J. COMBIN. MATH. COMBIN. COMPUT. 125 (2025) 9337--9352

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Research on optimization strategy of dynamic planning method in resource balancing in online and offline integrated teaching in higher education

Linglanxuan Kong[1], Dongtao Han[2,✉]

[1] *Personnel Department, Shanghai Customs University, Shanghai, 201204, China*
[2] *School of Government, Shanghai University of Political Science and Law, Shanghai, 201701, China*

ABSTRACT
Starting from the essence of dynamic programming algorithms, the terminology in dynamic programming algorithms, the applicability conditions of the algorithms, and common sub-problem models are summarized. The Belman optimal algorithm is used to split the multilevel problems in dynamic planning into simple single-level problems and solve them one by one, combined with the function approximation structure to approximate the performance index function, to construct the adaptive dynamic planning algorithm, and to apply it in the resource balancing optimization of integrated teaching. The results show that the adaptive dynamic programming algorithm has better resource balancing effect than other algorithms, and the number of convergence and running time are reduced by 6-53 times and 48.92-90.34 seconds respectively. The introduction of the adaptive dynamic programming algorithm improved the resource balancing accuracy of university teaching and learning management by 4.0%-17.4% in each subject group. As the number of resources increased, the time consumption required when balancing resources decreased by 50%-83.33% for test groups 3, 4 and 5, and the efficiency of the test improved by 75%-100%. This shows that the algorithm proposed in this paper is effective when dealing with balancing online and offline teaching resources in higher education.

Keywords: Adaptive Dynamic Programming Algorithm, Bellman Optimality Principle, Functional Approximation Structure, Teaching Resource Balancing

1. Introduction

Driven by the development of "Internet+Education", the blended teaching mode, which integrates online teaching and traditional offline teaching, has emerged [1-2]. Blended teaching breaks the

✉ Corresponding author.
E-mail address: handt123@126.com (D. Han).
Received 15 January 2024; Revised 10 March 2024; Accepted 30 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-511

limitations of traditional education, integrates teaching resources, realizes effective communication between teachers and students, and comprehensively promotes the improvement of students' comprehensive quality [3]. However, due to the short research time of online and offline blended teaching mode, there are still many problems [4]. In order to solve the drawbacks of hybrid teaching, colleges and universities should increase the investment in education, make every effort to protect the teaching-related support, mobilize the enthusiasm of teachers and students in the classroom, guide students to consciously carry out the construction of the knowledge system, integrate high-quality teaching resources, give full play to the optimization of the function of the hybrid teaching configuration, improve the assessment and evaluation mechanism, and promote the scientific evaluation of teaching, so as to maximize the advantages of on-line and off-line teaching to promote the innovation of educational reforms in colleges and universities. In order to maximize the advantages of online and offline teaching, promote the reform and innovation of university education, and let education truly serve the majority of students [5-8].

Online-offline integrated teaching, also known as "blended teaching", is an improvement and innovation of the current teaching mode [9]. This model is student-centered, pays more attention to the cultivation of students' independent ability, and meets students' individualized learning needs; at the same time, blended teaching can not only give play to the interactivity and supervision of traditional teaching, but also show the advantages of online guided learning and sharing of educational resources [10-12].

In the traditional teaching mode, students acquire knowledge in the classroom with the help of textbooks or complete practical training in limited class time, with limited opportunities to participate in teaching [13]. Online and offline blended teaching breaks the limitations of time and space, and students independently choose the learning content and control the learning progress, which truly realizes the tailor-made teaching [14-15]. On the one hand, the open learning space and time are more conducive to the improvement of students' professional skills. Blended teaching returns the initiative of the classroom to the students, who are free to arrange their learning progress according to their own mastery and deepen their mastery of knowledge [16-18].

Compared with traditional offline classroom teaching, online-offline integrated teaching is more in-depth [19]. The most popular online and offline integrated teaching mode mainly includes, first, including pre-course preparation, practical operation, practice report, practice verification and other links, in the pre-course preparation, the teacher can through the network education platform for classroom teaching of important and difficult to formulate the thinking questions, prompting the students in the pre-study process to independently think about exploring the answers to the questions [20-21], and second, classroom teaching, the teacher should be placed in the focus on the practical operation link, urging students to exercise to cultivate solid practical hands-on ability, practice report should be through the online teaching. Secondly, during classroom teaching, teachers should focus on practical operation, urge students to exercise independently to cultivate solid practical hands-on ability, and the practice report should be completed through the online teaching system [22-23], and thirdly, based on the results of the practice and the report assessment and scoring, so that students can learn to analyze the strengths and weaknesses of the whole teaching process by themselves, thus effectively improving the efficiency and quality of teaching [24].

Starting from the basic theory of dynamic programming algorithms, the multi-stage decision-making problems in dynamic programming are outlined, and the common sub-problem models in dynamic programming are summarized. The comparison with other algorithms highlights the characteristics of dynamic programming in terms of space consumption and time efficiency in solving practical problems. Using the Bellman optimization principle combined with the inverse analysis strategy, the multi-stage problem is split into simple single-stage problems and solved one by one to get the optimal decision of the resource balancing process. The performance of the adaptive dynamic programming algorithm is compared and analyzed with that of the genetic algorithm and ant colony algorithm in terms of operation time and convergence, and the application effect of the algorithm of this paper in the balancing of teaching resources is evaluated.

2. Adaptive dynamic planning algorithm based on the balance of teaching resources

2.1. Dynamic Planning Algorithm

2.1.1. Multi-stage decision-making. Multi-stage decision making [25] refers to the fact that a problem can be divided into multiple interconnected stages with a chain-like structure in a particular order. At each stage, a decision needs to be made, the decision of the previous stage will directly affect the state of the next stage, and the state of the next stage depends on the results of the previous stage. All stages of decision-making will eventually form a decision sequence, to solve the multi-stage decision optimization problem is to find a decision sequence to make the problem optimal for a certain indicator function.

