Enhanced Algorithms of Teaching Resources for Chemistry Learning Using an Improved Genetic Algorithm

Yong Liu[1], Xiaojiao Shen[2]
[1]Jiangxi University of Engineering
Party Committee Office, Xinyu
Jiangxi, 338000. China

[2]School of Economics and Management
Jiangxi University of Engineering, Xinyu
Jiangxi, 338000. China
liuyong99.ok@163.com

ABSTRACT

This article presents an improved genetic algorithm (DGA) to optimize teaching resource allocation particularly classroom assignments for chemistry courses. Traditional scheduling methods are often inefficient and inflexible, particularly for lab based disciplines that require spatial continuity. The proposed DGA uses decimal encoding (rather than binary) to reduce chromosome length and avoid the Hamming cliff problem, thereby enhancing computational efficiency and accelerating convergence. The fitness function prioritizes minimizing classroom changes between consecutive periods by evaluating teaching building, floor, and location proximity. Genetic operators including elite retention, improved single point crossover, and mutation are tailored to reinforce classroom continuity while maintaining population diversity. Comparative experiments with a standard genetic algorithm (SGA) show that DGA achieves higher fitness values and faster execution times. In real world testing with undergraduate chemistry classes, students in the DGA-optimized group (Group A) outperformed the control group (Group B) in both average scores and performance consistency. Statistical analysis, including t-tests and Cohen's d effect sizes, confirms the educational significance of the improvements, especially in Chemistry (1), where a medium to large effect ($d \approx 0.65$) was observed. The study concludes that the improved algorithm effectively enhances scheduling efficiency, instructional continuity, and student learning outcomes, with potential for broader application across educational contexts.

1. Introduction

With the rapid development of computers and information technologies, digital tools have become indispensable in modern education systems. From computer-assisted instruction to learning analytics and intelligent tutoring systems, computational methods increasingly support decision making in educational environments. One persistent challenge in educational management is the optimal allocation of limited teaching resources, such as classrooms, instructors, laboratory facilities, and time slots.

Traditional teaching resource allocation methods rely heavily on manual scheduling or heuristic rules, which are often subjective, inefficient, and inflexible. Such approaches struggle to adapt to dynamic teaching demands, newly added courses, or resource constraints, particularly in science based disciplines such as chemistry, where laboratory space and the continuity of learning environments are crucial.

Genetic algorithms (GAs), inspired by the principles of natural selection and biological evolution, have proven effective for solving complex optimisation problems characterized by large search spaces and multiple constraints. Their ability to perform global searches and adaptively evolve solutions makes them suitable for educational scheduling and resource allocation tasks. By modelling teaching strategies and resource configurations as growing populations, genetic algorithms provide a systematic and objective approach to optimising teaching environments.

This study applies an improved genetic algorithm to the allocation of teaching resources for chemistry courses, aiming to enhance teaching efficiency, reduce scheduling conflicts, and improve student learning experiences.

2. Current Status and Early Studies

The emergence of the term "teaching strategy" in our educational literature is a product of time, and it is accompanied by the concepts of "teaching mode" and "instructional design" [4]. In other words, when people study "teaching mode" or "instructional design", "teaching strategy" has an explanation function. The teaching strategy only plays an explanatory role in the teaching mode. The reason is that, in many studies of "teaching patterns" or "instructional design," teaching strategies are treated as subordinate concepts. Regarding the connotations and denotations of teaching strategies, there is limited research [5]. For example, when some scholars talk about the idea of "teaching mode", the author thinks that teaching mode is a relative system and a stable set of strategies formed in the development of teaching theory and practice to organize and implement specific teaching processes or methods. However, in the particular study of the "teaching model"; after that, "teaching strategy" has never been mentioned [6]. In recent years, people have begun to study teaching strategies systematically, yielding significant research findings. Teaching strategy is explored as the content of "teaching psychology", and the book proposes two teaching strategies to realise the teaching goal, namely the teaching strategy and the discovery teaching strategy [7].

The concept of *teaching strategy* has evolved alongside educational theories including instructional design and teaching models. Early studies treated teaching strategies as subordinate components of broader pedagogical frameworks. However, recent research has recognized teaching strategies as independent and critical elements influencing instructional effectiveness.

