Research Article

Improved Genetic Algorithm for English Information Education and Teaching Quality Impact Analysis

Yuanbo Li1 and Wanhe Song2

1 Department of Basic Teaching, Hebei Vocational College of Rail Transportation, Shijiazhuang 050050, China
2 Department of Basic Teaching, Hebei Vocational University of Industry and Technology, Hebei, Shijiazhuang 050091, China

Correspondence should be addressed to Wanhe Song; songwanhe5918@163.com

Received 27 May 2022; Accepted 2 July 2022; Published 31 July 2022

Academic Editor: Qiangyi Li

Copyright © 2022 Yuanbo Li and Wanhe Song. 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.

With the increase of the number of college students, in order to improve the quality of teaching, the professional setting is more and more detailed, the course content is more and more rich, the information teaching system will be increasingly perfect. This paper proposes an improved genetic algorithm for English informatics education scheduling system to address the problem of the contradiction between the rapidly increasing number of students and the limited teaching resources in the current English scheduling in universities. The system first proposes an improved parallel mechanism which computes the probability of population changing evolutionary strategies according to the algebra of population maintaining the optimal solution invariant. The evolution strategy needed by the population can be obtained by fuzzy reasoning according to the evolution degree and individual difference of the population so as to get rid of the local extreme value and search for the global optimal value. Secondly, an improved algorithm initialization method, fitness transformation method, and competition strategy are proposed to adjust the balance between convergence speed and population diversity. Finally, based on Baldwin evolution theory, a new local search strategy is proposed. The algorithm has been tested on famous international data sets and a school’s English curriculum. Experimental results show that this algorithm has higher efficiency and quality than other algorithms with better performance, and the proposed teaching system can improve the teaching quality of English information education.

1. Introduction

Due to the development of information technology platforms, the teaching system can be upgraded by making full use of information technology, thus adding to the improvement of teaching quality. Improving and optimizing the teaching scheduling method according to the teaching schedule, content, and mode constitute a worthy research direction to improve the teaching quality. At present, the traditional manual scheduling method commonly used in domestic universities is mainly set according to the previous scheduling experience. However, the traditional manual scheduling method lacks complete systematic theoretical support and computer data modeling. In actual operation, it often has disadvantages such as involving many, communicating a lot, being slow, low efficiency, being error-prone, and subjective consciousness [1]. With the increase in the number of college students, the refinement of majors, and the diversification of courses, our familiar manual course scheduling has gradually failed to do its job effectively [2]. It can optimize the course scheduling system of colleges and universities and rationalize the planning and design of various courses through computer algorithms [3]. It can greatly improve the efficiency of teaching managers, effectively optimize teaching management, and improve teaching quality.

The automatic class scheduling problem is proved to be a nondeterministic polynomial (NP) complete problem [4]. Due to the complexity of this kind of problem, more and more scholars have paid attention to and studied it. In 2002, Britain and Italy teamed up to organize the five-yearly International Timetabling Competition (ITC), which aims to
provide researchers with a universally accepted benchmark against which the gap between research and practice is bridged. Curriculum scheduling is usually defined as the rational arrangement of limited educational resources under certain constraints.

The importance and complexity of curriculum arrangement have attracted the attention and research of many experts and scholars for a long time. Some foreign scholars put forward the corresponding mathematical model of curriculum arrangement, which laid a certain mathematical foundation for future research. A domestic scholar proved that the problem of curriculum arrangement is NP complete, so that people have a preliminary theoretical understanding of the complexity of curriculum arrangement. However, before the 1990s, most of the solutions relied on the experience of educational administrators, with low versatility and practicability. In the 1990s, more and more intelligent optimization algorithms have been used to solve the curriculum scheduling problem, and the corresponding results have been achieved. In recent years, the solutions include genetic algorithm [5], simulated annealing algorithm [6], ant colony algorithm [7], tabu search algorithm [8], and hybrid algorithm [9]. However, these schemes are deficient in consideration of constraints and are basically aimed at the scheduling problem of the general class teaching mode, which is not suitable for solving the scheduling problem of English information education.

