Dynamic Programming

Introduction & Water Allocation as Sequential Process

Introduction

Dynamic Programming (DP) proposed by Bellman, transforms a sequential or multistage decision problem that may contain many interrelated decision variables into a series of single-stage problems, each containing only one or a few variables.

In other words, the dynamic programming technique decomposes an N-decision problem into a sequence of N separate, but interrelated, single-decision sub-problems.

Decomposition is very useful in solving large, complex problems by decomposing a problem into a series of smaller sub-problems and then combining the solutions of the smaller problems to obtain the solution of the entire model composition.

The reason for using decomposition is to solve a problem more efficiently which can lead to significant computational savings.

As a rule of thumb, computations increase exponentially with the number of variables, but linearly with the number of sub-problems.

Water Allocation Problem

Consider a quantity of water Q that can be allocated to three water users, denoted by index j =1, 2, and 3. The problem is to determine the allocation x_j to each user j that maximizes the total net benefits.

Let the gross benefit resulting from an allocation of x_j to user j is approximated by the function

$a_j[1-\exp(-b_jx_j)]$ where a_j and b_j are known positive constants.

Let the costs be defined by the concave function $c_jx_j^{d_j}$, where c_j and d_j are known positive constants and $d_j<1$.

Water Allocation Problem

$$\text{Maximize} \quad \sum_{j=1}^{3}\left\{ a_j[1-\exp(-b_jx_j)] - c_jx_j^{d_j} \right\}$$

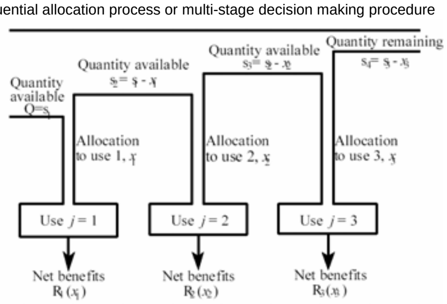
$$\text{Subject to} \quad \sum_{j=1}^{3} x_j \le Q$$

$$x_j \ge 0$$

This problem cannot be solved by LP or Lagrange multipliers method but readily solvable by DP

DP - Water Allocation Problem

First step in DP is to structure the allocation problem as a sequential allocation process or multi-stage decision making procedure



DP - Water Allocation Problem

Allocation to each user is considered a decision ***stage*** in a sequence of decisions

When a portion x_j of the total water supply Q is allocated at stage j, this results in net benefits as

$$R_j(x_j) = a_j[1-\exp(-b_jx_j)] - c_jx_j^{d_j}$$

A state variable s_j is defined as the amount of water available to the remaining (4 - j) users or stages.

Finally, a state transformation function $s_{j+1} = s_j - x_j$ defines the state in the next stage as a function of the current state and the current allocation or decision

DP - Recursive Equations

Allocation Problem restated as (with 3 Decision variables)

$$f_1(Q) = \underset{x_1 \,\&\, 0 \le x_1 \le Q}{maximum} \left\{ R_1(x_1) + \underset{x_2 \,\&\, 0 \le x_2 \le Q - x_1 - s_2}{maximum} \left[R_2(x_2) + \underset{x_3 \,\&\, 0 \le x_3 \le s_2 - x_2 - s_3}{maximum} R_3(x_3) \right] \right\}$$

Transform it into 3 separate problems each with only one decision variable

$$f_1(Q) = \underset{x_1 + x_2 + x_3 \le Q \,\&\, x_1, x_2, x_3 \ge 0}{maximum} \left[R_1(x_1) + R_2(x_2) + R_3(x_3) \right]$$

Let function $f_3(s_3)$ equal the maximum net benefits derived from use 3 given a quantity s_3 available for allocation to that use.

