

# Dynamic Programming in Stage Coach Problem and Cargo Loading Problem

S. Sathyapriya<sup>1</sup>, S. Swathi<sup>2</sup>, S. Srivarshini<sup>3</sup>, P. Bhavatarini<sup>4</sup>

<sup>1</sup>Assistant Professor, Department of Mathematics, Tamil Nadu, India

<sup>2,3,4</sup>Sri Krishna Arts and Science College, Coimbatore, Tamil Nadu, India

## ABSTRACT

The main aim of the paper is to use application of dynamic programming in stage-coach problem and cargo loading problem. Dynamic programming is a useful mathematical technique for making a sequence of interrelated decisions. It provides a systematic procedure for determining the optimal combination of decisions. Stage-coach problem in which the objective is to find the shortest distance and the corresponding path from a given source node to a given destination node in a given distance network. Cargo loading problem is an optimization problem in which a logistic company is left with the opinion of loading a desirable combination of items in a cargo subject to its weight or volume or both constraints.

**Keywords :** Dynamic Programming, Stage-Coach Problem, Cargo Loading Problem.

## I. INTRODUCTION

The term dynamic programming was originally used in the 1940s by Richard Bellman to describe the process of solving problems where one needs to find the best decisions one after another. In contrast to linear programming, there does not exist a standard mathematical formulation of “the” dynamic programming problem. Rather, dynamic programming is a general type of approach to problem solving, and the particular equations used must be developed to fit each situation. Therefore, a certain degree of ingenuity and insight into the general structure of dynamic programming problems is required to recognize when and how a problem can be solved by dynamic programming procedures. These abilities can best be developed by an exposure to a wide variety of dynamic programming applications and a study of the characteristics that are common to all these

situations. A large number of illustrative examples are presented for this purpose.

## II. METHODS AND MATERIAL

**Dynamic programming** (also known as **dynamic optimization**) is a method for solving a complex problem by breaking it down into a collection of simpler sub problems, solving each of those sub problems just once, and storing their solutions. The next time the same sub problem occurs, instead of re computing its solution, one simply looks up the previously computed solution, thereby saving computation time. Dynamic programming algorithms are often used for optimization. A dynamic programming algorithm will examine the previously solved sub problems and will combine their solutions to give the best solution for the given problem.

**Stage i :** Each sub problem of the original problem is known as a Stage i.

**Alternative  $m_i$ :** In a given stage i, there may be more than one choice of carrying out a task. Each choice is known as Alternative  $m_i$

**State variable  $x_i$ :** A possible value of a resource within its permitted range at a given stage i is known as State variable  $x_i$

**Recursive function  $f_i(x_i)$ :** A function which links the measure of performance of interest of the current stage with the cumulative measure of performance of the previous stage with the cumulative measure of performance of the stage as a function of the state variable of the current stage is known as Recursive function  $f_i(x_i)$  of the Current stage.

**Best Recursive function value:** in a given stage i, the lowest (minimization problem) / highest (maximization problem) value of the recursive function for a given value of  $x_i$  is known as the Best Recursive function value.

**Best alternative in a given stage i :** in a given stage i, the alternative corresponding to the best recursive function value for a given value of  $x_i$  is known as the Best alternative for that value  $x_i$ .

**Backward recursive function:** Here the computations begin from the last stage/sub problem, and this stage will be numbered as stage 1, while the first sub problem will be numbered as the last stage. Since the recursion proceeds in a backward direction, this type of recursive function is known as **Backward recursive function**

**Forward recursive function:** while defining the stages of the original problem, the first sub problem will be numbered as stage 1 and last sub problem will be numbered as the last stage. Then, the recursive function will be defined as per this assumption. This type of recursive function is known as **Forward recursive function**.