As a self-organizing and adaptive artificial intelligence technology, genetic algorithm (GA) provides an effective tool for general combinatorial optimization problems. However, with the continuous expansion of modern information technology applications and the complexity and diversity of practical engineering problems, the shortcomings of the traditional genetic algorithm gradually appear. In the intelligent course scheduling problem, the NP problem of complex combinatorial optimization with multiple constraints is difficult to solve by traditional genetic algorithm. In order to design a more efficient genetic algorithm, a kind of fuzzy adaptive parallel genetic algorithm (FAPGA) is proposed based on previous research.

The innovations of this paper are as follows:

1. The algorithm framework in this paper adopts fuzzy adaptive parallel mechanism. This mechanism improves the algorithm’s ability to overcome problems such as difficult selection of \( P_c \) and \( P_m \) values and premature convergence, and can adaptively adjust the evolutionary strategy according to individual differences between chromosomes and population evolution.

2. This paper adopts the population initialization method and competition strategy based on distance adjustment. The initialization method can distribute chromosomes in different regions of the solution space to ensure the diversity of initial solutions. As a constraint condition, competition strategy continuously adjusts the diversity lost in the process of evolution, so that the population can maintain rich diversity and set a reasonable search range in the process of evolution.

3. This paper adopts the improved fitness transformation method. This method uses a more stable scale standard and can control the fitness transformation non-linearly, so that the new fitness after transformation is closer to the requirements of algorithm evolution.

4. Local search algorithm adopts Baldwin learning local search strategy based on simulated annealing. This strategy can fully explore the region of individual solution space and improve the local search ability of the algorithm.

This paper consists of four main sections: Section 1 provides an introduction, Section 2 presents the methodology, Section 3 analyzes and discusses the results, and Section 4 concludes the paper.

2. Methodology

In this section, the coding method of the algorithm is first introduced, then the basic idea of the parallel mechanism of fuzzy adaptive is explained, followed by the description of the initialization method and the competition strategy of the algorithm, and finally the improved scale transformation method and the local search strategy are explained. FAPGA takes the parallel mechanism as the basic framework of the algorithm and integrates various improved methods into different links of the algorithm to form a new improved genetic algorithm.

2.1. Fuzzy Adaptive Parallel Mechanism

2.1.1. Basic Principle of Parallel Mechanism. The introduction of parallel mechanism in genetic algorithm can effectively combine the natural parallelism of GA and the fast concurrency of computer, which is an important direction to improve the performance of algorithms proposed in recent years. There are four main types of genetic algorithm parallel models: master-slave model, fine-grained model, coarse-grained model, and hybrid model. The parallel mechanism proposed in this paper adopts coarse-grained model. The mechanism expands the search scope through the parallel evolution of multiple populations instead of a single population, in which each population becomes an evolutionary unit and evolves independently according to different evolutionary strategies. When the genetic generation of the algorithm reaches a certain period, individual migration will be carried out; that is, a certain number of excellent individuals will be exchanged between various groups. The parallel mechanism configures different algorithm parameters for different populations so as to effectively control the independent evolution characteristics of populations. The diversity in the process of population evolution is guaranteed, and the coordination among various groups is realized through migration, which improves the performance of the algorithm.
2.1.2. Evolution Strategy of Parallel Mechanism. At present, the improvement of genetic algorithm can be summarized as coding, micro genetic strategy (improvement of genetic operator and parameter selection), and macro genetic strategy (improvement of algorithm operation mechanism). This paper studies the macro multi-population parallel mechanism rather than specific evolutionary strategy, so the parallel mechanism adopts the simpler evolutionary strategy proposed in literature [10] and makes improvements. Specifically, \( P_m \) with a larger value and \( P_c \) with a smaller value are called an exploration strategy. In terms of genetic operation, mutation operation is selected first, then crossover operation is selected, and finally selective replication operation is performed. This evolutionary strategy is beneficial for jumping out of local optimum and exploring new solution space. FAPGA uses the common population as a vessel for receiving and developing excellent individuals from other populations. First, we give the definition of evolutionary potential.