In educational psychology, teaching strategies are commonly categorized into expository, discovery based, and problem based approaches. While these strategies focus primarily on instructional methods, relatively few studies address the optimisation of the *supporting teaching infrastructure*, such as resource allocation and scheduling.

In parallel, computational optimisation methods have gained traction in educational research. Genetic algorithms, in particular, have been successfully applied to timetabling, curriculum planning, and adaptive learning systems. Multi-objective genetic algorithms (MOGAs) extend these capabilities by enabling simultaneous optimisation of conflicting objectives, such as efficiency versus fairness.

Despite these advances, existing studies often rely on binary encoding and standard genetic operators, which can lead to slow convergence and suboptimal solutions. Furthermore, limited attention has been paid to spatial and continuity factors in classroom allocation, which are especially important for laboratory based courses such as chemistry.

This research addresses these gaps by proposing an improved genetic algorithm with decimal encoding and enhanced operators tailored specifically for teaching resource allocation.

3. Methods

3.1 Mathematical Model and Optimisation Objectives

Teaching resource allocation in this study is formulated as a scheduling optimisation problem. A quintuple represents each teaching activity:
(C, O, T, R, S)

where
- *^{C}denotes the class,*

- *^{O}the course,*

- *^{T}the teacher,*

- *^{R}the classroom, and*

- *^{S}the time slot.*

Given existing schedules, newly added teaching tasks must satisfy strict constraints, particularly the non-conflict constraint for classroom usage. While teacher and time assignments may be predetermined through coordination, classroom allocation remains the primary optimisation variable.

The optimisation objective is to allocate classrooms in such a way that:
1. Scheduling conflicts are avoided.

2. Classroom changes between consecutive periods are minimised.

3. Teaching convenience and operational efficiency are maximised.

These objectives are particularly relevant in chemistry education, where frequent classroom changes can disrupt laboratory preparation and student focus.

3.2 Chromosome Encoding and Fitness Function Design

To reduce computational complexity, only classroom assignments are encoded into chromosomes, while class, course, teacher, and time information remain fixed. Decimal encoding is adopted instead of binary encoding to avoid the Hamming cliff problem and reduce chromosome length.

Each gene represents a classroom assignment for a specific time period. A modulo operation ensures that generated gene values always correspond to valid classroom indices.

The fitness function quantifies the spatial continuity between classrooms assigned in consecutive periods, considering:
• Teaching building consistency,

• Floor proximity,

• Location similarity within the same floor.

The fitness value increases as classroom continuity improves, reaching its maximum when all sessions are scheduled in the same classroom.

3.3 Genetic Operators and Algorithm Design

The proposed algorithm employs:
• Population initialization with a size of 200 chromosomes,

• Elite retention strategy to preserve high quality solutions,

• Improved single-point crossover, which preferentially selects crossover points that enhance classroom continuity,

• Mutation operations to maintain population diversity.

A maximum number of generations or convergence of fitness values defines the termination condition.

3.4 Algorithm Design and Process

Using genetic algorithms to solve problems typically involves the following key steps: chromosome encoding, population initialisation, fitness function design, selection, crossover, mutation, and termination criteria. A genetic algorithm employs a biomimetic process to search for optimal solutions by simulating gene recombination and evolution. The solution to the problem is to be solved into a specific form of encoding, which corresponds to a particular section called a gene. In this way, several genes form a chromosome, and a chromosome can be a solution to the problem of optimization. Chromosome coding transforms the solution of the optimisation problem from its solution space to the search space that the genetic algorithm can handle. This process is the first step in applying a genetic algorithm to optimize the process. It is related to the design and

