An Algorithmic Model Building Scheme Based on Dynamic Programming Algorithms

Zhiyong Li, Weiling Chen
National Demonstration Center for Experimental Economics and Management Education on Guangdong University of Finance & Economics GuangZhou 510320, China

Abstract. Dynamic programming algorithm is a basic tool for solving multi-level decision-making problems, which is widely used in social economy, engineering technology and optimal control. Aiming at the incommensurability of multi-objective programming indices, the relative membership degree matrix of allocation schemes for quantitative and qualitative indices is established, and the analysis method of multi-objective and multi-stage dynamic programming problems is given by using vector and matrix unification. The decision maker determines the decision coefficients of each constraint indicator according to the specific problem and performs weighting processing. The weights of different state variables for different constraint indicators are integrated, the comprehensive weights of state variables are solved, and the original static constraint indicators are transformed into dynamic variables that can influence the decision results. The research shows that the algorithm transforms the multi-stage decision problem into a single-stage decision problem, and solves the optimal value of the expected value through the mixture of random expectation models in a single stage. As long as it is used properly, dynamic planning of this traditional method can be a better way to solve complex problems.

1. Introduction (Heading 1)
Dynamic programming was first used to solve some optimal problem. This problem refers to the existence of such activities, which can be divided into several interrelated stages, in which decisions need to be made at each stage [1]. The process of work has obvious sequence, so a dynamic programming model for solving the problem can be established. Iterative dynamic programming can discretize the continuous system from time and space perspectives [2]. All state variables corresponding to different time periods are discretized into a group of grids, and different feasible values of control variables are applied to calculate in each time period. Finally, a complete control strategy is found to optimize the performance index of the non-linear system. The reverse is also true. That is, the state is ineffective [3-5]. The basic principle of the dynamic programming method is that the state should have no aftereffects, but when the state is defined properly, the problem can be changed to make it have no aftereffect. Therefore, it can be solved by the dynamic programming method, and the minimum or maximum of the objective function is applicable [6]. The number of iterations of the algorithm increases. The predicted task completion time is closer to the actual completion time, so that the task assignment is effectively and reasonably performed, the task completion time is greatly shortened, and the task assignment efficiency is improved [7].

Dynamic programming vector parameterization is a direct method for solving optimal control problems [8]. It transforms infinite-dimensional optimal control problems into finite-dimensional optimization problems. In the early parameterization methods, the optimal control problem is discretized at a finite time point, and the control variables and state variables are taken as independent solving
variables. The Hungarian method requires that every element in the coefficient matrix be non-negative [9-11]. Therefore, when the objective function is maximized, the method of changing the objective function coefficient is usually used. It is easy to not converge or fall into local optimal solution. In addition, the training set needed is huge [12]. The most important thing is that the training time of samples is long and the time complexity is high [13]. Therefore, in the cloud environment, a task allocation method for reasonably assigning tasks is found. The basic idea of the dynamic programming algorithm is to decompose the problem to be solved into several sub-problems, first solve the sub-problems, and then obtain the solutions of the original problems from the solutions of these sub-problems [14]. For the optimal solution of this compression error function, it is easier to determine the global optimal solution, which has been applied in many fields. The dynamic programming method is a branch of operations research and is a mathematical method to optimize the multi-stage decision process [15-17].

2. Materials And Methods

By solving the problem in stages, decisions will be made at each stage, and the combination of these decisions becomes the strategy of the problem [18]. In multi-stage decision-making, the strategy has a certain range [19]. The optimal strategy is the one that achieves the optimal effect for all the allowable policy sets. When dynamic programming is applied to solve the scheduling problem, the location of a workpiece can not be determined in the middle stage [20]. It is calculated from the largest feasible set after all feasible sets have been computed. An optimal sub-policy can be similarly defined for a sub-process. According to the inefficiency of the state, when determining the optimal sub-strategy of the sub-process (referred to as the optimal sub-strategy), the decision before the segment has no effect on this, and the initial state and the decisions on the previous segment only affect the state [21]. That is, any level of the optimal decision is taken as the initial level, and the state of this level is taken as the initial, so that the remaining decisions up to the final level will be the most for the multi-level process starting from this level. Excellent decision making [22].