2.1.2. Dynamic programming algorithms. Stages: Stages are attributes of the problem. Usually a multi-stage decision-making problem can be divided into several interconnected stages in the order in which they are processed so that the problem can be solved sequentially in the order in which they are divided. Multiple stages are generally present in a multi-stage decision optimization problem. Stage variables are often represented by the letter k. Stages have two characteristics: first, there is a sequential relationship between stages, and second, there is a dependency relationship between stages.

State: State is an attribute of stages. The state is used to describe the objective conditions that each stage is currently in. Each stage usually contains multiple states.

The state variable f_k is used to denote the state in which stage k is located, and the set of values of state f_k is denoted by F_k and is called the set of states, where $f_k \in F_k$, the set of states in stage 3 of the multistage decision problem $F_k = \{D_1, D_2\}$.

Decision making: when the state of a stage is determined, different choices can be made, and the next state can be deduced from the stage

A choice that can be deduced from a state in that stage to a state in the next stage is called a decision.

The decision variable at stage k state f_k is denoted by $v_k(f_k)$, and the set of permissible decisions at stage k state f_k is denoted by $V_k(f_k)$. Clearly there are $v_k(f_k) \in V_k(f_k)$. two choices, i.e., two decisions, to be made from stage 2 state C_1, which can be deduced from stage 3 state D_1, D_2. That is:

$$k = 2, f_2 = C_1, V_2(f_2) = \{C_1 \to D_1, C_1 \to D_2\} \tag{1}$$

Strategy: The combination of decisions chosen at each stage is called a strategy and is often represented by the letter p:

$$p_k(f_k) = (V_k(f_k), V_{k+1}(f_{k+1}), V_{k+2}(f_{k+2}) \ldots V_n(f_n)) \tag{2}$$

State Transfer: In dynamic programming, the state of the previous stage and the corresponding decision choices will determine the state of the problem in the current stage. That is, if the state f_k of stage k and the decision $v_k(f_k)$ are determined, then the state f_{k+1} of stage $k+1$ of the problem can also be determined. State transfer is the process of deriving state f_{k+1} in stage $k+1$ from state f_k and decision $v_k(f_k)$ in the first k stages of the problem, and is often expressed in the following equation:

$$f_{k+1} = T_k(f_k, v_k(f_k)) \tag{3}$$

Indicator function and optimal indicator function [26]: the indicator function is used to measure the advantages and disadvantages of the strategy chosen for the problem, while the optimal indicator function is the value of the indicator function when the problem adopts the optimal strategy, and the value of the optimal indicator function is the solution of the problem. Generally commonly used function $g_k(f_k)$ represents the optimal metric value when the problem starts from a state f_k in stage k and reaches the end of the problem using the optimal strategy $p_k(f_k)$ of the process. The metric function $g_3(C_1)$ of the problem represents the shortest path from the current point C_1 to the end point E, where the goal of the problem is to solve $g_1(A)$.

There are generally two ways to calculate the optimal indicator function: if the initial state of the problem is determined, it can be calculated by starting from the initial state to the goal state in a downward push: if the goal state of the problem is determined, it can be calculated by starting from the goal state backward push. In general, the backward push is used to calculate, at this time the planning equation is:

$$g_k(f_k) = opt \quad (g_{k+1}(T_k(f_k, v_k(f_k))) + C(T_k(f_k, v_k(f_k)))) \tag{4}$$

where f_k is the state the problem is in at stage k, $v_k(f_k)$ is the decision variable when the state is f_k at stage k, and $V_k(f_k)$ is the corresponding set of allowed decisions. $T_k(f_k, v_k(f_k))$ is the process of computing the state of stage $k+1$ from the states and decisions $v_k(f_k)$ of the first k stages of the problem, i.e., $f_k + 1 = T_k(f_k, v_k(f_k))$. $C(T_k(f_k, v_k(f_k)))$ is the cost of the decisions made from stages k to $k+1$. $g_k(f_k)$ denotes the value of the best metric to reach the end of the process using the optimal strategy starting from the state f_k of the k th stage of the problem, which can be time, space, path, yield, etc.

2.1.3. Conditions for the applicability of dynamic programming algorithms:

1) Optimal substructure property. Dynamic programming algorithms to solve multi-stage decision-making problems, the first step is often to describe the structure of the optimal solution to the problem, if the optimal solution of a problem is composed of the optimal solution of the subproblem, the problem is said to satisfy the optimal substructure property. The nature of the optimal substructure is the basis of dynamic programming algorithms, any solution structure does not satisfy the nature of the optimal substructure of the problem, it can not be solved by dynamic programming methods.

2) No backwardness. When the multi-stage decision-making problem is divided into a number of stages, for a certain stage, the state of each stage before it will not be able to affect the decision made in the current stage. The current stage can only through the current state of the decision-making process to the future development of the process will not depend on the state of the previous stage, which is the absence of posteriority.

3) Overlapping subproblems. When the algorithm repeatedly calculates the same sub-problem in the process of running, it means that the sub-problems in the problem have overlapping. The subproblems of the problem that can be solved by the partition algorithm cannot overlap, and if the subproblems overlap, there are a large number of repetitive calculations in the partition algorithm, which leads to the time inefficiency of the algorithm.

2.1.4. Patterns in the design of dynamic programming algorithms. The conventional design steps of the dynamic programming algorithm are as follows:

1) Define subproblems: Based on the characteristics of the problem, the problem is divided into several subproblems. There is a sequential relationship between the sub-problems and the current sub-problem can be solved given a relatively small sub-problem solution.

