# Exploring Intelligent Instructional Strategies Tailored to Individual Differences

Jing Bai

University of International Relations Beijing, China

baijing@uir.edu.cn

**Abstract.** In current teaching practice for introductory computer science courses, the traditional lecture model, characterized by a "uniform pace and standardized output", often struggles to accommodate individual differences arising from learners' varied prior knowledge and cognitive rhythms. Consequently this approach frequently leads to gaps in the development of computational thinking and a decline in learning motivation. To address this teaching dilemma, this paper introduces the Markov Decision Process as a quantitative analytical framework to construct an adaptive guidance model tailored to the foundational cultivation of computational thinking. By formally modeling learners' cognitive states, deriving instructional intervention strategies, and establishing incentive-based feedback mechanisms, this model achieves the dynamic optimization of learning pathways. This research not only provides a methodological reference for adaptive learning path planning in large-scale educational environments, but also paves the way for the in-depth reform of interdisciplinary teaching models and the development of corresponding software systems in the future.

**Keywords:** Introductory Courses, Intelligent Education, Reinforcement Learning

## 1 Introduction

Within the current educational framework for computer science, introductory courses serve as the key to unlocking professional understanding, bearing the critical mission of cultivating computational thinking and fostering information literacy. However, the teaching difficulty of introductory computer courses exhibits multi-dimensional characteristics [1]. First, the knowledge graph of such courses spans multiple dimensions, including computer hardware architecture, software principles, algorithm design, and programming fundamentals [2]. The content is complex and highly abstract, making it difficult for beginners to construct a comprehensive cognitive framework within limited course time [3]. Second, learners from diverse backgrounds exhibit significant differences in their foundational knowledge, logical reasoning skills, and areas of interest, making it challenging for traditional, standardized teaching models to meet these individualized learning needs [4]. Furthermore, because introductory courses often priori-


C. V. Garces et al. (eds.), *Proceedings of the 6th International Conference on Internet, Education and Information Technology (IEIT 2026)*, Atlantis Highlights in Social Sciences, Education and Humanities 53,
https://doi.org/10.2991/978-94-6239-733-0_33

tize the exposition of concepts, learners often fall into the misunderstanding of fragmented rote memorization, which makes it difficult to transform discrete theoretical knowledge into practical ability to solve real-world problems [5]. Moreover, these introductory courses are often simply classified as "basic classes", leading some learners to underestimate their significance and lack the internal motivation for independent exploration. At the same time, due to the time limitation of course, teachers are often unable to analyze the actual cases in depth, which further exacerbates the disconnection between theory and practice. Therefore, in the reform of introductory computer science education, the core challenge to be solved urgently lies in how to ensure comprehensive coverage of the subject content while taking into account the cultivation of knowledge depth, and provide appropriate and personalized support for learners in this process.

Traditional introductory courses typically follow a linear pedagogical path, and teachers uniformly arrange the content and progress according to the syllabus. However, this static design cannot adapt to the diverse cognitive rhythms of individual students. For learners with weak foundational knowledge, too fast input of knowledge may lead to misunderstanding. For learners with relevant learning backgrounds, long-term repetition of basic concepts will weaken their learning motivation. In the face of the above challenges, the education sector has put forward a variety of exploration directions for the teaching reform of introductory computer courses.

1. Case-driven teaching transforms abstract concepts into specific problems by introducing real-world cases (such as artificial intelligence ethics and network security events), which significantly improves learners' participation.

2. The flipped classroom mode encourages learners to complete knowledge transfer before class through the combination of pre class video learning and classroom discussion. This mode focuses on high-level thinking activities and collaborative exploration, and improves the interactive density of the classroom.

3. Some universities have implemented virtual simulation experiments, allowing learners to intuitively experience data coding, network topology construction and other processes in the simulation environment. This helps to make up for the lack of pure theory teaching, so that learners intuitively understand the interaction mechanism between hardware and software.

Although these explorations have achieved positive results in some aspects, their limitations cannot be ignored: case-driven teaching is often difficult to systematically cover all knowledge modules of the course. Flipped classroom is difficult to ensure the synchronization of learning progress, and has high requirements for learners' autonomous learning ability. Virtual simulation experiment is limited by the ratio of platform resources and teachers, so it is difficult to achieve large-scale personalized guidance. Therefore, the existing methods have not fundamentally solved the structural contradiction between "unified teaching supply" and "diversified learning needs" in the introductory courses. How to achieve dynamic and accurate personalized learning support in large-scale teaching is still an unsolved problem. The essence of this dilemma is that the teaching system lacks the ability to perceive learners' real-time knowledge state and the decision-making mechanism to dynamically adjust teaching strategies based on this state.