**Definition 1.** The evolutionary potential of individual \( i_x \) revealed by local search is defined by the following formula:

\[
V(i_x) = \omega(1 - Z^z)(f'(i_x) - f_{\text{max}}).
\]

In (1), \( \omega \) represents the weight of evolutionary potential, \( z \) represents the temperature attenuation coefficient, and \( \lambda \) represents the search times of new solutions. \( f'(i_x) \) represents the optimal solution of chromosome \( i_x \) after local search, and \( f_{\text{max}} \) represents the maximum population fitness. The individual's evolutionary potential is related to the search times of new solutions and the fitness of the optimal solution, and the fewer the search times, the greater the evolutionary potential.

The public population first interoperates with excellent individuals. At the end of crossover, all individuals are searched according to the local search strategy, and the evolutionary potential is calculated. Then, the fitness is updated according to (2). Finally, chromosomes of population size and number are selected according to fitness.

\[
f(i_x) = f(i_x) + V(i_x).
\]

The purpose of common population is to discover individuals with evolutionary potential and to discover new solutions. Individuals with high evolutionary potential are more likely to be selected for the next generation.

2.1.3. Adaptive Judgment of the Probability of Strategy Transformation. In the process of population evolution, if the population is stagnant or trapped in the local optimal solution, the evolution strategy should be changed according to the population evolution. In this paper, the evolutionary algebra \( G_f \), whose optimal solution remains unchanged, is introduced to measure the evolutionary state of the population, and (3) is used to calculate the probability of population changing strategy.

\[
U_{cb} = \frac{(G - G_m)}{G} \cdot \frac{1}{1 + \exp\left[\beta(2(G_f/G_{\text{max}}) - 1)\right]}
\]

In (3), \( G_m \) is the current evolutionary algebra and \( G \) is the total evolutionary algebra. \( \beta \) is the control parameter and has a value of 6 in the text. \( G_f \) is the algebra of population stagnation, and \( G_{\text{max}} \) is the maximal stagnation algebra.

The solid curve in Figure 1 represents the rule of \( U_{cb} \) transformation with stagnation algebra when evolutionary algebra is invariant. The nonlinear probability can control the evolution strategy of population transformation more reasonably. \( U_{cb} \) increases slowly in the early stage of evolutionary stagnation, leaving some time for the population to get rid of the stagnation by existing evolutionary strategies. With the increase of stasis algebra, the algorithm determines that the population is in a stagnant state, which significantly improves the \( U_{cb} \) compared to the probability of linear control and avoids the population wasting time on the existing evolution strategy. Finally, in order to keep the balance between population exploration and algorithm convergence speed, \( U_{cb} \) decreases linearly with the further increase of evolutionary algebra when the stagnation algebra is constant.

2.1.4. Evolutionary Strategy of Fuzzy Reasoning. In this paper, fuzzy reasoning is used to adjust the evolution strategy of population according to the individual difference and evolution status. Equation (4) is adopted to represent the population evolution.

\[
E_1 = \frac{(f_{\text{max}} - f_{\text{avg}})}{f_{\text{max}}} \in [0, 1].
\]

The size of \( (f_{\text{max}} - f_{\text{avg}}) \) is often used as a measure of the degree of population evolution. The method is indeed effective, but the shortcoming is that the fitness function is designed according to the specific problem, and its value varies with the specific problem. In addition, the maximum and minimum fitness of the population are also changing during the algorithm calculation, so it is difficult to determine the appropriate threshold to judge the evolution degree of the population. Equation (4) is more general to measure the degree of population evolution. Equation (5) represents the individual difference of the population.

\[
E_2 = \frac{1}{T} \sum_{i=1}^{T} \left(\frac{f(i_x) - f_{\text{min}}}{f_{\text{max}} - f_{\text{min}}} \right) \in [0, 1].
\]

The smaller the value of \( E_2 \) is, the more dispersed the population fitness is, the greater the population difference is, and the better the diversity is. Otherwise, the population diversity is poor.

In (4) and (5), \( f(i_x) \) represents the fitness of individual \( i_x \), and \( f_{\text{max}} \) is the maximum fitness of the population. \( f_{\text{min}} \) is the minimum population fitness, \( f_{\text{avg}} \) is the average population fitness, and \( T \) is the number of population chromosomes.
The fuzzy variables $E_1$ and $E_2$ are divided into three language sets [large, medium, and small]. The evolution strategy is divided into three language sets: exploration, common, and development. Table 1 shows the fuzzy rules of the algorithm inference evolution strategy based on $E_1$ and $E_2$. If $E_1$ is "large" and $E_2$ is "small," this indicates that the population has a low degree of evolution and good diversity and the evolutionary strategy should be "development." If $E_1$ is "small" and $E_2$ is "large," this indicates that the population has a high degree of evolution and poor diversity and the evolutionary strategy should be "exploration." Other rules can be obtained by similar reasoning according to the above rules.