2) Select the state: the objective situation of the subproblem is represented by the state, and the state must satisfy no posteriority.

3) Determine the state transfer equation: the process of determining the state of the current stage through the state of the previous stage of the current stage and decision-making choices is state transfer. Decision-making and state transfer have a direct connection, if the stage is determined to determine the range of decisions that can be selected, then the state transfer equation of the problem can be written naturally.

4) Finding boundary conditions: the initial or ending conditions for the iteration of the state transfer equation. Dynamic programming algorithm does not have a unified standard model can be used for all problems, the actual problem is different, the algorithmic model of dynamic programming is also different, therefore, the need for specific problems to be specifically analyzed.

2.1.5. Common Subproblem Models

I: Input is $X_1, X_2, \ldots, X_n$, divide the subproblem by $X_1, X_2, \ldots, X_i$.

II: Inputs $X_1, X_2, \ldots, X_n$ and $Y_1, Y_2, \ldots, Y_n$, divide the subproblems by $X_1, X_2, \ldots, X_i$ and $Y_1, Y_2, \ldots, Y_j$.

III: Divide the subproblem by $X_i, X_{i+1}, \ldots, X_j$ when the input is $X_1, X_2, \ldots, X_n$.

IV: When the input is a tree structure, the stage is divided by subtrees.

Tree dynamic programming is dynamic programming built on a tree structure. The subproblems are more obvious in the tree structure, and generally the subtree with each node as the root node is the subproblem. The state transfer equation is established based on the relationship between the father node and the child node, and the optimal value of the father node is deduced from the optimal value of the child node. Finally, the optimal solution of the problem is finally obtained by passing information to the root node through the child nodes of the root, and solving for the root.

2.2. Adaptive Dynamic Programming Fundamentals

2.2.1. Bellman Optimality Principle. Dynamic programming (DP) method [27] is an effective tool for solving optimal control problems. The Bellman optimization principle involves splitting the multistage problem into simple single-stage problems and solving them one by one. In the problem solving process, the use of reverse analysis strategy, that is, for a stage of decision-making, it depends on the current state and will affect the later decision-making, each stage of the optimal decision-making composition constitutes the optimal strategy of the problem. Linear, nonlinear dynamic problems can be solved using the idea of the principle of optimality, for the discrete nonlinear system assuming that phase k has state variable $x(k)$, the decision is $u(k)$, you can get the following dynamic equations:

$$x(k+1) = F(x(k), u(k)), k = 0, 1, \cdots \tag{5}$$

where: $x(k) \in R^n$ is the state variable and $u(k) \in R^m$ is the decision variable. This equation for moving from $x(k)$ to $x(k+1)$ is called the state transfer equation. Define the “cost” of transferring the system from state k to state $k+1$ with the following cost function:

$$J(x(i), i) = \sum_{k=i}^{\infty} \gamma^{k-i} l(x(k), u(k), k), k = i, i+1, \ldots \tag{6}$$

where: $l(x(k), u(k), k)$ represents the utility function, $\gamma (0 < \gamma \le 1)$ is the discount factor and is satisfied, the cost function can also be called the performance indicator function. In the case of initial state determination, the control objective is to minimize the above equation by solving $u(k)$. The optimality principle of problem solving is the idea of reverse recursion, to minimize the cost function in stage k, the minimum cost of the current moment and the next moment needs to be cumulative:

$$J^*(x(k)) = \min_{u(k)}\{l(x(k),u(k)) + \gamma J^*(x(k+1))\} \tag{7}$$

The corresponding decision $u(k)$ at the stage of satisfying k in the above equation is the optimal decision at the current moment and is represented by the following equation:

$$u^*(\mathrm{k}) = \arg\min_{u(t)}\left\{(x(k),u(k)) + \gamma J^*(x(k+1))\right\} \tag{8}$$

For continuous nonlinear systems:

$$J(x(t),t) = \int_t^\infty l(x(\tau),u(\tau))d\tau \tag{9}$$

$$x(t) = F(x(t),u(t)), t = 0,1,\cdots \tag{10}$$

where: $F(x,u)$ represents the continuous system state function. The cost function of this system is:

It has the same meaning as the cost function of the discrete system, i.e., find an optimal response to minimize the cost function. The way to solve the continuous system is to discretize it and find the solution by using the idea of discrete system, both will theoretically converge to the same solution as long as the sampling time is small enough. According to the optimality principle, the Hamilton-Jacobi-Bellman (HJB) equation can be obtained:

$$\begin{aligned} -\frac{\partial J^*}{\partial \mathrm{t}} &= \min_{u\in U}\left\{l(x(t),u(t),t) + (\frac{\partial J^*}{\partial x(\mathrm{t})})^{\mathrm{T}} F(x(t),u(t),t)\right\} \\ &= l(x(t),u^*(t),t) + (\frac{\partial J^*}{\partial x(t)})^{\mathrm{T}} F(x(t),u^*(t),t) \end{aligned} \tag{11}$$

The above equation is a first order nonlinear partial differential equation of $J^*(x(t),t)$ to $x(t),t$. Motor vector control is a nonlinear system, and the optimal control strategy is obtained by solving the HJB function. However, it is not easy to solve the first order nonlinear partial differential equation, and the problem of "dimensional disaster" occurs with the growth of state variables and decision making.

