An Integrated Application of Educational Technologies for "Algorithm Design and Analysis" Course

Xiong Luo, Zhaoshun Wang, Junwei Zhou, and Yi Chen
School of Computer and Communication Engineering
University of Science and Technology Beijing (USTB)
30 Xueyuan Road, Haidian District, Beijing 100083, China
e-mail: xluo@ustb.edu.cn

Abstract—**In this paper, according to the teaching requirement of "Algorithm Design and Analysis" course, we discuss the educational technologies for this course based on our teaching practice. Starting from the analysis of the general deficiencies under current teaching mode, we give some improvements for it. To address the problems, we propose an integrated application method of educational technologies for "Algorithm Design and Analysis" course by combining problem-base-based "filling teaching", multi-level heuristic teaching, curriculum-design-based comprehensive training, and so on. The teaching practice in our course shows the effectiveness of those integrated educational technologies. The satisfactory teaching effect is achieved.**

Keywords-algorithm design and analysis; integrated application; heuristic teaching

I. Introduction

"Algorithm Design and Analysis" is an important course for professional studying related to computer science and technology and other information technology subject. In this course, we mainly instructs the way to solve actual problems which often encountered in computer science and engineering fields . The students would be asked to learn and understand the basic principle, method, and technology in the design and analysis of various algorithms. Through "Algorithm Design and Analysis" course, it is to cultivate the ability of making correct analysis of complex algorithm. Meanwhile, it is also to train their logical thinking ability and imagination. So it can help the students to understand the development of the theory of algorithms. In addition, we also encourage the students to use the knowledge of algorithm to solve practical problems in their disciplines. In so doing, it can cultivate the students' ability of independent scientific research and engineering practice applications. To learn this course, there are two prerequisite course, i.e., "Data Structure" and "Computer Program Design".

After learning this course, the students should be capable to design and analyze an algorithm by using some classic methods, e.g. recursive and divide-and-conquer, greedy method, dynamic programming, backtracking, branch and bound, and so on.

Because this course mainly relates to the theory of algorithm, it is so abstract that it is rather difficult to teach and understand. Accordingly it is more difficult for students to apply those design strategies of algorithm to solve practical problems. As for these difficulties, the students' learning enthusiasm is not high. At present, for this course, in order to overcome the above problems, there has been already some discussions on teaching methods, and they have played a certain positive role in studying [1-4].

In this paper, by combining with our actual teaching experience, we summarize some educational technologies. Through an integrated application of those technologies, teaching effect is greatly improved, and we also get a better evaluation from the students.

II. Statement of Current Teaching Mode

As mentioned above, there are some common problems under current teaching mode of "Algorithm Design and Analysis" course, which can be summarized as follows.

1) Incomplete basic theoretical knowledge

In general, during the process of designing an algorithm for a problem, we firstly need to analyze the characteristics of problem and abstract the logic structure of the corresponding data. According to the logic structure, we can select the physical implementation structure. Then we establish the corresponding mathematical model to get the algorithm we need. During the whole process, while abstracting and establishing the relationship between the logical structure and physical structure, it needs the skillful use of knowledge of "Data Structure" course. And while establishing the mathematical model of processed problem, it needs some basis of mathematical reasoning analysis. It is a comprehensive application process of theory. But the basic theoretical knowledge of some students are not strong enough, in the early stage of algorithm design, they will encounter a "bottleneck". This is a noticeable weak link of learning process in the "Algorithm Design and Analysis" course. Thus it has affected the enthusiasm of students.

2) A gap between the theory and the practice

In our teaching , some students can understand the theoretical knowledge of the algorithm, they also have a clear understanding of basic principles of algorithm design strategies, e.g. dynamic programming, the branch and bound, but while designing algorithms for some practical applications, they do not have a idea of how to do. It is another weakness of students in designing practical algorithms. Because the students are lack of depth perceptual understanding for algorithm, moreover, they do not have enough programming process, they can not have a global understanding for the implementation of the algorithm.

3) Limitations in comprehensive application of algorithm

Many of our students can understand the basic design methods of algorithm, and they are able to take advantage of one to solve a simple problem. However if they encounter some special problems, which require integrated application of a variety of algorithm design methods, they are often unable to proceed it and can not really apply their knowledge. Furthermore, it is lack of a comprehensive problem-solving skills. All those problems are due to students' lack of training for comprehensive application.

