates genetic recombination, mutation, and survival-of-the-fittest selection. The unique characteristics of GA primarily stem from their simulation of biological evolution, particularly Darwin's theory of natural selection. Compared to PSO and ACO, which draw inspiration from collective social behaviours like bird flocks or ant colonies, GA's core features manifest in its encoding scheme, operation operators, and the balance between exploration and exploitation. GA: typically employs a fixed-length string (chromosome) to represent a solution. This string may be binary, integer, floating-point, or symbolic (encoding varies by problem). Throughout the process, the algorithm manipulates these encodings rather than the solutions themselves. At the beginning of the algorithm, the set of solutions to the issue is regarded as the initial population. Individuals in the population continuously evolve through genetic operations such as crossover and mutation, and selection operations are performed based on the fitness of individuals. Individuals with higher fitness are more likely to be retained and passed on to the next generation, thus driving the population to continuously update and optimise. Throughout this process, crossover and mutation operations can introduce new gene combinations, enabling interaction among population members. As the number of evolutions accumulates, the overall fitness of the population gradually improves, eventually converging to the optimal solution or an approximate optimal solution (Liu and Wang, 2020). The detailed steps of the GA approach are as bellow.

1 Select an encoding for the problem set and give an initial population with *N* chromosomes. Convert the problem parameters into gene code chains using a certain code system. Each code chain represents an individual, which is a solution to the optimisation problem. Taking binary encoding as an example, a defined correlation exists between the encoding parameters and the resulting encoding, where the encoding parameter μ represents binary encoding or real-number encoding, μ_{max} and μ_{min} are the maximum and minimum values of μ respectively, a is the encoded value, and n is the encoding length.

$$\mu = \mu \frac{a}{2^n - 1}\left(\mu min_{max}\right)_{min} \tag{1}$$

2 Calculate the fitness function value for every chromosome in the population.

3 If the termination criterion is met, halt execution; else, compute probability *P*, and select *N* chromosomes randomly from the old population according to this probability distribution to form a new population. This selection method mainly selects based on the size relationship between the fitness of individuals. The calculation method is as follows, where *q* represents the selection probability of the optimal individual, and *r* denotes its rank position.

$$p_i = \frac{q}{1-(1-q)^p}(1-q)^{r-1} \tag{2}$$

4 Obtain a crossover set of N chromosomes through crossover. The calculation of the crossover operation is as follows, where $\overline{X}$ and $\overline{Y}$ stand for two individuals.

$$\begin{cases} \overline{X} = r\overline{X} + (1-r)\overline{Y} \\ \overline{Y} = r\overline{Y} + (1-r)\overline{X} \end{cases} \tag{3}$$

5 With a small mutation probability, a gene of a chromosome is mutated to form a new population *new_pop*(t+1). Let $t = t + 1$, *pop*(t) = *new_pop*(t), and repeat step (2).

