MULTISTAGE ALGORITHM ON CAPITAL BUDGETING AND INVENTORY PRODUCTION MODELING

BY

OYEKAN, EZEKIEL ABIODUN
M. TECH/SSSE/2000/652

A THESIS SUBMITTED TO THE POSTGRADUATE SCHOOL IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTER OF TECHNOLOGY (M. TECH) IN MATHEMATICS IN THE DEPARTMENT OF MATHEMATICS / COMPUTER SCIENCE, SCHOOL OF SCIENCE & SCIENCE EDUCATION, FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGERIA.

APRIL 2006

CERTIFICATION

This thesis titled: MULTISTAGE ALGORITHM ON CAPITAL BUDGETING AND INVENTORY PRODUCTION MODELLING by EZEKIEL ABIODUN OYEKAN meets the regulation governing the award of the degree of Master of Technology in Mathematics, Federal University of Technology, Minna and is approved for its contribution to knowledge and literary presentation.

Date 07/04/2006

Dr. N.I. Akinwande
Supervisor

Date 07-04-2006

Dr. L.N. Nzeako
Head of Department

Date 24/8/06

Prof. G.D. Momoh
Dean, School of Science
and Science Education

Date 07/04/06

Prof. Francis Otunta
External Examiner

12/11/07

Date

Prof. J.A.Abalaka
Dean, Postgraduate School

DECLARATION

I hereby declare that this work embodied in this thesis is an original work carried out by me, OYEKAN, EZEKIEL ABIODUN. It has never been presented elsewhere for the award of any degree. All works related to the field of study, prior to the present studies, have been duly acknowledged and referred.

______________________________ ____________________

OYEKAN, EZEKIEL ABIODUN Date

DEDICATIONS

To my late wife; Mrs. Abayomi Abosede Oyekan. My Dear beloved; who departed forty two days after delivering a set of triplets. Also to Shalom and the triplets: Faith, Hope and Love.

ACKNOWLEDGEMENT

I waited patiently for the Lord and He inclined unto me and heard my cry (Psalm 40:1).

I am deeply indebted to Dr. N. I. Akinwande who doubles as my Major supervisor and God inspiring person towards the completion of this project. His encouragement really motivated me to the end.

I am most grateful to Prof. K. R. Adeboye as the mathematical mentor of the Department. His educative roles during the project work are highly appreciated.

I am profoundly grateful to the Head of Department Dr. N. L. Ezeako and other staff of Maths/Computer Department of Federal University of Technology Minna for their encouragements and invaluable contribution to the success of this project.

I am grateful to Dr. Victor Waziri who stood by me during this ardous period.

This project can not be concluded without acknowledging my uncle, Mr. N. T. Oyekan; who assisted in sundry ways for the completion of the project.

To Dn. S. A. Adekojo, Dn. Adeniran James, Dn. Jacobs Morenikeji, Miss V. O. Afolabi, Mr. Akinloye O. A. and Miss. Okunade Mary; I am grateful.

Last but not the least, to all my unmentioned colleagues scattered both at my working place and elsewhere.

TABLE OF CONTENTS

CHAPTER THREE - BUDGETING CAPITAL AND INVENTORY MODELS

CHAPTER FOUR - COMPUTATIONAL ANALYSIS OF THE CAPITAL BUDGETING AND INVENTORY MODELS

CHAPTER FIVE – CONCLUSION AND RECOMMENDATION

Abbreviation of terms

CGM	Conjugate Gradient Method
ECGM	Extended Conjugate Gradient Method
OC	Optimal Control
DP	Dynamic Programming
ILP	Integer Linear Programming
LP	Linear Programming
LPP	Linear Programming Problems
MP	Mathematical programming
DS	Distribution Supervision

Glossary

Inventory:	All stored manufactured products for future sale
Capital budget:	Money invested for projects in a corporation or company
Dynamic programming:	Another expression for multistage programming
Shortest-route:	Shortest distance from a source to a destination
Acyclic Network:	A unitary route from destination to the source
Multistage method:	A method that involves many computational processes from a state zero to the nth state via many stages or routes
Network:	Many nested routes from a unitary state through other nested states and through to the final destination

List of Figures page

ABSTRACT

This research is a study of the one-dimensional constraints problems using the multistage methods. A variety of dynamic programming applications are explored in some depths. The essential features in the dynamic programming models are clearly treated. The dynamic programming approach attacks an optimization problem with multifold constraints and many variables by splitting the problem into sequence of stages in which lower-dimension optimization takes place. In contrast, most linear and nonlinear programming approaches attempt to solve such problems by considering all the constraints simultaneously.

CHAPTER ONE

GENERAL INTRODUCTION TO OPTIMIZATION MODELLING

1.1 Background to the study

In this chapter, we shall have a general overview on optimization modelling. We will thereafter state the aims and objectives of our research work. First let us conceptualize the word **optimization** via a simple definition.

Definition: (optimization)

Rao (1977) defined optimization as a process of obtaining the best result under some given circumstance in design construction and maintenance of any engineering system.

Thus, optimization maybe viewed as a process of obtaining the best solution of a given problem that would enable us make optimum decision be it in managerial, engineering constructional frames or any other ventures worth optimizing in other to take plausible decisions.

1.2 Concept of modelling

It is always necessary to make some constructural idealized or physical representations of some given problems before putting their solution into decision taking in large scale. To this wise, the representative of the true structure of the problem is often regarded as a model.

Construction of a model helps to remove the complexities and possible uncertainties attending a decision-making problem into a logical framework amenable to comprehensive analysis. Such a model clarifies the decision alternative and their anticipated effects, indicates the data that are relevant for the alternatives, and leads to informative conclusions. In short, modelling is a vehicle for arriving at a well-structured view of reality.

The word "model" has several shades of meaning, all of which are relevant to optimization problem solving. First a model may be a representation of reality, such as a small-scale model airplane or locomotive. The second, "model" may imply some sort of idealization, often embodying a simplification of details, such as model plan for urban redevelopment. Finally, "models" may be used as a verb, meaning to exhibit the consequential characteristics of the idealized representation.

It must be formulated to capture the crux of the decision-making problem. At the same time, it must sufficiently be free of burdensome minor details that lend it to finding an improved solution that is capable of implementation. Striking a proper balance between reality and manageability is no mean trick in most applications, and for this reason model-building can be arduous.

1.3 Types of models

According to Chase & Aquilano (2000), there are several classifications of models but the usual types of models are: physical (e.g. airplane model), analog (e.g. a scale where the deflection of a spring or beam represent weight), schematic models (electrical circuits diagrams, organization charts), symbolic models (computer code or mathematical models representing a bank teller or a machine).

Also in Everett E. Adam Jr. (1993), models are classified as follows:

(a) Verbal models – Verbal or written models express in words the relationships among variables. Verbal models are descriptive. Suppose a passing motorist asks you to give directions to the nearest gas station, if you tell him the way, you are giving a verbal model. If you write the directions in words (not pictures), you are giving a descriptive model.

(b) Schematic models – Schematic models show a pictorial relationship among variables. If you give the passing motorist a map showing the way to the nearest gas station, you will be giving a schematic model.

(c) Iconic models – These are scaled physical replicas of objects or process. Architectural models of a new building, aeroplane, highway engineering replicas of a proposed overpass system are iconic models.

(d) Mathematical / Symbolic models – Mathematical models show functional relationships among variables by using mathematical symbols and equations.

However, there are three (3) basic types of model used in operation research namely:

1. Iconic model
2. Analogue and
3. Symbolic / Mathematical models (this may be discrete or continuous depending on whether the variables involved are discrete or continuous).

Modelling:- This is an act of building a model. Thus, mathematical modelling is an act or skill to problem solving such that the real life problem or phenomenon is logically represented using the notion of mathematics and thereafter apply relevant solution technique to solve the model. For example, force = mass x acceleration is (a mathematical model) .

1.5 Processes of modelling

In this section, we outline in summarized steps the modelling processes. The structures for a good discrete optimization model are:

(a) Formulating the problem,

(b) Building the model

(c) Performing the analysis

(d) Interpreting the result of the model

(e) Validating the model

(f) Implementing the findings and updating the model.

(a) Formulating the problem

At this stage, you must diagnose the statement of the problem's elements. These include the controllable or decision variables, the uncontrollable variables, the restrictions or constraints on the variables, and the objectives for defining a good or improved solution.

In the formulation process, you must establish the confines of the analysis. Determining the limits of a particular analysis is mostly a matter of judgement

(b) Building the model

After formulating the problem; you now get down to the fine detail. Decide on the proper data inputs and design the appropriate information outputs. Identify both the static and dynamic structural elements, and devise mathematical formulas to represent the interrelationships among these elements. Some of these interdependencies may be posed in terms of constraints or restriction on the variables. Some may take the form of probabilistic evolutionary system. The time horizon may also be chosen for the completion of the model. The choice of this horizon in turn influences the nature of the constraints imposed, since, with along enough horizon, it is usually possible to remove any short-run restrictions by an expenditure of resources.

(c) Performing the analysis

Given the initial model, along with its parameters as specified by historical, technological, and judgemental data, you next calculate a mathematical solution. Solution means values for the decision variables that optimize one of the objectives and give permissible levels of performance on any other of the objectives.

In certain cases, a major part consists of determining the sensitivity of the solution to the model speculations, and the particular to the accuracy of the input data and structural assumptions. Because sensitivity testing is so essential a part of the validation process, one must be careful to build one's model in such a way as to make this process computationally tractable.

(d) Interpreting the model.

This stage of modelling process simply involve the interpretation of the result of the model formulated. The major variables formulated into either objective function or constraints must be easily and clearly identifiable.

(e) Validating the model.

Undoubtedly, the first version of any large mathematical model will inevitably contains many flaws. These must necessarily and thoroughly tested in order to try to identify and correct them. This ensures the elimination of the major flaws and if possible, the minor ones as wel; in order to get an improved model. According to Hillier & Lieberman(2001):The process of testing and improving a model to increase its validity is referred to as **model validation**. It is difficult however to describe how model validation is done, because the process depends greatly on the nature of the problem being considered and the model being used.

(f) Implementing the findings and updating the model.

This is the modelling stage in which the performing analyses is put into practical implementation. It is common for operational research to be used repeatedly in the analysis of decision problems. Each time, the model must be revised to take account of both the specifics of the problem and current data. Hence models must be documented in details so as give forum for future plans updating.

1.5 Statement of the problems

This research is for the study of discrete optimization models involving multistage problems solving. As we can not study all discrete optimization problems having similar features, we have the two problems as our statements of problems:

Problem one (P1) (The capital budgeting problem)

Consider a certain utopia corporation with the given problem of annually budgeting several million pounds for land development, and for building shopping centres, complexes, and industrial parks. The corporation is now planning to invest up to #10 million on one or more of three large projects. The data for these projects are contained in table (1.1) below:

Investment level y	Project 2		Project 3		Project 4	
	Cost $I_2(y)$	Value $R_2(y)$	Cost $I_3(y)$	Value $R_3(y)$	Cost $I_4(y)$	Value $R_4(y)$
0	0	0	0	0	0	0
1	3	8	4	9	6	17
2	5	13	5	13	7	18
3	7	18	6	18	8	21
4	8	19	9	19	9	22
5	9	21	10	23	10	24

Table 1.1: The utopia Corporation data problem; Investment cost in units #1 000 000 and present values in units of #100 000.

From the table, observe that each of the three projects can be developed at any of the five different investment levels. For example, the corporation can choose to invest #3million, #5million, #7million, #8million, or #9million in project 2. If the investment choice is level 1, namely, #3million for project 2, then present value of future earnings is estimated to be $R_2(1) = \#0.8\text{million}$ if, instead, the investment choice is level 5, namely #9million, then the value for earnings rises to $R_2(5) = \#2.1\,\text{million}$. A similar interpretation applies for the other two projects.

The problem is; how should the utopia corporation project capital be distributed in other to make maximum profit?