In order to address these problems appeared during the teaching process of this course, through the teaching practice and exploration of several years, we make a summary of some educational technologies and teaching methods. It has achieved good teaching effect.

Fig. 1 is a schematic diagram of our proposed teaching scheme for "Algorithm Design and Analysis" course based on an integrated application of the following educational technologies.

III. PROBLEM-BASE-BASED "FILLING TEACHING"

One of the main reasons that students are difficult to master the basic theory and application techniques of algorithm, is that the lack of application examples. During the process of teaching, if the teachers just purely explain the theory of knowledge, or selectively introduce a small amounts of examples involved some simple algorithm applications, students may be able to understand the basic knowledge of teaching in class, but after class when they encounter problems with other complicated types, they still can't flexibly use the knowledge they've learned [5]. Therefore, under our problem-base-based "fill teaching" mode, the teachers aim at implementing the "fill teaching" in a problem base with a large number of examples. We can explain and analyze some of the primary classic examples of problem in detail, and directly point out the difficulties and the key design steps for some secondary examples in the problem base. Especially in the beginning stage of learning algorithm design, we advocate the teaching of a large number of algorithm design examples, the teacher even can guide students to do understanding memories of some key design strategies. It has been proved that it plays an important role in the later in-depth study of algorithm.

In addition, it should be highlighted that we need to carefully select and design a problem base. This can effectively avoid the appearance of tedious and negative reaction of students while implementing the teaching based on a large number of examples. In our teaching practice, we often choose two main kinds of problems in the base. One is the training problem base of Collegiate Programming Contest. Now it is selected mainly from ACM International Collegiate Programming Contest (ACM-ICPC). ACM-ICPC is the year contest sponsored by the Association for Computing Machinery (ACM) to show students' innovation ability, team spirit, programming under pressure, and analysis and problem solving ability. It has now been developed into one of the most influential university computer contests [6]. The topics of this contest include: computational geometry, number theory, combinatorics, searching algorithm, dynamic programming, etc. And there are a lot in common between those topics of ACM-ICPC and the teaching content of "Algorithm Design and Analysis" course [7]. Another source of problems is derived from testing problems of companies. Now a large number of IT companies will arrange the testing and interview for candidates, in which there are many problems that are relevant to the issue of our course. After the training for these actual exam problems of companies, there will be a reference and guiding significance in the students' later employment. So the students have a strong interest in it.

IV. MULTI-LEVEL HEURISTIC TEACHING

The heuristic teaching method mentioned here refers to the students' learning process based on experience. The teachers guide the students to solve, analyze, criticize, judge, and conclude, by asking questions to them. Thus the students can comprehend by drawing an analogy or inferring other things from one fact. It makes the experience of students gradually expand, and lets the students to think more flexibly.

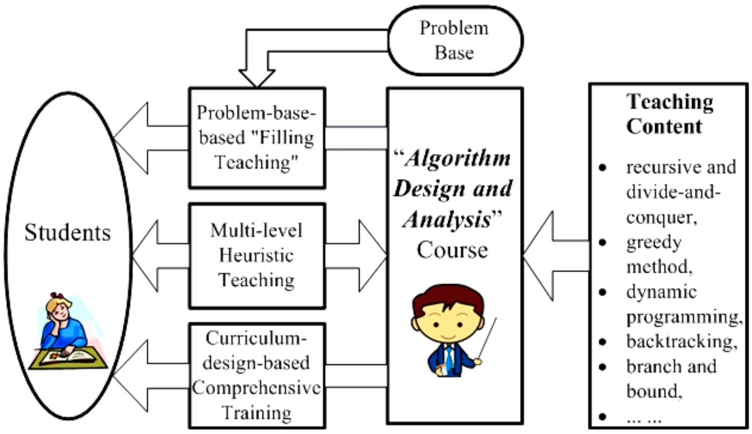
Problem Base
Problem-base-based "Filling Teaching"
Multi-level Heuristic Teaching
Curriculum-design-based Comprehensive Training
Students
"Algorithm Design and Analysis" Course
Teaching Content
recursive and divide-and-conquer,
greedy method,
dynamic programming,
backtracking,
branch and bound,
… …

Figure 1. Schematic diagram of teaching scheme for "Algorithm Design and Analysis" course.