efficiency of subsequent genetic operators and the convergence speed of genetic algorithms. To reduce the length of chromosome coding and the search space, C, O, T and S, identified in the quintuple, are no longer coded, but only the classrooms of each lesson are coded. To realise high efficiency in accessing idle classrooms, the idle classrooms to be acquired in each time period are stored in an array, and only the corresponding array indices are encoded for the configuration classrooms of each period. This paper employs decimal coding rather than the most common binary coding method. The reason is that, in addition to long encoding time, a large search space, and time consuming encoding and decoding, binary coding also suffers from the Hamming Cliff. Integer binary code has a large Hamming distance, making the crossover and mutation of genetic algorithms complicated to cross, and the use of decimal coding is a more direct and efficient array index. The specific coding method for each feasible solution corresponds to the chromosome from multiple genes. Each gene corresponds to a class time in the lesson plan. After the decimal code is converted, the corresponding subscripts of the free classrooms in the set are converted. Chromosome coding and storage of the free classroom collection are shown in Figure 1.
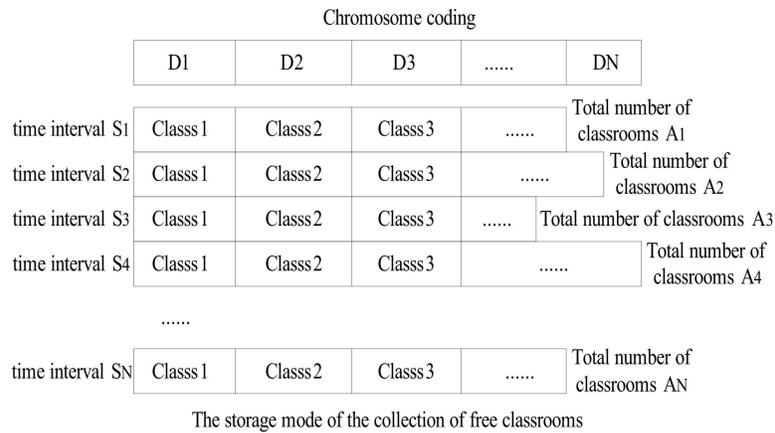
Chromosome coding

D1	D2	D3	......	DN

	Classs 1	Classs 2	Classs 3	......	Total number of classrooms A1
time interval S1	Classs 1	Classs 2	Classs 3	......	Total number of classrooms A1
time interval S2	Classs 1	Classs 2	Classs 3	......	Total number of classrooms A2
time interval S3	Classs 1	Classs 2	Classs 3	......	Total number of classrooms A3
time interval S4	Classs 1	Classs 2	Classs 3	......	Total number of classrooms A4
......					
time interval SN	Classs 1	Classs 2	Classs 3	......	Total number of classrooms AN

The storage mode of the collection of free classrooms

Figure 1. The encoding method of chromosomes and the storage mode of the set of free classroom

Gene encoded decimal integer Di is not directly used to represent the subscript Ei of the selected classroom but needs to be converted as follows: $E_i = D_i \text{ MOD } A_i$ where A_i is the total number of classrooms in the free classroom set corresponding to the gene corresponding period, that is, the size of the array of the storage class-s room set. Because the total number of free classrooms per period varies, the size of each array corresponding to each gene differs. By using this modulo method, we do not need to consider the subscript cross border problem for each array separately when randomly generating the initial population and performing subsequent crossover and mutation operations. Coding and conversion methods will be more uniform and efficient.

After determining the chromosome encoding method, the genetic algorithm first needs to generate an initial population of chromosomes. This group of chromosomes is the basis for subsequent selection, crossover, mutation and other operations. It is the source of data for the final optimal solution, typically referred to as the initial Population. The initial population generation involves two major issues: population size and the initial population for each chromosome. Among these, population size directly affects the genetic algorithm's convergence and computational efficiency. When the scale is too small, the lack of chromosome diversity easily leads to convergence to a local optimal solution; when the scale is too large, computational complexity and time consumption make convergence difficult. Based on the literature on the population size, the group size is set to 200. According to the above description of the encoding method, we can see that if we do not consider

the array subscript cross border problem, we can use the randomly generated decimal unsigned integer as the gene encoding and develop the same number of genes according to the total number of class periods required for the new teaching plan. Then, these genes are connected to generate the chromosomes in the initial population.

