KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY
KUMASI

INSTITUTE OF DISTANCE LEARNING

A BUS REPLACEMENT MODEL FOR THE STATE TRANSPORT COMPANY, KUMASI.

THESIS SUBMITTED TO KNUST MATHEMATICS DEPARTMENT IN

PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD

OF

MASTER OF SCIENCE DEGREE IN INDUSTRIAL MATHEMATICS

BY

KPARIB YEN-WON DOUGLAS (B Ed)

DECEMBER 2009.

DECLARATION

I hereby declare that, except for the specific references, which have been duly acknowledge, this work is the result of my own field research and it has not been submitted either in part or whole for any other degree elsewhere.

Kparib Yen-won Douglas	…………………..	………….
PG1836107		
Student name & ID	Signature	Date

Certified by:

Mr. Kwaku Darkwah	…………………..	………………
Supervisor	Signature	Date

Certified by:

Prof E. Badu	…………………..	………………
Dean of IDL	Signature	Date

Certified by:

Dr. S.K Amponsah	………………..	…………….
Head of Department	Signature	Date

ABSTRACT

The State Transport Company, like many service organizations, faces the problem of how long a bus should be on the road before it is replaced.

The aim of this thesis is therefore to determine a schedule of disposals and replacements of the Higher bus, taking into account the revenue generated, operating cost and the salvage values, such that the total cost of these activities is minimized.

Data was collected from the State Transport Company Office in Kumasi on the revenue generated, operating cost, and the salvage values on the bus with time.

The problem was solved by using dynamic programming.

It was found out that the company should always dispose its buses when they are two years old.

DEDICATION

I give thanks and praises to the Most High God on whose grace alone bring all good things into fruition.

This work is dedicated to my parents: Mr. Bane Kurug and Mrs. Ndambil Tilbire my brother Kparib Peter and not forgetting my wife Mrs. Kparib Elizabeth and children Mavis, Leticia and Ebenezer Kparib and anyone who contributed to the success of this project.

ACKNOWLEDGEMENT

I would like to express my sincere appreciation primarily to my supervisor Mr. Kwaku Darkwah for his patience and tolerance, which he exhibited, in supervising this project.

My deepest gratitude also goes to my friends especially, Mr. Amoako Samuel and Ameyaw Eric (KNUST), and others for their interaction and advice.

My immense appreciation also goes to Mr. Appiah Kubi (manager, S.T C., Kumasi), Mr. Conrad Halm Lutterodt (Senior Service Engineer, S T C) and Mr. Quaye Enoch (Accountant, S T C) who together acted as resource persons and provided me with vital information concerning the research.

TABLE OF CONTENTS

LIST OF FIGURES

Figure *Page Number*

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

1.1 HISTORICAL BACKGROUND OF THE STUDY

Dynamics Programming (DP) is a simple yet powerful approach for solving certain types of sequential optimization problems. Most real-life decision problems are sequential (dynamic) in nature since a decision made now usually affects future outcomes and payoffs. An important aspect of optimal sequential decisions is the desire to balance present costs with the future costs. A decision made now that minimizes the current cost only without taking into account the future costs may not necessarily be the optimal decision for the complete multi-period problem [Mahmut, 2000].

The State Transport Company, like many service organizations, faces the problem of how long a bus should be on the road before it is replaced. The method of dynamic programming model will be used to solve this problem.

1.1.1 TRANSPORTATION.

Transport or transportation is the movement of people and goods from one place to another. The term is derived from the Latin ‘trans’ (across) and ‘portare’ (to carry). Industries, which have the business of providing transport equipment, transport services or transport, are important in most national economies, and are referred to as transport industries.

The field can be divided into infrastructure, vehicles, and operations.

(i) Infrastructure consists of the fixed installations necessary for transport, and may be roads, railways, airways, waterways, canals and pipelines or terminals such as airports, railway stations, bus stations and seaports.

(ii)Vehicles traveling on the network include automobiles, bicycles, buses, trains, people and aircraft.

(iii)Operations deal with the way the vehicles are operated, and the procedures set forth for this purpose include the financing, legalities and policies Martin, (1997).

1.1.2. TYPES OF TRANSPORTATION

(i) Human-power transport

Human-powered Transport is the transport of person(s) and /or goods using human muscle power. Like animal-power transport, human-power transport has existed since time immemorial in the form of walking, running and swimming.

(ii) Animal-power

Animal-powered Transport is the use of working animals (also known as "beasts of burden") for the movement of people and goods. Humans may ride some of the animals directly, use them as pack animals for carrying goods, or harness them, singly or in teams, to pull(or haul) sleds or wheeled vehicles. Animals are superior to people in their speed, endurance and carrying capacity. Prior to the industrial revolution, they were used for all land transport, impracticable for people, and they remain an important mode of transport in less developed areas of the world.

(iii)Air (Aviation)

A fixed-wing aircraft, commonly called airplane or aeroplane, is an aircraft where movement of the wings in relation to the aircraft is not used to generate lift. The term is used to distinguish it from rotary-wing aircraft. The movement of the lift surfaces relative to the aircraft generates lift. A heliplane is both fixed-wing aircraft ranging from small trainers and recreational aircraft to large airliners and military cargo aircraft. Two necessities for aircraft are airflow over the wings for lift, and an area for landing.

The majority of aircraft also need an airport with infrastructure to receive maintenance, restocking, refueling and for the loading and unloading of crew, cargo and passengers. While the vast majority of aircraft land and take off on land, some are capable of taking off and landing on ice, snow and calm water.

The aircraft is the second fastest method of transport, after the rocket. Commercial jets can reach up to 875 kilometers per hour, single-engine aircraft 175 km per hour. Aviation is able to quickly transport people and limited amounts of cargo over longer distances, but incur high costs and energy use; for short distance or in inaccessible places helicopters are used.