## CHARACTERISTIC OF DYNAMIC PROGRAMMING:

The basic features which characterize the dynamic programming problem are as follows: -

- i) The problem can be subdivided into stages with a policy decision required at each stage. A stage is a device to sequence the decisions. That is, it decomposes a problem into Sub-problem such that an optimal solution to the problem can be obtained from the optimal solutions to the sub-problems.
- ii) Every stage consists of a number of states associated with it. The states are the different possible conditions in which the system may find itself at that stage of the problem.
- iii) Decisions at each stage converts the current stage into state associated with the next stage.
- iv) The state of the system at a stage is described by a set of variables, called state variables.
- v) When the current state is known, an optimal policy for the remaining stages is independent of the policy of the previous ones.
- vi) To identify the optimum policy for each state of the system, a recursive equation is formulated with n stages remaining, given the optimal policy for each state with (n-1) stages left.
- vii) Using recursive equations approach each time the solution procedure moves backward stage by stage for obtaining the optimum policy of each state for that particular stage, till it attains the optimum policy beginning at the initial stage.

## APPLICATION OF DYNAMIC PROGRAMMING :

The dynamic programming can be applied to many real life problems. Two samples of those applications is taken:

1. Stage-coach problem
2. Cargo loading problem

**THE STAGE COACH PROBLEM: -**

**Stage-coach problem** or **shortest-path problem** in which the objective is to find the shortest distance and the corresponding path from a given source node to a given destination node in a given distance network.

The stage-coach problem is a problem specially constructed to illustrate the features and to introduce the terminology of dynamic programming. It concerns a mythical fortune seeker in Missouri who decided to go west to join the gold rush in California during the mid-19th century. The journey would require traveling by stagecoach through unsettled country where there was serious danger of attack by marauders. Although his starting point and destination were fixed, he had considerable choice as to which states (or territories that subsequently became states) to travel through en route.

**III. RESULTS AND DISCUSSION**

**PROBLEM :**

**PRESS SHOP SECTION:**

- A :** Initial BEMCO machine
- B :** Machine with line balancing and change of hydraulic to mechanical
- C :** Tool modification and machine parts reduction
- D :** Automation and VFD Variable Frequency Drive (stoke speed controller)

**POLISHING SECTION:**

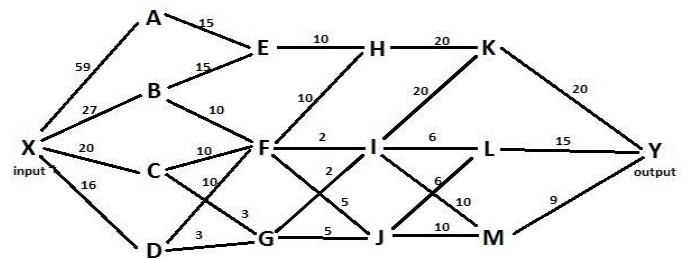
- E :** Initial SAM machine
- F :** Tool combination and automation
- G :** Polishing duration setter

**PRE-ASSEMBLY:**

- H :** Initial NARENDRA machine
- I :** POKEYOKE (machine improvement activity)
- J :** Automation and machine part reduction

**ASSEMBLY:**

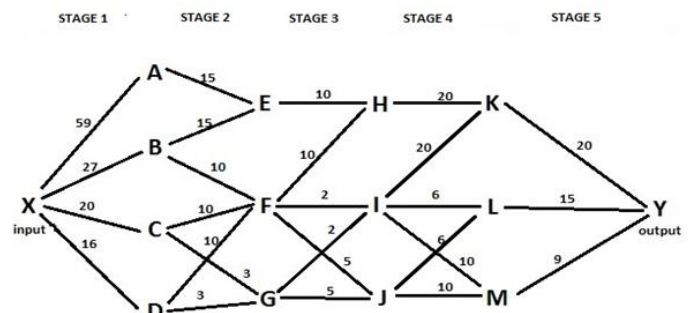
- K :** Initial SUNRAY machine
- L :** Auto spring load, vacuum pressure gauge provision
- M :** Machine design modification and auto spraying system
- X :** INPUT
- A, B, C, D :** PRESS SHOP
- E, F, G :** POLISHING
- H, I, J :** PRE-ASSEMBLY
- K, L, M :** ASSEMBLY
- Y :** DISPATCH



A distance network consists of 15 nodes which are distributed as shown in above figure. Find the shortest path from node X to node Y and also the corresponding distances.

**SOLUTION:**

Each pair of adjacent vertical columns of nodes is treated as a stage. Therefore, there are 5 stages in this problem. Since the stages are defined from right to left, backward recursive function is to be used.