2.2.2. Adaptive dynamic programming. Adaptive Dynamic Programming [28] uses a function approximation structure to approximate the performance index function and realize the forward solution of the optimization problem. The structure of ADP method mainly consists of evaluation network, action network network and model network. Where the evaluation network is the approximation structure of $J(x(k))$, the action network is the approximation network of the relationship between $x(k)$ and $u(k)$, and the model network represents the controlled system. Two strategies are used to realize ADP: Heuristic Dynamic Programming (HDP) and Quadratic Heuristic Programming (DHP), and the schematic diagrams of HDP and DHP are shown in Fig. 1 and Fig. 2, respectively. It can be seen that the network structure of HDP and DHP is basically the same, the difference lies in the cost function output of the two. HDP evaluates the network output cost function, while DHP outputs the cost function gradient $\frac{\partial J(x(k))}{\partial x(k)}$.

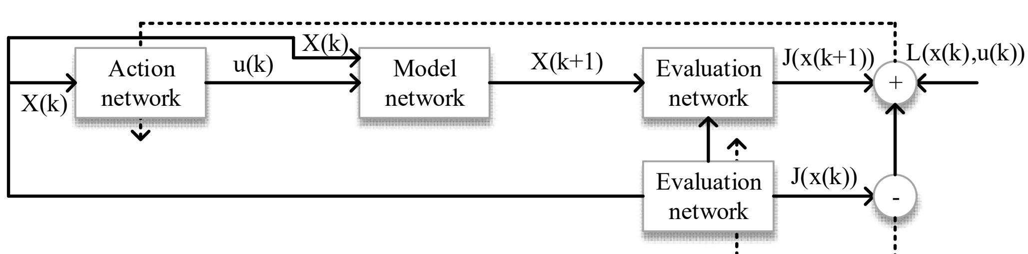

Fig. 1. HDP schematic diagram

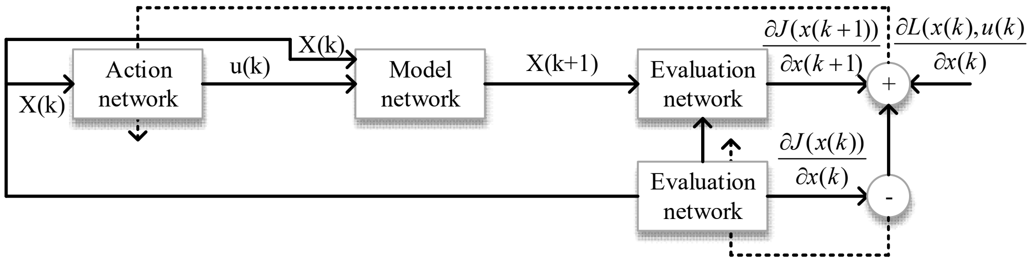

Fig. 2. DHP schematic diagram

Adaptive dynamic programming evolved gradually from the beginning when it had to be trained offline to being able to learn adaptively online. For continuous nonlinear systems, the cost function is shown in Equation (9) and the state equation is shown below:

$$x = f(x) + g(x)u, x(t_0) = x_0 \tag{12}$$

The HJB functions are as follows:

$$H\left(x,u,J_x\right) = l\left(x,u\right) + J_x^T\left(f(x) + g(x)u\right) \tag{13}$$

where: J_x is the partial derivative of the cost function with respect to x. When the optimal control strategy is found, the HJB function is equal to 0. A neural network is generally used to fit the evaluation network and the action network:

$$\begin{aligned} J(x) &= W_c^T \phi_c(x) + \varepsilon_c \\ u(x) &= W_a^T \phi_a(x) + \varepsilon_a \end{aligned} \tag{14}$$

where W_c and W_a are the evaluation network and action network weights, Φ_c and Φ_a are the activation functions of the evaluation network and action network. ε_c and ε_a are the errors of evaluation network and action network. Substituting the output of the evaluation network into the above equation, when $H(x,u,J_x)$ is gradually approaching 0, it means that the evaluation network gradually reaches the optimal cost, and the output of the action network will reach the optimal strategy decided by the evaluation network. According to the above idea synchronously update the two network weights to achieve online adaptive.

2.2.3. DHDP Fundamentals. Discrete nonlinear systems:

$$x(t+1) = f[x(t), u(t)] \tag{15}$$

$x(t)$ is the system state variable, and $u(t)$ is the system control variable. The system indicator function, or called the cost function:

$$J(t) = \sum_{k=1}^{\infty} \alpha^{k-1} r(t+k) \tag{16}$$

r is the utility function, which represents the reward and punishment of the system, and $\alpha (0 < \alpha < 1)$ is the discount factor, which represents the effect of r on the indicator function at different times. In order to solve the optimization problem consisting of equation (14), it is necessary to find a suitable decision sequence $u(t)$ that minimizes the indicator function J. According to Bellman's optimality theory, that is, J^* is the minimum cost sum of t moments and $t+1$ moments, and the corresponding t moments $u(t)$ also reach the optimum:

$$u^*(t) = \arg\min_{u(t)} \{ r(x(t), u(t)) + \alpha J^*(x(t+1)) \} \tag{17}$$

DHDP uses the structure of evaluation network, action network and model network, the DHDP schematic is shown in Figure 3:

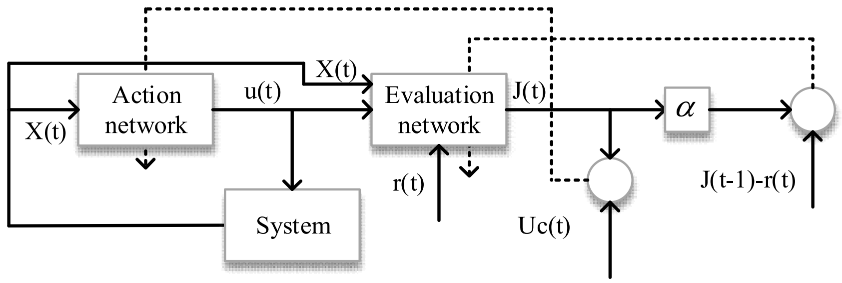

Fig. 3. Schematic diagram of DHDP

The role of the evaluation network in the figure is to evaluate the error function of the network by calculating the indicator function based on the state variables and training it based on the error function:

$$\begin{aligned} E_c(t) &= e_c^2(t)/2 \\ e_c(t) &= \alpha J(t) - [J(t-1) - r(t)] \end{aligned} \tag{18}$$