(iv) Rail transport

Rail transport is the transport of passengers and goods along railways(or railroads), consisting of two parallel steel rails, generally anchored perpendicular to beams(termed sleepers or ties) of timber, concrete or steel to maintain a consistent distance apart, or gauge.

The Greeks constructed stone railways in the 6th century BC, while the first iron rails were laid in 1768 with steam engine introduced in 1804(Martin, 1997). High-speed rail was introduced by Shinkansen in 1964 (Martin, 1997). Rail transport remains the most energy efficient land transport, and is used for long distance freight and all distances of passenger transport. In cities, rapid transit and Trans are common parts of public transport.

(v) Water (Ship) transport

Ship transport is the process of transport by barge, boat, ship or sailboat over a sea, ocean, lake, canal or river. A watercraft is a vehicle designed to float on and move across (or under) water. The first craft were probably types of canoes cut out from tree trunks.

In the 1800s the first steam ships were developed using a steam engine to drive a paddle wheel or propeller to move the ship.

The steam is produced using wood or coal. Now most ships have an engine using a slightly refined type of petroleum called bunker fuel. Transport by water is less costly than air transport for trans-continental shipping. Short sea shipping and ferries remain viable in coastal areas.

(vi) Pipe-line transport

Pipeline transport sends goods through a pipe. Most commonly, liquid and gas are sent but pneumatic tubes can send solid capsules using compressed air.

(vii) Cable transport

Cable Transport is a broad mode where vehicles are pulled by cables instead of an internal power sources. It is most commonly used at steep gradient. Typical solutions include aerial tramway, elevators, escalator and ski lifts; some of these are categorized as convey or transport (Stopford et al, 1997).

(viii) Road transport

The most common road vehicle is the automobile, a wheeled passenger vehicle that carries its own motor. Other users of roads include buses, trucks, motorcycles, bicycles and pedestrians. As of 2002, there were 590 million automobiles worldwide. The first forms of road transport were horses, oxen or even humans carrying goods over dirt tracks that often followed game trails. John Loudon Mcdam designed the first modern highways of inexpensive paving material of soil and stone aggregate known as Macadam during the industrial revolution. Coating of Cobblestones and wooden paving were popular during the 19th century while tarmac and concrete paving became popular during the 20th.

(ix) Inter-modal freight transport and inter-modal passenger transport

Inter-modal freight transport is the combinations of multiple modes of transportation for single shipment containers. It allows seamless integration of sea, rail and road transport and have reduced transshipment costs.

Inter-model passenger transport is a journey performed by using several modes of transport; since all human transport normally starts and ends with walking, all passenger transport can be considered inter-modal.

Public transport may also involve the intermediate change of vehicle, within or across modes, at transport hub, such as a bus or railway station (Lay, et al, 1992).

1.1.3 ROAD TRANSPORT SERVICE PROVIDERS IN GHANA

Looking at the demand of transportation service in Ghana, there are many transport service providers to meet the demands. These are:

(i) Ghana Private Road Transport(G.P.R.T.U)

Ghana Private Road Transport Union (G.P.R.T.U) was established in 1966 as a commercial private road transport service providers union. The main aim is to provide for the welfare of its members and passengers.

G.P.R.T.U has it National Secretarial Office in Accra, which is made up of National Chairman, National Vice Chairman, General Secretary, two Deputies General Secretaries, the first and second Trustees, Finance Officers.

It also has Regional and District Secretarial Offices in all the ten (10) Regions and the Districts capitals respectively. At the Regional Secretarial Office we have the; Regional Chairman, Vice Chairman, Secretary, first and second Trustees.

The other officers are the Industrial Relation Officer, Finance Officer, Security Department, Mediation and Arbitration department, Welfare department.

Members of GPRTU enjoy the following benefits: (i) legal services (ii) loans (iii) accident and sick policies. It also assists its members to purchase buses. This is done in collaboration with PHC Motors of Ghana Ltd, Agricultural Development Bank, Prudential bank (urban buses) and Stanbic bank etc. Members are made to pay some percentage of the total cost of the vehicle and the remaining amount is paid by installment.

Ghana Private Road Transport Union has stations in all the major towns and villages in Ghana. At the station, we have the Chairman, Vice Chairman, Secretary, Treasure, Financial Secretary, Organizer, and Discipline Committee, Welfare committee, Supervisor, Book-man or Sales-girl.

Apart from Ghana GPRTU also operates in these West Africa countries; Togo, Cote d’Ivoire, Benin, Burkina Faso and Nigeria [Khaleepha, 2008].

(ii) Progressive Transport Owners Union (PROTOA)

It is a split group from G.P.R.T.U. Like the G.P.R.T.U, they also have National Chairman, National Vice Chairman, General Secretary, two Deputies General Secretaries, and the first and second Trustees, Finance Officers.

It also has Regional and District Secretarial Offices in all the ten (10) Regions and the Districts capitals respectively. The PROTOA operates in all the towns and villages in the country. It members also enjoy the same benefits like G.P.R.T.U members.

(iii) Metro Mass Transport Service

The government funds this. The Metro buses operate in all the ten regions of Ghana. The company has offices in all the regional capitals.

(iv) O.A Transport Service

The Director General is Mr. Opoku Agyemang. The headquarters is in Kumasi and has Offices in all the ten (10) Regions and the Districts capitals respectively. It operates in many of the major towns in Ghana.

(v) M-Plaza Transport Service

They have offices only in the regional capitals of Ghana but operate in all the towns and villages in the country.

1.2 BACKGROUND OF STUDY TO THE STATE TRANSPORT COMPANY

The State Transport Company (ST C).

The Intercity S T C began in 1909 as a government transport department to cater for the needs of central government. In 1965, it was made a body corporate by legislative instrument L I414 of 9th March, 1965 to run commercial passenger services and was then called the State Transport Corporation.

In January 1968, the government also created a haulage division to cater for the haulage of both wet and dry cargo. It was handed over to STC to manage as a bulk haulage division, to function alongside the passenger division.