## 2 Reinforcement Learning-Driven Teaching Model of Introductory Computer Courses

In this context, reinforcement learning provides a new paradigm for the teaching optimization of introductory courses. Its core idea is to realize adaptive decision-making through the closed-loop mechanism of "state-action-reward". The learning process of learners is modeled as a Markov decision-making process, in which the system state includes the mastery degree and cognitive load level of learners in various knowledge dimensions (such as information representation, system structure, network protocol, algorithmic thinking, etc.). Teaching actions cover intervention strategies such as resource recommendation, difficulty adjustment, and exercise type selection. The reward signal is composed of multiple indicators such as learners' knowledge gain, task completion quality, and concept retention rate. Through continuous interaction and strategy optimization under this framework, the system can dynamically identify learners' cognitive state and dynamically generate the optimal knowledge path for each learner. For example, more visual cases and analogy explanations are provided for those with weak abstract thinking, and theoretical progression is accelerated and open inquiry tasks are added for those with better logical basis, so as to realize the adaptive teaching of introductory courses in the true sense.

Markov decision process provides a systematic modeling method for the design of introductory computer courses through key elements such as state space S, action set A, reward function R, state transition probability P, discount factor γ, and strategy π.

1. The state space S of the model describes the learning status of learners in the course. For the introductory computer courses, the core content can be divided into several knowledge modules, including basic computer concepts, operating system principles, network foundation, data structure, algorithm thinking training, database foundation and information security concepts. The learning state of each learner can be used as a vector $[s_1, s_2, \cdots, s_N] \in S$, where $s_i \in [0,1]$ reflects the learners' mastery level of the *i*th knowledge module, 0 indicates complete incomprehension, and 1 indicates comprehensive mastery. The update of learners' state depends on the completion of classroom assignments, experimental report performance and online test results, which can reflect the learning effect in real time and provide reference for subsequent teaching decisions.

2. Action space A defines the learning activities available to students in the course system. The actions of introductory computer courses can include many types, such as "watching theoretical video explanation", "participating in hands-on experiments", "completing comprehensive programming exercises", "reviewing and consolidating what has been learned", "participating in online discussions or tests", etc. Different actions correspond to different learning objectives and difficulty levels, so that the system can recommend the most appropriate learning tasks according to the current knowledge state of learners and finally achieve targeted and hierarchical teaching guidance.

3. Reward function R is an important part of the model, which provides quantitative feedback for learners' learning behaviors. In order to encourage effective learning, the reward mechanism can be set as follows: complete basic tasks or experiments to obtain basic positive rewards$+r_1$; complete more difficult comprehensive exercises to obtain

higher rewards $+r_2$ $(r_2 > r_1)$. Learners can get $+r_3$ extra rewards for completing cross module projects. Additional rewards can be attached to successful completion of the task for the first time to encourage efficient learning. For the wrong or unfinished tasks, we can appropriately punish $-r_4$ to urge learners to correct their mistakes, but the punishment should not be too heavy to avoid attacking the enthusiasm of learning. In addition, learners who preview course content or actively participate in classroom activities can also receive small rewards $+r_5$ to maintain their learning motivation. Through this design, learners' learning behavior can be dynamically guided, and gradually optimize the learning path.

4. The state transition probability $P(s'|s, a)$ is used to describe the new knowledge state that learners may reach after performing a specific learning behavior in the current knowledge state. Because the introductory computer courses cover a wide range of contents, including several knowledge modules such as basic computer concepts, operating systems, network principles, data structures and algorithms, and there is a correlation between different knowledge points, the changes of learners' learning state are highly nonlinear and dynamic, and it is difficult to accurately predict through a simple mathematical model. In order to deal with this complexity, we can use learners' past homework scores, experimental reports, test scores and classroom interaction data to carry out statistical analysis, and estimate the impact of different actions on state transition. At the same time, reinforcement learning algorithms, such as DQN, can also be introduced to gradually learn the optimal learning strategies by simulating the interaction between learners and the course environment, without accurately modeling each state transition. In addition, the discount factor $\gamma$ is used to control the trade-off between short-term and long-term learning benefits. Higher $\gamma$ values emphasize learners' long-term mastery of knowledge and ability accumulation, while lower $\gamma$ values pay more attention to the improvement of recent test scores or experimental completion.