In order to ensure the convergence of the indicator function to the target, the evaluation network adopts the idea of back-propagation, and the network weights are updated by calculating the first-order bias inversion of the error to the weights:

$$\begin{aligned} w_c(t+1) &= w_c(t) + \Delta w_c(t) \\ \Delta w_c(t) &= -\lambda_c \frac{\partial E_c(t)}{\partial w_c(t)} \end{aligned} \tag{19}$$

$w_c(t)$ is the evaluation network weights and $\lambda_c(\lambda_c > 0)$ is the learning rate of the evaluation network, which decays linearly with t.

The action network is designed to indirectly back-propagate the error between the indicator function J and the target value. The target value of the indicator function $U_c(t)$, and setting $U_c(t)$ to 0 represents the target value of the indicator function training is 0. The state variables are used as inputs to the action network, and the control signals output by the network are used as inputs to the model network, so that the action network can be put into use in either linear or nonlinear models. The error function of the action network is as follows:

$$\begin{aligned} E_a(t) &= e_a^2(t)/2 \\ e_a(t) &= J(t) - U_c(t) \end{aligned} \tag{20}$$

The weight update algorithm is similar to the evaluation network:

$$\begin{aligned} w_a(t+1) &= w_a(t) + \Delta w_a(t) \\ \Delta w_a(t) &= -\lambda_a \frac{\partial E_a(t)}{\partial w_a(t)} \end{aligned} \tag{21}$$

$w_a(t)$ represents the action network weights, and $\lambda_a(\lambda_a > 0)$ represents the learning rate, which decays linearly with t. The action network and evaluation network of DHDP are constructed using PIDNN, which is shown in Eqs. (19) and (21), and the training updates of network weights w_a and

w_c are adjusted according to the feedback self-learning, which needs to compute the gradient change, which is computationally intensive and slow.

3. Effectiveness of Adaptive Dynamic Programming Algorithms in Balancing Teaching Resources

3.1. Application of Adaptive Dynamic Programming Algorithms to Class Scheduling

3.1.1. Resource balance results for scheduling quality. The genetic algorithm, ant colony algorithm and adaptive dynamic programming algorithm are used for experimental testing respectively, with the same test cases, and the quality of the scheduling is judged according to the number of soft constraints satisfied in the class schedule, and the comparison is made after ten runs of each. The comparison results of the three algorithms are shown in Table 1. All three algorithms for class scheduling effectively avoid the occurrence of conflict phenomenon. The ant colony algorithm satisfies the constraints better than the genetic algorithm, but underperforms compared to the adaptive dynamic programming algorithm. The quality of the class schedule discharged using the adaptive dynamic programming algorithm is superior compared to the two algorithms.

Table 1. Comparison results of three algorithms

Program	Genetic algorithm	Ant colony algorithm	Adaptive dynamic planning algorithm
The number of times of hard constraint	0	0	0
A course has two classes a week, and the course is less than a day	38.74	26.59	17.35
Difficult courses schedule bad times	35.12	21.33	16.42
The number of times a teacher has less than a special requirement	9.71	4.28	1.05

3.1.2. Comparison of Algorithm Runtime and Adaptation. The running time and adaptability of the three algorithms are compared and analyzed when the scheduling units are 60, 120, 240, 480, and 960. The results of comparison of algorithm running time and adaptation are shown in Table 2. As can be seen from the table, the three algorithms have more running time as the number of scheduling units increases. In the case of the same scheduling units, the ACO algorithm takes the longest time and the adaptive dynamic programming algorithm is the shortest. The fitness values of the three algorithms increase with the growth of scheduling units, and the fitness of the algorithms gradually stabilizes when the number of scheduling units reaches a certain number. The adaptive dynamic programming algorithm has a higher fitness value than the other algorithms for the same number of scheduling units. In summary, the adaptive dynamic programming algorithm has a short scheduling time and its adaptability value is much larger than other algorithms, so it is verified that the use of adaptive dynamic programming algorithm to solve the scheduling problem is feasible and effective. For example, in the scheduling unit at 960, the time used by the algorithm in this paper is reduced by 23.82s and 65.72s compared with the genetic algorithm and the ant colony algorithm; and its adaptability is improved by 52.83 and 21.77, respectively.

Table 2. The algorithm runs time and fitness comparison results

Comparative content	Scheduling unit	Genetic algorithm	Ant colony algorithm	Adaptive dynamic planning algorithm
Algorithm time(s)	60	28.053	34.264	27.668

	120	55.489	66.049	52.816
	240	113.456	108.316	85.893
	480	268.002	300.35	233.453
	960	631.484	673.387	607.667
Algorithm fitness	60	174.364	205.496	216.722
	120	200.139	206.765	225.955
	240	211.388	218.256	271.736
	480	244.322	268.182	316.954
	960	289.786	320.842	342.612

3.1.3. Convergence comparison of algorithms. In order to compare the performance of the algorithms, given a convergence value, the three algorithms are analyzed for reaching that convergence value. Given the operation time and adaptation value, set the scheduling unit to 526, the algorithms run until 150 seconds to stop the operation, the adaptation value of 250 to stop the operation, each algorithm is run 15 times, and the number of iterations and the adaptation value at the time of stopping the operation are recorded. The results of convergence comparison of the algorithms are shown in Table 3. When stopping the operation, the ant colony algorithm (98.634 times) and the genetic algorithm (105.815 times) run fewer iterations than the adaptive dynamic programming algorithm (123.258), which shows that the three algorithms are the most efficient and the ant colony algorithm is the least efficient when scheduling classes. This is related to the multi-stage decision making and dynamic planning strategy of the adaptive dynamic programming algorithm. In terms of adaptability, the adaptive dynamic programming algorithm has a higher adaptability value (281.163) than the ant colony algorithm (249.559) and the genetic algorithm (235.514), which again validates the argument that a mixture of the two algorithms can make up for the inadequacy of a single algorithm in solving the class scheduling problem.