One traditional way to measure individual similarity is the Hemming distance. Let $i_x$ and $i_y$ be two distinct individuals, and the distance $D(i_x, i_y)$ between them can be defined as follows:

$$D(i_x, i_y) = \sum_{z=1}^{l} |i_x^z - i_y^z|.$$  \hspace{1cm} (6)

In (6), $l$ is the length of chromosome gene and $i_x^z$ and $i_y^z$ represent the $z$-th gene position of individuals $i_x$ and $i_y$, respectively. For binary code scheme, if $i_x^z \neq i_y^z$, then $|i_x^z - i_y^z| = 1$; otherwise, $|i_x^z - i_y^z| = 0$.

Different gene loci in binary code have different effects on individuals; that is, the actual distance between two individuals with a very small Hamming distance may be very large in the solution space. Therefore, Hamming distance cannot fully explain the differences between individuals. Weighted Hamming distance can be used to assign weights to genes at different positions based on the Hamming distance. The calculation formula is shown as follows:

$$D_w(i_x, i_y) = \sum_{z=1}^{l} m_z |i_x^z - i_y^z|.$$  \hspace{1cm} (7)

where $m_z$ represents the weight of gene position $z$. In this article, $m_z = 2^z$. Suppose $x_1 = 100101$, $x_2 = 1000000$, $x_3 = 000100$, and the weighted Hamming distance between the three individuals is 2. $D_w(i_1, i_2) = 2^0 + 2^2 = 5$, $D_w(i_1, i_3) = 2^0 + 2^3 = 33$, $D_w(i_2, i_3) = 2^2 + 2^3 = 36$. It can be seen that two individuals with the same Hamming distance have different weighted Hamming distance, which can better measure the difference between individuals.

The distance between individuals is defined by the ratio of the weighted Hamming distance to the maximum weighted Hamming distance as follows:

$$d(i_x, i_y) = \frac{D_w(i_x, i_y)}{\sum_{z=1}^{l} 2^z}.$$  \hspace{1cm} (8)

where $l$ is the length of chromosome gene. The formula of individual similarity determination is shown in Equation (9).

$$S(i_x, i_y) = \begin{cases} 1, & d(i_x, i_y) < \alpha, \\ 0, & d(i_x, i_y) \geq \alpha. \end{cases}$$  \hspace{1cm} (9)

In (9), $\alpha$ represents the distance threshold of individual similarity. If the distance is less than this threshold, the algorithm considers $i_x$ and $i_y$ to be similar; otherwise, they are not similar. Each individual in the group needs to conduct individual similarity judgment with other individuals to calculate individual similarity $r(i_x)$.

$$r(i_x) = \sum_{y=1}^{T} S(i_x, i_y).$$  \hspace{1cm} (10)

It can be seen that $r(i_x)$ represents a similar cumulative value of individual $i_x$ and other individuals. The larger the value of $r(i_x)$, the more the similar individuals in the group. If $r(i_x)$ is greater than the similarity threshold $\eta$, the individual $i_x$ is called a similar individual.

In the process of population evolution, a fixed value of distance threshold $\alpha$ will affect the algorithm process. FAPGA adopts (11) to change the distance threshold $\alpha$ along with the progress of the algorithm.

$$\alpha = \alpha_2 + (\alpha_1 - \alpha_2) \left(1 - \frac{t}{A}\right).$$  \hspace{1cm} (11)

In (11), $\alpha_1$ is the maximum distance threshold, $\alpha_2$ is the minimum distance threshold, $\alpha$ represents the current evolutionary algebra, and $A$ represents the total evolutionary algebra. At the early stage of evolution, the value of $\alpha$ was larger, and more similar individuals were adjusted to maintain population diversity. As the algorithm progresses, the distance between individuals shrinks, and the degree of similarity between individuals becomes higher and higher. If a fixed value of $\alpha$ is used, more individuals will be judged as similar individuals in the later stage of the algorithm, and adjusting more individuals will slow down the convergence rate of the algorithm. With the increase of the evolutionary
algebra, the value of $a$ should be continuously reduced to avoid affecting the convergence speed of the algorithm.