3 Genetic algorithm optimisation based on adaptive neighbourhood and wolf swarm algorithm

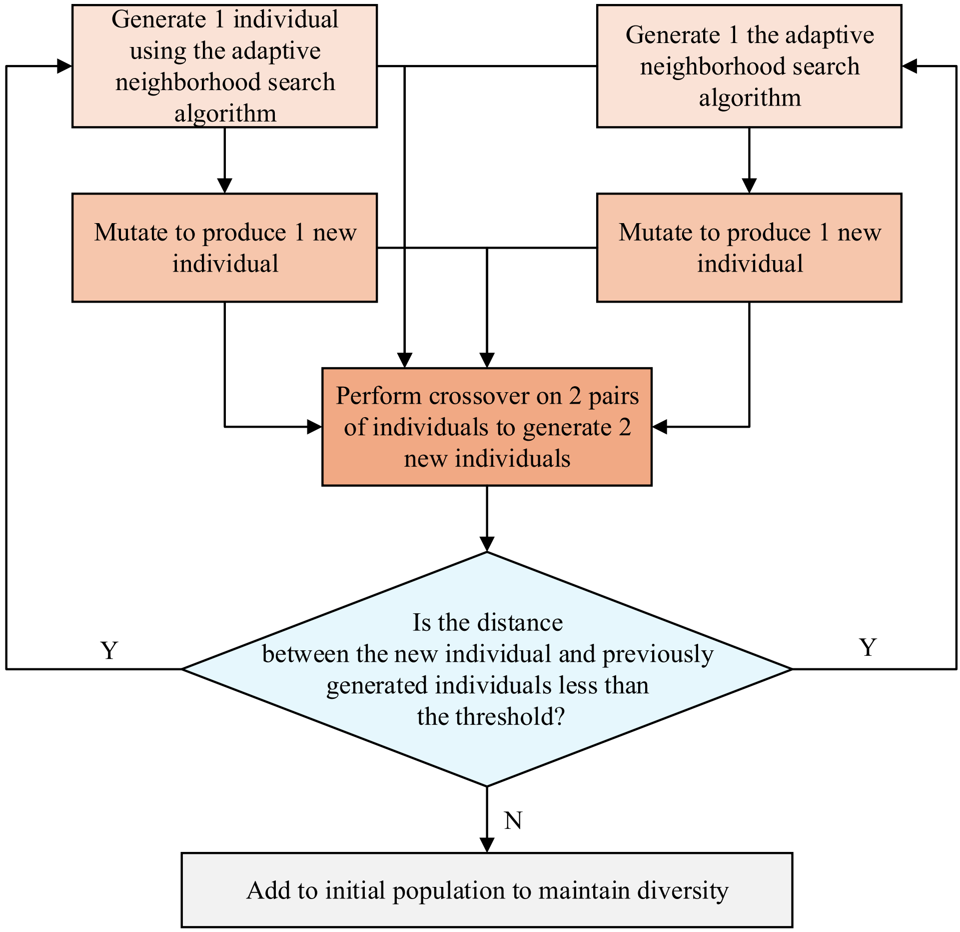
According to what has been mentioned before, the operational solution process of GA is essentially a simulation of the evolutionary process of living things, simulating the genetic operation in biological evolution to solve the problem. However, the local search ability of GA is poor, often converging early when there may be more optimal solutions locally, which cannot guarantee the quality of the optimal solution, and in the process of crossover and mutation operations, the randomness is strong, and the optimisation process lacks efficiency. In the late stage of evolution, with the high degree of population similarity, it is more difficult to enrich the diversity of populations through crossover and mutation, and to further expand the search range, and it is easy to fall into the local optimal solution. Therefore, this paper optimises GA based on adaptive neighbourhood and wolf pack algorithms (Xiu et al., 2020). Firstly, ANM is used to initialise the population. The improved roulette wheel selection method is adopted for selection to improve the goodness of the initial population. Adaptive greedy crossover and parallelism crossover are introduced into the crossover operator, and the wolf attack operation of the wolf pack algorithm is introduced to offer guidance on conducting the crossover operation in the GA and enhance the algorithm's optimisation efficiency. To reduce mutation uncertainty in the GA, inversion and variable neighbourhood search methods are integrated into the mutation operator, and further expand the search range of the mutation operation to improve the convergence of the algorithm.

1 Establishment of the initial population in light of the adaptive neighbourhood approach. To simultaneously optimise the initial population's fitness and genetic diversity, during population initialisation, we introduce an adaptive neighbourhood search approach to create the initial candidate solutions, as shown in Figure 1. The detailed generation approach is as bellow: randomly select a student as the first student to be served in the sequence, i.e., student 1. Then, based on the distance matrix, find the student closest to student 1, i.e., student 2, but do not directly put student 2 into the chromosome encoding sequence. Randomly give a value x between $\gamma\theta$. Multiply this value x by the distance d between student 1 and student 2 to get a data L. Draw a circle with student 2 as the centre and L as the radius, and search all students within this range to form a set. Randomly select one customer as student 2 and put it into the chromosome encoding sequence. Repeat this process until all students are included in this sequence, generating a chromosome that includes all students.

2 Chromosome encoding and chromosome decoding. This paper reduces the possible misreading and errors caused by direct encoding by establishing a correspondence between gene symbols and actual genes. According to the characteristics of large-scale tasks and dynamic changes, the resource occupation of tasks is encoded using a task-container mapping representation. The chromosome's total dimension matches the complete task count. Every chromosome gene encodes as a positive integer denoting the assigned resource number, where the value indicates which resource is allocated to the corresponding task.

3 Design of adaptive crossover and parallel crossover operators based on the greedy algorithm (Zhao et al., 2021). To further boost the performance of the GA's crossover operation and hasten its convergence speed. In the early stage of population evolution, the population is not highly similar, so strong crossover is performed to enrich the diversity of the population, expand the search range, and quickly filter out chromosomes with poor fitness. In the late stage of population evolution, weak crossover is performed, and while protecting the excellent characteristics, strong mutation is used to expand the local search capability of the population and escape from local optima.

Figure 1 Improved population initialisation (see online version for colours)



For a certain population $X_{ik} = \{x_{1l}, x_{12}, \ldots, x_{tn}\}$, where k is the chromosome amount, and i is the encoding at the i^{th} position of the k^{th} chromosome, the steps to calculate the adaptive crossover probability of this population are as follows.