Like dynamic programming, stochastic dynamic programming mainly uses the actual situation that can reflect the phenomena in the real world to be expressed as the abstract direction in mathematics. When optimizing the system, the parameters of stochastic dynamic programming algorithm are shown in Table 1 and Figure 1.

| Tab.1 Parameters of stochastic dynamic programming algorithm |
|-------------------------------------------------------------|
|                | Random | Feedback |
| State variable  | 1.10   | 2.60     |
| Decision variables | 0.52   | 0.19     |
Fig. 1 Parameters of stochastic dynamic programming algorithm

The calculation method of dynamic programming determines that this method can provide a large amount of information. When the external conditions change, the original calculation results can be used to quickly get a new optimal strategy. Each of these sub-problems is a much simpler optimization problem than the original one. The solution of each sub-problem is only based on the results of its next sub-problem, so that the optimal solution of the original problem can be obtained once. As a general principle, when the sum of processing time and installation time is greater than the sum of weight, the reverse dynamic programming method is better. On the contrary, the forward dynamic programming method is better. It also obtains the two-point nature of the optimal solution of the problem, which can improve the computational efficiency of the algorithm, help to understand the problem and solve more complex models. Therefore, for a deterministic discrete segment decision process, according to the optimization principle, the basic equations of its dynamic programming can be given, the random variables can be analyzed to judge its independence, the relationship between state variables and decision variables can be found, and the target can be established. The function and the corresponding recursive equation [23-25].

Through the equation, we can establish some corresponding functional relationship, also known as the recursive relationship of dynamic programming. Iterative dynamic programming algorithm is applied to solve this problem. The penalty factor was 0.03, and the time period was divided into nine sections. The state grid point is selected as N = 0.6, the control grid point R = 16, the shrinkage factor r = 0.19, the initial control area is taken as 0.1, and 20 iterations are used. The control and status curves are shown in Figure 2 and Figure 3 below:
The core of dynamic programming is Bellman optimality principle, which first transforms a multi-level decision problem into a series of single-level decision problems, and then from the last state to the initial state. That is to say, when solving multi-level decision-making problems, we should start from the end to the beginning, and recurse backward. The state of the stage can be described by some characteristic of the stage, and the process of decision-making can be explained by the evolution of the state of each stage. A state of a stage can also be considered as taking a "value" of the state. The variables describing the state are called state variables; the dynamic programming model is applicable to the assignment problem of minimizing or maximizing the objective function of the optimal assignment problem. Each state can have a variety of strategies, which correspond to different configuration modes and different number of observable nodes, among which the maximum number of observable nodes is called the optimal value of the state. Given the state of a certain phase, and the decision corresponding to this state is selected, the state of the next segment can be completely determined, which is the certainty of the decision process. The decision made in the current state, due to the influence of the random variable factor, the next state cannot be completely determined, so that the new state also exhibits a random nature. Therefore, how to better apply the staged idea to deal with random variables in stochastic dynamic programming. Finally, we present the algorithm for solving the dynamic programming model of the standard assignment problem.

Energy diffusion exists in single target detection, which has a greater impact on the performance of the algorithm in multi-target detection. When the target is not intersected or adjacent, the impact is not so great, but once there is intersection or adjacency between the detected targets, the extreme value method is difficult to detect and track the target. The uncontrollable factors are not transferred by human subjective will. According to no aftereffect, the income in the current stage will only depend on the current state, but has nothing to do with the past state and decision-making. The state can be a value or a combination of multiple values. Secondly, for the dynamic programming model of deterministic multi-stage decision-making process, when applying the basic equations, not only the nature of the optimal decision described by the inference is used, but also the necessary and sufficient conditions of the theorem are used. Theorem is the basic theorem of dynamic programming. The parallel algorithm based on variable splitting decomposes a large-scale optimization problem into a series of small and easy-to-solve sub-problems. It is one of the basic methods for designing parallel algorithms. Both search and heuristic algorithms are very effective. In order to know the approximate degree of the approximate optimal order and the optimal order, the analysis error bound is indispensable to use such a method. In the descending algorithm using the function gradient, a lot of work is to calculate the function value and function. Numerical, reasonable parallel calculation of function values and gradient values can greatly improve computational efficiency.