2.2. Improved Adaptive Scaling. GA can adopt different selection strategies, such as breeding pool selection and Boltzmann selection [11], but many selection strategies are based on proportional selection methods.

$$f'(i_n) = \frac{f(i_n)}{\sum f(i_n)}$$

$$N = \beta \frac{f_{\min} - f_{\max}}{\log_{10} 95}$$

In literature [12], (12) was used for fitness transformation. In (12), $f_{\min}$ represents the minimum population fitness, $f_{\max}$ represents the maximum population fitness, $N$ represents the temperature of the simulated annealing method used in the literature, and $\beta$ represents the attenuation coefficient. Equation (12) can modify the fitness according to the evolution period, avoiding the “premature” convergence due to excessive selection of super individuals in the early stage of the algorithm. However, the defect is that after the transformation of the original fitness equation (12), the new fitness gap between individuals becomes too small, and the probability of excellent individuals being selected in the early stage of evolution decreases too much, which affects the ability to search for excellence.

FAPGA performs fitness transformation through (13) and proportional selection operation based on the transformed fitness.

$$f'(i_n) = f(i_n) + Gf_{\text{avg}}$$

In (13), $f_{\text{avg}}$ represents the mean value of population fitness, and $G$ is an important parameter of fitness transformation. The calculation method is shown in the following equation:

$$G = \frac{1}{1 + e^{(2\alpha 1 - 1)}} \in [0, 1].$$

In (14), $A$ represents the current evolution algebra, $a$ represents the total evolution algebra, and the value of $g$ is 6.

2.3. Baldwin Learning Local Search Strategy Constructed Based on Simulated Annealing. Firstly, the neighborhood structure $l$ is defined as follows:

$$l = \delta \prod_{x=1}^{p} L_x = \delta \prod_{x=1}^{p} \frac{\theta(u_x(\text{max}) - u_x(\text{min}))}{2^l - 1}$$

In (15), $u_x(\text{max})$ and $u_x(\text{min})$, respectively, represent the upper and lower bounds of the domain of decision variable $u_x$, and $p$ represents the dimension of decision variable (i.e., problem dimension). The gene length is represented by $l$, $\delta$ is the range coefficient of the search space, and $\theta$ is the coefficient controlling the size of neighborhood structure.

Assume that the dimension of the solution problem is 1, and the corresponding function of the problem is represented by $f(u_x)$. As shown in Figure 2(a), the binary coding string represented by chromosomes with gene length $l$ can partition the domain $[u_x(\text{min}), u_x(\text{max})]$ of decision variable $u_x$ into $(2^l - 1)$ uniform regions. Each region represents a neighborhood structure, that is, a line segment of length $L(u_x)$. The function of solving the problem dimension 2 is represented by $f(u_x, u_{(x+1)})$. As shown in Figure 2(b), the definition domains of decision variables $u_x$ and $u_{(x+1)}$ are partitioned into $(2^l - 1)$ uniform regions, and there are $(2^l - 1)^2$ neighborhood structures in the solution space, where each neighborhood structure represents a plane matrix with an area of $L(u_x) \times L(u_{(x+1)})$. By analogy, the $z$-dimensional solution problem contains $(2^l - 1)^z$ neighborhood structures, each representing a $K$-dimensional cube of volume $\prod_{x=1}^{P} L(u_x)$. In this way, a one-to-one relationship is established between the target solution space represented by the solution problem and the search space of GA.

The circular entity represents the individual, the dotted matrix represents the neighborhood structure in the solution space, the arrow represents the search direction, and its serial number represents the current search times. Figure 3(a) represents the neighborhood structure that can be searched by an individual through the first search, and Figure 3(b) represents the neighborhood structure that can be searched theoretically through the second search. The above strategies enable individuals to explore neighborhood structures in multiple directions. The scope of neighborhood structures can be adjusted by (15), so as to establish a local search strategy with controllable search scope and intensity. Most existing studies have adopted local search strategies based on Lamarckian evolution theory [13]. Literature [14] uses mountain climbing as the local search strategy of the algorithm for detailed search. Literature [15] added simulated annealing random disturbance decision variables to genetic algorithm for local search. Although the above methods enhance the local optimization ability of the algorithm, they cannot effectively distinguish whether the new individual is the local optimal solution. The strategy of this paper is based on Baldwin’s theory of evolution, in which an individual discovers a new optimal solution through local search without changing the original individual’s genotype. This method not only achieves the purpose of searching new optimal solution, but also reduces the risk of population falling into local optimum.