5. Strategy $\pi(a|s)$ defines the probability distribution of learners' choice of various learning activities in a specific learning state, which is the core basis of learning path planning. In the introductory computer courses, there are significant differences in the basis and interests of different learners. Strategy optimization can ensure that the system can intelligently select appropriate learning tasks according to each learner' current knowledge modules, such as arranging theoretical learning, hands-on experiments or comprehensive exercises. Through continuous updating and optimizing strategies, the system can form the optimal strategy, so that learners can realize the comprehensive mastery of knowledge points and the steady improvement of ability in the whole course learning process from the initial state. The model not only provides personalized learning path planning for introductory computer courses, but also provides theoretical support and operational methods for the introduction of adaptive teaching mechanism and the improvement of intelligent education system.

## 3 Theoretical Significance and Practical Value of Intelligent Teaching Strategy

This paper reconstructs the teaching of introductory computer courses from the perspective of reinforcement learning. Specifically, Markov decision process is introduced as a quantitative analysis framework to construct an adaptive guidance model suitable for computational thinking enlightenment. The model realizes the dynamic planning of guidance path by formalizing the cognitive state of learners, deducting teaching intervention strategies, and setting of incentive feedback mechanism. Its theoretical value and practical significance are reflected in many levels.

1. Theoretically, this method transforms the educational process from experience-driven art to data-driven science. For a long time, how to realize differentiated oriented teaching in introductory computer courses has been an outstanding core problem in the field of education. The difficulty lies in the significant differences between learners in the aspects of prior knowledge, cognitive style and learning rhythm, which is difficult for the traditional teaching model to make a dynamic response. This paper implements intelligent teaching for learners, and transforms the problem into a formalized decision-making process: Taking the real-time state of learners as the input and the optimal teaching strategy as the output, pursuing the maximization of long-term learning benefits in the dynamic balance of exploration and utilization, enriching the technical path of intelligent education system, and opening up a new problem domain for the follow-up research of educational data mining and learning analysis.

2. At the practical level, the mechanism can effectively alleviate the cognitive load caused by the "scattered knowledge points" of introductory courses, make the learning sequence more in line with the individual cognitive development law through dynamic path planning, help zero foundation learners build confidence, and enable learners with prior knowledge to obtain sufficient challenges and growth space. More importantly, the reinforcement learning system has the ability of continuous evolution, and the accumulated teaching data can promote the continuous optimization of the model. This closed-loop teaching based on state perception and instant feedback is helpful to transform learners from the object of passively accepting knowledge to the core subject of continuous optimization, so as to cultivate learners' metacognitive ability and self-regulation ability while maintaining learners' learning motivation.

## 4 Conclusion

This paper takes the introductory computer courses as an example to verify the theoretical feasibility and system implementation value of the personalized teaching model based on Markov decision process. In the future, the state space in the model needs to further integrate dynamic differentiation features, such as learning style, cognitive load and emotional state, in order to depict a more realistic picture of students. At the application level, it will promote the evolution of the prototype system to an extensible intelligent teaching platform, and realize the deployment and feedback closed loop in the real environment by connecting with the online education platform and the enterprise training system. Finally, in the digital education ecology, we should strengthen the capture of large-scale learning behavior data to form a data-driven closed loop of teaching optimization, and provide power for the sustainable development of intelligent education.

## Reference


1. Kose U, Arslan A. Optimization of self-learning in Computer Engineering courses: An intelligent software system supported by Artificial Neural Network and Vortex Optimization Algorithm[J]. Computer Applications in Engineering Education, 2017, 25(1): 142-156.
2. Du Plessis J P, Van Biljon J A, Tolmie C J, et al. A model for intelligent computer-aided education systems[J]. Computers & Education, 1995, 24(2): 89-106.
3. Chen L, Chen P, Lin Z. Artificial intelligence in education: A review[J]. IEEE access, 2020, 8: 75264-75278.
4. Wang W. Intelligent Platform for Educational Resources of Computer Basic Courses in the Digital Education Environment[J]. innovation, 2023, 6: 7.
5. O'Shea T. Intelligent systems in education[J]. Introductory Readings in Expert Systems, Gordon and Breach Science Publishers, New York, 1982: 147-176.