3. Result Analysis and Discussion
As the complexity of the system increases and the role of instability factors becomes more and more obvious, there are often such variables, which are distributed in different periods and are independent of each other, and there are also such variables. The key of using dynamic programming method to study
scheduling problem is to study the nature of the optimal solution of the problem. As long as the nature of the optimal solution to be solved is found, it is possible to find a more ingenious method of using dynamic programming to solve the problem, while establishing the dynamic programming model. In addition to properly dividing the problem into several stages, it must be carried out according to the following basic requirements, which are also the basic conditions for constituting the basic model of dynamic programming as shown in Table 2 and Figure 4. As with state variables, it is often also possible to use a number, a set of numbers, or even a vector as a decision variable to describe the decision. The time period is the value of the time, the time period is different from the random variables, and the independent is subject to different distributions. The strategy is for decision-making, and the strategy can be said to be a set of decision-making sequences at various stages.

Tab.2 Basic Theorems of Conditions for Constructing Dynamic Programming Models

| Selection of state variables | State | Transfer |
|-----------------------------|-------|---------|
| Determining Decision Variables | 6.20  | 5.90    |

Fig.4 Basic Theorems of Conditions for Constructing Dynamic Programming Models

Among the common index functions, the index function is the sum of the index functions of each segment:

\[ V_k,n = \sum_{j=2}^{n} v_j(x_j,u_j) \]  \hspace{1cm} (1)

Index function: \( v_j(x_j,u_j) \)

The optimal solution of the whole process includes the optimal strategy and the corresponding optimal index function. For sequential solution, it is assumed that the definitions of stage ordinal number \( k \) and state variable \( KX \) remain unchanged, state transition variance:

\[ X_k = T^{-1}(Xk + 1,uk) \]  \hspace{1cm} (2)

According to the division of each segment and the law of evolution between segments, write the state transition equation:

\[ X_{k+1} = T(X_k, U_k) \]  \hspace{1cm} (3)
Decision variables: UK  
state variable: $X_k$

The basic idea of the function space iteration method is to take the number of segments (steps) as variables, first seek the optimal strategy under different segments, and then select the best from these optimal solutions, so as to determine the optimal number of segments at the same time. The steps are as follows:

$$f_{1(i)} = C_{in}$$
$$f_{1(N)} = 0$$ (4)

Selection of an initial function: $f_{i(i)}$

In the dynamic programming, it is assumed that the target to be detected has special frame data, and the target motion range is a dimension plane, then the state variable of the target is as shown in the equation:

$$x(k) = [x(k), y(k)]$$ (5)

X direction in the plane: $x(k)$
y direction in the plane: $y(k)$

The statistical properties of each resolution unit are represented by:

$$Z_{ij}(k) = \sum A(k) + W_{ij}(K)$$ (6)

Zero-Mean Complex Gauss White Noise with Independent and Identical Distribution: $W_{ij}$
Uniform Distribution of Complex Random Variables: $A(k)$

The dynamic programming algorithm actually traverses all possible state sequences, and finally finds the only optimal state sequence, taking this sequence as the real trajectory of the target:

$$I(X_k) = \text{MAX} (X_{k-1})$$ (7)

Valid state: $x_{k-1}$

Find all final states that meet the following criteria:

$$x_{k} = x_{k}; I(x_{k}) > VT$$ (8)

Termination state: $X_K$
Threshold: $VT$

The detection probability refers to the difference between the position of at least one state in all states and the true position of the target in two resolution units, and the probability that the value function of the final state exceeds the detection threshold:

$$i_t = r^* + \pi^*_t + \alpha(\pi_t - \pi^*_t) + \beta y_t$$ (9)

In the detection of the target, the probability that the maximum noise state value function exceeds the detection threshold:

$$I = \left[ \sum_{j=1}^{p} \omega_j^m y_j^m \right]^{1/m}$$ (10)

Detection probability: $it$  
False alarm probability: $Ii$

The number of operations performed by an algorithm is obviously related to the size of the example. The size of the example is measured by the size of the input. The complexity of the algorithm is a function of the size of the input. Nowadays, scientists have a consensus that the algorithm for solving a problem is effective only when its complexity function is a polynomial function of input scale. The selection of state variables depends on specific problems, and there are different selection methods, but all of them must satisfy an important property: the follow-up process (called post-sub-process, referred to as sub-process) starting from a certain stage state cannot be affected by the previous evolution process. Dynamic programming algorithm cannot be directly applied to target state, so the premise of realizing
dynamic programming algorithm is to discretize the target state space, and at each stage of dynamic programming, the idea of combining differential evolution with neural network is applied to solve stochastic optimization problems. First, the initial sample is randomly generated. The initial sample is optimized by differential evolution algorithm to determine the input and output data, and the continuous state is changed into a discrete state. This is done at each stage until all phases are over, at which point the target space is split into different units. Optimal strategy and optimal target value training, while providing the optimal value for the previous phase of the current phase.