In stage 1, the possible alternative is only one, i.e. node Y. In the same stage (stage 1), the possible state variables are nodes K, L, M.

The recursive function  $f_1(x_1)$  for given combination of the state variable,  $x_1$  and alternative,  $m_1$  in the first stage is:

$$f_1(x_1) = d(x_1, m_1)$$

Where  $d(x_1, m_1)$  is the distance between node  $x_1$  and node  $m_1$ .

The recursive function for given combination of the state variable,  $x_i$  and alternative,  $m_i$  in the stage  $i$  for  $i$  more than 1 is presented below:

$$f_i(x_i) = d(x_i, m_i) + f_{i-1}(x_{i-1} = m_i)$$

Where  $d(x_i, m_i)$  be the distance between the nodes  $x_i$  and  $m_i$ ,  $f_i(x_i)$  be the shortest distance between from node  $x_i$  in the current stage  $i$  to the destination node in the stage 1.

**STAGE 1:** the recursive function  $f_1(x_1)$  for stage 1 is

$$f_1(x_1) = d(x_1, m_1)$$

**CALCULATIONS**

State variable $x_1$	Alternative $m_1$	$f_1(x_1)^*$	$m_1^*$
	Y		
K	20	20	Y
L	15	15	Y
M	9	9	Y

**STAGE 2 :** The recursive function  $f_2(x_2)$  for given combination of the state variable,  $x_2$  and alternative,  $m_2$  in the second stage is:

$$f_2(x_2) = d(x_2, m_2) + f_1(x_1 = m_2)$$

**CALCULATIONS**

State variable $x_2$	Alternative $m_2$			$f_2(x_2)^*$	$m_2^*$
	K	L	M		
H	20+20=40	-	-	40	K
I	20+20=40	6+15=21	10+9=19	19	M
J	-	6+15=21	10+9=19	19	M

**STAGE 3:** The recursive function  $f_3(x_3)$  for given combination of the state variable,  $x_3$  and alternative,  $m_3$  in the third stage is:

$$f_3(x_3) = d(x_3, m_3) + f_2(x_2 = m_3)$$

**CALCULATIONS**

State variable $x_3$	Alternatives $m_3$			$f_3(x_3)^*$	$m_3^*$
	H	I	J		
E	10+40=50	-	-	50	H
F	10+40=50	2+19=21	5+19=24	21	I
G	-	2+19=21	5+19=24	21	I

**STAGE 4:** The recursive function  $f_4(x_4)$  for given combination of the state variable,  $x_4$  and alternative,  $m_4$  in the fourth stage is:

$$f_4(x_4) = d(x_4, m_4) + f_3(x_3 = m_4)$$

**CALCULATIONS**

State variable $x_4$	Alternatives $m_4$			$f_4(x_4)^*$	$m_3^*$
	E	F	G		
A	15+50=65	-	-	65	E
B	15+50=65	10+21=31	-	31	F
C	-	10+21=31	3+21=24	24	G
D	-	10+21=31	3+21=24	24	G

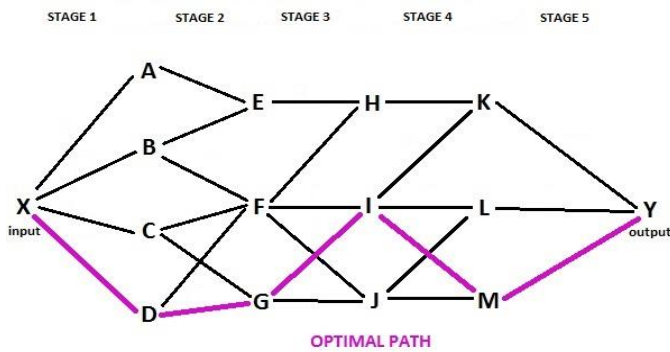
**STAGE 5:** The recursive function  $f_5(x_5)$  for given combination of the state variable,  $x_5$  and alternative,  $m_5$  in the fourth stage is:

$$f_5(x_5) = d(x_5, m_5) + f_4(x_4 = m_5)$$

CALCULATIONS

State variable $x_5$	Alternatives $m_5$				$f_5(x_5)$	$m_5^*$
	A	B	C	D		
X	65+59 =124	31+27 =58	24+20 =44	24+1 6=40	50	D