Problem two (P2) (The inventory model)

In this problem, we are considering the inventory holding costs together with an assigned explicit smoothing cost incurred by changing production at different period. Here, we consider a hypothetical manufacturing productive company, with given quantified amount of inventory storage at given periodic levels. For each period t, the cost incurred depends on the production quantity x_t, the ending inventory level i_t and the previous period's production quantity x_{t-1}:

$$C_t(x_t, i_t, x_{t-1}) = C(x_t) + 1.i_t + 1.(x_t - x_{t-1})^2 \qquad 1.1$$

with the cost of productivity defined as fixed in this wise:

$$C(0) = 0,\ C(1) = 15,\ C(2) = 17,\ C(3) = 19,\ C(4) = 21,\ C(5) = 23 \qquad 1.2$$

for all periods. In equation (1.1), the third function to the right is a quadratic expression representing the smoothing cost; which is given by the square of the fluctuation in production levels over two successive periods. In this illustration, the cost impact of a variation in production is quadratic, and an upward fluctuation of a given magnitude is as costly as a downward fluctuation of the same magnitude. The required demand and the feasible production and inventory levels are

$$D_t = 3 \qquad (\textit{stationary demand}) \qquad 1.3$$

$$x_t = 0,1,2,........,5 \qquad i_t = 0,1,..........,4 \textit{ and no ending inventory} \qquad 1.4$$

The problem now is to find the optimal cost production policies and to make comparison with the cost of production smoothing.

1.6 Aims and Objectives

The problems before us are categorized into two discrete problems; capital budgeting and inventory production problems. Thus, we must have two broad goals as our aims which are:

Aims

The aims of this research are:

(1) To construct the optimal network tables that would enable us obtain the optimal distribution for the capital budget projects

(2) To construct multistage tables of analysis that would enable us to compute the numerical optimal results of the networks.

Objectives

The problems before us are dynamical programming problems. Therefore our objectives towards achieving our aims are:

(i) To derive recursive formulas that would enable us solve the problems numerically.

(ii) To use the derived recursive formulas to obtain different computational optimal policy stages and to represent each stage of gain in tabular forms.

1.7 Significance of study to knowledge

Most of the optimization problems research has been based on continuous problems solving solutions. This discrete research work therefore gives a glimpse of the application of optimization theory into economic solving problems. It is anticipated that the research would help in managerial decisions takings in our productive companies and project planning ventures.

1.8 Methodology of study

Multistage programming problems do not have fixed algorithms for solving them. As there are devised human approaches to human economy needs, so are the mirage of deviated methods obtain to acquire the needs. Multistage stage problems are therefore derived according to the problems at hand. Thus, we shall derive recursive formulas to achieve our aims.

1.9 Scope of study

These models are few vector variables and of one constraint.

Conclusion of chapter one

We have developed our line of action towards achieving our quest of solutions. Towards these ends, we did give periscopic visualization of our modellings problems as multistage programming problems. The statements of the problems have been clearly spelt out; so also are the aims and objectives.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

Many versions of multistage programming abound; in Wagner (1989) multistage algorithm is treated to some large extend under various subtopics. Multistage algorithm is a dynamic programming method which is applicable mostly in construction and production companies for excellent decisions taking to expedite projects completion within shortest time in order to minimize costs and maximize profits. The structures of these decisions taking always take effects from such models known as dynamic programming. Dynamic programming is a legitimate branch of mathematical programming (MP) despite the fact that it differs in many fundamental respects from the methods of earlier optimization manipulating techniques that enable optimal solutions. In the first place, it may not require an algebraic model at all and we shall shortly examine a number of problems in which algebraic symbols make no appearance whatever. More importantly, DP is centred on a single very simple- but broadly applicable principle around which a general approach to problem solving has evolved. It is no one method but involves the use of techniques from a variety of sources- including linear programming (LP), integer linear programming (ILP) and the rest. DP is particularly appropriate for problems in which decisions are to be taken in sequence, with each decision being dependent in some way on the one before it. Surprisingly as it may seem, DP could actually be used to solve all optimization problems based on both linear and non-linear systems of equations- everyone can be viewed as a sequential decision problem. We note in passing, however, that other mathematical programming models maybe more efficient solving certain class of problems. Nonetheless, DP is also important in its own right. We shall investigate other classes of DP in an important class of decision problems that other methods can rarely- if ever-address. These are stochastic problems in which chance and uncertainty play so large a part that it can not be ignored and must be built into the decision model. First, though, we consider a classical routing model.

2.2 SHORTEST_ROUTE PROBLEM

Our first illustration of the DP is the shortest route problem; we do this using a life company problem which we shall address as Waziri Constructional Company (WCC). In this problem, we investigate the problem face daily by the Distribution Supervisor (DS). It is the responsibility of the DS to recommend suitable routes for WCC's drivers to follow when delivering customers' goods. In a typical case, the DS has no more information than the map in figure 2.1 below and will usually estimate from factory to customer from a quick inspection of the route network. However, it is clear that this rough–and-ready method may often lead to longer journeys than are necessary and may thus result in significant avoidable cost, Management will naturally wish to find a reliable way of identifying the shortest route between factory and customer, whatever the customer's location. Though there is several solution methods for problems of this sort, most depend on ideas of a DP type. A classic DP treatment is first presented below; a schematic version of the DS's route is presented in figure 2.2. We regard the driver's route from Chanchaga factory to customer's site as sequence of routing decisions, each taken as a specified stage of the journey. A decision is made at Chanchaga governing the initial direction taken by driver. This means that, at the second stage of the journey, the driver will find himself at A., B or C-where a second choice of direction must be made. At stage 3, the driver will is at D, E or F and faces a final decision-whether to travel to G or H before ending the journey at L. The numbers attached to the links in the network are distances in miles.

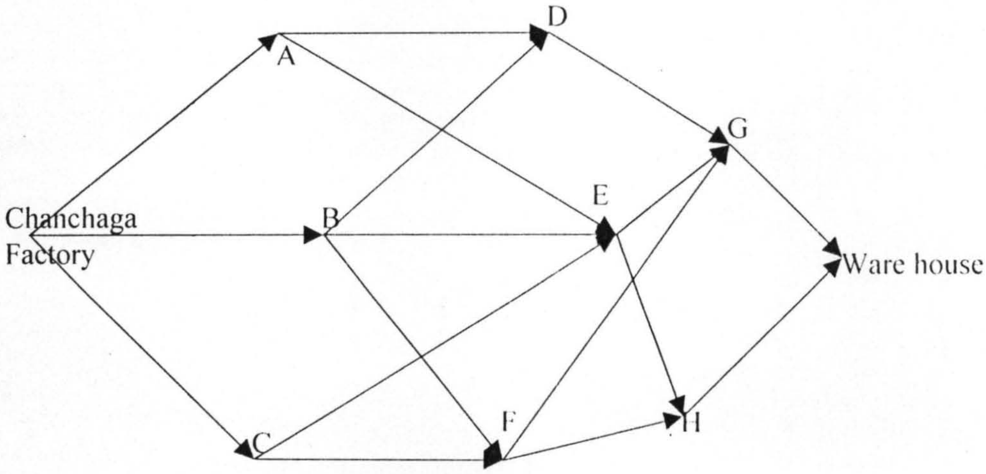


Figure 2.1: Network route between factory and Costumers

In some typical DP approaches, the analysis of the problem proceeds from the last decision stage to the first stage of the analytical shortest route. In other words, we start with end point of the problem and work our way backwards to the start of the problem. We adopt this significant approach in computing our acyclic optimal policies in this project. In our example, there are three decision stages, as shown in figure 2.2, and so our discussion begins at stage three (3). At this stage, the driver must find himself in one of the three locations: D, E or F. We consider the choice of routes he may take from Table 2.1. The table will repay close inspection. We note that there are 3 locations appropriate for stage 3-D, E and F. Depending at which one of these locations the driver finds himself, a decision must then be made as to whether to travel to G or to H.. Note that according to driver to the route diagram there is no alternative to the road to G if the driver finds himself in D. In each of E and F, o the other hand, two possibilities arise.

The calculations of remaining distances are evident from figure 2.2. If the driver is at F, for example, and selects the road to G, then the total length of the rest of the journey- from F to G To I- will be 23+21 miles. These simple computations are enough to establish an optimal decision for each location and corresponding optimal distance for the remaining part of the journey. Table 2.2 below summarises the results of this analysis. These will be used to determine optimal routing decisions at stage 2.

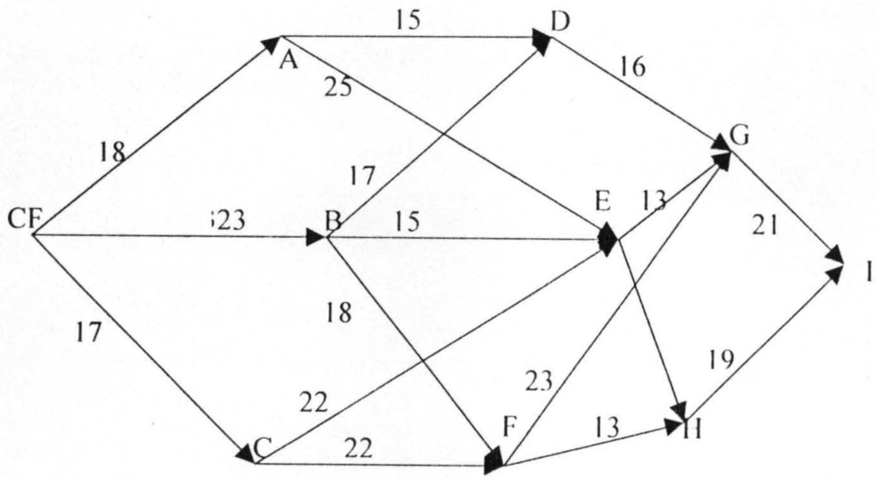


Figure 2.2: Network route travel distance (miles)

Having completed the stage 3 analysis, we now go back to the earlier stage- stage 2. In other words, we suppose that the driver has arrived at A, B, or C and select a suitable route to D, E or F.

Table 2.1 WCC's shortest-route problem: stage 3

Location	Decision	Remaining distance (miles) Now +later	Optimal decision	Optimal distance (miles)
D	Go to G	16+21	Go to G	37
E	Go to G	13+21		
	Go to H	17+19	Go to G	34
F	Go to G	23+21		
	Go to H	13+19	Go to H	32

Table2.2: stage results

Location	Optimal Decision	Optimal distance (miles)
A	Go to G	37
E	Go to G	34
F	Go to H	32

Table 2.3: Stage 2 analysis

Location	Decision	Renaming distance (miles) Now +later	Optimal decision	Optimal distance (miles)
A	Go to D	15+17		
	Go to E	25+14	Go to D	52
B	Go to D	17+37		
	Go to E	15+34		
	Go to F	18+32	Go to E	49
C	Go to E	20+34		
	Go to F	22+32	Go to E or F	54

2.3 Bellman's principle of optimality

This principle is concerned with the manner with which the optimal policy is considered after reaching a particular state of a given stage. Bellman's states that:

Any optimal decision sequence is such that, whatever the stage and state of the problem, the decision-maker will always act in a manner which is optimal with respect to that stage and state.

Table 2.4 stage 2

Location	Optimal decision	Optimal distance
A	Go to D	52
B	Go to E	49
C	Go to E or	54

Table 2.5 Stage 1 analysis

Location	Decision	Renaming Distance (miles) Now+later	optimal decision	Optimal distance (miles)
Chanchaga	Go to A	18+52	Go to A	70
	Go to B	23+49		
	Go to C	17+54		

At any stage in the decision process our analysis of a particular decision depends on knowing what will be done at the next stage when we arrive there. Such knowledge exists because we have already undertaken the relevant analysis and have the optimal results available. By Bellman's principle, these are all we need to know about the succeeding stage. Stage 2 results are summarized in

Table2.4. As you will now realize, these will shortly be used in the analysis of stage 1.

The analysis of stage 1 follows an identical pattern. Note, however, that we need consider only one location. There is no doubt about the driver's whereabouts at stage1; the journey begins at Chanchaga town, we are now in position to present the solution of the problem as a whole in DP; this is done by backtracking through the tables of stage results, incorporating appropriate optimal decisions at each stage. for our shortest –route problem, we begin at Table 2-5; it clear from the stage 1 analysis that the optimal route length is 70 miles and that the driver should select the road to A as the first leg of his journey, this decision will place him in A at sage 2, of course, whereupon –by reference to Table 2.2-he will now to continue via D. In D at sage 3, however, Table 2-2 provides the information necessary for an optimal end to his trip: he should reach 1 by way of G. To summarize, the driver's route is:

Chanchaga-A-D-1 with the distance travelled minimized at 70 miles.