It was later incorporated in June, 1995 as a Limited Liability Company under Ghana's companies code 1962 Act 179 in the name State Transport Company Limited.

In June, 2000 the assets of the company were purchased and organized into new company, Vanef STC by a consortium of business concerns known as the Vanef Consortium.

This was after the STC was put on divestiture under the Ghana government's systematic program to make previously owned state institutions more viable by divesting them to worthy private investors.

The Vanef consortium is a parent Limited Liability holding company that oversees the operations of the two main business units:

(i) the Passenger Service Division (PSD), and

(ii) the VCL haulage division. These business units are jointly and severally owned by the government of Ghana and the Vanef consortium with the consortium owning majority shares in all the units. Vanef STC took over the running of business of the erstwhile State Transport Company (STC) on June 1, 2000.

In 2003, the name of the company was changed to Intercity STC Coaches Limited.

The board and management comprise individuals of varied and rich backgrounds all bringing their expertise together to make Intercity STC a supremely successful corporate entity, dominating her chosen markets with superior products and services.

(iii) Routes

Intercity S T C has the farthest reach and widest network on the local market.

With the exception of Koforidua, intercity STC runs services to and from all Regional Capitals in Ghana. The Company's parcels service, the package Express, also delivers package to all of these destinations. Apart from Regional Capitals Intercity STC also runs services to and from other local destinations, and three (3) West Africa countries namely Cote d'Ivore, Benin and Burkina faso.

The Company can boast of nineteen (19) local and three (3) international stations.

These include:

Accra, Kumasi, Takoradi, Tamale, Bolgatanga, Bawku, Tema, Tudu, Aflao, Berekum, Cape-coast, Dormaa-Ahenkro, Fambisi, Hohoe, Ho, Kpandu, Paga, Sunyani,Wa, Abidjan, Ouagaougou and Cotonou. Intercity STC currently operates in four countries (4): Ghana, Cote d'Ivoire, Burkina faso and Benin. The Company also operates through Togo to and from Benin, although it does not have a station in Togo.

Maintenance workshops are strategically located in the major cities, are mandated to inspect, and maintains all company buses passing through those cities.

Information technology is an integral component of STC. I T systems support the purchasing of tickets, scheduling of buses, processing of traffic information, operations planning, payroll, finance functions, monitoring of revenue and other functions. (Intercity STC, 2004).

CREW ON A BUS

Many buses need at least the driver, mate and second driver.

STC has a station and a workshop in Kumasi. The station is located at Adum, where both loading and off-loading of passengers and freight is done. Here, we have the station manager, book-men or sale-girls and the parcel section.

The workshop is positioned at Oforikrom, where we have the Regional manager, accountant, principal operations officer, administrator and engineer. This is where all maintenances work on the buses is done. Any bus from a journey reports at the workshop after off loading the passengers.

1.3. STATEMENT OF THE PROBLEM

According to the engineer of STC there is no policy on the number of years a bus should be used after which it is disposed of. They only dispose off their buses when they are no longer road worthy. That is when the buses are from four to six years old.

Since STC has no documented policy for the disposal of their buses, the problem is that the company could be running at a loss as the buses are kept up to the fourth year and sixth year before they are disposed.

1.4. OBJECTIVE OF STUDY

The objectives of the study are:

(1) to develop a bus replacement policy for the State Transport Company and

(2) recommend the findings to STC.

1.5 METHODOLOGY

The research will be that of direct approach and hence information obtained will be through interviews aimed at obtaining responses to the following questionnaires; the type of buses they used, cost of new bus, age (yr), operating cost, and salvage value of a bus after years of usage.

The method of dynamic programming model will be used. Dynamic programming is concerned with making a sequence of interrelated decisions. Dynamic programming (DP) is a simple yet powerful approach for solving certain types of sequential optimization problems. A personal hand calculator will be used for the calculations. Search on the internet was used to obtain relate literature.

The main Library at KNUST and the department of mathematics Library was consulted in the course of the project.

1.6. THESIS ORGANIZATION

Chapter one covers the historical background of transport service, brief discussing of the methodology and the objective of the study.

Chapter two contains the literature review and methods.

Chapter three contains the data collection, analysis and discussion.

Chapter four covers the conclusion and recommendation.

CHAPTER TWO

INTRODUCTION

This chapter looks at optimization techniques, recursive algorithm, dynamic programming- literature survey and application to various problems, literature survey of replacement models.

2.1 OPTIMIZATION TECHNIQUES

Optimization techniques are designed to maximize profit or minimize cost of any business operation. There are specialized techniques under the optimization model for specific problems.

The models include:

(i) Linear Programming: It is best handled by the simplex algorithm, and also solves linear models. Linear programming (LP) is a technique for optimization of a linear objective function, subject to linear equality and inequality constraints. Linear Programming determines the way to achieve the best outcome (such as maximum profit or minimum cost) in a given mathematical model, given some list of requirements represented as linear equations (Alexander,1998).

(ii) Integer Programming: It solves the same mathematical model as that of linear programming, but with the additional restriction that some of the decision variables must have integer values.

(iii) Dynamic programming: Dynamic programming works on the principle of finding an overall solution by operating on an intermediate point that lies between where we are now and where we want to go. The procedure is recursive in that the next intermediate point is a function of the point already visited.

A prototypical problem that is suitable for dynamic programming has the following properties:

(a) The problem can be decomposed into a sequence of decisions made at various stages.

(b) Each stage has a number of possible states.

(c) A decision takes one from a state at one stage to some state at next stage.

(d) The best sequence of decision also known as policy at any stage is independent of the decisions made at prior stages.

(f)There is a well-defined cost for traversing from state across stages. Moreover, there is a recursive relationship for choosing the best decision.