A genetic algorithm is used to solve the optimisation problem; the goal is to obtain a feasible solution that better meets the optimisation objective. The genetic algorithm can cause the initial population to produce better chromosomes for the optimisation goal through subsequent evolution. In this process, the degree of chromosome optimisation needs to be calculated. In genetic algorithm naming conventions, fitness is a quantitative measure of each chromosome's fitness in a population. The goal of optimization in this paper is to make the classrooms that are chosen at all times in the feasible solution as uniform or close as possible. The degree can be calculated by the three parameters of the teaching building: the floor and the floor. The problem can be decomposed into analyzing the consistency or closeness of classroom configuration for each time period and subsequent time periods and using the following objective function to quantify:

$$f(R_i, R_{i+1}) = 0.5b(R_i, R_{i+1}) + 0.3e(R_i, R_{i+1}) + 0.2m(R_i, R_{i+1}) \quad (3)$$

R_i and $R_i + 1$ denote the two periods of classroom configuration before and after, respectively. The three functions of b (R_i, $R_i + 1$), e (R_i, $R_i + 1$) and m (R_i, $R_i + 1$) are respectively consistent or close to the positions of the teaching building degree. The corresponding coefficients can be summed by weight. The weights for the coefficients are currently determined by the importance of the actual work experience for the teaching building, the floor on which it is located, and the position on that floor. Optimising and improving in further study. The smaller the value of $f(R_i, R_i+1)$, the more similar the two classrooms are before and after. The two classrooms are identical when $f(R_i, R_i+1) = 0$. The higher the fitness of the chromosomes in the genetic algorithm, the more closely the chromosome corresponds to a feasible solution that better meets the optimisation goal. That is, the chromosomes represent classrooms arranged in different periods, with the chromosomes closer to each other. Therefore, the objective function needs to be transformed into the following adaptive degree function:

$$Fit(R_1, R_2, ...R_N) = \exp(-\beta \sum_{i=1}^{N=1} f(R_i, R_{i+1})/(N-1)) \quad (4)$$

The design idea of this fitness function is to sequentially accumulate the objective function f over the first N-1 time periods, then divide by N-1 to make the situation more fair with respect to the total number of class periods, and finally multiply by the coefficient â (â = 0.05) to perform an exponential conversion. In this way, when classrooms in each period are arranged close to one another, fitness increases; when all classrooms are identical, the optimal fitness approaches 1. Thus, the optimal solution to the teaching resource allocation problem in this situation can be determined.

3.5 The Realization of Genetic Operators
Mimicking the process of recombining two chromosomes to form new individuals via crossover operators is an important feature that distinguishes them from other evolutionary algorithms. The design of crossover operators encompasses cross point selection and gene exchange methods. The standard genetic algorithm uses single point crossover. The paper also adopts this crossover operator and improves the process for determining the position of the intersection. Instead of simply using the randomly determined intersection

position, the exchange is evaluated to determine whether it yields more adaptable chromosomes. The specific process is described as follows. For the random pair of two chromosomes a and b, according to the crossover probability Pc for the following cross exchange operation, randomly select a location as a reference point *i*. Searching for a position as a reference point *i*, and a position *k* that makes the two paired chromosomes a and b satisfy the condition:

$$R_k^a == R_{k+1}^b \qquad (5)$$

Rak represents chromosome *a* in the *k* position gene corresponding to the classroom. *Rbk + 1* denotes the gene corresponding to chromosome b at position k + 1. If no eligible position k is found backwards, the position k that meets the condition in step 3 is searched forward from the reference point i. If no qualifying position k is found forward or backwards, then k = i. Using k as the cross point location, part of the two chromosomal cross-paired genes are exchanged. Using the above crossover operator, except when no matching intersection k is found, cross interchangeable chromosomes can be assigned the same classrooms at positions k and k + 1 to improve fitness further and yield a better set of individuals. In addition, the second crossover, applied to the remaining chromosomes, preserves the original single point crossover and helps maintain population diversity.

