Reform based on classification teaching of an *Algorithm Design and Analysis* course

Zhenzhang Liu† & Feng Wang‡

Hubei University of Automotive Technology, Shiyan, Hubei, People's Republic of China†
Nanjing University of Science and Technology, Nanjing, Jiangsu, People's Republic of China‡

ABSTRACT: *Algorithm Design and Analysis* is one of the core courses for an IT-related major. The course involves many difficult concepts, complex mathematical formulae and theoretical deductions. Traditional teaching does not take into account students' individual learning abilities and career directions and, hence, an urgent need to improve the situation. The work reported in this article explored the use of classification teaching to improve teaching and learning. Classification teaching involves classifying students based on various criteria and streaming them into different classes according to their classifications, where they may be taught differently. Classification teaching was adopted for teaching *Algorithm Design and Analysis*. This required changes to training targets, teaching methods and assessments. Statistical data show that the scores for each component of the assessment and the total score significantly improved after the adoption of classification teaching. Thus, the classification teaching mode can play an important role in improving teaching.

INTRODUCTION

Algorithm Design and Analysis is an important specialised foundation course for IT-related majors. The course introduces some basic algorithms and analyses their complexity. Algorithms covered include the partition, greedy, dynamic programming, backtracking, and branch and bound algorithms [1]. By learning and practising the use of algorithms, students develop the ability to analyse complex problems and to use algorithms in their solution.

Algorithm design and analysis is a course that combines theory with practice. Students are not only required to possess a solid foundation of higher mathematics and data structures, but also a strong ability for abstract thinking and computer programming. The traditional teaching of algorithm design and analysis pays most attention to mathematical reasoning, proving correctness and the complexity analysis of algorithms [2-3].

This course features many concepts; a wide range of complex mathematical formulae and much theoretical deduction. Because of the expansion of enrolment in higher education in China, many students are weak in theoretical analysis and mathematics.

Foundation courses, such as data structures and program design, are not well mastered by such students and this causes many difficulties with their learning. They lose interest in studying, and this makes the teaching unsatisfactory. This implies that, faced with such students, colleges and universities should change their teaching methods and course organisation to fully realise the students' learning potential and to improve teaching.

In classification teaching, students are classified into different categories according to their foundation knowledge and ability. Different categories of students can be set different training targets and teaching plans. The teaching may employ different teaching methods and assessments. At present, classification teaching is widely used in institutions of higher education for courses such as, college English and mathematics, and has produced remarkably good results [4]. Given the discussion above, it is appropriate to investigate the reform of the algorithm design and analysis course based on classification teaching.

REVIEW OF CLASSIFICATION TEACHING

The Need for Classification Teaching of Algorithm Design and Analysis

Algorithm design and analysis is taken by students majoring in computer science and technology, software engineering, information and computer science. The teaching goal of the course is to develop an in-depth understanding of basic algorithms, and to develop the ability to analyse complex algorithmic problems. Teaching content includes common algorithm design, analysis and programming. It lays a good foundation for students who intend to work in software system development.

The computer school produces a unified syllabus and teaching plan for the course. Students are divided into different classes that share the same teaching plan and teaching content; differing only in class times and instructors. Students from different majors have different knowledge and mathematical foundations. So, course requirements will differ between majors. Also, students' abilities and career directions differ. Traditional teaching does not meet students' development needs and does not take into account differences in students' abilities.

Basis for the Classification of Students

Considering the above arguments, classification teaching was adopted for lgorithm design and analysis. This involves changes to the training targets, teaching methods and assessments. Two teaching classes were established to allow for student differences in knowledge, ability and career orientation. One class was a conventional class and the other was a practical class.

Students majoring in information and computer science usually have good mathematical and logical reasoning ability. After graduation, they will probably continue further study or find jobs related to research or complex algorithm design. By comparison, most software engineering students after graduation will work in software companies, enterprises or government departments. They will likely work on software design and development, computer application development or system maintenance [5].

Generally speaking, students majoring in information and computer science attend the conventional teaching class, and the software engineering students attend the practical teaching class. However, students' personal wishes will be taken into consideration in that software engineering students are free to select the conventional class.

CLASSIFICATION TEACHING OF ALGORITHM DESIGN AND ANALYSIS

Choosing Appropriate Teaching Material

In order to achieve the classification teaching targets, the teacher must choose appropriate teaching materials. The book, *Algorithm Design and Analysis*, edited by Professor Wang Xiaodong is recommended for the conventional class. This book is highly perceptive and offers various mathematical methods. It is rigorous in reasoning and analysis and the concise algorithm code is easy to understand.