First, compute the mean fitness score across the population's chromosomes f_{ave}, as shown below, where f_{ave} is the average fitness value of the chromosomes in the present population, k is the number of chromosomes in the population, f_k is the fitness value of the k^{th} chromosome in the population.

$$f_{ave} = \frac{\sum_{y=1}^{k} f_k}{k} \tag{4}$$

Second, calculate the adaptive crossover probability for each chromosome in the population, as shown below, where pc is the adaptive crossover probability based on the chromosome individual, pc' is the basic crossover probability, f_{max} represents the maximum fitness value of the chromosomes in the current population, and f_{min} is the minimum fitness value of the chromosomes in the current population.

$$Pc = \begin{cases} Pc' \times cos\left(\dfrac{f_k - f_{ave}}{f_{max} - f_{ave}} \times \dfrac{\pi}{2}\right) \\ Pc' \times \sin\left(\dfrac{f_{avve} - f_k}{f_{ave} - f_{min}} \times \dfrac{\pi}{2}\right) \end{cases} \tag{5}$$

Finally, to enhance population evolution speed and boost algorithmic optimisation performance, a certain gene segment in the elite population *Bestpop* is copied and passed to other chromosomes *otherpop* through a mapping elimination method. *otherpop* will also copy the corresponding gene segments to *Bestpop*. The fitness values of both after crossover are calculated. According to the fitness values, the number of crossovers between *Bestpop* and *otherpop* is determined. The similarity between the two is calculated using the variance of the values. When the similarity reaches a certain threshold, to avoid excessive concentration in the entire population, the crossover will end in advance.

4 Improvement of the mutation operator in light of the reversal operation. The mutation operation of GA is given a certain mutation probability *Pm*. For the goal of improving the efficiency of GA mutation and enhance the controllability of mutation, avoid destroying the building blocks formed by crossover, the reversal operation is introduced to enhance the local seek ability of the approach. This paper focuses on a certain chromosome encoding sequence $X_{ik} = \{x_{il}, x_{ij}, \ldots, x_{ir}\}$, and performs the reversal operation on it. That is: randomly select a customer x_{in} between x_{ir} and x_{il}, select the next population's encoding x_{ij} according to the distance matrix and the neighbour probability matrix, exchange the next encoding number x_{ir} of x_{ir+1} and x_{ij}, and perform the reversal operation on the encoding between x_{ij} and x_{ir+1}.

4 Network teaching resource management optimisation scheduling based on improved genetic algorithm

4.1 Establishment of mathematical model for network teaching resource management optimisation scheduling

The quality of network teaching resources, in addition to whether their content meets the sharing requirements proposed by teachers and students, also needs to consider many other factors, including the transmission bandwidth during the progress of facilitating networked sharing of instructional materials, the operational dynamics of digital pedagogical resources, their networked ecosystem, and stakeholder engagement patterns. These factors directly affect the effect of sharing network teaching resources, and need to be considered in the optimisation process. Transmission bandwidth of network teaching resource sharing. Sharing network teaching resources is generally divided into two ways: one is to download network teaching resources and then apply them to the teaching process. The other is to share network teaching resources online during the teaching process.

Current deployment state of web-based teaching materials. The openness of the network leads to the dynamic development of network teaching resources, with the addition of new teaching resources and the failure of expired teaching resources occurring frequently. The network teaching resources found by general search engines may become inaccessible due to the resource providers revoking the related resources, so the impact of the present working status of network teaching resources needs to be considered during the optimisation process. Predict the possibility of accessing a specific network teaching resource at a certain moment using Google's PageRank algorithm (Park et al., 2019) and probability statistical methods, and use the calculated value as the parameter for the present working status of the network teaching resource. Accounting for the collective influence of these factors on networked educational resource sharing, we formulate the unit resource sharing cost function within the enhanced GA framework for resource information optimisation as bellow.

$$F_i = \frac{1}{X_i^2} e^{x_i} + x_i^{c_i} + (1 - S_i) D_i X_i^2 + H_i X_i \tag{6}$$

where X_i represents the reindexed identifier of the networked educational resource, c_i is the sharing cost situation of the resource, subject to user payment willingness and resource monetisation status. The value is set to 0 if the resource is available for free or if the user is willing to pay for sharing it, and it is assigned a value of 2 when the resource requires payment and the user is unwilling to pay. S_i is the predicted working status value of the resource, a decimal between 0 and 1. Normally, the value of a normal resource is greater than 0.6. D_i is the bandwidth condition of the resource sharing, represented by an integer between 1 and 5. The smaller the value, the more stable the bandwidth connection. H_i is the digital resource hosting infrastructure, employing 0 and 1 to denote the exact network environment and distinct network environments, individually. Based on the chromosome's representation, the fitness operation for the chromosome is defined as the aggregate employment cost of all resources encoded within it.