2.4 Computational advantages

The preceding activity-the method of identifying, evaluating and comparing all the solutions to a problem-is usually known as complete enumeration. Where the solution set is small, as it is in WCC routing problem, complete enumeration is a viable approach .indeed, it is frequently the simplest and best way of solving the problem. Unfortunately, for larger problems-even many smaller ones- the number of solutions is so great that enumerating them all is a practical impossibility. This, of course, is the fundamental reason for the uses of mathematical programming in business in decision-making. There are simply too many options to consider every one individually and we must therefore find means of dealing with them en masse and selecting the ones-that deserve closer attention. There are in fact 12 routes from north town to 1 in Fig. 14-2, each involving 4 legs of journey. Complete enumeration approach must therefore require, in addition to the identification of the routes:

36 additions

And 11 route -to-route comparisons.

The first of these numbers follows because each route necessitates 3 additions. For example:

18+15 +16+21

To calculate the distance from Chanchaga to 1 via A, D and G., the second is a result of the fact that we choose the optimal route by making a series of pair wise comparisons between their lengths; for example, between Chanchaga-A-D-G-1, and Chanchaga –A-E-G-1, and so on. Now, consider the equivalent numbers of operations in the DP approach. These are shown in Table 14-6.

2.5 Characteristics of all dynamic models

The common characteristics of all dynamic programming models according to (Wisniewski and Darcy, 1992) are expressing the decision by means of recursive formulae. These Recursions are idealized as components that are discretely in stages. These stages are categorized as multistage.

Now, there are two simple rules that can be applied mechanically to all problems so as to expose their dynamic properties. Experience is the beat teacher. In other words, multistage recursion formulation is an art. The most important approach is the application of Bellman's principle of optimality; which stated earlier.

Definition 2.1 (Principle of Optimality)

An optimal policy must have the property that regardless of the route taken to enter a particular state, the remaining decisions must constitute an optimal policy for leaving that state- (Wagner, 1989).

To buttress this fundamental principle, let us consider the stagecoach as in Wagner (ibid).

Example 3.2 (Application of the DP)

Adamu's fame spread wide. He received an urgent request from the Russian Tsar Kazim to find a route to Rastor from Vladivostok. In this case, the Tsar's wife made a strip, and the expense associated with each leg of the journey is for protection against attacks by Cossacks and local tribesmen. The numbers shown on each arc in the figure below are the rubbles that must be paid for protection on the route. Find an optimal routing. Display your calculations in tables.

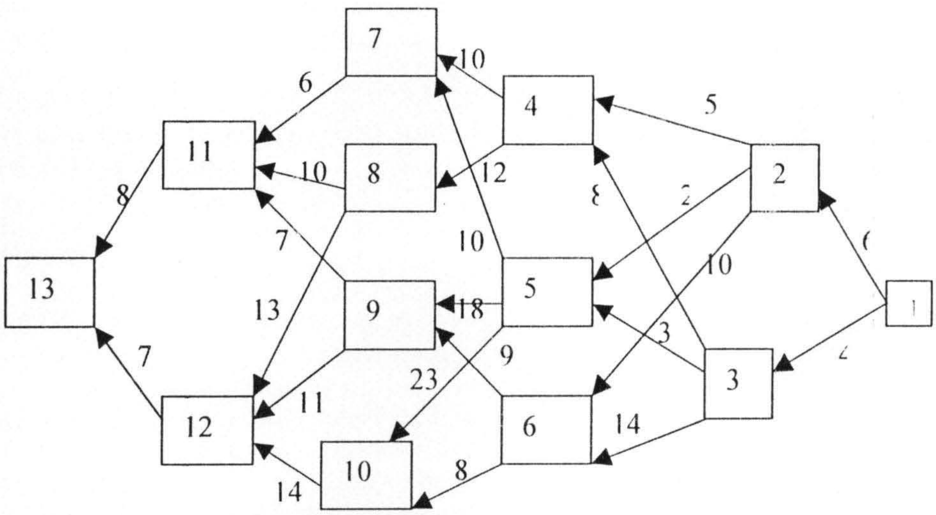


Figure 2.3: Shortest network route

Ion this computational route analysis, we have 5-stages to go before the acquisition of the final result starting from the end.

$n = 0$

$\because f_0(1) = 0,\ n = 0,\ j_0(13) = Stop$

We solving this problem using a recursive formula defined as:

$$f_n(s) = \min_{s,j}[C_{s,j} + f_{j-1}(s)];\ \text{for j=1, 2, 3, 4, 5}$$

Now for stages computation; we start from the last stage in this sequential order:

Stage one, n=1

For n=1,

$$f_1(11) = C_{11,13} + f_0(13) = 8 + 0,\quad d_1(11) = 13$$

$$f_1(12) = C_{12,13} + f_0(13) = 7 + 0,\quad d_1(12) = 13$$

j/n	13	$d_1(s)$	$f_1(s)$
11	8+0	13	8
12	7+0	13	7

Table 2.6: First stage

Stage two, n=2

$$f_2(7) = C_{7,11} + f_1(11) = 6 + 8,\quad d_2(7) = 11$$

$$f_2(8) = \min(C_{8,11} + f_1(11), C_{8,12} + f_1(12)) = 18,\quad d_2(8) = 11$$

$$f_2(9) = \min(C_{9,11} + f_1(11), C_{9,12} + f_1(12))$$

$$= \min(17 + 8; 11 + 7) = 15,\quad d_2(9) = 11$$

$$f_2(10) = (C_{10,12} + f_1(12))$$

$$= 14 + 7 = 21,\quad d_2(10) = 12$$

j/s	11	12	$d_2(s)$	$f_2(s)$
7	6+8		11	14
8	10+8	13+7	11	18
9	7+8	11+7	11	15
10		1+7	12	21

Table 2.7: Stage 2

Stage three; n=3

$$f_3(4) = \min(C_{4,7} + f_2(7), C_{4,12} + f_2(8))$$

$$= \min(10 + 14, 12 + 18) = 24; \quad d_3(4) = 7$$

$$f_3(5) = \min(C_{5,10} + f_2(7), C_{5,8} + f_2(8), C_{5,9} + f_2(9), C_{5,10} + f_2(10))$$

$$= \min(10 + 14, 18 + 18, 23 + 15, 14 + 21) = 24; \quad d_3(5) = 7$$

$$f_3(6) = \min(8 + 21, \; 9 + 21) = 29; \quad d_3(6) = 10$$

j/s	7	8	9	10	$d_3(s)$	$f_3(s)$
4	10+14	12+18			7	24
5	10+14	18+18	23+15	14+21	7	24
6			8+21	9+18	10	27

Table 2.8: stage three

Stage four; n=4

The computational processes below summarize second to the last stage for the acquisition of the shortest route. The table 2.9 is the numerical results.

$$f_4(2) = \min(5 + 24, \; 2 + 24; 10 + 29) = 29; \quad d_4(2) = 4$$

$$f_4(3) = \min(8 + 24, \; 3 + 24; 14 + 29) = 27; \quad d_4(3) = 5;$$

j/s	4	5	6	$d_4(s)$	$f_4(s)$
2	5+24	2+24	10+27	5	26
3	8+24	3+24	14+27	5	27

Table 2.9: Stage 4

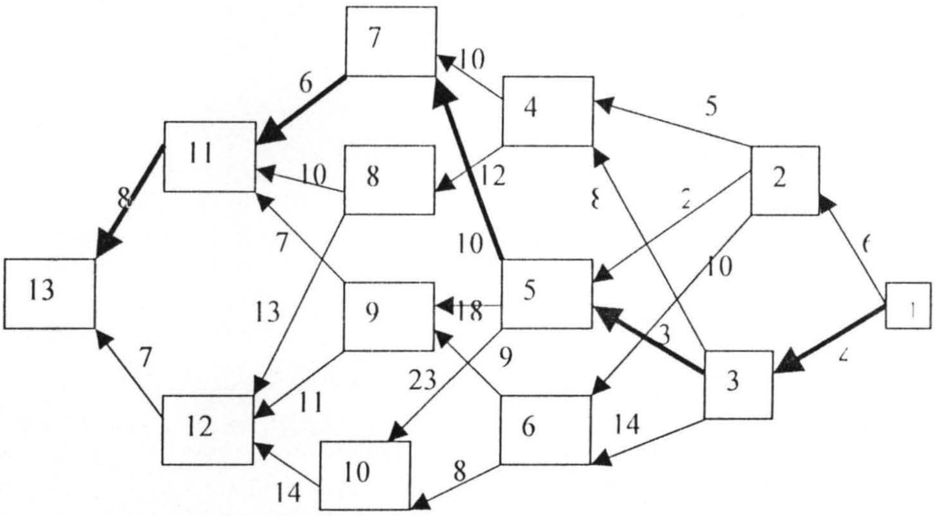
Last Stage five; n=5

$$\begin{aligned} f_5(1) &= \min(C_{1,2} + f_4(2); C_{1,3} + f_4(3) \\ &= \min(29 + 6,\ 4 + 27) = 31;\ d_5(1) = 3 \end{aligned}$$

j/s	2	3	$d_5(s)$	$j_5(s)$
1	6+26	4+27	3	31

Table 2.10: Stage 5

From table 2.10, we see that the optimal cost at minimum consideration is 31 000 rubies. This is achieved when he takes the route through states 1-3-4-7-11-13 as shown by the heavy arrows in figure 2.4 below.



2.4 Figure: Shortest network route for final decision

With the above example we come to the end of a shortest route problem.

2.6 The Mathematics of Dynamics programming

We have deliberately avoided the mathematical procedures for the DP models. Although every DP can be formulated algebraically, it is an unfortunate truth that this often serves to obscure the analysis rather than clarify it. Here, we present earlier models in more formal manner.

2.7 Distribution of materials

Consider a hypocritical problem of distributing crates of eggs to certain four consumers' stores. The distributor suspects that in order to maximize his overall profit, he should not put his all his eggs in one shopping Basket. Hence he wants to find an optimal distribution of eggs.

To make clear the appropriate model structure, we pose an obviously simplified numerical example, and afterwards we summarize the approach with more general mathematical notation.

Let y_j denote the number of crates shipped to store j and $R_j(y)$ the resulting net profit for store j when $y_j = y$. Observe the number of crates with their associated cost at each of the four stores table 2.11 below:

	Net profit			
Number of Crates, y	Store 1 $R_1(y)$	Store 2 $R_2(y)$	Store 3 $R_3(y)$	Store 4 $R_4(y)$
0	0	0	0	0
1	6	3	2	5
2	10	10	6	9
3	14	15	14	13
4	16	19	20	17
5	18	21	22	21
6	20	22	24	25

Table 2.11

If all the crátes are shipped to store 1, that is Store 1 $y_1 = 6$, total profit is Store 1 $R_1(6) = 20$. Should the distributor ship all crates to one store, and then he should ship all of them to store 4 where the profit is Store 1 $R_4(6) = 25$. But clearly he can do better by distributing the crates to more than one store. For example, by letting $y_1 = y_2 = 3$, his total profit equals 29 ($R_1(3) + R_2(3) = 14 + 15 = 29$. In general notation, then, the distributor's decision problem can be formulated as:

$$\text{Maximize } \sum_{j=1}^{n} R_j(y_j) \qquad 2.1$$

Subject to:

$$\sum_{j=1}^{n} y_j = N \qquad 2.2$$

$$y_j = 0,1,2,3......,N \qquad 2.3$$

In this formulation, a crate can not be split for sale.