Attach different additional values to each point of the target's current frame moving into the search window. So the idea of direction weighting is simply to assign different values to points with different possibilities and smaller values to points with less possibilities. Points with higher probability are given larger weights. When dynamic programming is applied to solve scheduling problems, it is not in the middle stage that the location of a workpiece can be determined, but after all feasible sets have been calculated. Specifically, the distance between the optimal individuals of each population is compared at each time of evolution. If the distance between them is too small or less than a predetermined value, only those with good performance will be retained. Poorly performing populations will be marked and the marked population will be reinitialized before the next cross mutation. Starting from the largest feasible set (ie, the totality of all the artifacts), trace all the feasible sets back. It is the essential difference between the state of dynamic programming and the state of the system that usually describes it. Since the state of each phase of the phase has been recorded in the optimal combination state, it is convenient to determine the configuration position of the previous phase. When determining the state, the state must contain sufficient information given by the problem to make it satisfy the latter. Effectiveness.

Dynamic programming does not have a standard expression, but a mathematical expression for a specific problem, so there is no uniform processing format. It must be based on the characteristics of the problem itself, using flexible mathematical skills to deal with. The final solution of global optimization is the synthesis of local optimal solutions. Because of the natural parallelism, a large-scale optimization problem can be decomposed into several smaller-scale problems. In the process of iteration from an initial value, smaller-scale problems can be solved in parallel at different nodes with the same calculation steps. Therefore, for each iteration, the strategy iteration method is more complex than the function iteration method, and the amount of calculation is also large. However, the number of iterations of the strategy iteration method tends to be less than the iteration of the function iteration method, that is, the convergence is faster. Especially when there is more experience with practical problems, you can choose an initial strategy that is better than the optimal strategy, and the convergence is faster. Furthermore, since optimization scheduling is inevitably considered to be moving toward an optimal solution, it is often overlooked that some complicated factors and the relationship between these factors can express these complex factors and their relationships. A difficult problem, so it can be expressed in a more complex correspondence, and it can be comprehensively grasped as a whole. Specifically, the strength of these targets is first calculated, and the intensity is compared. The maximum intensity is regarded as the target, and the rest are treated as noise. Finally, it performs single target detection and tracking, and removes the value of this target in the measurement data.

4. Conclusion
In this paper, the algorithm model of dynamic programming algorithm is studied. The dynamic programming calculation is realized. The dimension is reduced, the calculation amount is reduced and the calculation efficiency is improved by the pre-configuration criterion, the symmetry criterion and the survival of the fittest criterion. The numerical results show that the algorithm can accurately determine the number and location of the global optimal allocation. The steps of solving this kind of problem by iterative dynamic programming are given. The new optimization algorithm, which can run on computer by means of software implementation of numerical iteration, solves the problem of decision-making optimization under the influence of different constraints. Differential evolution algorithm can find the optimal value by increasing the diversity of sub-populations and not sharing the peak value between sub-populations. By analyzing the nature of the optimal solution of the problem, the scope of the
iteration in the dynamic programming process can be reduced, thereby reducing the scope of the enumeration; selecting the appropriate state variables can simplify the problem and facilitate the identification of each state. Recursive relationship between. And according to the actual situation, it can make a more reasonable judgment on the constraint ability of each constraint index, and avoid the static thinking problem of the previous model on the constraint index. It is an effective tool to solve the problem that the feasible solution set is a discrete solution set objective function and constraint function non-micro optimization problem.

References

[1] Zolfpour-Arokhlo M, Selamat A, Hashim S Z M, et al. Modeling of route planning system based on Q value-based dynamic programming with multi-agent reinforcement learning algorithms[J]. Engineering Applications of Artificial Intelligence, 2014, 29(3):163-177.

[2] Fafianie S, Bodlaender H L, Nederlof J. Speeding Up Dynamic Programming with Representative Sets: An Experimental Evaluation of Algorithms for Steiner Tree on Tree Decompositions[C]// Springer US, 2015:636-660.

[3] Tang Y, You R, Kan H, et al. Cache-oblivious wavefront: improving parallelism of recursive dynamic programming algorithms without losing cache-efficiency[J]. ACM SIGPLAN Notices, 2015, 50(8):205-214.

[4] Ping S, Li-Ping W, Zhi-Qiang J, et al. Application of two multi-dimensional dynamic programming algorithms in optimization of cascade reservoirs operation[J]. Journal of Hydraulic Engineering, 2014, 45(11):1327-1335.