2.2. RECURSIVE PROGRAMMING

Dynamic programming and many useful algorithms are recursive in structure. To solve a given problem the algorithm calls a subroutine recursively one or more times to deal with closely related sub-problems. Recursive algorithms typically follow a divide-and-conquer approach in the sense that they break the problem into several sub-problems that are similar to the original problem but smaller in size, solve the sub-problems recursively, and then combine these solutions to create a solution to the original problem.

2.2.1. DIVIDE-AND-CONQUER ALGORITHM

The divide-and-conquer paradigm is a recursive algorithm and it involves three steps at each level of the recursion.

(i)Divide the problem into number of sub-problems.

(ii)Conquer the sub-problems by solving them recursively. If the sub-problem sizes are small enough, however, just solve the sub-problems in a straightforward manner.

(ii)Combine the solutions to the sub-problems into the solution for the original problem

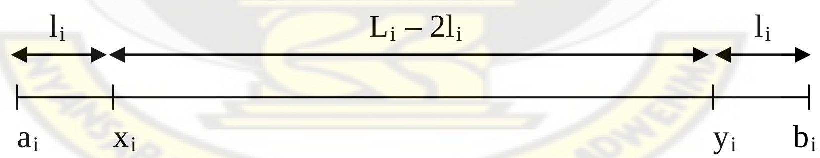
For example, consider the minimization problem below:

Minimize $f(x) = x^4 - 5x + 2$ subject to $-1 \leq x \leq 1$, by reducing the interval of uncertainty to less than 10% of the original and using the Fibonacci search algorithm with

$F_0 = F_1 = 1$; $F_{n-2} + F_{n-1} = F_n$, $n \geq 0$ (Amponsah, 2006). Thus F=[1,1,2,3,5,8,13,21,…].

We choose n such that $\frac{1}{F_n} < \frac{10}{100} = 0.1$. Now for n=6, $\frac{1}{F_n} = \frac{1}{13} = 0.076923 < 0.1$ and hence we shall make six applications of the Fibonacci numbers as follows:

Let [a, b]=[-1,1], then $L_1 = b_1 - a_1 = 1 + 1 = 2$. Using the formula $l_i = \frac{F_{n-(i+1)}}{F_{n-(i-1)}} L_i$ calculate the interval of reduction l_i such that the point x_i and y_i divide L_i into three sections with $x_i = a_i + l_i$ and $y_i = b_i - l_i$, $i = 1, 2, \ldots$



Evaluate f(xi) and f(yi) and select the point that gives the minimum evaluation.

For sub problem $i=1$, $n=6$, $L_1 = 2$.

Using the formula $l_i = \frac{F_{n-2}}{F_n} L_i$, $l_1 = \frac{5}{13} \times 2 = 0.76923$.

Hence $x_1 = a_1 + l_1 = -1 + 0.76923 = -0.23077$ and $y_1 = b_1 - l_1 = 1 - 0.76923 = 0.23077$.

$f(x_1)$=3.15668, $f(y_1)$=0.848985. Since $f(x_1) > f(y_1)$ we discard $[a_1, x_1]$ and set $a_2 = x_1$, $b_2 = b_1$.

For subproblem *i=2*

Put $[a_2, b_2]$=[-0.23077, 1] and L2 = 1 + 0.23077 = 1.23077.

$$l_2 = \frac{f_{6-3}}{f_{6-1}} L_2 = \frac{3}{8} \times 1.23077 = 0.461538.$$

Hence $x_2 = a_2 + l_2$ = -0.23077+0.461538 = 0.23077 and $y_2 = b_2 - l_2$ = 10.461538 = 0.538462, $f(x_2)$ = 3.15668 and $f(y_2)$ = -0.608245.

Since $f(x_2)>f(y_2)$ we discard the interval $[a_2, x_2]$ and put $[x_3, b_3]$=[0.23077, 1].

For subproblem *i=3*

$L_3 = b_3 - a_3$ = 0.76923 and $l_3 = \frac{F_{6-(3+1)}}{F_{6-(3-1)}} \times L_3 = \frac{2}{5} \times 0.76923$=0.307692.

Hence $x_3 = a_3 + l_3$ = 0.229719 + 0.3076923 = 0.538462 = y_2 and $y_3 = b_3 - l_3$ = 1-0.3076923 = 0.692308.

$f(x_3)$ = -0.608245, $f(y_3)$ = -1.23182, since $f(x_3) > f(y_3)$ we discard the interval $[a_3, x_3]$ and put $[a_4, b_4]$=[0.58462, 1].

Continuing we have the as final interval of uncertainty to be and min.$f(x_6)$ = -1.71814 occurs as x_6 = 0.846154 (Amponsah, 2006)

The current interval of uncertainty becomes the search domain for the solution of the current subproblm. After the solution of each subproblem the interval of uncertainty is reduced further.

The interval of uncertainty is a division of the original domain and that solution in that interval is the conquest of the interval. This continues until we obtain the final interval of uncertainty that satisfies the formulation condition. The optimal solution to the original problem is then determined. Thus the optimal solution is a conquest of the final part division of the original domain by use of the recursive formula.

2.2.2. GREEDY ALGORITHM

A greedy algorithm is a recursive algorithm that follows the problem solving heuristic approach and makes locally optimal choice at each stage in the computation with the hope of finding the global optimum.

An optimization problem can be solved by greedy algorithm, if the problem has two ingredients (properties):

(i) Greedy choice property

The first key ingredient is the greedy-choice property: a globally optimal solution can be arrived at by making a locally optimal (greedy) choice. In order words when we are considering which choice to make, we make the choice that looks best in the current stage of the problem, without considering the results from subsequent choices to be made. Here is where greedy algorithms differ from dynamic programming. In dynamic programming, we make a choice at each step, but the current choice usually depends on the solutions to previous sub problems. Consequently, we typically solve dynamic programming problems in a bottom-up manner, progressing from smaller sub problems to larger sub problems. In a greedy algorithm, we make whatever choice seems best now and then solve the sub problem arising after the choice has been made. The choice made by a greedy algorithm may depend on choices so far, but cannot depend on any future choices or on the solutions to other sub problems. Thus, unlike dynamic programming, which solves the sub problems bottom up, greedy strategy usually progresses in a top-down fashion, making one greedy choice after another, reducing each give problem instance to a smaller one.