Hence for various discrete values of s_3 between 0 and Q, one can determine the value of $f_3(s_3)$ as $f_3(S_3) = \underset{x_3 \& 0 \le x_3 \le S_3}{maximum} R_3(x_3)$

7

DP - Recursive Equations

Since $s_3 = s_2 - x_2$, equation, $f_1(Q)$, can be rewritten in terms of only x_1, x_2, and s_2:

$$f_1(Q) = \underset{x_1 \,\&\, 0 \le x_1 \le Q}{maximum} \left[R_1(x_1) + \underset{x_2 \,\&\, 0 \le x_2 \le s_2}{maximum} \left[R_2(x_2) + f_3(s_2 - x_2) \right] \right]$$

Now let the function $f_2(s_2)$ equal the maximum net benefits derived from uses 2 and 3 given a quantity s_2 to allocate to those uses. Thus for various discrete values of s_2 between 0 and Q, one can determine the value of $f_2(s_2)$ where

$$f_2(s_2) = \underset{x_2 \,\&\, 0 \le x_2 \le s_2}{maximum} \left[R_2(x_2) + f_3(s_2 - x_2) \right]$$

Finally, since $s_2 = Q - x_1$, equation $f_1(Q)$, can be written in terms of only x_1 and Q: $f_1(Q) = \underset{x_1 \,\&\, 0 \le x_1 \le Q}{maximum} \left[R_1(x_1) + f_2(Q - x_1) \right]$

8

DP - Recursive Equations

$$f_1(Q) = \underset{x_1 \,\&\, 0 \le x_1 \le Q}{maximum} \left[R_1(x_1) + f_2(Q - x_1) \right]$$

$$f_2(s_2) = \underset{x_2 \,\&\, 0 \le x_2 \le s_2}{maximum} \left[R_2(x_2) + f_3(s_2 - x_2) \right]$$

$$f_3(S_3) = \underset{x_3 \& 0 \le x_3 \le S_3}{maximum} R_3(x_3)$$

$f_1(Q)$ is the maximum net benefits achievable with a quantity of water Q to allocate to uses 1, 2 and 3. This cannot be solved without a knowledge of $f_2(s_2)$. Similarly, $f_2(s_2)$ cannot be solved without a knowledge of $f_3(s_3)$.
Fortunately, $f_3(s_3)$ can be found using above equation without reference to any other maximum net benefit function $f_j(s_j)$.
Once the value of $f_3(s_3)$ is determined, the value of $f_2(s_2)$ can be computed, which will allow determination of $f_1(Q)$, the quantity of interest.

9

DP - Recursive Equations

Recursive sets of equations are fundamental to dynamic programming.

It is sometimes easier and quicker to solve numerous single variable problems than a single multivariable problem.

Each recursive equation represents a stage at which a decision is required, hence the term "multistage decision-making procedure".

10

Discrete Dynamic Programming (DDP)

When the state variables or quantity of water available s_j at stage j and the decision variables or allocations x_j to use j are allowed to take on only a finite set of discrete values, the problem is a discrete dynamic programming problem (DDP).

The solution will always be a global maximum (or minimum) regardless of the concavity, convexity, or even the continuity of the functions $R_j(x_j)$.

Obviously, the smaller the difference or interval between each discrete value of each state and decision variable, the greater will be the mathematical accuracy of the solution when the x_j are actually continuous decision variables.

Solving discrete dynamic programming problems to find the value of the objective function, and also the values of the decision variables that maximize or minimize the objective function, is best done through the use of tables, one for each stage of the decision-making process.

11

DP - Water Allocation Problem - Example

For the previous problem, let $Q = 5$, and $a_j = 100, 50, 100$; $b_j = 0.1, 0.4, 0.2$; $c_j = 10, 10, 25$; and $d_j = 0.6, 0.8, 0.4$ for $j = 1, 2, 3$, respectively.