The adaptive dynamic programming algorithm has the shortest running time when the fitness value reaches 250. Since the adaptive dynamic programming algorithm uses a dynamic programming algorithm with multi-stage decision making in the initial stage of the search, a large amount of search time can be saved in the initial stage compared to the ACO algorithm. Using the ACO algorithm in the later stages of the search avoids consuming a lot of time on useless iterations compared to the genetic algorithm, while the ACO algorithm also has better parallel search capability. Therefore, the adaptive dynamic programming algorithm for solving the scheduling problem is superior to the ant colony algorithm and the genetic algorithm in terms of operational efficiency and rationality of the class schedule.

Table 3. The convergence of the algorithm is compared

Performance comparison	Algorithm	Iteration number	Fitness
The performance comparison of the given operation time algorithm(The scheduling unit is 526,time=150 s)	Genetic algorithm	105.815	235.514
	Ant colony algorithm	98.634	249.559
	Adaptive dynamic planning algorithm	123.258	281.163
Performance comparison	Algorithm	Iteration number	Running time
The proposed fitness value algorithm is compared(The scheduling unit is 526,fitness=250)	Genetic algorithm	147.69	251.245
	Ant colony algorithm	101.019	209.819
	Adaptive dynamic planning algorithm	94.818	160.904

3.2. Application of Adaptive Dynamic Programming Algorithm in Teaching Resource Balancing

3.2.1. Experiment on balanced division of resources for management of disciplinary types. Scientific and effective teaching management methods are an important factor in ensuring teaching quality. In the teaching process, it is necessary to focus on optimizing the dynamic optimization problem of teaching quality management in colleges and universities. The dynamic planning algorithm is based on the optimization of the process of online and offline integration of teaching management in colleges and universities, and the optimization is mainly carried out from the point of view of two-way dynamic programming in this method.

In order to test the accuracy of the adaptive dynamic programming algorithm, 25 courses of the same type of data were used for sequential numbering test, in which a group of 5, divided into (subject 1~subject 5) 5 different educational disciplines to test and take the median. The results of the algorithm grouping optimization accuracy comparison are shown in Figure 4. The results of the analysis show that by introducing the optimization measures of the adaptive dynamic programming algorithm, the accuracy of the college teaching management has been improved in all the discipline groups. The optimized algorithm demonstrated higher accuracy than the pre-optimization in all subject groups tested, with subject 1 showing the most significant accuracy increase of 17.4% and subject 4 showing the lowest increase of 4.0%. The increase in the remaining three disciplines ranged from 9.6% to 12.1%. It can be seen that the introduction of the adaptive dynamic planning algorithm resulted in a significant increase in the balancing effect of discipline type management resources.

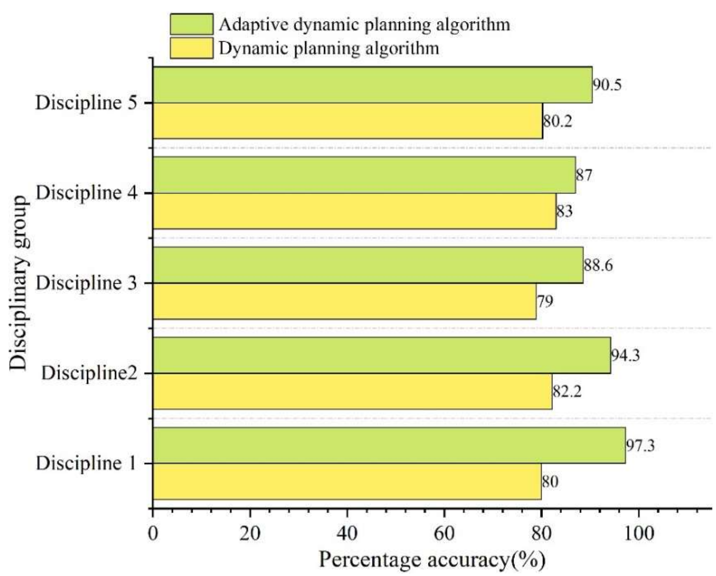

Fig. 4. Algorithm optimization accuracy comparison

To further demonstrate the characteristics of the algorithm after optimization, Fig. 5 shows the comparison curve plots of the overall pre-optimization and post-optimization at various stages for 25 datasets accompanied by data volume and time variations. As can be seen from the figure, the adaptive planning algorithm takes longer time than the planning algorithm with the same test data nodes. This shows that the computational performance of the algorithm is significantly improved after optimization.