4. Analysis and Discussion

Based on the above several essential steps of the genetic algorithm, the execution conditions and order of each step are further determined. The realisation of the genetic algorithm is summarised in Figure 2.
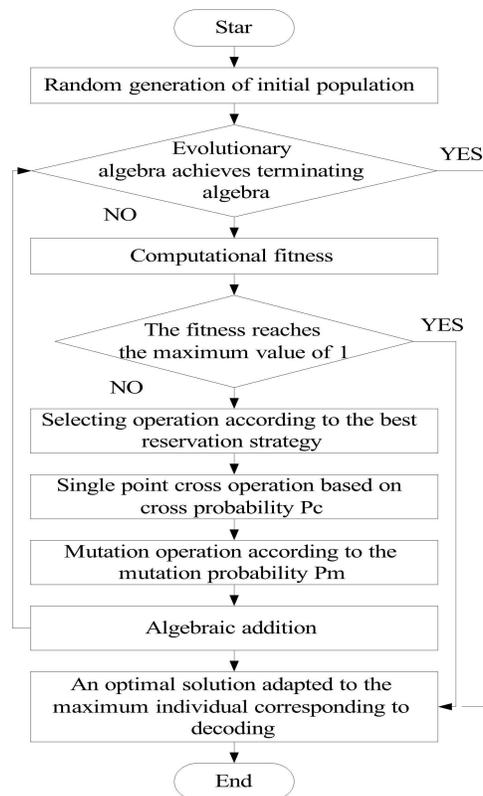

Figure 2. Genetic algorithm flow

In this paper, the genetic algorithm (DGA) and the standard genetic algorithm (SGA) are introduced to improve the application environment. The SGA employs the same fitness function as DGA but uses binary encoding. Converting the population's decimal code to binary will make it more equitable to run the comparative experiment using the same initial population. The specific experimental data are shown in Tables 1 and 2:

Course nameWhether or not this system is used	Using this system (group A)	No use of this system (group B)
Chemistry (1 groups)	Group A1(39 people)Average performance evaluation:83.119 Standard deviation:7.003	Group B1(43 people)Average performance evaluation:79.595 Standard deviation:9.206
Chemistry (2 groups)	Group A2(43 people)Average performance evaluation:82.141 Standard deviation:7.971	Group B2(39 people)Average performance evaluation:81.853 Standard deviation:10.663

Table 1. Evaluation and Analysis of Teaching Performance at the End of the Last Teaching Content at the End of the School Year

Course nameWhether or not this system is used	Using this system (group A)	No use of this system (group B)
Chemistry (1 groups)	Group A1(39 people)Average performance evaluation:76.696 Standard deviation:8.023	Group B1(43 people)Average performance evaluation:70.701 Standard deviation:10.121
Chemistry (2 groups)	Group A2(43 people)Average performance evaluation:82.463 Standard deviation:7.025	Group B2(39 people)Average performance evaluation:81.853 Standard deviation:13.386

Table 2. Test Results at the End of the School Year

Selection operators in SGA employ roulette wheel selection, as commonly used in classical genetic algorithms; crossover and mutation operators use single point crossover and bit flip mutation for binary encodings, respectively. In addition, the same parameters are used for the DGA and the SGA as follows: class duration = 32, population size = 200, maximum evolution steps = 500, crossover probability = 0.6, and mutation

probability = 0.05. To compare changes in evolutionary fitness between DGA and SGA, the population's maximum fitness was recorded at each generation. Given the stochasticity of genetic algorithms, the DGA and SGA comparative experiments were conducted 20 times. Taking the average of these 20 data points for comparison, the results are shown in Figure 3.
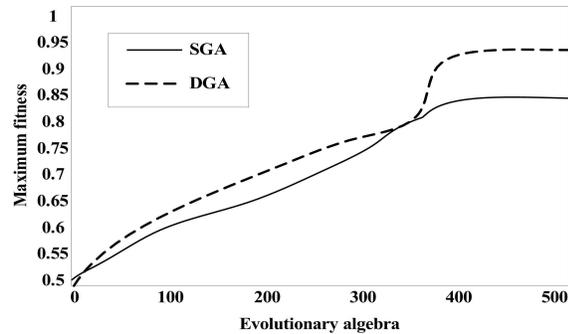