Now our desire is to transform equations (2.1) through to (2.3) into a DP problem To achieve this, let us view how the optimization table 2.11 can viewed as finding a best-profit route in an acyclic network. Now all the decision variables relate to a single time period; thus viewing the quantities y_j sequentially, starting with store 4, then store 3, store 2 and then store 1, we can categorize the problem into 4 stages. This multistage characterization makes it possible to draw a network analogous to the distributional problem before us. The network distribution is:

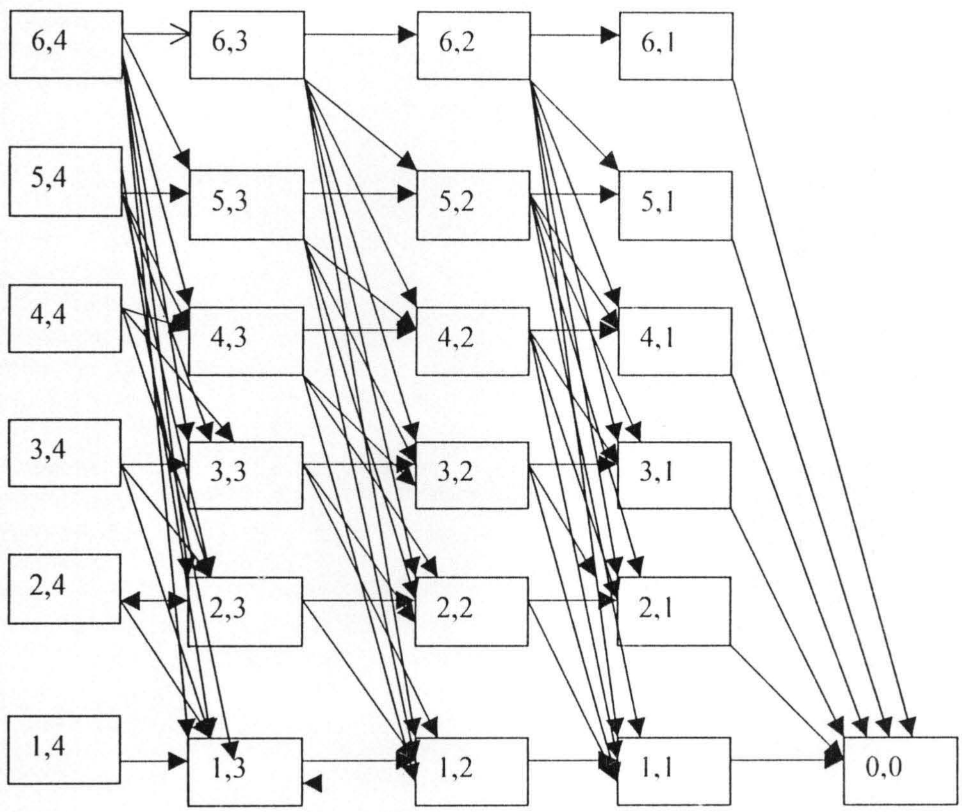


Figure 2.5: The Distribution DP network.

To convert the problem statement (2.1) through (2.3), and its best route in acyclic network equivalent, into a multistage (DP) version, define:

$$g_j(n) = (\textit{profit when } n \textit{ crates are ditributed to store } 1, \textit{ store } 2, \ldots\ldots, \textit{store } n \qquad 2.4$$

$$y_j(n) = (\textit{a distribution amount for store } j \textit{ that yields } g_j(n) \qquad 2.5$$

The g indicates company's goal, the letter n refers to the number of crates to be distributed. And the index j denotes just a store. The quantity $g_j(n)$ is simply the value of the best-profit route from Node (n,j) to the terminal Node (0,0). By definition, the terminal Node (0,0) has the value $g_0(0) = 0$. Now since only a single arc leads out each node (n, 1) for stage $j = 1$, the value of the best-route at each of these nodes is $R_1(n)$. Thus continuing onto the nodes for stage $j = 2$, to illustrate, consider the node (3, 2), the value of the best-profits route is calculated as

$$g_2(3) = \max imize[R_2(0) + g_1(2), R_2(1) + g_1(1), R_2(2) + g_1(0)]$$

$$= \max imize[0 + 10, 3 + 6, 10 + 0] = 10$$

From this illustration, the logical computational procedure that underlies the computation of the optimal route in the acyclic figure 2.5 is conveniently expressible as:

$$g_j(n) = \max_y[R_j(y) + g_{j-1}(n - y_j), \text{ j=0, 1, 2, ...}$$

$$g_0(n) = 0 \quad for \quad j = 0$$

The computation starts with the last stage $j = 1$ since the value for $g_1(n)$ are trivially easy to find. In particular, the profit of store 1 increases as more crates are shipped, and so

$$g_1(n) = R_1(n)$$

$$y_j(n) = n, \qquad for \ n = 0,1,2,........,6$$

Table 2.12 gives the summary of the first stage best-profit allocations.

N	$y_1(n)$	$g_1(n)$
0	0	0
1	1	6
2	2	10
3	3	14
4	4	16
5	5	18
6	6	20

Table2.12 Optimal policy for the first stage

Computation of stage two optimal policies

$$\begin{aligned} g_2(0) &= [R_2(0) + g_1(0)] \\ &= [0 + 0] = 0; \quad R_2(1) = 0;\ g_2(0) = 0 \end{aligned}$$

$$\begin{aligned} g_2(1) &= \max_{1}[R_2(0) + g_1(1), R_1(1) + g_1(0) \\ &= \max_{1}[0 + 6, 3 + 0] = 6; \qquad R_2(1) = 0;\ g_2(1) = 6 \end{aligned}$$

$$\begin{aligned} g_2(2) &= \max_{2}[R_2(0) + g_1(2), R_1(1) + g_1(1), R_1(2) + g_2(0) \\ &= \max_{2}[0 + 10, 3 + 6, 10 + 0] = 10; \qquad R_2(2) = 0,2;\ g_2(2) = 10 \end{aligned}$$

$$\begin{aligned} g_2(3) &= \max_{3}[R_2(0) + g_1(3), R_1(1) + g_1(2), R_1(2) + g_2(1), R_1(3) + g_2(0) \\ &= \max_{3}[0 + 14, 3 + 10, 10 + 6, 15 + 0] = 16; \qquad R_2(3) = 2;\ g_2(3) = 16 \end{aligned}$$

$$\begin{aligned} g_2(4) &= \max_{4}[R_2(0) + g_1(4), R_1(1) + g_1(3), R_1(2) + g_2(2), R_1(3) + g_2(1), R_2(4) + g_1(0) \\ &= \max_{4}[0 + 16, 3 + 14, 10 + 10, 15 + 6, 19 + 0] = 21; \qquad y_2(4) = 2;\ g_2(4) = 21 \end{aligned}$$

$$\begin{aligned} g_2(5) = \max_{4}[&R_2(0) + g_1(5), R_1(1) + g_1(4), R_1(2) + g_2(3), R_1(3) + g_2(2), R_2(4) + g_1(1), \\ &R_2(5) + g_1(0)] \\ = \max_{4}[&0 + 20, 3 + 18, 10 + 16, 15 + 14, 19 + 10, 21 + 6, 22 + 0] = 29; \ y_2(5) = 3,4;\ g_2(5) = 29 \end{aligned}$$

The optimal control table 2.13 for stage two is displayed below:

y/n	0	1	2	3	4	5	6	$y_2(n)$	$g_2(n)$
0	0+0							0	0
1	0+6	3+0						0	6
2	0+10	3+6	10+0					0,2	10
3	0+14	3+10	10+6	15+0				2	16
4	0+16	3+14	10+10	15+6	19+0			3	21
5	0+18	3+16	10+14	15+6	19+6	21+0		3,4	25
6	0+20	3+18	10+16	15+14	19+10	21+6	22+0	3,4	29

Table 2.13: Optimal polices for stage two.

Stage three; j=3

The computational procedures for stage three optimal policies are hereunder given:

$$g_3(0) = \max_0[R_3(0) + g_2(0)]$$
$$= \max_0[0+0] = 0;\ y_3(0) = 0;\ g_3(0) = 0$$

$$g_3(1) = \max_1[R_3(0) + g_2(1), R_3(1) + g_2(0)]$$
$$= \max_1[0+6,\ 2+0] = 6;\ y_3(1) = 0,\ g_3(1) = 6$$

$$g_3(2) = \max_2[R_3(0) + g_2(2), R_3(1) + g_2(1),\ R_3(2) + g_2(0)]$$
$$= \max_2[0+10,\ 2+6,\ 6+0] = 10;\ y_3(2) = 0,\ g_3(2) = 10$$

$$g_3(3) = \max_3[R_3(0) + g_2(3), R_3(1) + g_2(2),\ R_3(2) + g_2(1), R_3(3) + g_2(0)]$$
$$= \max_3[0+16,\ 2+10,\ 6+6, 14+0] = 10;\ y_3(3) = 0,\ g_3(3) = 16$$

$$g_3(4) = \max_4[R_3(0) + g_2(4), R_3(1) + g_2(3),\ R_3(2) + g_2(2), R_3(3) + g_2(1), R_3(4) + g_2(0)]$$
$$= \max_4[0+21,\ 2+16,\ 6+10, 14+6, 20+0] = 21;\ y_3(4) = 0,\ g_3(4) = 21$$

$$g_3(5) = \max_4[R_3(0) + g_2(5), R_3(1) + g_2(4), R_3(2) + g_2(3), R_3(3) + g_2(2), R_3(4) + g_2(1),$$
$$R_3(5) + g_2(0)]$$
$$= \max_4[0 + 25,\ 2 + 21,\ 6 + 16, 14 + 10, 20 + 6, 22 + 0] = 26;\ y_3(4) = 4,\ g_3(4) = 26$$

$$g_3(6) = \max_6[R_3(0) + g_2(6), R_3(1) + g_2(5), R_3(2) + g_2(4), R_3(3) + g_2(3), R_3(4) + g_2(2),$$
$$R_3(5) + g_2(1),\ R_3(6) + g_2(0)]$$
$$= \max_6[0 + 29,\ 2 + 25,\ 6 + 21, 14 + 16, 20 + 10, 22 + 6,\ 24 + 0] = 30;$$
$$y_3(4) = 3,4,\ g_3(4) = 30$$

y/n	0	1	2	3	4	5	6	$y_3(n)$	$g_3(n)$
0	0+0							0	0
1	0+6	2+0						0	6
2	0+10	2+6	6+0					0	10
3	0+16	2+10	6+6	14+0				0	16
4	0+21	2+16	6+10	14+6	20+0			0	21
5	0+25	2+21	6+16	14+10	20+6	22+0		4	26
6	0+29	2+25	6+21	14+16	20+10	22+6	24+0	3,4	30

Table 2.14: Optimal policies for stage three

Stage Four

The stage four follows the computational pattern as the proceedings patterns and its optimal policy table is presented in table 2.15 below:

y/n	0	1	2	3	4	5	6	$y_3(n)$	$g_3(n)$
0	0+0							0	0
1	0+6	5+0						0	6
2	0+10	5+6	9+0					2	11
3	0+16	5+10	9+6	13+0				0	16
4	0+21	5+16	9+10	13+6	17+0			0,1	21
5	0+26	5+21	9+16	13+10	17+6	21+0		1	26
6	0+30	5+26	9+21	13+16	17+10	21+6	25+0	1	31

Table 2.15: Optimal policies for stage four

We will at this last lap of computational stage, calculate the optimal route for the distribution of the six crates route from the general optimal solution network below. Firstly, we give the general table analysis for the stages.

Crates	$j=1$		$j=2$		$j=3$		$j=4$	
Available n	$y_1(n)$	$g_1(n)$	$y_2(n)$	$g_2(n)$	$y_3(n)$	$g_3(n)$	$y_4(n)$	$g_4(n)$
0	0	0	0	0	0	0	0	0
1	1	6	0	6	0	6	0	6
2	2	10	0,2	10	0	10	2	11
3	3	14	2	16	0	16	0	16
4	4	16	3	21	0	21	0,1	21
5	5	18	3,4	25	4	26	1	26
6	6	20	3,4	29	3,4	30	1	31

Table 2.16: General optimal policies for all stages

As a general analysis, observe that table (2.16) that when six crates are available for distribution to stores 4, 3, 2 and 1, the optimal store decisions is $y_4(6) = 1$. This implies that $n = 6 - 1 = 5$ crates are made available for the first three stores; hence the optimal Store 3 decision is $y_3(5) = 4$ crates. As a result, $n = 5 - 4 = 1$ crate is made available for the first two stores. Therefore, the optimal Store 2 decision is $y_2(1) = 0$, so that the optimal store 1 decision is $y_2(1) = 1$. The associated total profit is $g_4(6) = 31(= 6 + 0 + 20 + 5)$.

The information in table 2.16 is helpful for sensitivity analysis. For example, suppose that after the crate is dispatched to store 4, one of the remaining five crates is destroyed, thereby leaving only four crates for distribution to Stores 3, 2, and 1. Verify that the optimal distribution is $y_3(4) = 0,\ y_2(4) = 3$ and $y_1(1) = 1$, with smaller profit of 26.