“Algorithm Design and Analysis" course has a close relationship with practice. Therefore, in the classroom teaching, in addition to inspire students to actively think about the basic principles of the algorithm, teachers also need to inspire students to select and apply the algorithm in the actual problems.

For example, in the teaching of design method divide-and-conquer, we usually explain a classic problem, i.e., "closest pair of points problem”. This problem needs to design a divided-conquer algorithm to find two points in a set of n points, so that the distance between those two points is minimum. We will generally start with analysis under the one-dimensional condition and discuss how to identify these two points on the X-axis. Then, on this basis, we can extend that method to find the solution under the two-dimensional case, in which the "closest pair of points problem” will be solved in a plane. In general, it is the end of this algorithm teaching here. But in order to further inspire the students’ algorithm design thinking, starting from the requirements of some popular network games, we can raise a new question, i.e., in the three-dimensional space, how to design a divide-and-conquer algorithm to deal with the "closest pair of points problem”? By guiding the students to think deeply about this issue, whether they can really find out the corresponding algorithm or not, during this process, the student greatly enhance the understanding of practice application of divide-and-conquer algorithm.

V. Curriculum-design-based Comprehensive Training

Curriculum design is an important training part for students to use the learned knowledge of algorithm. In this course, it is very important to guide students to use all kinds of algorithm design techniques to solve practical problems. It can not only train their practical programming ability, but also train their ability of analyzing and solving problems. It helps to develop students' creative thinking and innovation ability as a whole.

In the curriculum design, considering the different application background, we provide a large choice of many comprehensive questions which are carefully planed by us.. To solve these problems, it needs a comprehensive application of graph theory, searching technologies, dynamic programming, backtracking, recursive and divide-and-conquer, and branch and bound method. It is a certain challenge for students who are required to use programming language, e.g. C++ or Java, and other software, to deal with the problems they selected within the range mentioned. They need to discuss and analyze the time and space complexity of the proposed algorithms. Finally, they are required to submit a comprehensive testing report within two weeks. After receiving all the students' reports, the teachers will randomly select some students to explain their own ideas in class. Through the curriculum-design-based comprehensive training, the students’ theory and practice level of “Algorithm Design and Analysis” course is enhanced, and their comprehensive application ability has been improved.

VI. Conclusion

In our teaching practice and exploration of several years in “Algorithm Design and Analysis” course, it has been proved that, with an integrated application of problem-base-based "filling teaching", multi-level heuristic teaching, and curriculum-design-based comprehensive training technologies, it fully encourages the students to initiate their own learning activities, makes the class more interesting and efficient, and improves the students' learning interest. On the other hand, the teaching method is more flexible, so that it makes the students easier to understand the teaching content.

Acknowledgment

This work was jointly supported by the Project of National Characteristic Major Construction of Computer Science and Technology under Grant TS1Z007 and the Project of Beijing Characteristic Major Construction of Computer Science and Technology under Grant 20083.

References

[1] S. Yan, B. Liu, L. Feng, and N. Ji, "Some thoughts on "algorithm design and analysis" teaching reform," in Proc. Int. Conf. Ind. Mechatronics Autom., Wuhan, China, May 2010, pp. 595-597.

[2] X. Xia and L. Gui, "Teaching method innovation in "algorithm analysis and design"," in Proc. Int. Conf. E-Bus. E-Gov., Shanghai, China, May 2011, pp. 766-768.

[3] Z. Wang, "The research on teaching ideas of "data structure and algorithm" in non-computer major," Adv. Intell. Soft Comput., vol. 140 AISC, pp. 249-254, 2012.

[4] X. Chen, D. Xu, and F. Fang, "A multi-level practice teaching system for courses of algorithm and program design," Res. Explor. Lab. , vol. 31, no. 8, pp. 319-322, August 2012. (in Chinese)

[5] L. Torrey, "Teaching problem-solving in algorithms and AI," in Proc. Natl. Conf. Artif. Intell., Toronto, Canada, July 2012, pp. 2363-2368.

[6] ACM International Collegiate Programming Contest, http://en.wikipedia.org/wiki/ACM_International_Collegiate_Programming_Contest.

[7] Y. Lin and X. Liang, "Curriculum reform of algorithm design and analysis based on ACM-ICPC," Comput. Knowl. Technol., vol. 8, no. 7, pp. 1708-1709, March 2012. (in Chinese)