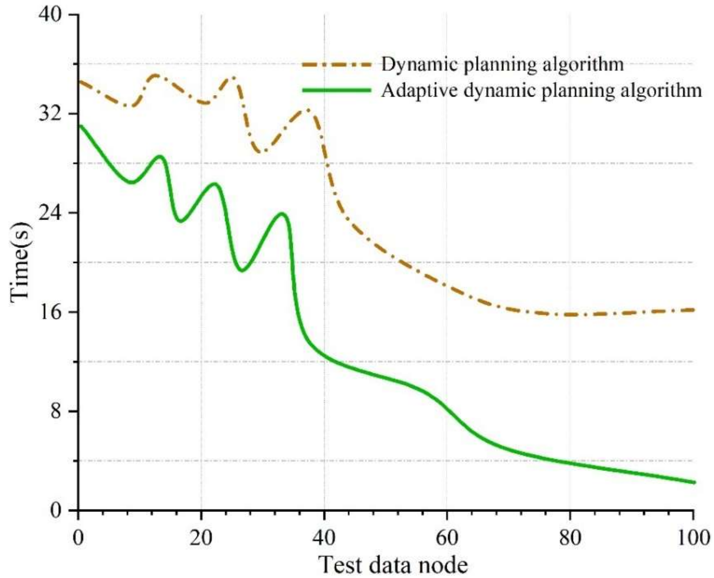

Fig. 5. Algorithm optimization curve

3.2.2. Multidisciplinary resource balance merger experiment. After passing the algorithmic accuracy measure described above, the overall instructional management merger experiment is to be conducted. The minimum score problem of the sports event merger test is a very typical test of the practical application of decision monotonicity, which meets the requirements of quadrilateral inequality and monotonicity in the process of state transfer, which fundamentally reduces the number of states involved in the process of state transfer, and achieves the purpose of reducing the complexity of the problem in the time required to solve the problem on the basis of the qualitative purpose. This method is often used for course scheduling optimization, in which multiple programs need to be combined in order to minimize costs or maximize benefits. The study selected a number of datasets divided into different types of courses and optimized the data by merging and combining them so that the total score of each group meets the minimum requirements, and the results of the experimental data for course scheduling optimization are shown in Table 4. It can be seen through the test that when the amount of data is small, the difference between the time required before and after optimization is not significant. Along with the increase in the amount of data, the effect of optimization is significantly enhanced and the time required is significantly shortened, especially when the course schedule is intensive. At this time, the efficiency of test groups 3, 4 and 5 is increased by 50%, 66.67% and 83.33% respectively, and the time required after optimization is significantly reduced.

Table 4. The course arrangement optimizes the test results

Test content	Test group				
	1	2	3	4	5
Test data number	1, 13, 4, 2, 7	2, 6, 8, 3, 4, 1, 12	7, 13, 14, 8, 9, 12, 4, 2, 5, 6, 3, 0	13, 11, 12, 15, 19, 16, 13, 11, 18, 9, 7, 2, 17, 8, 14	16, 7, 1, 9, 15, 11, 25, 22, 24, 21, 18, 19, 16, 20, 3, 24, 17, 12, 23, 10, 20, 8
Output result	45	68	194	1206	3754
Optimize the pre-run time(s)	0.06	0.06	0.06	0.06	0.06
Optimized running time(s)	0.06	0.06	0.03	0.04	0.05
Efficiency added(%)	0.00	0.00	50	66.67	83.33
Effect	No obvious change		Obvious		

3.2.3. Teaching resource planning test results. Another test was conducted to solve the monotonicity of inequalities and decisions. The post office test is one of the current popular methods which functions to maximize the allocation of teaching resources, and this test can be used to derive reasonable teacher working hours, design the most efficient scheduling plan, arrange teachers according to the number of students, and so on. The test selected 25 test data for post office problems and the results of the teaching resource planning test are shown in Table 5. The results show that the optimized algorithm is fully applicable. When the data is relatively small, the time before and after optimization is still not much different. And along with the increase in the amount of data, the alternating increase in the number of villages involved and the number of post offices to be established, the time of the optimization project is obviously reduced, while the efficiency of the optimization has been significantly improved. Especially, the effect of the fifth group of tests is obvious, with a 100% improvement.

Table 5. Teaching resource planning test results

Test content		Test group				
		1	2	3	4	5
Output result	Village number	1	2	8	28	27
	Post number	4	9	13	282	292
Output result		8	15	57	18934	24512
Running time(s)	Preoptimize	0.08	0.07	0.07	0.15	0.16
	After optimization	0.06	0.04	0.04	0.08	0.08
Efficiency added (%)		33.33	75.00	75.00	87.50	100.00
Effect		General	More obvious	More obvious	More obvious	Obvious

4. Conclusion

This paper introduces the Bellman optimization principle on the basis of the dynamic planning algorithm and constructs an adaptive dynamic planning algorithm. The balance of resources and the application effect of the algorithm in the online and offline integrated teaching of higher education are analyzed.

1) The adaptive dynamic planning algorithm can be applied to resource balancing problems such as class scheduling in the process of teaching, and the quality of its class schedule is better. In addition, with the increase of class scheduling units, the running time of the algorithm is more, but compared with other algorithms, the computing time is the shortest, and the adaptive ability is stronger, the advantage is more obvious. Especially in terms of adaptability, the adaptability value of this paper's algorithm (281.163) is 31.60 and 45.65 higher than that of ant colony algorithm and genetic algorithm, respectively, which once again verifies that this paper's algorithm is better in terms of operational efficiency and schedule rationality when solving the scheduling problem.

2) After the introduction of the optimization measures of adaptive dynamic programming algorithm, the accuracy of college teaching management is improved by 4.0%-17.4% in all subject groups. It can be seen that the optimized algorithm demonstrated higher accuracy than the pre-optimization in all subject groups tested. The optimization effect of the model is more obvious when the amount of data processed by the model is larger.

References

[1] Yen, S. C., Lo, Y., Lee, A., & Enriquez, J. (2018). Learning online, offline, and in-between: comparing student academic outcomes and course satisfaction in face-to-face, online, and blended teaching modalities. Education and Information Technologies, 23, 2141-2153.

[2] Sari, I. K. (2021). Blended learning sebagai alternatif model pembelajaran inovatif di masa post-pandemi di sekolah dasar. Jurnal Basicedu, 5(4), 2156-2163.