(ii) Optimal substructure.

A problem exhibits optimal substructure if an optimal solution to the problem contains within it optimal solutions to subproblems. This property is a key ingredient of assessing the applicability of dynamic programming as well as greedy algorithm. Optimal substructure varies across problem domains in two ways:

(a) how many subproblems are used in the process of computing the optimal solution to the original problem, and

(b) how many choices we have in determining which subproblem(s) to use in an optimal solution process.

Example

A customer went to the sorcery shop. He paid for the items bought and was to receive a change of 41 cents.

However, the sales clerk had the following denominations of coins:

(1) 25 cents (quarter) denominations.

(2) 10 cents (dime) denominations..

(3) 5 cents (nickel) denominations..

(4) 1 cent denominations.

The sales clerk is to give the minimum number of coins that will be equal the change of 41 cents.

The problem is which denominations should she select and how many coins of each selected denomination should be used to give the minimum of coins for the 41 cents change.

Greedy choice option: The coin with the highest denomination is chosen at each step.

Optimal sub structure: The problem of selection of a coin denomination is a sub problem. The choice of highest coin denomination possible is an optimal solution to the sub problem. Since the solution of the original problem is the count of the number of coin denominations selected at the various sub problems the problem possesses an optimal substructure.

Sub problem 1:

Select coin to reduce change of 41 cents.

Greedy solution: Choose highest coins denomination of 25 cents

Solution of sub problem 1 is 1 coin of 25 cents.

The remaining change is 41 - 25 = 16.

Sub problem 2:

Select coin to reduce 16 cents.

Greedy solution: Choose highest coins denomination of 10 cents.

Solution of sub problem 2 is 1 coin of 10 cents.

The remaining change is 16- 10 = 6.

Sub problem 3:

Select coin to reduce change of 6 cents.

Greedy solution: Choose highest coins denomination of 5 cents

Solution of sub problem 3 is 1 coin of 5 cents.

Remaining change: 6 - 5 = 1.

Sub problem 4:

Select coin to reduce change of 1 cents.

Greedy solution: Choose highest coins denomination of 1 cent

Solution of sub problem 1 is 1 coin of 1 cent.

Remaining change: 1 - 1 = 0.

The solution is that the sales clerk should give one 25 cents, one 10 cents, one 5 cents and one 1 cent as the change.

2.3. THE DYNAMIC PROGRAMMING METHOD

Dynamic programming is a general approach to solution of optimization problems and uses ingredients of recursive programming.

Richard Ernest Bellman first developed dynamic programming in the late 1950's (Bellman, 1957, Bellman and Kalaba 1965).

Bellman's principle of optimality states that 'An optimal policy (set of decisions) has the property that whatever the initial state and decisions are , the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision'.

It was first devised for time dependent optimization problem in which decisions had to be taken in sequence at various time (stages). However, it has been extended to solve constrained, linear or non-linear and integer optimization problems.

Bellman, described a way of solving problems where you need to find the best decisions one after another. In forty-odd years since this development, the number of applications of dynamic programming has increased enormously. The main characteristics of the model are that solutions are obtained in a sequence at various stages in the solution process and are dependent on the outcome in previous solution process. Each solution process results in a value and the ultimate aim of the process is maximizing some objective function (usually a sum or product terms) resulting from the individual solution processes (Smith et al, 1997).

Dynamic programming is applicable when the sub problems are not independent, that is, when sub problems share the results of other sub problems. In this context, a divide-and-conquer algorithm does more work than necessary, repeatedly solving the common sub sub problems. A dynamic programming algorithm solves every sub problem just once and then saves its answer in a table, thereby avoiding the work of re-computing the answer every time the sub problem is encountered.

Dynamic programming is typically applied to optimization problems. In such problems, there can be many possible solutions. Each solution has a value, and we wish to find a solution to the problem, as opposed to the optimal solution, since there may be several solutions that achieve the optimal value (Cormen, et al 2002).

The dynamic programming algorithm can be broken into a sequence of four steps:

(i) Characterize the structure of an optimal solution.

(ii) Recursively define the value of optimal solution

(iii) Compute the value of optimal solution in a bottom up fashion.

(iv) Construct an optimal solution from computed information.

2.4. DYNAMIC PROGRAMMING SURVEY

The term dynamic programming was originally used in the 1940s by Richard Ernest Bellman to describe the process of solving problems where one needs to find the best decisions one after another(Adda, et al, 2003).

Beaumont (2007) used dynamic programming to identify an optimal strategy to find solution to a contract bridge tournament. The contract bridge tournament comprises several rounds of matches in which players compete as pairs for ‘master points’ awarded for each match won or drawn and for being highly placed at the end of the tournament. In the second and subsequent rounds, pairs are matched against other pairs that have been approximately equally successful so far. The optimal strategy is a function of a pair’s ability.

Daniel (2007) used dynamic programming to find the best-scoring set of beat times that reflects the tempo as well as corresponding to moments of a high ‘onset strength’ in a function derived from audio. This very simple and computationally efficient procedure is shown to perform well on the MIREX-06 beat tracking training data, achieving an average beat accuracy of just under 60% on the development data.

Nicole and Quenez (1995) also used dynamic programming to determine a solution for the problem of pricing contingent claims or options from the price financial market. In this situation, there is a price range for the actual market price of the contingent claim. The maximum and minimum prices are studied using stochastic control methods. The main result of this work is the determination that the maximum price is the smallest price that allows the seller to hedge completely by a controlled portfolio of the basic securities. A similar result is obtained for the minimum price (which corresponds to the purchased price).