[5] Bahig H M, Khedr A Y. MonitTDPA: A tool for monitoring the tracing of dynamic programming algorithms[J]. Computer Applications in Engineering Education, 2017, 25(2):179-187.

[6] Rubensson E H, Rudberg E. Chunks and Tasks: A programming model for parallelization of dynamic algorithms[J]. Parallel Computing, 2014, 40(7):328-343.

[7] Grossi E, Lops M, Venturino L. A Novel Dynamic Programming Algorithm for Track-Before-Detect in Radar Systems[J]. IEEE Transactions on Signal Processing, 2013, 61(10):2608-2619.

[8] Liu D, Wang D, Yang X. An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs[J]. Information Sciences, 2013, 220(1):331-342.

[9] Kim Y, Ahn S J, Hwang P L, et al. Coordinated Control of a DG and Voltage Control Devices Using a Dynamic Programming Algorithm[J]. IEEE Transactions on Power Systems, 2013, 28(1):42-51.

[10] Liu D, Li H, Wang D. Neural-network-based zero-sum game for discrete-time nonlinear systems via iterative adaptive dynamic programming algorithm[J]. Neurocomputing, 2013, 110(Complete):92-100.

[11] Boaro M, Fuselli D, Angelis F D, et al. Adaptive Dynamic Programming Algorithm for Renewable Energy Scheduling and Battery Management[J]. Cognitive Computation, 2013, 5(2):264-277.

[12] Russo M, Sforza A, Sterle C. An exact dynamic programming algorithm for large-scale unconstrained two-dimensional guillotine cutting problems[J]. Computers & Operations Research, 2014, 50:97-114.

[13] Wei Q, Liu D. Stable iterative adaptive dynamic programming algorithm with approximation errors for discrete-time nonlinear systems[J]. Neural Computing & Applications, 2014, 24(6):1355-1367.

[14] Jiang D R, Powell W B. An Approximate Dynamic Programming Algorithm for Monotone Value Functions[J]. Operations Research, 2014, 63(6):págs. 1489-1511.

[15] Zhang Z, Zhang S, Wang Y, et al. Use of parallel deterministic dynamic programming and hierarchical adaptive genetic algorithm for reservoir operation optimization[J]. Computers & Industrial Engineering, 2013, 65(2):310-321.

[16] Feinberg E A, Huang J. The value iteration algorithm is not strongly polynomial for discounted dynamic programming[J]. Operations Research Letters, 2014, 42(2):130-131.
[17] Yi W, Kong L, Yang J. Thresholding Process Based Dynamic Programming Track-Before-Detect Algorithm[J]. IEICE Transactions on Communications, 2013, E96.B(1):291-300.

[18] Rong A, Figueira, José Rui. A reduction dynamic programming algorithm for the bi-objective integer knapsack problem[J]. European Journal of Operational Research, 2013, 231(2):299-313.

[19] Chebil K, Khemakhem M. A dynamic programming algorithm for the Knapsack Problem with Setup[J]. Computers & Operations Research, 2015, 64(C):40-50.

[20] Rong A, Figueira, José Rui. Dynamic programming algorithms for the bi-objective integer knapsack problem[J]. European Journal of Operational Research, 2014, 236(1):85-99.

[21] 

[22] Xu J, Zeng Z, Han B, et al. A dynamic programming-based particle swarm optimization algorithm for an inventory management problem under uncertainty[J]. Engineering Optimization, 2013, 45(7):851-880.

[23] Qin X, Wang S, Zheng D. A Dynamic Programming Track-Before-Detect Algorithm Based on Local Linearization for Non-Gaussian Clutter Background[J]. Chinese Journal of Electronics, 2016, 25(3):583-590.

[24] Itzhaky S, Singh R, Solar-Lezama A, et al. Deriving divide-and-conquer dynamic programming algorithms using solver-aided transformations[J]. ACM SIGPLAN Notices, 2016, 51(10):145-164.

[25] Elshqeirat B, Soh S, Rai S, et al. A Dynamic Programming Algorithm for Reliable Network Design[J]. IEEE Transactions on Reliability, 2014, 63(2):443-454.

[26] Salari M, Reihane M, Sabbagh M S. Combining ant colony optimization algorithm and dynamic programming technique for solving the covering salesman problem[J]. Computers & Industrial Engineering, 2015, 83:244-251