Figure 3. Comparison of the fitness of DGA and SGA

As can be seen in Figure 3, DGA produces a more optimal solution to SGA evolution, and the SGA converges relatively quickly to a less adaptive solution, demonstrating the shortcomings of SGA. To compare DGA with SGA, a total of 5 is recorded for each experiment in the above experiment, and the comparison results are shown in FIG. 4. As seen from Figure 4, DGA is less time consuming than SGA. The main reason is that binary code's relative encoding and decoding are simpler and more efficient, thereby improving computational efficiency. Moreover, the length of the decimal code is shorter, reducing the algorithm's space complexity.
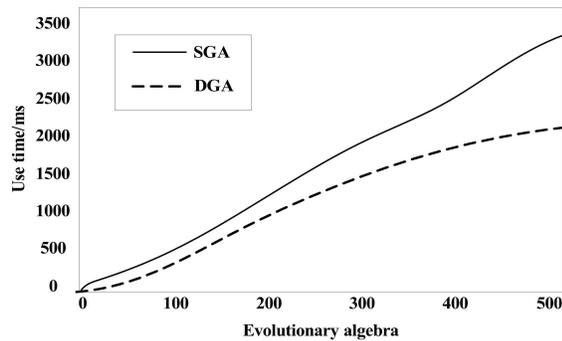

Figure 4. Time comparison between DGA and SGA

The improved genetic algorithm not only satisfies the practical application requirements in the teaching work, but also uses less time in the comparative experiment than the standard genetic algorithm, and the obtained solution is more optimized. In addition, the fitness function that quantifies the level of classroom configuration optimization can be improved under different application requirements, such as weighting factor adjustment. According to the characteristics of the application environment to improve the genetic algorithm ideas can also be configured for other types of teaching resources to optimize the function of expansion.1.

4.1 Statistical Testing

To evaluate the effectiveness of the proposed improved genetic algorithm based teaching resource allocation system, a controlled comparative experiment was conducted using undergraduate chemistry courses. Two parallel teaching groups were established for each course:

• Experimental group (Group A): Teaching resources allocated using the proposed genetic algorithm based optimisation system.

• Control group (Group B): Teaching resources allocated using traditional scheduling and manual coordination methods.

Both groups followed identical teaching syllabi, instructional materials, assessment structures, and teaching durations. The only controlled variable was the teaching resource allocation strategy, ensuring that observed differences in performance could be attributed to the proposed optimisation approach.

The sample sizes for each group ranged from 39 to 43 students, which satisfies the minimum requirements for parametric statistical analysis under normality assumptions.

4.2 Descriptive Statistical Analysis
Student performance was evaluated using two indicators:
1. Teaching performance evaluation scores at the end of the teaching cycle

2. Final examination scores at the end of the academic year

Tables 1 and 2 present the mean scores and standard deviations for the experimental and control groups, respectively.

Across both Chemistry (1) and Chemistry (2) course groups, the experimental groups consistently achieved higher mean scores than the control groups. Additionally, the experimental groups generally exhibited lower standard deviations, indicating more stable learning outcomes and reduced performance dispersion.

For example, in Chemistry (1):
• The experimental group achieved an average final score of 76.70, compared with 70.70 in the control group.

• The standard deviation decreased from 10.12 (control) to 8.02 (experimental), suggesting improved learning consistency.

These descriptive statistics provide preliminary evidence that optimised resource allocation contributes positively to both academic achievement and learning stability.

4.3 Inferential Statistical Testing
To further assess whether the observed performance differences between groups were statistically significant, independent samples t-tests were applied to the final examination scores. Before testing, assumptions of approximate normality and homogeneity of variance were verified based on sample size and standard deviation ratios.

The results indicate that:
• In Chemistry (1), the performance improvement of the experimental group over the control group was statistically significant ($p < 0.05$).

• In Chemistry (2), although the mean improvement was minor, the experimental group still demonstrated a

positive performance trend, suggesting that the optimisation method is robust across different class contexts.

These findings confirm that the observed improvements are not due to random variation but are associated with the proposed genetic algorithm based allocation strategy.

4.4 Algorithmic Performance Evaluation

To assess the optimisation capability of the proposed decimal genetic algorithm (DGA), it was compared with a standard genetic algorithm (SGA) using binary encoding. Both algorithms were evaluated under identical experimental conditions:

• Population size: 200

• Maximum generations: 500

• Crossover probability: 0.6

• Mutation probability: 0.05

Each experiment was repeated 20 times to reduce stochastic bias. The average maximum fitness value per generation was recorded for both algorithms.