2.4. Algorithm Flow. The algorithm flow is shown in Figure 4.

2.5. Architecture of Teaching Quality Evaluation System. The architecture of the teaching quality evaluation system is designed from the perspective of the functional requirements of the teaching quality evaluation system. The system mainly includes the design of the client side and the design of the system service side. It can be divided into four parts: web
server, database, PC terminal, and mobile terminal, as shown in Figure 5. When the system is running, the client sends a request to the server, and the web port accepts the request and retrieves the corresponding data from the database. Finally, the required data is sent to the client.

The main role of the client is to preprocess the data related to education evaluation and send the processed data information to the server for analysis and storage by means of web technology. This process is based on the client/server architecture. This architecture allows for a reasonable distribution of tasks between the client and the server. It makes full use of the advantages of the hardware environment at both ends and enables the sharing of information resources and data processing on the network. After accessing the network, the client can log in to the teaching quality evaluation system to modify the account-related information and obtain the evaluation information needed for teaching.

In this system, the database is a data collection warehouse for storing and managing data. As the core part of the teaching evaluation system, the data will have a direct impact on the normal operation of the whole system. Before designing the database, we consider the correlation between each data table and import the data table into the database to facilitate the quick and accurate retrieval of data later. The database management system used in this system is MySQL database, which is popular in the field of web development for its low cost, high performance, ease of use, and multi-platform advantages.

In the data preprocessing stage, more than 2000 pieces of teaching quality evaluation information were extracted from the teaching evaluation system, and the extracted data were summarized into a data table according to students’ grade, major, gender, teachers’ age, academic title, and courses. With the help of association rule data mining method, we can find the approximate relationship between teachers’ information and their ratings and adjust the results to students’ related information. By setting the support level $\alpha = 3\%$ and confidence level $\beta = 5\%$, the data is inputted into

---

**Figure 2:** Illustration of population initialization method.

**Figure 3:** Illustration of population initialization method. (a) First search. (b) Second search.
the system to obtain the strong rule term, and finally the relationship between a certain indicator and the evaluation score can be obtained.

3. Experiment and Analysis

In this paper, algorithm validation experiments are carried out on famous data sets and then on an English class schedule in a school. Each example was run independently 30 times to get the best result and compared with the best results obtained internationally. The experimental hardware environment is Intel Core i5 2.6 GHz, the memory is 2 GB, and C++ programming is adopted.

3.1. Lewis 60 Example Solution. The Lewis contains 60 examples with only hard constraints, and the difficulty of hard constraints is greater than that of other public data sets and real class scheduling problems. Each example has at least one feasible solution. Lewis pointed out that algorithms can be compared fairly when they focus only on resolving hard conflicts. The sample group was evenly divided into three groups, each containing 20. Specific size is as follows: small (200 | n | 225 or less, or less | m | 5∼6), medium (390 | n | 425 or less, or less | m | 10 and 11), and large (1000 | n | 1075 or less, or less | m | 25 to 28). n represents the number of courses to be arranged, and M represents the number of available rooms. Figure 6 shows a comparison between the proposed algorithm and five other algorithms [16–20] with better results.

If all courses can be arranged in the schedule without conflict, that is, the result is 0, this indicates that the feasible solution of the sample can be obtained directly. The best result obtained in previous studies was that of literature [20], which obtained 54 feasible solutions, while the method in this paper can obtain 59 feasible solutions.