Bush, et al (1999) describes a compile-time analyzer that detects dynamic errors in large, real-world programs. The analyzer traces execution paths through the source code, modeling memory and reporting inconsistencies.

Zeqing and Shin (2006) introduced and studied properties of solutions for functional equations arising in dynamic programming of multistage decision processes.

Quansong, et al (2006) used dynamic programming to study the microbial community composition and its variations in environmental ecology. Clustering analysis of the Automated Ribosomal Interagency Spacer Analysis (ARISA) from different times based on the dynamic programming algorithm binned data revealed important features of the biodiversity of the microbial communities.

Slater (1964) uses dynamic programming to determine an optimal path from a number of alternatives paths, in order to move from a given initial state to a desired final position.

Norman and Clarke (2004) used stochastic dynamic programming model to examine the appropriateness of sending a lower order batsman into 'hold the fort' on a 'sticky wickets'. In cricket, a rain-affected pitch can make batting more difficult than normal. Several other conditions such as poor light or an initially lively pitch may also result in difficulties for the batsman. All these are referred to us 'sticky wickets'.

Mahmut (2000) used dynamic programming to get an optimal price for a car of a professor who had limited number of days to leave a country after his sabbatical leave.

2.5. BASIC CONCEPTS AND SOLUTION PARADIGM

(i) Stage variables denote times that decisions are made.

(ii)The State variables of a dynamic process completely specify the process and provide information on all that needs to be known in order to make a decision. Thus, State variables are a measure of the conditions that exist at the beginning of any stage of solution.

(iii) Decision variables (which may also be referred to as *control variables* or *policy variables*) are directly specified to obtain an optimal solution at each stage of a dynamic programming solution.

(iv) Stage Transformation Function: The rule or equation that relates the output state variable x_{n-1} for stage n to the input state variable x_n and the decision variable d_n.

(v)Return Function: A value (such as profit or loss) associated with making decision d_n at stage n for a specific value of the input state variable x_n.

These concepts are made use of by Varaiya, (1971), when he introduced the discrete-time dynamic programming a problem as

$$P1:\ Maximize \sum_{i=0}^{N-1} f_0(i, x(i), u(i)) + \Phi(x(N))$$

subject to

(i) dynamics: x(i+1)=f(i, x(i),u(i)), i=0,1,…,N-1,

(ii) initial condition: x(0) = $x_{0,}$

(iii) control $u(i) \in \Omega_i$, i=0,1,...,N-1.

In (*p1*), *i* is the stage variable and the state variable x(i) and control variable u(i) belong to arbitrary sets X and U respectively. X and U may be finite sets, or finite-dimensional vector spaces, or even infinite-dimension spaces, $x_0 \in X$ is fixed. The Ω_i are fixed subsets of U. Finally, $f_0(i,.,.)$: $X \times U \rightarrow R$, $\Phi : X \rightarrow R$, f(i,.,.): $X \times U \rightarrow X$ are fixed functions.

The main idea underlying DP involves embedding the optimal control problem(*p1*), in which the system starts in state x_0 at time zero, into a family of optimal control problems with the same dynamics, objective function, and control constraints as in (*p1*) but with different initial states and initial times. More precisely, for each $x \in X$ and *k* between zero and N-1, the kth sub problem is given by:

$$P2;\ \text{Maximize} \sum_{i=0}^{N-1} f_0(i, x(i), u(i)) + (\Phi x(N)),$$

subject to

(i) dynamics: x(i+1)=f(i, x(i),u(i)), i = k, k+1, …, N-1,

(ii) control $u(i) \in \Omega_i$, i=k, k+1,...,N-1.

Assume that an optimal solution to sub problem *p2* exists for all $0 \leq k \leq N-1$, and all $x \in X$.

Let V(k, x) be the maximum value of the objective of sub problem *p2*. We call V the (maximum) value function. Define V(N) at the Nth stage by $V(N, x) = \phi(x)$, then the maximum value function at the kth sub problem V(k,x) satisfies the backward recursion equation

$$V(k,x) = Max\{ f_0, (k,x,u) + V(k_1 f(k,x,u)) \setminus u \in \Omega_k \},\ 0 \leq k \leq N-1.$$

2.6. APPLICATIONS OF DYNAMIC PROGRAMMING TECHNIQUES

Dynamic programming can be used to solve integer programming problem that seeks to minimize or maximize an objective function value subject to some constraints. It can also be used as a search algorithm to find minimum or maximum distances and measures. A network search problems may not be formulated as integer programming problems. Thus dynamic programming has it own requirements and approach to optimization problems, which is different from linear programming, as stated earlier in the section above.

A prototypical problem that is suitable for dynamic programming has the following properties:

(i) The problem can be decomposed into a sequence of decisions made at various stages.

(ii) Each stage has a number of possible states.

(iii) A decision takes one from a state at one stage to some state at next stage.

(iv)The best sequence of decision also known as policy at any stage is independent of the decisions made at prior stages.

(v)There is a well-defined cost for traversing from state across stages. Moreover, there is a recursive relationship for choosing the best decision.

Dynamic programming has varied applications and solves problems in various fields.

We provide examples in the following section to illustrate some of the varied problems that dynamic programming can solve. These are;

(i) Production and inventory control problem,

(ii) The stagecoach problem (network problem),

(iii) Knapsack problem and

(iv) The equipment replacement problem.

The production and inventory control problem and the equipment replacement problem do not have integer programming formulation. The knapsack problem has integer programming formulation and the stagecoach problem is a network search problem.

The solution approach to all these problems will follow the four sequential dynamic programming steps even when the individual problem formulations are different.

The four sequence of steps used by dynamic programming in solving an optimization problem are detailed below as :

(i) Characterize the structure of an optimal solution.

(ii) Recursively define the value of optimal solution

(iii) Compute the value of optimal solution in a bottom up fashion.