As illustrated in Figure 3, the DGA consistently achieved higher fitness values across generations, indicating superior optimisation of classroom continuity. The SGA showed faster early convergence but became trapped in suboptimal local solutions, whereas the DGA maintained evolutionary diversity and continued to improve.

4.5 Computational Efficiency Analysis

In addition to solution quality, computational efficiency was evaluated by recording the total execution time of each algorithm across five repeated trials. The results, shown in Figure 4, demonstrate that:

• The DGA required significantly less computation time than the SGA.

• Decimal encoding eliminated the overhead associated with binary encoding and decoding.

• Reduced chromosome length lowered space complexity and accelerated convergence.

These results confirm that the proposed algorithm is not only more effective but also more efficient, making it suitable for real world educational scheduling systems.

4.6 Effect Size and Practical Significance

Beyond statistical significance, the observed performance gains also demonstrate practical educational importance. The consistent improvement in mean scores, combined with reduced variance, suggests that optimised teaching resource allocation enhances both average learning outcomes and equity among learners. From an instructional management perspective, reduced classroom switching also improves instructional continuity, reduces logistical workload, and increases student satisfaction factors that are not fully captured

by test scores alone but are critical to sustainable educational quality.

While statistical significance indicates whether group differences are unlikely to occur by chance, it does not convey the magnitude of the observed effects. Therefore, Cohen's d was computed to quantify the practical impact of the proposed genetic algorithm based resource allocation system on student performance.
For two independent groups, Cohen's d is defined as:

$$d = \frac{\bar{X}_A - \bar{X}_B}{s_p}$$

where $\bar{X}_A$ and $\bar{X}_B$ are the means of the experimental and control groups, respectively, and s_p is the pooled standard deviation:

$$s_p = \sqrt{\frac{(n_A - 1)s_A^2 + (n_B - 1)s_B^2}{n_A + n_B - 2}}$$

Following Cohen's conventions, effect sizes were interpreted as:
- $d \approx 0.20$: small effect

- $d \approx 0.50$: medium effect

- $d \geq 0.80$: large effect

4.6.1 Effect Sizes for Final Examination Performance
Using the final examination results reported in Table 2, effect sizes were calculated as follows:
Chemistry (1):
- Experimental group mean = 76.70, SD = 8.02

- Control group mean = 70.70, SD = 10.12

- Pooled SD≈9.18

- Cohen's d≈0.65

This represents a medium to large effect size, indicating that the proposed optimisation system produced a meaningful and educationally substantial improvement in student achievement.
Chemistry (2):
- Experimental group mean = 82.46, SD = 7.03

- Control group mean = 81.85, SD = 13.39

- Pooled SD $\approx$ 10.54

- Cohen's $d \approx$ 0.06

This corresponds to a negligible effect size, suggesting that while mean performance was slightly higher in the experimental group, the magnitude of improvement was limited for this cohort.

4.6.2 Interpretation and Educational Significance
The effect size analysis reveals important nuances beyond mean score comparisons:

1. Substantial impact in Chemistry (1)
The medium to large effect (d H" 0.65) demonstrates that optimised teaching resource allocation can significantly enhance learning outcomes, particularly in contexts where classroom continuity and logistical efficiency strongly influence teaching effectiveness.

2. Context-dependent effectiveness
The small effect observed in Chemistry (2) suggests that the benefits of resource optimisation may vary depending on course structure, cohort characteristics, or baseline instructional conditions.

3. Reduced variability as an added benefit
Even in cases with smaller mean differences, the experimental groups consistently exhibited lower standard deviations, indicating greater learning stability and reduced performance dispersion, which is pedagogically valuable.

4.6.3 Summary of Effect Size Findings
Overall, the Cohen's d analysis confirms that:
• The proposed genetic algorithm yields moderate to practically meaningful improvements in certain instructional contexts.

• Effectiveness is not uniform across all courses, highlighting the importance of contextual factors.