Values of Net Benefit Function, $R_j\ (x_j)$

x_i	$R_1(x_1)$	$R_2(x_2)$	$R_3(x_3)$
0	0	0	0
1	-0.5	6.5	-6.9
2	3.0	10.1	0.0
3	6.6	10.9	6.3
4	10.0	9.6	11.5
5	13.1	7.0	15.6

12

DP – Example (contd.)
Stage 1: Calculation of $f_3(s_3)$

$$f_3(s_3) = \underset{x_3\ \&\ 0 \le x_3 \le s_3}{maximum}\ [R_3(x_3)]$$

State s_3	$R_3(x_3)$								
	x_3:	0	1	2	3	4	5	$f_3(s_3)$	x_3^*
0		0						0	0
1		0	-6.9					0	0
2		0	-6.9	0				0	2
3		0	-6.9	0	6.3			6.3	3
4		0	-6.9	0	6.3	11.5		11.5	4
5		0	-6.9	0	6.3	11.5	15.6	15.6	5

DP – Example (contd.)
Stage 2: Calculation of $f_2(s_2)$

$$f_2(s_2) = \underset{x_2\ \&\ 0 \le x_2 \le s_2}{maximum}[R_2(x_2) + f_3(s_2 - x_2)]$$

State s_2	$R_2(x_2) + f_3(s_2 - x_2)$								
	x_2:	0	1	2	3	4	5	$f_2(s_2)$	x_2^*
0		0+0						0	0
1		0+0	6.5+0					6.5	1
2		0+0	6.5+0	10.1+0				10.1	2
3		0+6.3	6.5+0	10.1+0	10.9+0			10.9	3
4		0+11.5	6.5+6.3	10.1+0	10.9+0	9.6+0		12.8	1
5		0+15.6	6.5+11.5	10.1+6.3	10.9+0	9.6+0	7.0+0	18.0	1

DP – Example (contd.)
Stage 3: Calculation of $f_1(Q)$

$$f_1(Q) = \underset{x_1\ \&\ 0 \le x_1 \le Q}{maximum}[R_1(x_1) + f_2(Q - x_1)]$$

State $s_1 = Q$	$R_1(x_1) + f_2(Q - x_1)$								
	x_1:	0	1	2	3	4	5	$f_1(Q)$	x_1^*
5		0+18.0	-0.5+12.8	3.0+10.9	6.6+10.1	10.0+6.5	13.1+0.0	18.0	0

Keeping a record of the optimal allocation x_j^* associated with each state variable value makes it possible to backtrack through each successive table to find the optimal values of each decision variable.

From the Table, the maximum net benefits $f_1(Q)$ equal 18, and the allocation x_1^* that resulted in this maximum is 0.
Hence the optimal s_2, the quantity available to allocate to uses 2 and 3, is $Q - x_1 = 5 - 0 = 5$.

DP – Example (contd.)
Solution

From the Table for stage 2, the optimal allocation x_2^*, to user 2 given $s_2 = 5$, is 1.
Hence the optimal s_3, the quantity available to allocate to user 3, is $s_2 - x_2 = 5 - 1 = 4$.

From the Table for stage 1, the optimal allocation x_3^*, to user 3, given s_3=4, is 4.

The sum of the allocations, in this case, are equal to Q=5.

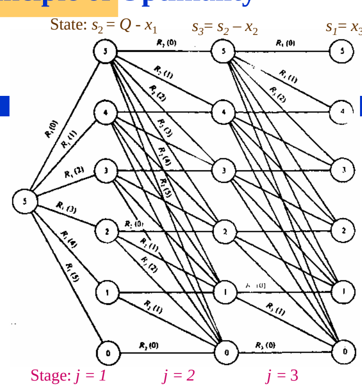
Principle of Optimality

To illustrate principal of Optimality upon which DP is base, let us represent the allocation problem by the network shown in figure.

Nodes of the network represent the states or quantities of water available to allocate to that and the following uses or stages.

Links represent the possible or feasible decisions given the state-variable value and stage. Associated with each link or allocation decision is a net benefit $R_j(x_j)$

Principle of Optimality



Principle of Optimality

The procedure just described is a process that moves backward through the network from stage 3 to stage 1 to obtain a solution

It is based on the principle that
No matter in what state of what stage one may be, in order for a policy to be optimal, one must proceed from that state and stage in an optimal manner.