Conclusion of chapter two

To conclude, always employ the recursion formulas to obtain the optimal solutions by beginning the computations at the final stage $j = 1$ and $g_1(0),\ g_1(1), \ldots\ldots\ldots\ldots, g_1(N)$. Then continue to finding $g_2(0),\ g_2(1), \ldots\ldots\ldots\ldots, g_2(N)$. You will then proceed in this same fashion for successively larger values of j until you finally found $g_s(N)$. You will discover that an actual optimal allocation by tracing back, beginning with $y_s(N)$, to obtain the values of y_j that together yields $g_s(N)$.

In the next chapter three, we shall treat two typical problems that would necessitate our derivation of the recursion formulas for capital budgeting and inventory production with smoothing models. The optimal analysis of these two models shall form the bedrock of chapter four

CHAPTER THREE

BUDGETTING CAPITAL AND INVENTORY MODELS

3.1 Introduction

In this chapter, we shall focus on the theoretical aspects of the budgeting capital investment and inventory models. The fourth chapter will obtain the numerical solutions based on these dynamical programming models. To this effect, networks analysis shall form the bases for the diagnosing of these essential models as the preceded chapter with its typical working examples.

3.2 The budgeting capital theoretical problem

In this section, you study the capital budgeting problem of a certain imaginary corporation. First we pose the problem of the corporation.

3.2.1 Statement of the problem

Consider a certain utopia corporation with the given problem of annually budgeting several million pounds for land development, and for building shopping centres, complexes, and industrial parks. The corporation is now planning to invest up to #10 million on one or more of three large projects. The data for these projects are contained in table (3.1) below:

Investment level y	Project 2		Project 5		Project 4	
	Cost $I_2(y)$	Value $R_2(y)$	Cost $I_3(y)$	Value $R_3(y)$	Cost $I_4(y)$	Value $R_4(y)$
0	0	0	0	0	0	0
1	3	8	4	9	6	17
2	5	13	5	13	7	18
3	7	18	6	18	8	21
4	8	19	9	19	9	22
5	9	21	10	23	10	24

Table 3.1: The utopia Corporation data problem; Investment cost in units #1000 000 and present values in units of #100 000.

From the table, observe that each of the three projects can be developed at any of the five different investment levels. For example, the corporation can choose to invest #3million, #5million, #7million, #8million, or #9million in project 2. If the investment choice is level 1, namely, #3million for project 2, then present value of future earnings is estimated to be $R_2(1) = \#0.8\text{million}$; if, instead, the investment choice is level 5, namely #9million, then the value for earnings rises to $R_2(5) = \#2.1\,\text{million}$. A similar interpretation applies for the other two projects.

3.2.2 Theoretical derivation of feasible recursive formula

The corporation also has the option of investing its resources in short-term securities. For expository convenience, let this option be designated as project 1 and assume that the commensurate economic returns of investing y million pounds are $R_1(y) = 2y$ hundred of thousand pounds. A short-term security investment can be at any set $y = 0,1,2,3,..........,10$.

For notational purposes, let $I_j(y)$ represent the investment cost of project j when the investment level $y = 2$ for project 4 requires an expenditure cost $R_2(y)$ =#7 million. You can use the same notation for project 1 by defining $I_1(y) = y$. Then the mathematical formulation of the utopia corporation problem is

$$\max imize \sum_{j=1} R_j(y_j) \qquad 3.1$$

subject to

$$\sum_{j=1}^{s} I_j(y_j) = K \text{ (available capital)} \qquad 3.2$$

$$y_j = 0,1,\ldots\ldots\ldots\ldots\ldots,10 \qquad 3.3$$

where y_j denotes the investment level for each project, $s = 4$ is the number of project $K = 10$ is the amount of available capital for investment. (As you will see, it is not necessary to add the restrictions $y_j \le 5$ for $j = 2,3,4$, and so these upper bounds are left out.

The network route is constructed in the figure 3.1 below. One column of the nodes appears for each project. The node designation is (k, j), where j refers to the project, and the value of k signifies an amount of capital available for possible investment in projects $1,2,3,\ldots\ldots\ldots\ldots, j$. Each arc leading out of Node (k, j) represents the decision about project j. For example, the arc from Node (6, 2) to Node (3, 1) represents the decision to invest $y_2 = 6 - 1 = \#3\,million$ in project 2 when capital available to spend on projects 2 and 1 is $\#6\,million$; the result of the decision is to leave $\#3\,million$ to spend on project 1. Hence according to figure 3.1, the value associated with this arc is $R_2(3) = 18$.

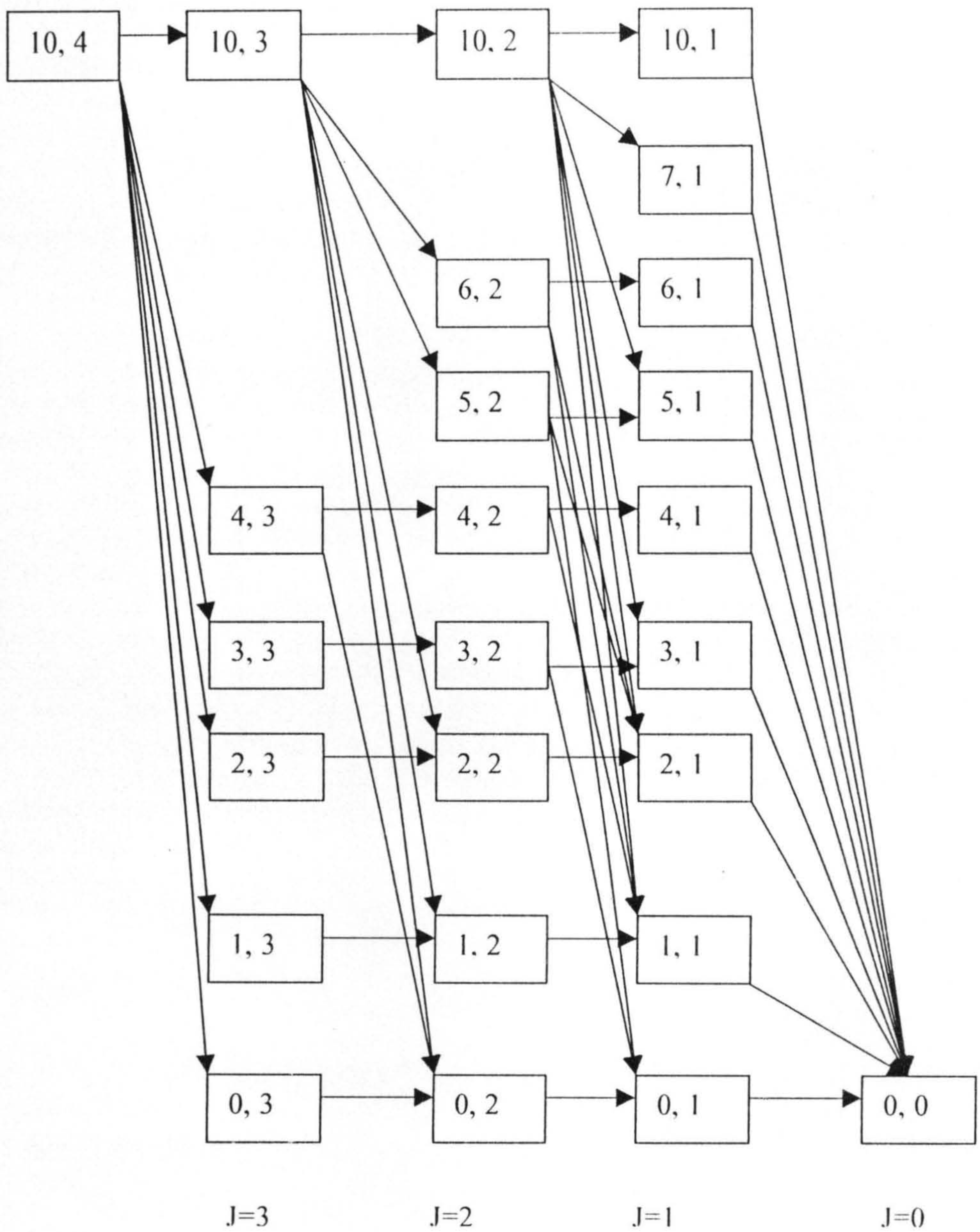


Figure3.1: Utopia Capital Network Representation

Figure 3.1 gives an exemplary effect of nonlinear function $I_j(y)$. Consider Node (10, 4); only six arcs emanate, since there are only six possible levels of investment. The same is observation holds for Nodes (10, 3) and (10, 2. Only level 0 investment arcs emanate from Nodes (K, 3), with K=0, 1, 2, 3, because any positive levels of investment require an expenditure of at least #4million, which is not feasible from these nodes. Similarly, three arcs leave Node (6, 2) because only investment levels 0,1, 2 are feasible project 2 when the available capital is 6. To summarize, given a Node (k, j), the only outward arcs permitted are those for values of y such that $I_j(y) \leq k$.

Now, finding the optimal investment is tantamount in figure (3.1) to solving for a route from Node (10, 4) to the terminal Node (0, 0) that gives the maximum present value. This idea can be captured by means of a dynamic programming recursive formula. To achieve this therefore, define

$$g_j(k) = (\textit{present value when capital}, k, \textit{is available to invest optimally on project } 1, \textit{project } 2, \textit{project } 3, \ldots\ldots\ldots\ldots, \textit{project } j, \ldots\ldots) \quad 3.4$$

$$y_j(k) = (\textit{an investment level for project } j \textit{ that yields } g_j(k)) \quad 3.5$$

In terms of network of figure 3.1, the quantity $g_j(k)$ is the present value of an optimal route from Node (0, 0).

The procedure for finding a best route assigns the value $g_0(0) = 0$, the puts $g_1(k) = R_1(k)$, and for each $j > k$, determines an arc that maximizes the sum of the immediate profit impact $R_j(y)$ and the profit from continuing optimally from the appropriate node at the next stage. To illustrate, at stage $j = 2$, the present value of a best route to the terminal from Node (5, 2) is calculated by

$$g_j(k) = \max_y\{R_1(0) + g_1(5), R_1(1) + g_1(2), R_1(2) + g_1(0)\} \quad 3.6$$

Each of the sums on the right of (3.6) is associated with an arc leading out of Node

(5, 2) and into either node (5, 1) or Node (0, 1). The general dynamic programming recursion is

$$g_j(k) = \max_y\{R_j(y) + g_{j-1}[k - I_j(y)]\}; \quad j = 1,2,................s \qquad 3.7$$

$$g_0(k) = 0; \qquad for\ j = 0. \qquad 3.8$$

where $k = 0,1,..........,K$, and the maximization is over only nonnegative integer values of y that satisfy $I_j(y) \leq k$.

We defer the numerical computation of the optimal plan to the next chapter as our numerical analysis.

3.3 The inventory model with production smoothing

In this section, we survey the inventory model. Thus, let us concept what is inventory through a definition

Definition 3.1 (Inventory)

Richard & Govindasami (1997) define in this manner:

Inventory is an idle stock of items for future use. The two key issues in inventory models are the quantity (how much) and the timing (when) of the orders.

3.3.1 The statement of the problem

In this problem, we are considering the inventory holding costs together with an assigned explicit smoothing cost incurred by changing production at different period. Here, we consider a hypothetical manufacturing productive company, with given quantified amount of inventory storage at given periodic levels. For each period t, the cost incurred depends on the production quantity x_t, the ending inventory level i_t and the previous period's production quantity x_{t-1}:

$$C_t(x_t, i_t, x_{t-1}) = C(x_t) + 1.i_t + 1.(x_t - x_{t-1})^2 \qquad 3.11$$

with the cost of productivity defined as fixed in this wise:

$$C(0) = 0,\ C(1) = 15,\ C(2) = 17,\ C(3) = 19,\ C(4) = 21,\ C(5) = 23 \qquad 3.12$$

for all periods. In equation (3.11), the third function to the right is a quadratic expression representing the smoothing cost; which is given by the square of the fluctuation in production levels over two successive periods. In this illustration, the cost impact of a variation in production is quadratic, and an upward fluctuation of a given magnitude is as costly as a downward fluctuation of the same magnitude. The required demand and the feasible production and inventory levels are

$$D_t = 3 \qquad (\textit{stationary demand}) \qquad 3.13$$

$$x_t = 0, 1, 2, \ldots\ldots\ldots, 5 \qquad i_t = 0, 1, \ldots\ldots\ldots\ldots, 4 \textit{ and no ending inventory} \qquad 3.14$$

3.3.1 The recursive formulation equation

Suppose we assume that the knowledge of the entering inventory is not sufficient to characterize the state of the system at the beginning of a period, then you will also need to know the production level in the previous period because this quantity affects the costs incurred in the current period. Hence, the state variable for this model must contain both the levels of entering inventory and the previous period's production. The problem still can be characterized as the finding of a least-cost route through an acyclic network. The network consists of a set node for each stage, but each node within a set represents possible values for both entering inventory and the previous period's production level. The appropriate node designation is (i, y, n), where y is the production level in the previous period. An arc out of Node (i, y, n) represents a feasible decision for current production x, has the associated cost $C(x)+1.(i+x-3)+1.(x-y)^2$, and leads into Node $(i+x-3, x, n-1)$.