The final results of the problem are traced in tables above backwards. Therefore, the shortest path is X-D-G-I-M-Y. Hence, the corresponding shortest distance = 40 minutes



**CARGO LOADING PROBLEM:**

Cargo loading problem is an optimization problem in which a logistic company is left with the opinion of loading a desirable combination of items in a cargo subject to its weight or volume or both constraints. Each item produces level of revenue. The objective is to load the most valuable cargo. It is also known as the **fly-away kit problem**, in which a jet pilot must determine the most valuable items to take aboard the jet and **knapsack problem** in which hiker must decide on the most valuable items to carry in back-pack.

The (backward) recursive equation is developing for the general problem of n-item W-ton vessel. Let  $m_1$  be the number of unit of items  $i$  in the cargo. The general problem is represented by following integer LP,

Maximize  $Z = r_1m_1 + r_2m_2 + \dots + r_nm_n$

Subject to,

$w_1m_1 + w_2m_2 + \dots + w_nm_n \leq W$   
 $m_1, m_2, \dots, m_n \geq 0$

The three elements of the model are

1. Stage is represented by item  $i, i=1,2,\dots,n$ .
2. The alternatives of stage  $i$  are represented by  $m_i$ , the number of units of item  $i$  included in the cargo.  
 $m_i = 0, 1, 2 \dots (\frac{W}{w_i})$
3. The state at stage  $i$  is represented by  $x_i$ , which is total weighted assigned.  
 $x_i = 0, 1, 2 \dots W$

The simplest way to determine recursive equation in a two-step process:

1. Express  $f_i(x_i)$  as a function of  $f_i(x_{i+1})$  as follows:

$f_i(x_i) = \max \{r_im_i + f_{i+1}(x_{i+1})\}, \quad i = 1, 2, \dots, n$

Where  $f_{n+1}(x_{n+1}) = 0$

2.  $x_{i+1} = x_i - w_im_i$

Thus, the proper recursive function is given as

$f_i(x_i) = \max \{r_im_i + f_{i+1}(x_i - w_im_i)\}, \quad i = 1, 2, \dots, n$

**PROBLEM:**

A Company is loading 5-ton vessel with one or more of four items. The following table gives the weight of package,  $w_i$  (in tons) and the unit revenue,  $r_i$  (in lakhs) for package of items  $i$ . How should the cargo be loaded to maximize the total return?

Package of items	$w_i$	$r_i$
Popular plus	1	3
Popular	2	22
Clip-on	3	42
Deluxe	4	17

**STAGE 4:**

**SOLUTION:**

W = 5 tons which indicates  $x_i = 0, 1, 2, 3, 4$

The package of items weight  $w_i$  and the maximum weight W all assume integer values; the state  $x_i$  can assume integer values only.

$$f_4(x_4) = \max [17m_4],$$

$$\frac{W}{w_i} = 5/1=5$$

17m <sub>4</sub>							Optimum solution	
x <sub>4</sub>	m <sub>4</sub> =0	m <sub>4</sub> =1	m <sub>4</sub> =2	m <sub>4</sub> =3	m <sub>4</sub> =4	m <sub>4</sub> =5	f <sub>4</sub> (x <sub>4</sub> )	m <sub>4</sub> *
0	0	0	0	0	0	0	0	0
1	0	17	17	17	17	17	17	1
2	0	17	34	34	34	34	34	2
3	0	17	34	51	51	51	51	3
4	0	17	34	51	68	68	68	4
5	0	17	34	51	68	85	85	5

**STAGE 3:**  $f_3(x_3) = \max [42m_3 + f_4(x_3 - 4m_3)],$

$$\frac{W}{w_i} = 5/4=1$$

[42m <sub>3</sub> + f <sub>4</sub> (x <sub>3</sub> - 4m <sub>3</sub> )			Optimum solution	
x <sub>3</sub>	m <sub>3</sub> =0	m <sub>3</sub> =1	f <sub>3</sub> (x <sub>3</sub> )	m <sub>3</sub> *
0	0	0	0	0
1	0+17=17	0	17	0
2	0+34=34	0	34	0
3	0+51=51	0	51	0
4	0+68=68	42+0=42	68	0
5	0+85=85	42+17=59	85	0