The above six algorithms are simulated on the following standard test functions (see Figure 7). $f_{10}$ is a high-dimensional function.

$$f_{10} = \sum_{x=1}^{D} \frac{x}{x}$$

(16)

As can be seen from Figure 7, within a small number of iterations, the convergence curve represented by the model in this paper presents a "steep" trend in the image solved by the function, indicating that the proposed algorithm has better convergence ability.
3.2. Experimental Verification Carried Out in a School’s English Schedule

(1) In the performance evaluation of the algorithm, the quality of English course scheduling scheme is compared from the aspects of excellence in saving time, uniformity of class hour distribution, optimization degree of course combination, classroom constraints, and time consumption. The improved genetic algorithm is compared with other five algorithms (see Table 2).

It can be seen from Table 2 that the improved genetic algorithm is higher than other five algorithms in terms of the degree of scheduling, the evenness of class hour distribution, the degree of course combination optimization, and the teacher constraint satisfaction.

(2) In terms of the efficiency of arranging English courses, the experiment is carried out from the aspects of students' satisfaction with course selection, classroom utilization rate, and satisfaction with the overall arranging rules (see Table 3).

In Table 3, two different data sets are selected to carry out the application experiment of the proposed algorithm. It shows that the rate of students' satisfaction with course selection reaches 100%, the average satisfaction rate of overall rules reaches 96.63%, the classroom utilization rate reaches 83.53%, and the course uniformity reaches 0.9.

(3) In terms of the satisfaction rate of settable rules for English course scheduling, the results are shown in Table 4.

Table 4 shows the satisfaction rate of English course scheduling under different English course scheduling rules. It basically satisfies the rule constraint condition of the current university intelligent course arrangement. The scheduling satisfaction rates for teacher unscheduled classes, limit class hours, global fixed points, conflicts between two classes, and class hour priorities reach a mean value of 98%.

3.3. Analysis of the Impact of the Information-Based Scheduling System on Teaching Quality

To apply the information-based scheduling system to the scheduling of English teaching and also to further verify the impact of the proposed model on the quality of English teaching, in the later stage of teaching, the course number was divided into two parts. One part is A, and the students in this part are scheduled and taught by the information-based scheduling system. The other part, B, was monitored to observe the students' natural evaluation of the quality of teaching and
learning in the case of the traditional scheduling system. After a period of 4 weeks, the learning effects of the two groups of students were compared and analyzed, so as to analyze whether the scheduling system in this paper had improved the students' English performance. The results of the learning evaluation comparison are shown in Table 5.

From Table 5, it can be seen that the teaching quality evaluation of using the information-based scheduling system is significantly higher than that of the traditional scheduling method. Thus, the system proposed in this paper can be applied to the informatization teaching of English to achieve the purpose of improving the teaching effect of the course.

### 4. Conclusion

A new improved genetic algorithm is proposed to solve the problems of local convergence, slow convergence speed, and low convergence accuracy of traditional genetic algorithm. Firstly, the algorithm framework adopts a fuzzy adaptive parallel mechanism, which improves the algorithm's ability to overcome the problems of difficult selection of $P_c$ and $P_m$ values and premature convergence. Secondly, the algorithm adopts a distance-adjusted population initialization method and a competition strategy to maintain a rich diversity of populations and set a reasonable search range during the evolutionary process. Finally, a new local search strategy is proposed to enhance the local optimization capability of the algorithm. The experimental results show that compared with the general genetic algorithm, this algorithm can overcome the difficulty of maintaining population diversity and solve the problem of falling into local optimal solution more effectively. In addition, the algorithm has higher efficiency and performance, and the teaching quality of English information education can be improved by using the proposed teaching system. However, due to the complexity of this algorithm, the generation path of optimized initial population still needs to be deeply explored.

### Data Availability

The labeled data set used to support the findings of this study is available from the corresponding author upon request.

### Conflicts of Interest

The authors declare that there are no conflicts of interest.
Acknowledgments

This work was supported by the Hebei Vocational College of Rail Transportation and Hebei Vocational University of Industry and Technology.

References

[1] B. Edo, T. Tadesse, and E. Mulugeta, "Students' and teachers' perceptions and experiences of course scheduling in undergraduate sports sciences program: an Ethiopian case study," Journal of University Teaching and Learning Practice, vol. 16, no. 3, pp. 248–266, 2019.

[2] A. R. Hurlbut, "Online vs. traditional learning in teacher education: a comparison of student progress," American Journal of Distance Education, vol. 32, no. 4, pp. 248–266, 2018.