In the present network, the network contains 18 nodes for $n=1$ and 21 nodes for $n \geq 2$. *Representing this network pictorially is a bit complicated as the arcs would cost aberration to viewing sight due to their jammed inter-connectedness in the network.*

Now if the planning horizon is N periods, the inventory level entering the initial period is i_0 and the production level immediately prior to the initial period is x_0, then the network optimization problem is to find a least-cost path from Node (i_0, x_0, N) to the terminal Node $(0,0,0)$. The familiar logic underlying a best route computation can be characterized by a recursive formula. Toward that end, let

$$f_n(i,y) = (\min imum\ policy \cos t\ entering\ inventory\ is\ at\ level\ i\ and\ previous\ production\ at\ level\ y\ with\ n\ more\ periods\ to\ go)$$

$$x_n(i,y) = a\ production\ level,\ yielding\ f_n(i,y)$$

The appropriate recursion can be written as

$$f_n(i,y) = \min_x imum[C(x)+1.(i+x-3)+1.(x-y)^2 + f_{n-1}(i+x-3, x)] \qquad 3.14$$

for $n = 1,2,\ldots\ldots\ldots\ldots\ldots, N$, where $y = 0,1,2,\ldots\ldots\ldots 5$ is a nonnegative integer in the range $y-3 \leq i \leq \min imum(4, 1+y)\ for\ n \geq 2$, and the minimization is over only nonnegative integer in the range $3-i \leq x \leq \min imum(5, 7-i)$. The computations are initiated with the values of

$f_1(i,y) = C(3-i)+1.(3-i-y)^2$ for $y = 0,1,2,\ldots\ldots\ldots 5$ and $x_1(i,y) = 3-i$. 3.15

and I is nonnegative integer in the range $y-3 \leq i \leq \min imum(3, 1+y)$.

In chapter four, we shall calculate the optimal policies and display the findings in tabular forms.

CONCLUSION OF CHAPTER THREE

In this chapter, we have derived the recursive formulas for budgeting capital distributions and the Inventory with smoothing models. We shall obtain their optimal policies as obtained in our literature review chapter in the next chapter. These optimal solutions for the two models shall serve as the analytical chapter for computational and data analysis chapter.

CHAPTER FOUR

COMPUTATIONAL ANALYSIS OF THE CAPITAL BUDGETING AND INVENTORY MODELS

4.1 Introduction

In this chapter, we shall obtain the optimal solutions to our two model problems, vis-à-vis, the capital budgeting and the inventory production models. To begin with, we analyze the utopia capital budgeting problem.

4.2 The Capital budgeting problem

Let us consider the capital budgeting table planning table 3.1; reproduced below for convenience.

Investment level y	Project 2		Project 3		Project 4	
	Cost $I_2(y)$	Value $R_2(y)$	Cost $I_3(y)$	Value $R_3(y)$	Cost $I_4(y)$	Value $R_4(y)$
0	0	0	0	0	0	0
1	3	8	4	9	6	17
2	5	13	5	13	7	18
3	7	18	6	18	8	21
4	8	19	9	19	9	22
5	9	21	10	23	10	24

Table 4.1: The utopia Corporation data problem; Investment cost in units #1000 000 and present values in units of #100 000.

Now, we optimal plan for the table 4.1 using the recursion (3.7) also produce for accessibility below:

$$g_j(k) = \max_y \{R_j(y) + g_{j-1}[k - I_j(y)]\}; \quad j = 1,2,\ldots\ldots\ldots\ldots\ldots, s \qquad 3.7$$

$$g_0(k) = 0; \qquad for\ j = 0. \qquad 3.8$$

where $k = 0,1,\ldots\ldots\ldots\ldots, K$, and the maximization is over only nonnegative integer values of y that satisfy $I_j(y) \leq k$.

Should we start with the project 1, the computation of $g_1(k)$ is straightforward, because the corporation invests in project 1 all the capital k that remains after its investments in the other projects:

$$g_1(k) = 2k \ \ and\ \ y\ (k) = k \ \ for\ \ k = 0,1,2,\ldots\ldots\ldots,10. \qquad 4.1$$

We omit tabulating these values because they are so readily computed. Thus continue with $j = 2$ to derive the computational scheme shown in table 4.1. There is a row in the table for each possible state, that is, for each amount of capital available. Rows for the values $k = 7,\ 8,\ 9$ have been omitted because they could never arise, as shown graphically in the network in figure 3.1. In the table 4.1, entries in which $I_2(y) \leq k$ fails to hold, have been shaded because these combinations of y and k are not feasible.

$R_2(y) + g_1[k - I_2(y)]$

Investment level stage n=2

y/x	0	1	2	3	4	5	$y_2(k)$	$g_2(k)$
0	0+0						0	0
1	0+2						0	2
2	0+4						0	4
3	0+6	8+0					1	8
4	0+8	8+2					1	10
5 6	0+1 0	8+4	13+ 0				2	13
10	0+1 2	8+6	13+ 2				2	15
	0+2 0	8+8	13+ 10	18+ 6	19+ 4	21+ 0	3	24

Table 4.2: Utopia Corporation $(j = 2)$

The first number of each entry inside the table is the value of $R_2(y)$, taken from table 4.1. The second entry is explained as follow:

Consider the entry for k=5 and y=1. At investment level $y = 1$, table 3.1 shows that the investment cost is $I_2(1) = 3$, which must be netted out of the available capital 5, thus leaving the amount k=2 for investment in project 1. According to equation (4.1), the second $g_1(2) = 2 * 2 = 4$, and so 4 appears as the second number in the table. In short, the second number is the quantity $g_1[k - I_2(y)]$, which is found by first calculating $k - I_3(1)$ from the quantities k, y, and the information in table 3.1, and then using the formula in (4.1).

The computational processes that gave us table (3.1) go as follow:

$$g_2(0) = \max(R_2(0) + g_2(0 - I_2(0), R_2(1) + g_2(0 - I_2(1)]$$

$$= \max(0 + 0, \infty) = 0;\ y_2(0) = 0\ ;\ g_2(0) = 0$$

$$g_2(1) = \max(R_2(0) + g_2(1 - I_2(0), R_2(1) + g_2(1 - I_2(1)]$$

$$= \max(0 + 2, 8 + g_1(-3);]$$

$$= \max(0 + 2, \infty] = 2,\ y_2(1) = 0 \qquad g_2(1) = 2$$

$$g_2(2) = \max(R_2(0) + g_2(2 - I_2(0), R_2(1) + g_2(2 - I_2(1)]$$

$$= \max(0 + 4, 8 + g_1(-1);]$$

$$= \max(0 + 4, \infty] = 4,\ y_2(2) = 0 \qquad g_2(2) = 4$$

$$g_2(3) = \max(R_2(0) + g_2(3 - I_2(0), R_2(1) + g_2(3 - I_2(1), R_2(2) + g_2(3 - I_2(2)]$$

$$= \max(0 + 6, 8 + 0, 3 + g_2(-2)]$$

$$= \max([0 + 6, 8 + 0, \infty] = 8;\ y_2(3) = 1;\ g_2(3) = 8$$

$$g_2(4) = \max(R_2(0) + g_2(4 - I_2(0), R_2(1) + g_2(4 - I_2(1), R_2(2) + g_2(4 - I_2(2)]$$

$$= \max(0 + 8, 8 + 2, 13 + g_2(-1)]$$

$$= \max([0 + 8, 8 + 2, \infty] = 8;\ y_2(4) = 1;\ g_2(4) = 10$$

$$g_2(5) = \max(R_2(0) + g_2(5 - I_2(0), R_2(1) + g_2(5 - I_2(1), R_2(2) + g_2(5 - I_2(2)]$$

$$= \max(0 + 10, 8 + 4, 13 + g_2(0)]$$

$$= \max([0 + 10, 8 + 4, 13 + 0] = 8;\ y_2(5) = 2;\ g_2(5) = 13$$

$$g_2(6) = \max(R_2(0) + g_2(6 - I_2(0), R_2(1) + g_2(6 - I_2(1), R_2(2) + g_2(6 - I_2(2), R_2(3) + g_2(6 - I_2(3)]$$

$$= \max(0 + 12, 8 + 6, 13 + g_2(1), 18 + g_2(-1)]$$

$$= \max([0 + 12, 8 + 6, 13 + 2, \infty] = 15;\ y_2(6) = 2;\ g_2(6) = 15$$

$$g_2(10) = \max(R_2(0) + g_2(10 - I_2(0), R_2(1) + g_2(10 - I_2(1), R_2(2) + g_2(10 - I_2(2), R_2(2) + g_2(10 - I_2(2), R_2(3) + g_2(10 - I_2(3), R_2(4) + g_2(10 - I_2(4), R_2(5) + g_2(10 - I_2(5)]$$

$$= \max(0 + 20, 8 + 6, 13 + 10, 18 + 6{:}19 + 4{:}21 + 2]$$

$$= \max([0 + 12, 8 + 6, 13 + 2, \infty] = 15;\ y_2(10) = 4;\ g_2(10) = 24$$

Thus with $g_2(10) = 24$, we are through with stage n=2.

Stage n=3

With the same technical computational processes as in stage n=2 and using the same recursion formula, the following numerical values for stage n=3 are reproduced:

$$g_3(0) = \max[R_3(0) + g_2(0 - I_3(0)]$$

$$= \max[0 + 0] = 0;\ y_3(0) = 0;\quad g_3(0) = 0$$

$$g_3(1) = \max[R_3(0) + g_2(1 - I_3(0), R_3(1) + g_2(1 - I_3(1)]$$

$$= \max[0 + 2] = 0;\ y_3(1) = 0;\quad g_3(1) = 2$$

$$g_3(2) = \max[R_3(0) + g_2(2 - I_3(0), R_3(1) + g_2(2 - I_3(1), R_3(2) + g_2(2 - I_3(2)]$$

$$= \max[0 + 4] = 0;\ y_3(2) = 0;\quad g_3(2) = 4$$

$$g_3(3) = \max[R_3(0) + g_2(3 - I_3(0), R_3(1) + g_2(3 - I_3(1), R_3(2) + g_2(3 - I_3(2); R_3(3) + g_2(3 - I_3(3)]$$

$$= \max[0 + 8] = 8;\ y_3(3) = 0;\quad g_3(3) = 8$$

$$g_3(4) = \max[R_3(0) + g_2(4 - I_3(0), R_3(1) + g_2(4 - I_3(1), R_3(2) + g_2(4 - I_3(2); R_3(3) + g_2(4 - I_3(3)]$$

$$= \max[0 + 10, 9 + 0] = 10;\ y_3(4) = 0;\quad g_3(4) = 10$$

$$g_3(5) = \max[R_3(0) + g_2(5 - I_3(0), R_3(1) + g_2(5 - I_3(1), R_3(2) + g_2(5 - I_3(2); R_3(3) + g_2(5 - I_3(3)]$$
$$= \max[0 + 13, 9 + 2, 13 + 0] = 13;\ y_3(5) = 0,2;\quad g_3(5) = 13$$

$$g_3(6) = \max[R_3(0) + g_2(6 - I_3(0), R_3(1) + g_2(6 - I_3(1), R_3(2) + g_2(6 - I_3(2);$$
$$R_3(3) + g_2(6 - I_3(3)]$$

$$= \max[0 + 15, 9 + 4, 13 + 2] = 15;\ y_3(6) = 0,2;\quad g_3(6) = 15$$

$$g_3(10) = \max[R_3(0) + g_2(10 - I_3(0), R_3(1) + g_2(10 - I_3(1), R_3(2) + g_2(10 - I_3(2);$$
$$R_3(3) + g_2(10 - I_3(3); R_3(4) + g_2(10 - I_3(4), R_3(5) + g_2(10 - I_3(5)]$$
$$= \max[0 + 20, 9 + 14, 13 + 10, 18 + 8, 19 + 4, 21 + 2] = 26; y_3(10) = 2;\quad g_3(10) = 26$$

$$R_2(y) + g_2[k - I_2(y)]$$

Investment level stage n=3

y/x	0	1	2	3	4	5	$y_3(k)$	$g_3(k)$
0	0+0						0	0
1	0+2						0	2
2	0+4						0	4
3	0+8						0	9
4	0+10	9+0					0	10
5	0+13	9+2	13+0				0,2	13
6	0+15	9+4	13+2				0,2	15
10	0+24	9+15	13+13	18+4	19+2	23+0	2	26