$$Fitness = -\sum_{i=1}^{n} F_i \tag{7}$$

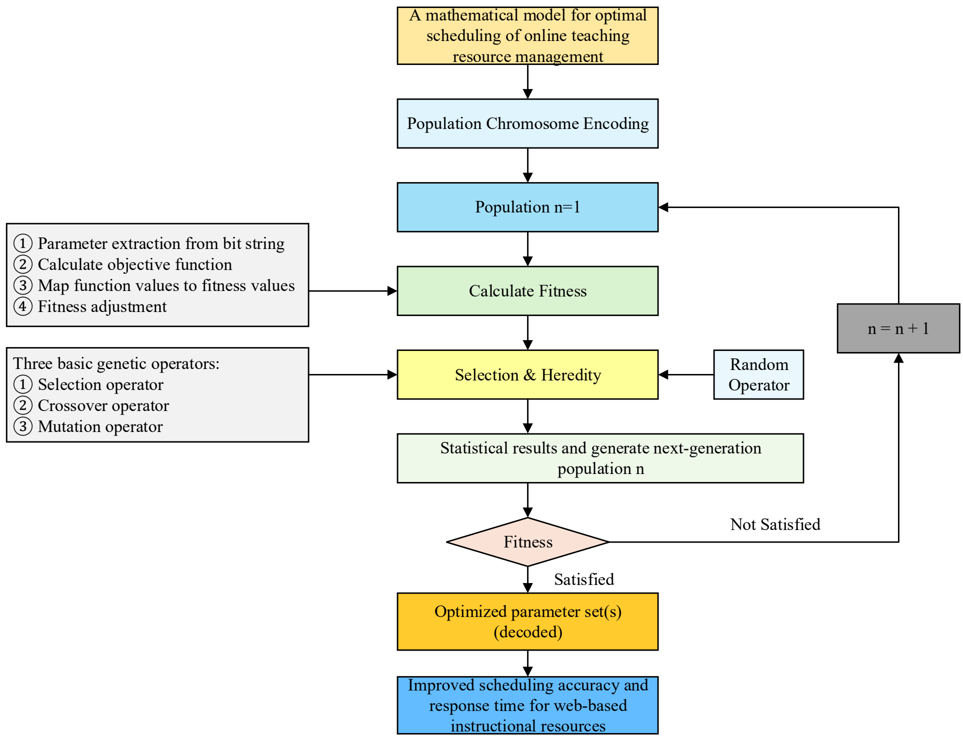
Under the premise that GA implements maximisation in solution space exploration, the fitness function is negated to reformulate the minimisation problem as an equivalent maximisation task. The primal aim is cost minimisation, achieved through maximisation of the negated aggregate sharing expenditures across the resource network, that is, the resource allocation accuracy reaches its global maximum.

$$Cost = \max\left(-\sum_{i=1}^{n} F_i\right) \tag{8}$$

4.2 Optimisation process based on the improved genetic algorithm

The process of optimising network teaching resource information using the improved GA embodies genotype representation. The process encompasses the generation of an initial population, assessment of individual fitness levels, selection of superior individuals, crossover operations to produce the next generation of individuals, individual mutation, along with other iterative steps, as shown in Figure 2. Through continuous evolution and reproduction, the evolutionary process terminates upon achieving either the target objective value or maximum generation threshold, and finally, the scheduling strategy that maximises the utilisation of digital learning resources is obtained. The optimisation steps based on the improved GA are as follows.

Figure 2 Optimisation of online teaching resource management in light of improved GA (see online version for colours)



1 Set the parameters of the improved GA. Let *N* denote the population cardinality and T_{max} represent the iteration upper bound, *Pc* stands for the crossover probability, and select the chromosome with the highest fitness value as the elite chromosome.

2 Encoding and optimisation of the initial population. Use the adaptive neighbourhood search approach to construct the initial population, that is, construct the chromosome according to the neighbour probability matrix, but to further enrich the diversity of the population, the adaptive neighbourhood seek approach is also employed to participate in the composition of the initial population.