(iv) Construct an optimal solution from computed information.

2.6.1. A PRODUCTION AND INVENTORY CONTROL PROBLEM

The problem is to minimize the sum of the production cost and inventory holding cost over a three-month period subject to demand, production capacity, warehouse capacity and inventory holding capacity. At any period, the ending inventory will be calculated as:

Ending inventory= beginning inventory +production – demand, during the period the total cost for each period is the sum of production cost and inventory holding cost for the month and is to be minimized for each period and over the entire duration.

There are three constraints:

The first constraint is that the ending inventory must be less than or equal to the warehouse capacity. The second constraint is that the production level in each period must not exceed the production capacity.

The third constraint is that the beginning inventory plus production must be greater than or equal to demand.

Suppose that we have developed forecasts of the demand for cars over three months and that we would like to decide upon a production quantity for each of the periods so that demand can be satisfied at a minimum cost. There are two costs to be considered: production costs and inventory holding costs. We will assume that production setup costs will be made each period and that setup costs will be constant. As a result, setup costs are not considered in the analysis.

We will allow the production and inventory holding costs to depend on quantity at hand and vary across periods. This makes our model more flexible since it also allows for the possibility of using different facilities for production and different storage capacity constraints, which may vary across periods (David, et, al, 1988).

Step 0: Variable definitions and data.

Let us adopt the following notation:

N= number of periods (stages in our dynamic programming formulation)

D_n=demand during stage n; n=1,2,…, N.

x_n=a state variable representing the amount of inventory on hand at the beginning of stage n; n=1,2,…,N.

d_n= decision variable for stage n. It is the production quantity for the corresponding period n:

P_n=production capacity in stage n:

W_n=storage capacity at the end of stage n;.

C_n=production cost per unit in stage n;

H_n=holding cost per unit of ending inventory for stage n.

In table 2.1 below, column one is the month, column two the demand(D_n) for the month, column three the production capacity(P_n), column four the storage capacity(W_n), column five the production cost per unit(C_n) and column six is the holding cost per unit(Hn) for the month.

Table 2.1: Data for the production and inventory control problem

Month	Demand(D_n)	production capacity(P_n)	storage capacity(W_n)	Production cost per unit($C_{n)}$	Holding cost per unit(H_n)
January	2	3	2	$175	$30
February	3	2	3	$150	$30
March	3	3	2	$200	$40

The beginning inventory for January is one unit.We will develop the dynamic programming solution for the problem covering N= 3 months of operation. These are January, February and March.

Step 1: The structure of an optimal solution.

Our first step in dynamic programming paradigm is to characterize the structure of an optimal solution.

We can think of each month in our problem as a stage in dynamic programming formulation.

In figure 1, stage 3 is January, stage 2 February and stage 1 March. The ending inventory of January is the beginning(x_2) of February and so on.

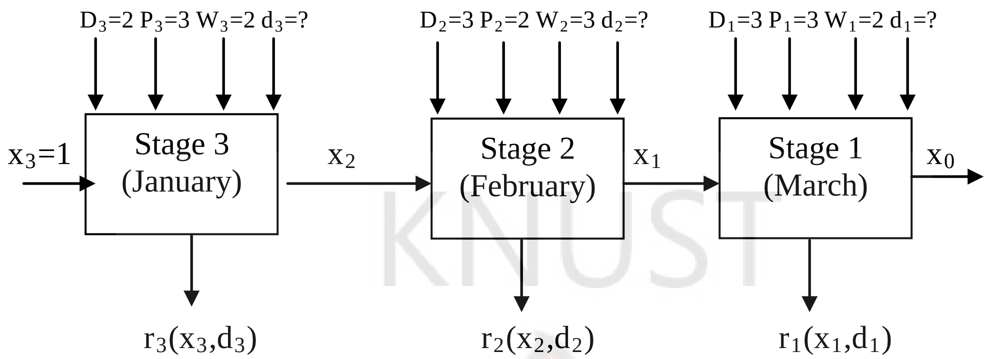


Figure 1. Schematic representation of the production and inventory control problem as a three-stage dynamic programming problem

(i)For stage 3(January)

We minimize the sum of production cost and inventory holding cost in the Month of January subject to demand (D_3=2), production capacity (P_3=3), warehouse capacity (W_3=2) and ending inventory, x_2. x_3=1 is beginning inventory.

The stage transformation function for the month January is of the form:

Ending inventory=beginning inventory (x_1=1)+production(P_3=3)-demand(D_3=2).

The return(objective) function for January is the sum of production and inventory holding costs in January and is given by $f_3(x_3)=\min.r_3(x_3,d_3)+f_2(x_2)$.

(ii) For stage 2 (February)

We minimize the sum of production cost and inventory holding cost in the Month of February subject to demand (D_2=3), production capacity (P_2=2), warehouse capacity (W_2=3) and ending inventory.

The stage transformation function for the month February is of the form:

Ending inventory=beginning inventory + production-demand (D_2=3), ie $x_1=x_2+P_2-D_2$.

This shows that the solutions(x_2) for the previous period (January) is needed to find the current solution x_1.

The return function for February is the sum of production cost and inventory holding cost in February. The inventory holding cost depends partly on the ending inventory(x_2) of the previous period.

(iii) The stage 1(March)

We minimize the sum of production cost and inventory holding cost in the of March subject to demand(D_1=3), production capacity(P_1=3), warehouse capacity(W_1=2) and ending inventory.

The stage transformation function for the month of March is of the form:

Ending inventory=beginning inventory + production-demand (D_1=3).

The return function for March is the sum of production and inventory holding costs in March.

<u>Step 2: A recursive solution</u>

In figure 1, we have numbered the periods backward; that is, stage 1 corresponds to March, stage 2 corresponds to February and stage 3 corresponds to January. The stage transformation functions being the equation:

Ending inventory=beginning inventory + production-demand.