For the practical class, *Algorithm Design and Analysis - C++ Language Description*, edited by Professor Chen Huilan is recommended. The simple descriptions of algorithms and analysis make the book easy for students to understand. Also, complete C++ source code for each algorithm helps students reproduce and verify algorithms.

Selection of Teaching Content

In the conventional teaching class, teachers concentrate on the kinds of algorithms, mathematical models of algorithms, classical algorithms, correctness proofs and complexity analysis. The data structures and program code to implement an algorithm are dealt with by simple explanation. Students can validate the algorithms by experiment after class. In order to raise students' interest, new trends in computing are introduced in class, along with recent developments in the field of algorithm design and analysis.

Teachers can introduce random and approximation algorithms by using simple board games. This would lay a good foundation for a subsequent specialised courses in artificial intelligence. Besides, the teacher can recommend excellent research papers that have the solutions to problems in different fields using algorithms. Hence, students will come to know the extensive use made of these classical algorithms in various fields.

Students in the practical class are mainly computer technology and software engineering majors. Their mathematical foundation and logical thinking abilities are generally low, and they usually have difficulty in understanding complex mathematical deductions and proofs. Thus, the main teaching task in the practical class is to teach basic algorithms and their application to different problems, as well as the simple analysis of the complexity of algorithms. The teacher should explain the implementation of an algorithm in detail, including the data structures and computer code. On the premise that understanding the implementation and application of algorithms is adequate, cumbersome mathematical deductions and proofs are avoided.

Rational Use of Teaching Content

Algorithms discussed in textbooks have wide applicability. Teachers should use examples from the textbooks explained in the forms of games and puzzles, as well as examples of the application of algorithms, to help students better understand. Sometimes one problem can be solved by a number of algorithms, for example, the *knapsack problem* (*combinatorial optimisation*). Students come to better understand algorithms in general by studying different algorithms that solve the same problem.

The International Collegiate Programming Contest (ICPC) organised by the Association for Computing Machinery (ACM) is an international competition for college students, in innovation, team work, programming, problem analysis and problem-solving [6]. It is the most influential computer programming contest and has become popular at Chinese colleges and universities. The questions set by ACM-ICPC store are well-designed, interesting and involve algorithm design. They emphasise the optimisation of algorithms. They provide good case studies for the algorithm design and analysis course.

MEASURES TO IMPROVE TEACHING

Appropriate Use of Multimedia Courseware

Multimedia technology plays an important role in classroom teaching. Compared with traditional blackboard teaching, it allows teachers to cover more material in a given period. However, multimedia courseware cannot be used for teaching all the content of the algorithm design and analysis course, especially in conventional class teaching. In conventional class teaching, there are many mathematical derivations involving algorithm modelling, proof and complexity analysis. For this, blackboard teaching is still essential. When using a blackboard, the teacher can alter the teaching pace, e.g. slowing down if more consideration is required. The appropriate teaching method, i.e. one that is traditionally blackboard- or multimedia-based, should be chosen according to the class and teaching content.

Demonstrating Algorithms Using Animation

An algorithm is an abstract concept often involving complex data structure and requiring a strong logical reasoning ability to understand. Students with weak logical reasoning ability find difficulty in thoroughly understanding algorithms. By using animation software, the teacher can display the operation and behaviour of an algorithm using graphics, animation and sound [7]. This vivid illumination of the theory and examples creates a huge impression on students and helps them to understand.

CLASSIFICATION TEACHING IN PRACTICE

The Design of Experiments

Computer experiments are an indispensable part of the algorithm design and analysis course. The course syllabus specifies five experiments based on five types of algorithm, viz. the partition, greedy, dynamic programming, backtracking, and branch and bound algorithms. These experiments help students master the theory of algorithms, and solve practical problems through the flexible application of algorithms [8].

The experiments are graded in the degree of difficulty. For each algorithm, there are elementary and integrated exercises. The elementary exercises cover relatively simple topics, which focus on verification of the working of an algorithm. Integrated exercises include advanced topics that are relatively complicated and require designing a solution. Students in the practical class can complete the elementary exercises with the help of the teacher. Students in the conventional class, with their better ability and interest in algorithm design can, under the guidance of a teacher, complete the integrated exercises, which foster their capacity for complex algorithm design.