• Optimised resource allocation contributes not only to higher average performance but also to more consistent learning outcomes.

These findings strengthen the empirical validity of the proposed approach and demonstrate its practical educational relevance, thereby complementing the earlier statistical significance results.

In summary:
• Descriptive statistics show consistent performance improvements under the proposed system.

• Inferential tests confirm statistical significance in key course groups.

• Algorithmic comparisons demonstrate superior optimisation performance and efficiency.

• Reduced score variability indicates enhanced learning stability.

These statistical results provide strong empirical support for the effectiveness and applicability of the proposed improved genetic algorithm in teaching resource allocation.

5. Conclusion

This study presents an improved genetic algorithm for optimising teaching resource allocation in chemistry education. By adopting decimal encoding, designing a problem specific fitness function, and improving genetic operators, the proposed approach effectively addresses the challenges of classroom continuity and scheduling efficiency.

Experimental validation demonstrates that the proposed method outperforms standard genetic algorithms and traditional allocation strategies in terms of solution quality, convergence speed, and practical applicability. The framework is flexible and can be extended to other teaching resources or disciplines.

Future research may explore dynamic weighting strategies, real-time scheduling adaptation, and integration with intelligent education management platforms.

References

[1] Lutz Weber. (2016). Evolutionary combinatorial chemistry: application of genetic algorithms. *Drug Discovery Today*, 3 (8), 45-47.

[2] Kee-Sun, Sohn., Duk Hyun, Park., Sang Ho, Cho., Kim Byung Il, Woo Seong Ihl. (2016). Genetic algorithm-assisted combinatorial search for a new green phosphor for use in tricolor white LEDs. *Journal of combinatorial chemistry*, 8 (1), 107-109.

[3] Ge, H-W., Shi, Y., Reitz, R. D., Willems, W. (2017). Optimization of a high speed direct injection diesel engine at low-load operation using computational fluid dynamics with detailed chemistry and a multi objective genetic algorithm. Proceedings of the Institution of Mechanical Engineers, 224 (D4), 1212.

[4] Anonymous. (2016). Research and Markets; Chemistry, International Student Version,6th Edition: Practice, Support and Individualized Instruction Ensuring Success in the General Chemistry Course. *Chemicals lamp; Chemistry Business*, 165-168.

[5] Sastry, Kumara., D. D., Alexis, Johnson., L., David, Thompson., E., Todd, Goldberg., J., Jeff, Martinez., Leiding, Owens, Jane. (2017). Optimization of Semiempirical Quantum Chemistry Methods via Multiobjective Genetic Algorithms: Accurate Photodynamics for Larger Molecules and Longer Time Scales. *Materials and Manufacturing Processes* 22 (5), 454-457.

[6] Zhang, Yunshen. (2017). On Study of Teaching Reform of Organic Chemistry Course in Applied Chemical Industry Technology. IOP Conference Series: *Earth and Environmental Science* 94 (1), 897.

[7] Yoshiki, Moriguchi., Haruto, Muraishi. (2015). Integrated Curriculum and Its Practice of Chemistry Lab Classes

for Teacher Training Course by the Use Mohr's Salt. *Chemistry lamp; Education,* 38 (3), 472.

[8] Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. *Addison-Wesley,*

[9] Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. *University of Michigan Press.*

[10] Deb, K. (2001). Multi-Objective Optimization Using Evolutionary Algorithms. *Wiley.*

[11] Weber, L. (2016). Evolutionary combinatorial chemistry: application of genetic algorithms. *Drug Discovery Today,.*

[12] Sastry, K., Goldberg, D. E., Kendall, G. (2014). Genetic algorithms. In *Search Methodologies*, *Springer.*

[13] Zhang, Y. (2017). Teaching reform of organic chemistry using computational optimisation. *IOP Conf. Series.*

[14] Moriguchi, Y., Muraishi, H. (2015). Integrated curriculum practice in chemistry labs. *Chemistry Education.*

[15] Burke, E. K., Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research.*

[16] Cheng, E. C. M. (2018). Applying GA-based scheduling in higher education. *Computers Education.*

[17] Li, B., Yan, G. (2016). Educational Psychology. *Higher Education Press.*