**STAGE 2:**

$$f_2(x_2) = \max [22m_2 + f_3(x_2 - 2m_2)] \frac{W}{w_i} = 5/2=2$$

[22m <sub>2</sub> + f <sub>3</sub> (x <sub>2</sub> - 2m <sub>2</sub> )				Optimum solution	
x <sub>2</sub>	m <sub>2</sub> =0	m <sub>2</sub> =1	m <sub>2</sub> =2	f <sub>2</sub> (x <sub>2</sub> )	m <sub>2</sub> *
0	0	0	0	0	0
1	0+17=17	0	0	17	0
2	0+34=34	22+0=22	0	34	0
3	0+51=51	22+17=39	0	51	0
4	0+68=68	22+34=59	44+0=44	68	0
5	0+85=85	22+51=73	44+17=61	85	0

**STAGE 1:**

$$f_1(x_1) = \max [10m_1 + f_2(x_1 - 3m_1)] \frac{W}{w_i} = 5/3 = 1$$

[10m <sub>l</sub> + f <sub>2</sub> (x <sub>l</sub> - 3m <sub>l</sub> )]			Optimum solution	
x <sub>1</sub>	m <sub>1</sub> =0	m <sub>1</sub> =1	f <sub>1</sub> (x <sub>1</sub> )	m <sub>1</sub> *
0	0	0	0	0
1	0+17=17	0	17	0
2	0+34=34	0	34	0
3	0+51=51	10+0=10	51	0
4	0+68=68	10+17=27	68	0
5	0+85=85	10+34=44	85	0

The optimum solution determined in the following manner:

$$x_{i+1} = x_i - w_i m_i$$

Given W= 5 tons, from stage 1, x<sub>1</sub> = 0 which means that 1<sup>st</sup> first item will not be loaded. This allocation leaves

$$x_2 = x_1 - w_1 m_1 = 0 - 3(0) = 0$$

Similarly,

$$x_3 = x_2 - w_2 m_2 = 0 - 2(0) = 0$$

$$x_4 = x_3 - w_3 m_3 = 0 - 4(0) = 0$$

$$x_5 = x_4 - w_4 m_4 = 0 - 1(5) = 5$$

Therefore, 5 units can be loaded to get maximum return of Rs.8 lakhs.

**IV. CONCLUSION**

Dynamic programming is a very useful technique for making a sequence of interrelated decisions. It requires formulating an appropriate recursive relationship for each individual problem. Stage - coach problem can be solved using dynamic programming approach by backward recursion to find the shortest path. Cargo loading problem which optimizes the total maximum return of weights.

**V. REFERENCES**

[1]. Kanti Swarup, PK. Gupta and Manmohan Singh, Operations Research, 4th Edition

(2011),Sultan Chand & Sons, Educational Publishers, New Delhi.

[2]. Prem Kumar Gupta and D.S.Hira, Operations Research, S. Chand & Company Pvt. Ltd, New Delhi.

[3]. Denardo,E.V."Dynamic Programming Theory and Applications, Prentice Hall, Englewood Cliffs,NJ,1982.

[4]. Smith,D.K: Dynamic Programming :A Practical Introduction, Ellis Horwood,London,199

**WEBSITE:**

[1]. [www.wolframalpha.com](http://www.wolframalpha.com)

[2]. [www.purplemath.com](http://www.purplemath.com)

[3]. [www.informs.org](http://www.informs.org)

[4]. <https://www.ime.unicamp.br/~andreani/MS515/capitulo7.pdf>

[5]. <https://www.ii.uib.no/saga/SC96EPR/stagecoach.html>

[6]. <https://books.google.co.in/books?isbn=81203327847>

**Cite this article as :** S. Sathyapriya, S. Swathi, S. Srivarshini, P. Bhavatarini, "Dynamic Programming in Stage Coach Problem and Cargo Loading Problem", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 6 Issue 1, pp. 342-348, January-February 2019. Available at doi : <https://doi.org/10.32628/IJSRST196151> Journal URL : <http://ijsrst.com/IJSRST196151>