3 Improved design of the selection operator based on ANM. Let the amount of chromosomes in the novel population be M (M is an integer multiple of 2 and 5, usually the initial number of chromosomes is set to 50, and 4M/5 chromosomes are chosen from the initial population adopting the roulette wheel method to form part of the population participating in the genetic operation. At the same time, ANM generates the remaining M/5 chromosomes, which together with the chromosomes selected by the roulette wheel method form the population participating in the genetic operation.

4 Improved design of the crossover operator based on the greedy algorithm and parallel crossover. Perform crossover operations using adaptive greedy crossover and parallel crossover. Adaptive greedy crossover first calculates the probability of crossover for each chromosome according to the adaptive crossover probability formula, and then compares whether the chromosome after crossover is better than the current chromosome when performing crossover operations. If it is not better, abandon the current crossover, randomly select the crossover point, and re-crossover until the crossover termination condition is met. Parallel crossover first selects the chromosome with the highest fitness value in the entire population to guide the entire parallel crossover. The remaining chromosomes must all perform crossover operations with it, up to a maximum of 3 times. When the fitness value of the chromosome after crossover is greater than the fitness value of the chromosome before crossover, or the chromosome that has undergone crossover is highly similar to the chromosome guiding the parallel crossover, the crossover will end.

5 Improved design of the mutation operator based on the reversal operation and variable neighbourhood search. Perform mutation operations using the reversal operation and variable neighbourhood search algorithm. First, the point x_{il} closest to the distance x_{ij} is determined using the Euclidean distance formula. If x_{il} is the point that performed the reversal operation in the first step, expand the search range. If not, perform the reversal operation. If the fitness value of the chromosome obtained after the reversal operation is greater than the fitness value of the current solution, the variable neighbourhood search will end. If the solution is not improved, expand the search range, that is, search in a larger neighbourhood. Adaptively expand the search range, that is, the search range is affected by the number of searches. Suppose $X_{ik} = \{x_{il}, x_{ij}, \ldots, x_{in}\}$ has already been searched 3 times, then the neighbourhood range for the fourth search is calculated as follows, where $D(x_{ij}, x_{in})$ represents the distance from all points in X_{ik} to x_{ij}, Length represents the shortest distance from all points to x_{ij}, Q represents the degree factor of adaptive range expansion. The meaning of this formula is that the search range extends from the point x_{ij} closest to it to the points around the distance $Q*Length$, where the calculation of Q is shown in equation (10), where n represents the number of times the neighbourhood is expanded, AND N represents the maximum number of times the neighbourhood is expanded.

$$D\left(x_{ij}, x_{in}\right) \in (Length, Q * Length] \tag{9}$$

$$Q = 1 + \sin\left(\frac{n}{N} * \frac{\pi}{2}\right) \tag{10}$$

6 Dynamic adjustment of crossover probability and mutation probability. By dynamically adjusting the crossover probability and mutation probability to cope with the degree of difference between chromosomes, as shown in equation (11), where $Fitness_{max}$ denotes the peak fitness score attainable by any single genotype, and $Fitness_{ave}$ corresponds to the arithmetic mean of chromosomal fitness values. When Δf is large, the algorithm tends to increase the crossover probability to promote population diversity; while when Δf is small, mutation probability amplification facilitates local optimum evasion and maintains population diversity. This adaptive strategy effectively enhances the global search ability of GA, enabling

it to find optimal solutions more efficiently in complex problems, thus overcoming the defect of GA being prone to fall into local optimum.

$$\Delta f = Fitness_{max} - Fitness_{ave} \quad (11)$$

7 Apply probabilistic mutation operators to offspring chromosomes to preserve population diversity. Upon satisfying termination criteria, the iterative process terminates and returns the optimised resource allocation set. Otherwise, repeat the evolutionary process starting from step (2).

The enhanced digital learning resources are delivered through a page-navigable interface to instructors and learners. The sequence represents the prioritised ordering of network teaching resources encoded in the optimal chromosome. This optimisation framework holistically integrates four critical dimensions - pedagogical content quality, economic cost factors, network transmission bandwidth, and real-time operational status – to generate a comprehensive resource ranking system. By quantifying the collective influence of these multidimensional parameters on sorting algorithms, it significantly enhances the efficiency of educational resource selection for both instructors and learners.