Table 4.3: Utopia Corporation $(j = 3)$

Stage n=4

We are now in the final stage of our computational processes which is on project 4. Here goes the computational procedure:

$$g_4(0) = \max[R_4(0) + g_3(0 - I_4(0); R_4(1) + g_3(0 - I_4(1); R_4(2) + g_3(0 - I_4(2);$$
$$R_4(3) + g_3(0 - I_4(3); R_4(5) + g_3(0 - I_4(5)]$$

$$= \max[0 + 0] = 0;\ y_4(0) = 0;\ g_4(0) = 0$$

$$g_4(1) = \max[R_4(0) + g_3(1 - I_4(0); R_4(1) + g_3(1 - I_4(1); R_4(2) + g_3(1 - I_4(2);$$
$$R_4(3) + g_3(1 - I_4(3); R_4(5) + g_3(1 - I_4(5)]$$

$$= \max[0 + g_3(1); 17 + g_3(-3), 18 + g_3(-6), ..., 24 + g_3(-9)]$$

$$= \max[0 + 2; \infty; \infty; \infty; ....; \infty]$$

$$= \max[0 + 2] = 2;\ y_4(1) = 0;\ g_4(1) = 2$$

$$g_4(2) = \max[R_4(0) + g_3(2 - I_4(0); R_4(1) + g_3(2 - I_4(1); R_4(2) + g_3(2 - I_4(2);$$
$$R_4(3) + g_3(2 - I_4(3); R_4(5) + g_3(2 - I_4(5)]$$

$$= \max[0 + g_3(2); 17 + g_3(-4), 18 + g_3(-6), ..., 24 + g_3(-8)]$$

$$= \max[0 + 4] = 4;\ y_4(2) = 0;\ g_4(2) = 4$$

$$g_4(3) = \max[R_4(0) + g_3(3 - I_4(0); R_4(1) + g_3(3 - I_4(1); R_4(2) + g_3(3 - I_4(2);$$
$$R_4(3) + g_3(3 - I_4(3); R_4(5) + g_3(3 - I_4(5)]$$

$$= \max[0 + g_3(3); 17 + g_3(-3), 18 + g_3(-5), ..., 24 + g_3(-7)]$$

$$= \max[0 + 8] = 8;\ y_4(3) = 0;\ g_4(3) = 8$$

$$g_4(4) = \max[R_4(0) + g_3(4 - I_4(0); R_4(1) + g_3(4 - I_4(1); R_4(2) + g_3(4 - I_4(2);$$
$$R_4(3) + g_3(4 - I_4(3); R_4(5) + g_3(4 - I_4(5)]$$

$$= \max[0 + g_3(4); 17 + g_3(-2), 18 + g_3(-4), ..., 24 + g_3(-6)]$$

$$= \max[0 + 10, \infty, \infty, ..., \infty] = 10;\ y_4(4) = 0;\ g_4(4) = 10$$

$$g_4(5) = \max[R_4(0) + g_3(5 - I_4(0); R_4(1) + g_3(5 - I_4(1); R_4(2) + g_3(5 - I_4(2);$$
$$R_4(3) + g_3(5 - I_4(3); R_4(5) + g_3(5 - I_4(5)]$$

$$= \max[0 + g_3(5); 17 + g_3(-1), 18 + g_3(-3), ..., 24 + g_3(-5)]$$

$$= \max[0 + 13, \infty, \infty, ..., \infty] = 13;\ y_4(5) = 0;\ g_4(5) = 13$$

$$g_4(6) = \max[R_4(0) + g_3(6 - I_4(0); R_4(1) + g_3(6 - I_4(1); R_4(2) + g_3(6 - I_4(2);$$
$$R_4(3) + g_3(6 - I_4(3); R_4(5) + g_3(6 - I_4(5)]$$

$$= \max[0 + 15); 17 + 0, 18 + g_3(-2), ..., 24 + g_3(-4)]$$

$$= \max[0 + 15, 17 + 0, \infty, ..., \infty] = 13;\ y_4(6) = 1;\ g_4(5) = 17$$

$$g_4(10) = \max[R_4(0) + g_3(10 - I_4(0); R_4(1) + g_3(10 - I_4(1); R_4(2) + g_3(10 - I_4(2);$$
$$R_4(3) + g_3(10 - I_4(3); R_4(5) + g_3(10 - I_4(5)]$$
$$= \max[0 + g_3(10),; 17 + g_3(4), 18 + g_3(3), 21 + g_3(2), 22 + g_3(1), 24 + g_3(0)]$$

$$= \max[0 + 24, 17 + 10, 18 + 8, 21 + 4, 22 + 2, 24 + 0] = 27;$$
$$y_4(10) = 1;\ g_4(10) = 27$$

$$R_2(y) + g_2[k - I_2(y)]$$

Investment level stage n=4

y/x	0	1	2	3	4	5	$y_4(k)$	$g_4(k)$
0	0+0						0	0
1	0+2						0	2
2	0+4						0	4
3	0+8						0	9
4	0+10						0	10
5	0+13						0	13
6	0+15	17+0					1	17
10	0+24	17+10	18+8	21+4	22+2	24+0	1	27

Table 4.4: Utopia Corporation $(j = 4)$

4.3 Decisional analysis

The stages have been numerically obtained using acyclic route computational processes. Stage 4 is shown in table 4.3. In the table, the optimal investment plan has a present value of 27. The plan itself is found by starting in table 4.3 with decision $y_4(10) = 1$ for project 4; that is, the corporation undertakes level 1 investment at a cost of #6 million pounds. This leaves #4 million to be allocated to project 3, 2 and 1. Now, we move to table 4.2, where $j=3$, you find that $y_3(4)=0$, so that project 3 is bypassed. Next, refer to table 4.1 where $j=2$. The corporation still has #4 million to allocate, and you ascertain that $y_2(4)=1$; this level 1 investment in project 2 requires an outlay of #3 million, leaving only #1 million for project 1. As a check on the recursive computations, note that the total present value from those decisions is $(17+0+8+2=27)$, which agrees with $g_4(10)$.

Now that the analytical computational processes have been executed and decisions taken on this section, we now proceed to the section of utopia corporation inventory model with production smoothing model.

4.4 The inventory model with production smoothing analysis

In section 3.3, we laid down a rigorous foundation on the problem of inventory model with production smoothing. We intend to do computational analysis on the given problem that we may take worthwhile model decisions. To do so, we reproduce the recursive formula for the problem hereunder:

$$f_n(i,y) = \min_x imum[C(x) + 1.(i+x-3) + 1.(x-y)^2 + f_{n-1}(i+x-3,x)] \qquad 4.4.1$$

where $y=0,1....,5$, i is a nonnegative integer in the range $y-3 \leq i \leq \min imum(4,1+y)$ for $n \geq 2$, and the minimization is over only nonnegative integer values in the range $y-3 \leq i \leq \min imum(5,7-i)$. The computations are initiated with the values of

$$f_n(i,y) = \min_x imum[C(x) + 1.(i+x-3) + 1.(x-y)^2]$$

4.4.2

$for\ y = 0,1,2,3,4,5\ and\ x_i(i,y) = 3-i$

and I is a nonnegative integer in the range $y-3 \le i \le \min imum(3,3+y)$. The calculations for n=1 using equations (4.4.2) are as follows:

when i=0

$$f_1(0,0) = C(3) + 1.(3-0-0)^2$$
$$= 19 + 9 = 28; \; x_1(0,0) = 3$$

$$f_1(0,1) = C(3) + 1.(3-0-1)^2$$
$$= 19 + 4 = 23; \; x_1(0,1) = 3$$

$$f_1(0,2) = C(3) + 1.(3-0-2)^2$$
$$= 19 + 1 = 20; \; x_1(0,2) = 3$$

$$f_1(0,3) = C(3) + 1.(3-0-3)^2$$
$$= 19 + 0 = 19; \; x_1(0,3) = 3$$

$$f_1(0,4) = C(3) + 1.(3-0-4)^2$$
$$= 19 + 1 = 20; \; x_1(0,4) = 3$$

when i=1

$$f_1(1,0) = C(2) + 1.(3-1-0)^2$$
$$= 17 + 4 = 21; \; x_1(1,0) = 2$$

$$f_1(1,1) = C(2) + 1.(3-1-1)^2$$
$$= 17 + 1 = 18; \; x_1(1,1) = 2$$

$$f_1(1,2) = C(2) + 1.(3-1-2)^2$$
$$= 17 + 0 = 17; \; x_1(1,2) = 2$$

$$f_1(1,3) = C(2) + 1.(3-1-3)^2$$
$$= 17 + 1 = 18; \; x_1(1,3) = 2$$

$$f_1(1,4) = C(2) + 1.(3-1-4)^2$$
$$= 17 + 4 = 21; \; x_1(1,4) = 2$$

when i=2

$$f_1(2,0) = C(1) + 1.(3-2-0)^2$$
$$= 15 + 1 = 16; \; x_1(2,0) = 1$$

$$f_1(2,1) = C(1) + 1.(3-2-1)^2$$
$$= 15 + 0 = 15; \quad x_1(2,1) = 1$$

$$f_1(2,2) = C(1) + 1.(3-2-2)^2$$
$$= 15 + 1 = 16; \quad x_1(2,2) = 1$$

$$f_1(2,3) = C(1) + 1.(3-2-3)^2$$
$$= 15 + 4 = 19; \quad x_1(2,3) = 1$$

$$f_1(2,4) = C(1) + 1.(3-2-4)^2$$
$$= 15 + 9 = 24; \quad x_1(2,4) = 1$$

$$f_1(2,5) = C(1) + 1.(3-2-5)^2$$
$$= 15 + 16 = 31; \quad x_1(2,5) = 1$$

$$f_1(3,0) = C(0) + 1.(3-3-0)^2$$
$$= 0 + 0 = 0; \quad x_1(3,0) = 0$$

$$f_1(3,1) = C(0) + 1.(3-3-1)^2$$
$$= 0 + 1 = 1; \quad x_1(3,1) = 0$$

$$f_1(3,2) = C(0) + 1.(3-3-2)^2$$
$$= 0 + 4 = 4; \quad x_1(3,2) = 0$$

$$f_1(3,3) = C(0) + 1.(3-3-3)^2$$
$$= 0 + 9 = 9; \quad x_1(3,3) = 0$$

$$f_1(3,4) = C(0) + 1.(3-3-4)^2$$
$$= 0 + 16 = 16; \quad x_1(3,4) = 0$$

$$f_1(3,5) = C(0) + 1.(3-3-5)^2$$
$$= 0 + 25 = 25; \quad x_1(3,5) = 0$$

Below is the general decision table 4.4 for n=1 stage recursion computation

Entering Inventory i	Y=0		y=1		y=2		y=3		y=4		y=5	
	$X_1(i,y)$	$f_1(i,y)$	$X_1(i,y)$	$f_1(i,y)$	$X_1(i,y)$	$f_1(i,y)$	$X_1(i,y)$	$f_1(i,y)$	$X_1(i,y)$	$f_1(i,y)$	$X_1(i,y)$	$f_1(i,y)$
0	3	28	3	23	3	20	3	19				
1	2	21	2	18	2	17	2	18	2	21		
2			1	15	1	16	1	19	1	24	1	31
3					0	4	0	9	0	16	0	25

Table 4.5: Inventory production smoothing model (n=1)

Stage n=2

When n=2, the recursion formula (4.4.2) enable us to obtain the initial computational values of $f_1(i,y)$ and the production quantity $x_i(i,y)$. Now, the recursion formula for the computation of other stages $n \geq 2$ is the recursion equation (4.4.1).