Principle of Optimality – Forward recursion

The procedure could just as well begin at stage 1 and proceed forward through the network.
In this case a function $f_j^{'}(s_j)$ would be defined as the total net benefit from uses 1 to j given s_j units of water to allocate to those uses.

$$f_1^{'}(s_1) = \underset{\substack{x_1 \\ x_1 \le s_1}}{Maximum} [R_1(x_1)]$$

Since the optimal s_1 is unknown, equation 1 must be solved for various discrete values of s_1 between 0 and Q. Next

$$f_2^{'}(s_2) = \underset{\substack{x_2 \\ x_2 \le s_2}}{Maximum} [R_2(x_2) + f_1^{'}(s_2 - x_2)]$$

where $f_2^{'}(s_2)$ is maximum net benefits from uses 1 and 2 with s_2 units of water available to allocate. Once again, this equation must be solved for various values of s_2 between 0 and Q.

Principle of Optimality – Forward recursion

Finally,

$$f_3^{'}(s_3) = \underset{\substack{x_3 \\ x_3 \le s_3 = Q}}{Maximum} [R_3(x_3) + f_2^{'}(s_3 - x_3)]$$

where $f_3^{'}(s_3)$ is maximum net benefits from uses 1, 2 and 3 with $s_3 = Q$ units of water available to allocate.

This process that moves forward from stage 1 to stage 3 is based on the principle that
No matter in what state of what stage one may be, in order for a policy to be optimal, one had to get to that state and stage in an optimal manner.

Bellman's Principle of Optimality (1957)

Backward recursion:
No matter in what state of what stage one may be, in order for a policy to be optimal, one must proceed from that state and stage in an optimal manner.
Forward recursion:
No matter in what state of what stage one may be, in order for a policy to be optimal, one had to get to that state and stage in an optimal manner.

While some problems can be solved equally well by either backward or forward-moving procedures, other problems may be solved by only one approach, but not both.
In either case, there must be a starting or ending point that does not depend on other stages in order to be able to define the first of the recursive equations.

Unlike other constrained optimization procedures, DDP methods are often simplified by the addition of constraints. For example, addition of lower and upper limits on each of the allocations x_j, could narrow the number of discrete values of x_j to be considered.

DP – Multiple State Variables

Suppose that in the preceding example allocation problem, the water was used for three different irrigated crops.

In addition to water, land is also required.
Assume that A units of land are available for all three crops and the u_j units of water are required for each unit of irrigated land containing crop j.

The management or planning problem is now to determine the allocations x_j of water and x_j/u_j units of land that maximize the total net benefits subject to the added restriction on land resources

$$maximize \sum_{j=1}^{3} R_j(x_j)$$

$$\sum_{j=1}^{3} x_j \le Q$$

$$\sum_{j=1}^{3} \frac{x_j}{u_j} \le A$$

$$x_j \ge 0 \quad j = 1,2,3$$

DP – Multiple State Variables – Contd.

Unlike the original problem, there are now two allocations to make at each stage: water and land.
Hence an additional state variable r_j is required to indicate the amount of land available for allocation to the remaining 4 - j crops.

The general recursive relation becomes.

$$f_j(s_j, r_j) = \underset{0 \le x_j \le s_j \ \& \ x_j \le r_j u_j}{maximum} [R_j(x_j) + f_{j+1}(s_j - x_j, \ r_j - \frac{x_j}{u_j})]$$

which must be solved for various discrete values of both state variables s_j and r_j ($0 \le s_j \le Q$ and $0 \le r_j \le A$).

DP – Curse of Dimensionality

Although the addition of a second state variable causes no conceptual difficulties, it does increase the required computational effort.

The larger the number of state variables, the more combinations of discrete states that must be examined at each stage. If done on a computer, this added dimensionality requires more computer time and storage capacity.

The existence of more than three state variables can exceed the computational capacity of a computer if many discrete values of each state variable are required.

This occurs because of the exponential increase in the total number of discrete states that have to be considered as the number of state variables increases.

This phenomenon is termed the **Curse of Dimensionality** of multiple-state-variable dynamic programming problems.

DP – Additional Applications

Three common DP applications in water resources planning are

- Water allocation
- Capacity expansion and
- Reservoir operation

The previous three-user water allocation problem illustrates the first type of application.

Other two applications will be discussed next.

Thank You