5 Experimental results and analyses

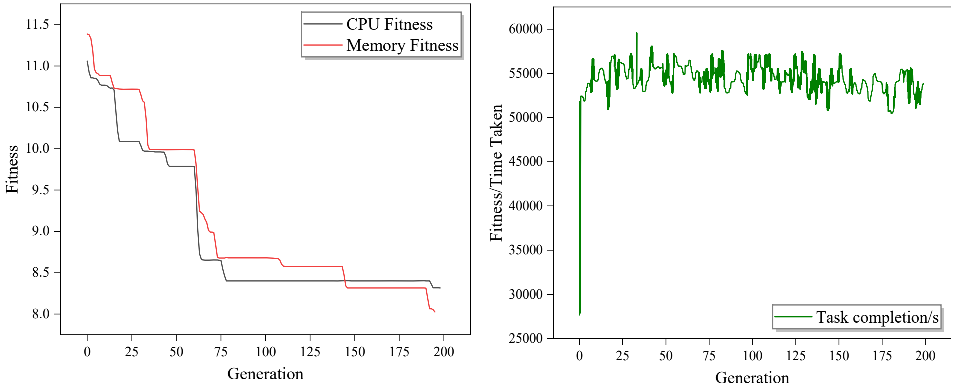
This paper uses the network teaching resource dataset in literature (Shi and Yang, 2020), which contains 12,563 user behaviour data and system performance data. User behaviour data includes student login time, study duration, access frequency, resource download records, etc. System performance data includes CPU, memory, bandwidth utilisation, etc. This paper divides the dataset into training set, test set, and validation set in a ratio of 6:3:1. The hardware environment used in the experiment is Intel Core i7-7500U CPU/8GB, the operating system is windows10, and the programming software is MATLAB 2021a. The population size is 200, the number of generations is 200, the crossover rate is 0.5, and the mutation rate is 0.02.

This paper first analyses the performance of the optimised GA algorithm (AWGA). GA, PSO, ACO, and WOA are selected as comparison algorithms. Each case is independently run 20 times to record its best solution (best) and average solution (mean). The comparative data are implied in Table 1. Compared with the experimental data of GA, PSO, ACO, and WOA, the AWGA algorithm shows better performance in both the best solution and the average value. Moreover, it reaches the known optimal solution with fewer cases. Therefore, SPAS-IGA algorithm demonstrates better optimisation ability and stable performance.

Table 1 Performance of different heuristic optimisation algorithms

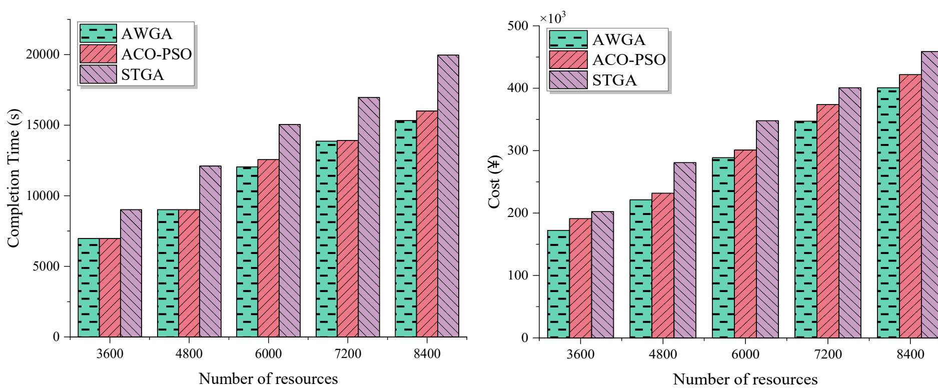
Algorithm	*GA*	*PSO*	*ACO*	*WOA*	*AWGA*
Best	458	431	425	413	406
Mean	471	442	437	429	418

Figure 3 CPU and memory load balancing and task completion per unit time (see online version for colours)



(a) (b)

Figure 4 Completion time and cost of consumption for different methods (see online version for colours)



(a) (b)

Figure 3 shows the load balancing of CPU and memory as well as the number of tasks completed per unit time under the condition of increasing teaching resources. It can be found that in the early stage of scheduling, the load balancing performance of CPU is better, while during the gradual optimisation process, the load balancing of memory gradually improves and becomes better than CPU, but ultimately both reach the optimal values of their performance. From the perspective of the number of tasks completed per unit time, it can reach about 55,000/s, which can meet the actual needs of network teaching resource scheduling.