We wish to express the fact that for n=2, the third entry in each box of the main part of the table 4.5 below comes from table 4.4. For example if y=0, $i=1$ and $i=3$, then the amount 18 is the value of $f_1(i+x-3,x) = f_1(1,3)$,contained the $i=1$ row and y=3 right column of table 4.4, Now, we begin the recursion computational processes:

$$f_n(i,y) = \min_x imum[C(x) + 1.(i+x-3) + 1.(x-y)^2 + f_{n-1}(i+x-3,x)]$$

Now, with n=2, we derive:

$$\begin{aligned} f_2(0,0) = \min[&[(0+(-3)) + ((0-0)^2 + f_1(-3,0)],[(15+(-2)) + ((1-0)^2 + f_1(-2,1)], \\ &[(17+(-1)) + ((2-0)^2 + f_1(-1,12],[(19+(0*3-3)) + ((3-0)^2 + f_1(0,3)], \\ &[(21+1)) + (16^2 + f_1(1,4)],[23+2]25 + f_1(2,5)] \end{aligned}$$

$$f_2(0,0) = \min[\infty, \infty, \infty, 19+9+19, 22+16+21, 25+25+31] = 47; x_2(0,0) = 3, f_2(0,0) = 47$$

$$f_2(1,0) = \min[[C(0)+(-2))+((0-0)^2+f_1(-2,0)],[(15+(-1))+((1-0)^2+f_1(-1,1)],$$
$$[(17+(0))+((2-0)^2+f_1(0,2],[(19+(1))+((3-0)^2+f_1(1,3)],$$
$$[(21+2))+(16^2+f_1(2,4)],[23+3]25+f_1(3,5)]$$

$$f_2(1,0) = \min[\infty, \infty, 17+4+20, 20+9+18, 23+16+24, 26+25+35] = 41;$$
$$x_2(1,0) = 2, f_2(1,0) = 41$$

As demonstrated in the above two iterates, it is not difficult to follow the same traditional pattern of computations to yield the n=2 stage table 4.5 underneath:

$$[C(x)+1.(i+x-3)+1.(x-y)^2+f_{n-1}(i+x-3,x)]$$

The recursive formula is used as always to obtain the needed data for determining the table appearing below:

Previous Production: y	Entering Inventory i	0	1	2	3	4	5	$x_i(i,y)$	$f_i(i,y)$
0	0				19+9+19	22+16+21	25+25+31	3	47
	1			17+4+20	20+9+18	23+16+24	26+25+25	2	41
1	0				19+4+19	22+9+21	25+16+31	3	42
	1			17+1+20	20+4+18	23+9+24	26+16+25	2	38
	2		15+0+23	18+1+17	21+4+19	24+9+16		2	36
2	0				19+1+19	22+4+21	25+9+31	3	39
	1			17+0+20	20+1+18	23+4+24	26+9+25	2	37
	2		15+1+23	18+0+17	21+1+19	24+4+16		2	35
	3	0+4+28	16+1+18	19+0+16	22+1+9			0,3	30.
3	0				19+0+19	22+1+21	25+4+31	3	38
	1			17+1+20	20+0+18	23+1+24	26+4+25	2,3	38
	2		15+4+23	18+1+17	21+0+19	24+1+16		2	36
	3	0+4+28	16+1+18	19+1+16	22+0+9			3	31
	4	1+9+21	17+4+15	20+1+4				2	25
4	1			17+4+20	20+1+18	23+0+24	26+1+25	3	39
	2		15+9+23	18+4+17	21+1+19	24+0+16		2	39
	3	0+16+28	16+9+18	19+4+16	22+1+9			3	32
	4	1+16+21	17+9+15	20+4+4				2	28
5	2		15+16+23	18+9+17	21+4+19	24+1+16		4	41
	3	0+25+28	16+16+19	19+9+16	22+4+9			3	35
	4	1+25+21	17+1+15	20+9+4				1,2	33

Table 4.6: Inventory production smoothing model (n=2)

The calculation for n=3 are shown in table 4.7. The format for this table also applies to larger values of n. The first number in each sum in the table represents the production, inventory and smoothing costs; these numbers will not change in tables for larger n. The second number in each sum is the cost of an optimal policy for the remaining stages; these numbers will change in tables for larger n.

Several illustrative optimal schedules for the planning horizon N=3 are given in table 4.7. Suppose the first month of the horizon is January, the previous month's level is $i_0 = 0$, and entering inventory is $i_0 = 1$. Then, referring to the second row in table 4.6 for (n=3, y=0, i=1), you find that an optimal production decision is $x_3(1,0) = 2$. Consequently, inventory entering Februarys (stage n=2) is $0(= i_0 + x - d = 1 + 2 - 3)$, and the optimal production decision is $x_2(0,2) = 3$ from the sixth row of table 4.5. This decision implies that March's entering inventory is 0, so that March's production level is $x_1(0,2) = 3$ from table 4.4. The total cost over the horizon is $f_3(1,0) = 50$.

Suppose, instead, that the previous month's production level is $x_0 = 4$ and entering inventory is $x_0 = 3$, then, $x_3(3,4) = 4$ in table 4.6 gives an optimal January decision. As a result, February's entering inventory is $4(= 3 + 4 - 3)$, and hence, the optimal production level in February is $x_2(4,4) = 2$ from table 4.5, and in March it is $x_2(3,2) = 0$ from table 4.4. The total cost over the horizon is $f_3(3,4) = 53$.

Revious Production: y	Entering Inventory i	0	1	2	3	4	5	$x_t(i,y)$	$f_t(i,y)$
0	0				28+38	38+39	50+41	3	66
	1			21+39	29+38	39+39	51+35	2	60
1	0				23+38	31+39	41+41	3	61
	1			18+39	24+38	32+39	42+35	2	57
	2		15+38	195+37	25+36	33+32	43+33	1	53
2	0				20+38	26+39	34+41	3	58
	1			17+39	21+38	27+39	35+35	2	56
	2		16+42	18+37	22+36	28+32	36+33	2	55
	3	4+41	17+38	19+35	23+31	29+25		0	45.
3	0				19+38	23+39	29+41	3	57
	1			18+39	20+38	24+39	30+35	2	57
	2		19+42	19+37	21+36	25+32	31+33	2	56
	3	9+47	20+38	20+35	22+31	26+28		3	53
	4	10+41	21+36	21+32	23+25			3	48
4	1			21+39	21+38	23+39	27+55	3	59
	2		24+42	22+37	222+36	24+32	28+33	4	56
	3	16+47	25+38	23+35	23+31	25+28		4	53
	4	17+41	26+36	24+32	24+25			3	49
5	2		31+42	27+37	25+32	25+32	29+33	2	57
	3	25+47	32+38	28+35	26+28	26+28		4	54
	4	26+41	18+36	29+32	27+25			3	52

Table 4.7: Inventory production smoothing model (n=2)

Observe the impact of the smoothing cost on the production schedules of table 4.8. For comparison, the analogous optimal schedules when the Utopia Corporation does not pay a smoothing cost are also known in Table 4.8. The smoothing cost reduces the amount of fluctuation in production levels from one period to the next. Observe that a peak production of $x = 5$ is never optimal. Note that in case $i_0 = 2$ *and* $x_0 = 4$, the production quantities are the same with and without smoothing costs, but the impact of smoothing factor causes the production to take place in two consecutive months, January and February.

Initial Inventory i_0	With smoothing- Previous production						Without smoothing		
	$x_0 = 0$			$x_0 = 4$					
	Jan.	Feb.	Mar.	Jan.	Feb.	Mar.	Jan.	Feb.	Mar.
0	3	3	3				4	5	0
1	2	3	3	3	3	2	5	0	3
2				4	3	0	4	0	3
3				4	2	0	0	3	3
4				3	2	0	0	5	0

Table 4.8: Inventory production smoothing model (n=3)

Conclusion of chapter four:

This chapter has indeed been quite hectic in both computational analytical processes. In the chapter, we have exposed the computational processes for both capital distribution of projects and the inventory production smoothing of two different corporations. The optimal capital was securely obtained; inventory production smoothing gives better estimate of production than without smoothing factor.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

This research has been a study of the one-dimensional constraints problems using the multistage methods. Thus, having seen a variety of dynamic programming application and explored a few in depth, it is necessary we summarize the essential features in the dynamical programming models.

The dynamic programming approach attacks an optimization problem with multifold constraints and many variables by splitting the problem into sequence of stages in which lower-dimension optimization takes place. In contrast, most linear and nonlinear programming approaches attempt to solve such problems by considering all the constraints simultaneously.

The dynamic programming approach casts a problem into the following structures:

(i) The decision variables with their associated constraints are grouped according to stages, and the stages are considered sequentially.

(ii) The information about previous stages relevant to selecting optimal values for current decision variables is summarized by a so-called state variable, which may be n-dimensional.

(iii) The current decision, given the present state of the system, has forcastable influence on the state at the next stage

(iv) The optimality of the current decision is judged in terms of its forecasted economic impact on the present stage of the system and all subsequent stages.

5.2 Recommendation

The research has been based on one constraint problem. Therefore, we suggest exploitation on the two constraints. In view of this, let us pose a two constraint problem and leave it for an onward further research accomplishment; the model can be written as follows:

(A) Consider the problem

$$\max imize \sum_{j=1}^{s} c_j y_j \qquad 5.1$$

Subject to

$$\sum_{j=1}^{s} a_{ij} y_j = b_i \quad for\ i = 0,1,2,3,........k \qquad 5.2$$

$$y_j = 0,1,2,3,............for\ each\ j$$

where each a_{ij} and b_i is a nonnegative.

(a) Formulate a dynamic programming recursion that appropriately generalizes

$$\begin{aligned} g_j(w) &= [R_j y + g_{j-1}(w - w_j y)] \ for\ j = 1,2,......N \\ g_0(0) &= 0 \ \ for\ \ j = 0 \end{aligned} \qquad 5.3$$

(b) Explain how optimization problem can be characterized by the recursion

$$F(n_1, n_2,................n_k) = \max imum[c_j + F(n_1 - a_1, n_2 - a_2,................n_k - a_k)] \qquad 5.4$$

where the maximization is over each value of $j = 1,2,3,..........,s$ that satisfies $a_{ij} \le n_j$ for every $j = 1,2,..........,k$.

© Explain why the recursion in part (b) is appropriate when linearity assumptions are dropped; for example, suppose that the objective function is

$$\sum_{j=1}^{5} c_j(y_j)$$

And at least one of the functions $c_j(y_j)$ is nonlinear.

Comment on the computational burden of applying the recursion (a) and (b) to problems of moderate size.

(B) Use the formulation in part (b) to solve the problem

$$\max imize\ 3y_1 + 4y_2 + 6y_3 + 8y_4 + 6y_5 \qquad 5.5$$

subject to

$$1y_1 + 2y_2 + 2y_3 + 3y_4 + 1y_5 \leq 3 \qquad 5.6$$

$$2y_1 + 1y_2 + 2y_3 + 1y_4 + 3y_5 \leq 4 \qquad 5.7$$

every y_j a nonnegative integer

(b) By how much does the objective function decrease when the right-hand-side constant in the constraint is 2, instead of 3? When the right- hand-side in the second constraints is 3, instead of 4? When the right-hand-side constants in both first and second constraints are decreased by 1?

(c) By how much does the objective function increase when the right-hand-side constant in the constraint is 4, instead of 3? When the right- hand-side in the second constraints is 5, instead of 4? When the right-hand-side constants in both first and second constraints are decreased by 1?

REFERENCES

Chase and Aquilano Initially (2000): Production & Operations Management (A life Cycle Approach, 4th Edition).

Everett, E. Adam, Jr. & Ronald J. Ebert (1993): Prentice- Hall of India Private Limited, New Delhi – 110001.

Frederick S. Hillier & Gerald J. Lieberman (2001): Introduction to Operations Research. McGraw-Hill Companies, Inc. (Seventh Edition)

Rao, S. S. (1978): Optimization theory and Applications. Wiley Eastern Limited.

Richard, B and Govindasami, N. (1997): Operations Research (Second Edition). Schaum's outline series McGraw-Hill

Wargner, H. M. (1989): Principles of Operations Research (Second edition): Prentice- Hall of India Private Limited, M.97 Connaught Circuss. New Delhi 110001.

Wisniewski, M. and Darcy, T. (1992): Mathematical programming optimization models for business and management decision-making. McGraw-Hill Book company.