Adopting an Structured Methodology for Experiments

One of the teaching targets of the algorithm design and analysis course is to develop a student’s ability to solve actual problems using algorithms. This lays a good foundation for students who will work on software system development, which requires a structured approach to problem-solving.

Requirements for the experiments should make students develop solutions by being guided to solve the problems in a structured, methodical way. Before doing an experiment on the computer, students must conduct a requirements analysis, algorithm design and data structure design. Only then do they write the algorithm code and design test data. Finally, they debug and prove the correctness of the algorithm on the computer. After the experiment, students complete a report on the experiment in the form of a software engineering report. Students of the conventional class also carry out a rigorous complexity analysis of the algorithm. By doing so, they develop good study habits and foster the ability of rigorous software development.

ASSESSMENT OF CLASSIFICATION TEACHING

The traditional method of assessment of the algorithm design and analysis course has two components, viz. a general assessment accounting for 30% of the total mark and a final assessment accounting for 70%. General assessment mark of 30% of the total includes the experiments (20%), and attendance and class discussion (10%). The final assessment is an examination on theory. As a result, there is more emphasis on theory than practice, which is not conducive to developing a student's practical ability in applying knowledge.

A different approach to assessment has been taken for classification teaching of the conventional and practical classes. The redesigned assessment method is shown in Table 1.

Table 1: The item and score of classification assessment.

Class	Experiment	Theory examination	Attendance and class discussion	Computer-based examination
Conventional	20	70	10	0
Practical	20	50	10	20

The conventional class assessment remains the same as for the traditional assessment method, i.e. the theory examination accounts for 70% of the total score, attendance and class discussion accounts for 10%, and experiments account for 20%. Hence, the conventional class retains an emphasis on theory. On the other hand, the practical class theory examination accounts for 50% of the total score and the difficulty is reduced compared with the conventional class examination. The other assessments for the practical class include a computer-based examination accounting for 20% of the total, attendance and class discussion accounting for 10% and the experiments accounting for 20%.

The examination on theory focuses on the assessment of algorithms. The computer-based examination awards points automatically using the ACM-ICPC on-line judging system, which focuses on programming ability and embodies the practical class teaching target of using algorithms to solve practical problems.

Emphasis differs in the theory examination between conventional and practical classes. The assessment components are shown in Table 2. With the conventional class there is more emphasis on algorithm analysis while, for the practical class, there is more emphasis on understanding algorithms and programming by which to solve actual problems. This assessment programme based on the teaching targets for the two classes is conducive in guiding students toward developing their respective professional competences.

Table 2: Assessment components of the theory examination.

Class	Algorithm			
	Thought	Design	Analysis	Understanding
Conventional	30	40	20	10
Practical	20	40	10	30

ANALYSIS OF THE EFFECT OF CLASSIFICATION TEACHING

The average scores for the algorithm design and analysis course in 2012, before classification teaching, were compared with the results in 2013, after classification teaching was introduced. The average scores cover the theory examination, a combined score for the experiments and computer-based examination, as well as attendance and class discussion. Note that these components have different maximum marks. The results are shown in Table 3, where the number after the oblique stroke is the maximum mark for that component.

Table 3: Effect of classification teaching.

	Theory examination	Experiments and computer-based examination*	Attendance and class discussion	Total score (100)
Before reform (2012)	46.6/70	13.9/20	8.3/10	68.7
Conventional class (2013)	53.5/70	14.4/20	9.3/10	77.1
Practical class (2013)	36.9/50	30.2/40	8.8/10	75.9

*Only *Practical class (2013)* does the *computer-based examination*

As can be seen from Table 3, the average total score has risen, from about 69 before the introduction of classification teaching to 77 for the conventional class, after introducing classification teaching, and 75 for the practical class. This shows that classification teaching has improved teaching overall.

A bar chart comparing the results before and after the introduction of classification teaching, is shown in Figure 1. In the bar chart the component scores have been converted to percentages to provide a better base from which to compare component improvements.