To further verify the effectiveness of the AWGA method proposed in this paper for network teaching resource scheduling, this paper selects STGA (Chen et al., 2021) and ACO-PSO (Karthick and Gomathi, 2022) as comparison methods. When the number of network teaching resources is 3,600, 4,800, 6,000, 7,200, and 8,400, the completion time and cost of different methods are shown in Figure 4. The number of user tasks increases, the completion time and cost of the three methods increase. When the number of network teaching resources is the same, the completion time and cost of the three methods are AWGA < STGA < ACO-PSO. The two indicators of STGA have a significant difference compared to ACO-PSO and AWGA, while the difference between ACO-PSO and AWGA is small. However, overall, AWGA has a faster scheduling completion time and lower cost. The reason for the difference in the performance of the above approaches is that STGA does not take into account the dynamic changes in user task resource requirements, and the algorithm cannot adapt to sudden changes in the environment, still having room for improvement in the scientific and rational aspects of teaching resource scheduling. Although ACO-PSO considers the dynamic changes in resource requirements, it uses the population at the time of environmental changes as the initial population, resulting in lower algorithm accuracy. AWGA, which is based on adaptive neighbourhoods and wolf pack algorithm for GA optimisation, can better adapt to environmental changes, and has a higher scheduling efficiency.

Table 2 Comparison of scheduling accuracy and response time of different methods

Indicator	*STGA*	*ACO-PSO*	*AWGA*
Scheduling accuracy rate	90.06%	95.17%	98.45
Response time (s)	1.08	0.71	0.23

The comparison of scheduling accuracy rate and response time of different methods is implied in Table 2. As the amount of scheduling tasks increases, the response time of AWGA remains at a low level. Compared with the baseline method, AWGA shows significant advantages when dealing with large-scale scheduling tasks. This is because traditional scheduling requires checking course information, teacher arrangements, and resource conditions one by one, which is time-consuming and error-prone. AWGA uses the efficient search and iteration mechanism of improved GA to quickly generate optimal solutions in a short time, greatly improving scheduling efficiency and being able to respond promptly to resource scheduling changes caused by course adjustments and temporary activities in smart campuses.

6 Conclusions

As the online education quickly growing, a proliferation of web-based pedagogical materials has been empirically observed. Developing robust frameworks for systematic resource governance and intelligent scheduling constitutes a pivotal challenge in advancing virtual education standards. To this end, this paper proposes an optimised management scheduling approach for online teaching resources in light of an improved GA. First, the GA is optimised based on adaptive neighbourhood and wolf pack algorithm. The quality of the initial population is improved by optimising the encoding and initial population. The average fitness value and the maximum and minimum fitness values of the population are introduced into the sine and cosine functions of this paper, and an adaptive crossover operation based on the greedy algorithm is designed. To improve the parallelism of the population, a parallel crossover operation is proposed by combining the wolf pack algorithm. This operation enhances the diversity of the population while ensuring that the crossover of the population has a certain guidance. The mutation operation of the population is completed by using the reversal operation and variable neighbourhood search algorithm, which expands the search range and enhances its local search capability. Then, a mathematical model with resource utilisation cost and scheduling accuracy rate as the objective is established. The improved GA is used to solve the mathematical model, thereby obtaining the list of online teaching resource information after optimisation. Experimental results show that the scheduling accuracy rate of the proposed approach is 98.45%, which is improved by 3.28%–8.39% compared with the benchmark approach. This approach significantly reduces resource allocation latency, improve the scheduling accuracy rate, reduce the user's waiting time, and better meet the diverse needs of different users for teaching resources.

Although parameter adjustment in the improved GA has some adaptive mechanisms, it can still be further optimised. By establishing a more refined dynamic parameter adjustment model, the key parameters of the GA, such as crossover probability and mutation probability, can be adjusted in real-time and accurately according to factors such as the real-time use of teaching resources, changes in network bandwidth, and user request characteristics, ensuring that the algorithm maintains optimal performance in different scenarios.

Declarations

All authors declare that they have no conflicts of interest.

References

Aguilar, S.J. (2020) 'A research-based approach for evaluating resources for transitioning to teaching online', *Information and Learning Sciences*, Vol. 121, No. 5, pp.301–310.

Chen, X., Yue, X-G., Li, R., Zhumadillayeva, A. and Liu, R. (2021) 'Design and application of an improved genetic algorithm to a class scheduling system', *International Journal of Emerging Technologies in Learning*, Vol. 16, No. 1, pp.44–59.

Comșa, I.-S., Molnar, A., Tal, I., Imhof, C., Bergamin, P., Muntean, G-M., Muntean, C.H. and Trestian, R. (2023) 'Improved quality of online education using prioritized multi-agent reinforcement learning for video traffic scheduling', *IEEE Transactions on Broadcasting*, Vol. 69, No. 2, pp.436–454.

Cui, J. (2023) 'Optimal allocation of higher education resources based on fuzzy particle swarm optimization', *International Journal of Electrical Engineering and Education*, Vol. 60, No. 2, pp.312–324.