[3] Abdullah, W. (2018). Model blended learning dalam meningkatkan efektifitas pembelajaran. Fikrotuna: Jurnal Pendidikan dan Manajemen Islam, 7(01), 855-866.

[4] Guo, Y., Liu, H., Hao, A., Liu, S., Zhang, X., & Liu, H. (2022). Blended learning model via small private online course improves active learning and academic performance of embryology. Clinical Anatomy, 35(2), 211-221.

[5] Dangwal, K. L. (2017). Blended learning: An innovative approach. Universal Journal of Educational Research, 5(1), 129-136.

[6] Suartama, I. K., Setyosari, P., & Ulfa, S. (2019). Development of an instructional design model for mobile blended learning in higher education. International Journal of Emerging Technologies in Learning, 14(16).

[7] Islam, S., Baharun, H., Muali, C., Ghufron, M. I., el Iq Bali, M., Wijaya, M., & Marzuki, I. (2018, November). To boost students' motivation and achievement through blended learning. In Journal of Physics: Conference Series (Vol. 1114, No. 1, p. 012046). IOP Publishing.

[8] Meulenbroeks, R. (2020). Suddenly fully online: A case study of a blended university course moving online during the Covid-19 pandemic. Heliyon, 6(12).

[9] Dakhi, O., JAMA, J., & IRFAN, D. (2020). Blended learning: a 21st century learning model at college. International Journal Of Multi Science, 1(08), 50-65.

[10] Hart, C. M., Berger, D., Jacob, B., Loeb, S., & Hill, M. (2019). Online learning, offline outcomes: Online course taking and high school student performance. Aera Open, 5(1), 2332858419832852.

[11] Xiao, J., Sun-Lin, H. Z., Lin, T. H., Li, M., Pan, Z., & Cheng, H. C. (2020). What makes learners a good fit for hybrid learning? Learning competences as predictors of experience and satisfaction in hybrid learning space. British Journal of Educational Technology, 51(4), 1203-1219.

[12] Saboowala, R., & Manghirmalani Mishra, P. (2021). Readiness of in-service teachers toward a blended learning approach as a learning pedagogy in the post-COVID-19 Era. Journal of Educational Technology Systems, 50(1), 9-23.

[13] Xu, Z., Yuan, H., & Liu, Q. (2020). Student performance prediction based on blended learning. IEEE Transactions on Education, 64(1), 66-73.

[14] Yustina, Y., Syafii, W., & Vebrianto, R. (2020). The effects of blended learning and project-based learning on pre-service biology teachersâ€™ creative thinking through online learning in the covid-19 pandemic. Jurnal Pendidikan IPA Indonesia, 9(3), 408-420.

[15] Idris, H. (2018). Pembelajaran model blended learning. Jurnal Ilmiah Iqra', 5(1).

[16] Dwiyogo, W. D. (2018). Developing a blended learning-based method for problem-solving in capability learning. Turkish Online Journal of Educational Technology-TOJET, 17(1), 51-61.

[17] Jin, Y. Q., Lin, C. L., Zhao, Q., Yu, S. W., & Su, Y. S. (2021). A study on traditional teaching method transferring to E-learning under the COVID-19 pandemic: From Chinese students' perspectives. Frontiers in Psychology, 12, 632787.

[18] Jalinus, N. (2021). Developing blended learning model in vocational education based on 21st century integrated learning and industrial revolution 4.0. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(8), 1239-1254.

[19] Haningsih, S., & Rohmi, P. (2022). The pattern of hybrid learning to maintain learning effectiveness at the higher education level post-COVID-19 pandemic. European Journal of Educational Research, 11(1), 243-257.

[20] Nortvig, A. M., Petersen, A. K., & Balle, S. H. (2018). A literature review of the factors influencing e-learning and blended learning in relation to learning outcome, student satisfaction and engagement. Electronic Journal of E-learning, 16(1), pp46-55.

[21] Singh, J., Steele, K., & Singh, L. (2021). Combining the best of online and face-to-face learning: Hybrid and blended learning approach for COVID-19, post vaccine, & post-pandemic world. Journal of Educational Technology Systems, 50(2), 140-171.

[22] Ashraf, M. A., Yang, M., Zhang, Y., Denden, M., Tlili, A., Liu, J., ... & Burgos, D. (2021). A systematic review of systematic reviews on blended learning: Trends, gaps and future directions. Psychology Research and Behavior Management, 1525-1541.

[23] Siripongdee, K., Pımdee, P., & Tuntıwongwanıch, S. (2020). A blended learning model with IoT-based technology: effectively used when the COVID-19 pandemic?. Journal for the Education of Gifted Young Scientists, 8(2), 905-917.

[24] Hediansah, D., & Surjono, H. (2020). Hybrid learning development to improve teacher learning management. JKTP: Jurnal Kajian Teknologi Pendidikan, 3(1), 1-9.

[25] Gu Qinghua,Wang Xueqing,Wang Dan & Liu Di. (2024). A Kriging-assisted multi-stage evolutionary algorithm for expensive many-objective optimization problems. Structural and Multidisciplinary Optimization(4),

[26] Cheachen Hanan Ali & Joad Ghufran Khalil. (2021). Using BAB to find Optimal Solution for Multi-Objective Function with Setup Time. Journal of Physics: Conference Series(3),

[27] Xiaojuan Li. (2025). Dynamic programming principle for stochastic optimal control problem under degenerate [formula omitted]-expectation. Systems & Control Letters105988-105988.

[28] Qingpeng Liang, Jiangping Hu, Linying Xiang, KaiboShi & YanzhiWu. (2024). Adaptive dynamic programming and distributionally robust optimal control of linear stochastic system using the Wasserstein metric. International Journal of Adaptive Control and Signal Processing(8),2810-2832.