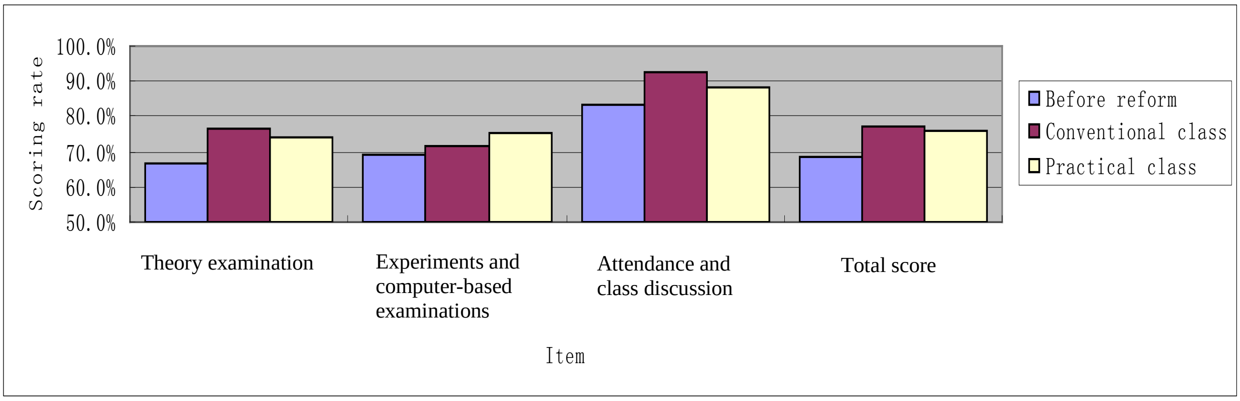

Figure 1: Comparison of the results before and after the introduction of classification teaching.

As can be seen from Figure 1, in the *Theory examination* component, students' average score has increased, from 66.5% before the teaching reforms to 76.4% and 73.8% for the conventional and practical classes respectively after the reforms; rises of 9.9% and 7.3%, respectively. For *Experiments and computer-based examination*, students' average score before teaching reform was 69.3%, increasing to 71.8% and 75.5% for the conventional and practical classes, respectively, after the teaching reforms; rises of 2.5% and 6.2%, respectively. For *Attendance and class discussion*, students' average score before teaching reform was 83%, increasing to 93% and 88% for the conventional and practical classes respectively after the teaching reforms; rises of 10% and 5%, respectively.

Therefore, the adoption of classification teaching has resulted in an improvement in each assessment component and in the total score for both conventional and practical classes. Also, the *Theory examination* score is higher for the conventional class than that for the practical class. The *Experiments and computer-based examination* score is higher for the practical class than that for the conventional. These latter two results conform to the main targets of the classified teaching reform.

CONCLUSIONS

Algorithm design and analysis is an important specialised foundation course for IT-related majors. Students of various majors have differing mathematical foundations, learning abilities and career directions. Therefore, classification teaching is an appropriate mode to adopt for the teaching of the course. This reform introduces new training targets, teaching methods and assessments. Analysis of the effect of the teaching reform shows that students' average score for each assessment component was higher after the adoption of classification teaching. This shows clearly the importance of classification teaching in improving teaching quality and learning outcomes. Also students' capacity for development in their chosen direction has improved after the adoption of classification teaching.

ACKNOWLEDGEMENTS

This research was supported by the Research and Development Project of Science and Technology of ShiYan (14K26) and Young University Teachers of Hubei Province into the Enterprise Action Plan (XD2014195).

REFERENCES

1. Chen, H.N., *Algorithm Design and Analysis - C++ Language Description*. Beijing: Publishing House of Electronics Industry, 64-205 (2012).
2. Sun, J., Wu, X.J. and Li, Y.Y., Research of computer algorithm design and analysis teaching method. *Science and Technol. Infor.*, **23**, 173-174 (2013).
3. Qin, D.H. and Chen, Z.Y., Research on algorithm design and analysis course teaching. *Computer Educ.*, **11**, 98-101 (2013)
4. Zhao, D.J. and Li, L., The current situation and enlightenment of graded teaching in university mathematics curriculum. *University Educ.*, **1**, 100-102 (2013)
5. Liu, W.Q., Discussion on algorithm design and analysis course teaching in software engineering. *J. of Luoyan Normal University*, 31, **8**, 101-103 (2012).
6. Li, H., Zhao, J.P., Li, Q. and Wu, Y., Curriculum reform of algorithm design and analysis based on ACM-ICPC. *Computer Educ.*, **7**, 88-91 (2013).
7. Zhang, W.S., Zhou, Q.Y. and Zhou, X.C., Research and application of algorithm animation. *Computer Applications and Software*, 25, **10**, 41-43 (2008).
8. Chen, X.J., Xu, D.F. and Fang, F.M., A multi-level practice teaching system for courses of algorithm and program design. *Research and Exploration in Laboratory*, 31, **8**, 319-322 (2012).