[3] P. S. Chen, G. Y. H. Chen, S. F. Lien, and W. T. Huang, "Using Scrum and unified modelling language to analyze and design an automatic course scheduling system," Journal of the Chinese Institute of Engineers, vol. 42, no. 6, pp. 534–543, 2019.

[4] T. Thepphakorn, S. Sooncharoen, and P. Pongcharoen, "Particle swarm optimisation variants and its hybridisation ratios for generating cost-effective educational course timetables," SN Computer Science, vol. 2, no. 4, pp. 1–12, 2021.

[5] M. Dener and M. H. Calp, "Solving the exam scheduling problems in central exams with genetic algorithms," Mugla Journal of Science and Technology, vol. 4, no. 1, pp. 102–115, 2018.

[6] S. L. Goh, G. Kendall, and N. R. Sabar, "Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem," Journal of the Operational Research Society, vol. 70, no. 6, pp. 873–888, 2019.

[7] P. Gore, P. Sonawane, and S. Potdar, "Timetable generation using ant colony optimization algorithm," Int. J. Innov. Res. Comput. Commun. Eng, vol. 5, no. 3, pp. 6033–6039, 2017.

[8] S. R. Sutar and R. S. Bichkar, "High school timetabling using tabu search and partial feasibility preserving genetic algorithm," International Journal of Advances in Engineering & Technology, vol. 10, no. 3, pp. 421 pages, 2017.

[9] A. Rezaeipanah, S. S. Matoori, and G. Ahmadi, "A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search," Applied Intelligence, vol. 51, no. 1, pp. 467–492, 2021.

[10] T. Li, Q. Xie, and H. Zhang, "Design of college scheduling algorithm based on improved genetic ant colony hybrid optimization," Security and Communication Networks, vol. 2022, Article ID 2565639, 2022.

[11] S. L. Yadav and A. Sohal, "Comparative study of different selection techniques in genetic algorithm," International Journal of Energy Sector Management, vol. 6, no. 3, pp. 174–180, 2017.

[12] Z. Han, D. Wang, F. Liu, and Z. Zhao, "Multi-AGV path planning with double-path constraints by using an improved genetic algorithm," PLoS One, vol. 12, no. 7, Article ID e0181747, 2017.

[13] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic local search-based differential evolution algorithms for optimization," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 6, pp. 3954–3967, 2021.

[14] D. Jia, C. Zhang, N. Wu, J. Zhou, and Z. Guo, "Autofocus algorithm using optimized Laplace evaluation function and enhanced mountain climbing search algorithm," Multimedia Tools and Applications, vol. 81, no. 7, Article ID 10299, 2022.

[15] J. K. Pattanaik, M. Basu, and D. P. Dash, "Dynamic economic dispatch: a comparative study for differential evolution, particle swarm optimization, evolutionary programming, genetic algorithm, and simulated annealing," Journal of Electrical Systems and Information Technology, vol. 6, no. 1, pp. 1–18, 2019.

[16] T. Mohammadpour, A. M. Bidgoli, R. Enayatifar, and H. H. S. Javadi, "Efficient clustering in collaborative filtering recommender system: hybrid method based on genetic algorithm and gravitational emulation local search algorithm," Genomics, vol. 111, no. 6, pp. 1902–1912, 2019.

[17] Y. Xia and Z. Fu, "Improved tabu search algorithm for the open vehicle routing problem with soft time windows and satisfaction rate," Cluster Computing, vol. 22, no. S4, pp. 8725–8733, 2019.

[18] L. Rodriguez, O. Castillo, M. Garcia, and J. Soria, "A new meta-heuristic optimization algorithm based on a paradigm from physics: string theory," Journal of Intelligent and Fuzzy Systems, vol. 41, no. 1, pp. 1657–1675, 2021.

[19] M. Battistutta, A. Schaerf, and T. Urli, "Feature-based tuning of single-stage simulated annealing for examination timetabling," Annals of Operations Research, vol. 252, no. 2, pp. 239–254, 2017.

[20] H. H. Hoos, F. Neumann, and H. Trautmann, "Automated algorithm selection and configuration (Dagstuhl Seminar 16412)," Dagstuhl Reports, vol. 6, no. 10, pp. 33–74, 2017.