Ge, R. and Chen, J. (2022) 'Analysis of college course scheduling problem based on ant colony algorithm', *Computational Intelligence and Neuroscience*, Vol. 20, No. 1, pp.79–84.

Grigorkevich, A., Savelyeva, E., Gaifullina, N. and Kolomoets, E. (2022) 'Rigid class scheduling and its value for online learning in higher education', *Education and Information Technologies*, Vol. 27, No. 9, pp.12567–12584.

Jhonlawi, B.A. (2024) 'Optimizing educational scheduling: ACO–based lecture schedule preparation application', *International Journal of Enterprise Modelling*, Vol. 18, No. 2, pp.52–62.

Kareem Awad, W., Zainol Ariffin, K.A., Nazri, M.Z.A. and Yassen, E.T. (2025) 'Resource allocation strategies and task scheduling algorithms for cloud computing: a systematic literature review', *Journal of Intelligent Systems*, Vol. 34, No. 1, pp.20–28.

Karthick, S. and Gomathi, N. (2022) 'Galactic swarm-improved whale optimization algorithm-based resource management in internet of things', *International Journal of Communication Systems*, Vol. 35, No. 3, pp.25–36.

Koch, F., Assunção, M.D., Cardonha, C. and Netto, M.A. (2016) 'Optimising resource costs of cloud computing for education', *Future Generation Computer Systems*, Vol. 55, pp. 473–479.

Li, W. and Li, W. (2024) 'Innovation of dynamic management model for teaching resources of business administration in cloud computing environment', *Journal of Computational Methods in Sciences and Engineering*, Vol. 10, pp.17–31.

Liu, S. and Wang, N. (2020) 'Collaborative optimization scheduling of cloud service resources based on improved genetic algorithm', *IEEE Access*, Vol. 8, pp.150878–150890.

Lu, J. (2022) 'Optimization simulation of balanced distribution of multimedia network modular teaching resources', *Mobile Information Systems*, Vol. 10, No. 3, pp.48–53.

Ni, B. and Xie, X. (2024) 'Distributed distribution and scheduling of teaching resources based on a random matrix educational leadership model', *Informatica*, Vol. 48, No. 8, pp.22–29.

Niu, L., Chen, X. and Xu, R. (2019) 'Quantitative analysis of the influence of learning resource scheduling in MOOC mode on traditional education and teaching', *International Journal of Continuing Engineering Education and Life Long Learning*, Vol. 29, No. 2, pp.21–32.

Park, S., Lee, W., Choe, B. and Lee, S.-G. (2019) 'A survey on personalized PageRank computation algorithms', *IEEE Access*, Vol. 7, pp.163049–163062.

Shi, Y. and Yang, X. (2020) 'A personalized matching system for management teaching resources based on collaborative filtering algorithm', *International Journal of Emerging Technologies in Learning*, Vol. 15, No. 13, pp.207–220.

Wang, H. (2023) 'Balanced allocation of teaching information resources based on discrete particle swarm optimisation algorithm', *International Journal of Computer Applications in Technology*, Vol. 73, No. 4, pp.304–312.

Wei, X. (2020) 'Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing', *Journal of Ambient Intelligence and Humanized Computing*, Vol. 7, pp.1–12.

Wu, M. (2022) 'A music teaching resource management model based on fuzzy clustering algorithm', *Mobile Information Systems*, Vol. 8, No. 1, pp.34–48.

Xiu, Y., Hao, Y., Yong, L. and Ren-rong, X. (2020) 'A clustering routing algorithm based on wolf pack algorithm for heterogeneous wireless sensor networks', *Computer Networks*, Vol. 167, pp.8–14.

Xu, L. and Yu, W. (2024) 'Design and implementation of artificial intelligence online learning platform based on resource scheduling technology', *Journal of Cases on Information Technology* (*JCIT*), Vol. 26, No. 1, pp.1–22.

Yang, Z. (2024) 'Research and implementation of education resource scheduling algorithm based on machine learning', *Procedia Computer Science*, Vol. 247, pp.271–280.

Zhao, K. (2021) 'A novel method for integration of online educational resources via teaching information remote scheduling', *Mathematical Problems in Engineering*, Vol. 20, No. 4, pp.45–53.

Zhao, Z. (2022) 'Optimization strategy of online teaching management mode based on cloud computing', *Curriculum and Teaching Methodology*, Vol. 5, No. 8, pp.75–81.

Zhao, Z., Zhou, M. and Liu, S. (2021) 'Iterated greedy algorithms for flow-shop scheduling problems: a tutorial', *IEEE Transactions on Automation Science and Engineering*, Vol. 19, No. 3, pp.1941–1959.