Thus, we have $x_{n-1}=x_n+d_n-D_n$,

x_3=1 for the inventory beginning of January.

$x_2= x_3 + d_3-D_3= x_3 + d_3-2$ for inventory ending January/beginning February.

$x_1= x_2 + d_2-D_2= x_2 + d_2-3$ for inventory ending February/beginning March.

$x_0= x_1 + d_1-D_1= x_1 + d_1-3$ for inventory ending March.

The return functions for each stage represent:

the sum of production and inventory holding costs for the month

i e $r_n(x_n,d_n)=C_nd_n+H_n(x_n+d_n-D_n)$.

(i)For stage 1: n=1(March),

$r_1(x_1,d_1)=200d_1 + 40(x_1+d_1-3)$ represents the total production and holding costs for the period. The production costs are \$200 per unit and the holding costs are \$40 per unit of ending inventory.

The other return functions are:

(ii)For February n=2

$$r_2(x_2, d_2)=150d_2+30(x_2+d_2-3) \text{ stage 2}$$

(iii)For January n=3,

$$r_3(x_3,d_3)= 175d_3+30(x_3+d_3-2) \text{ stage 3 .}$$

There are three constraints that must be satisfied at each stage as we perform the optimization procedure. The first constraint is that the ending inventory must be less than or equal to the warehouse capacity. Mathematically we have

$$x_n + d_n - D_n \leq W_n \text{ or } x_n + d_n \leq D_n + W_n \text{ ……(1).}$$

The second constraint is that the production level in each period must not exceed the production capacity. Mathematically we have

$$d_n \leq p_n \text{ …….(2)}$$

For each stage, we must have the constraint that requires beginning inventory plus production to be greater than or equal to demand.

Mathematically this constraint can be written as

$$x_n + d_n \geq D_n\text{……(3).}$$

The inventory problem is then formulated as:

$$f_n(x_n)=\min\{ r_n(x_n,d_n) + f_{n-1}(x_{n-1})$$

$$\text{subject to } x_n + d_n \leq D_n + W_n,$$

$$x_n + d_n \le D_n,$$

$$d_n \le P_n,$$

$$x_n, d_n \ge 0,$$

where $f_{n-1}(x_{n-1})$ is the minimum value of the return function of x_{n-1}.

Step 3:Computing stagewise the optimal costs.

(i)Computations for stage 1(March).

Our unknowns from the inventory problem are x_n, d_n. Since the problem is a discrete problem x_n, d_n are discrete. However, they should satisfy the constraints

$$x_n + d_n \le D_n + W_n,$$

$$d_n \le P_n,\ x_n + d_n \ge D_n.$$

For stage 1(March) we have n=1, $D_1=3$, $P_1=3$, $W_1=2$, $C_1=200$, $H_1=40$. From $d_1 \le p_1 = 3$, $d_1 = 0,1,2,3$ and $x_1 + d_1 \ge 3$ we get $x_1 = 0,1,2,3$. We use the values of x_1, d_1 to compute the minimum cost for stage 1.

Since we are attempting to minimize cost, we will want the decision variable d_1 to be as smaller as possible and still satisfy the demand constraint.

$$f_1(x_1) = \text{Min.}\{\ r_1(x_1,d_1) = 240d_1 + 40x_1 - 120$$

subject to

$x_1 + d_1 \le 5$ warehouse constraint,

$d_1 \le 3$, production constraint and

$x_1 + d_1 \ge 3$ demand constraint.

$x_1 = 0.$

$$f_1(0) = \min\left\{240\times 3 + 40\times 0 - 120 = 600,\ d_1 = 3\ \right\}.$$

Result $f_1(0) = 600$. Thus d^*_1 .

$x_1 = 1$, $d_1=2$, $d_1=3$

$$f_1(1) = \min.\begin{cases} 240\times 2+40\times 1-120=400,\ \mathrm{d}_1=2 \\ 240\times 3+40-120=640,\ \mathrm{d}_1=3 \end{cases}$$

Result $f_1(1)=400$. Thus $\mathrm{d}^*_1=2$.

$x_1=2$, $d_1=1$, $d_1=2$, $d_1=3$.

$$f_1(2) = \min.\begin{cases} 240\times 1+40\times 2-120=200,\ \mathrm{d}_1=1 \\ 240\times 2+80-120=440,\ \mathrm{d}_1=2 \\ 240\times 3+80-120=680,\ \mathrm{d}_1=3 \end{cases}$$

Result $f_1(2)=200$. Thus $\mathrm{d}^*_1=1$.

$x_1=3$, $d_1=0$, $d_1=1$, $d_1=2$.

$$f_1(3) = \min.\begin{cases} 240\times 0+40\times 3-120=0,\ \mathrm{d}_1=0 \\ 240\times 1+120-120=240,\ \mathrm{d}_1=1 \\ 240\times 2+120-120=480,\ \mathrm{d}_1=2 \end{cases}$$

Result $f_1(3)=0$. Thus $\mathrm{d}^*_1=0$.

Table 2.2 below contains x_1 and d_1 which take on values 0,1,2,3, and the values of $f_1(x_1)$.

M is used to represents no feasible solution and the last column is the optimal solution.

Table 2.2: Summary of results for March

		d_1				
		0	1	2	3	(d^*_1,f_1)
x_1	0	M	M	M	600	(3,600)
	1	M	M	400	640	(2,400)
	2	M	200	440	680	(1,200)
	3	0	240	480	720	(0,0)

Now let us proceed to stage 2.

(ii) Computations for stage 2(February)

$f_2(x_2)$= min.{ $r_2(x_2, d_2) + f_1(x_1)$= $150d_2 + 30(x_2 + d_2-3) + f_1(x_1)$= $180d_2 + 30x_2-90$ $+f_1(x_1)$ subject to $x_2+ d_2 \le 6$, $d_2 \le 2$, $x_2+ d_2 \ge 3$, and for each x_2 selected we calculate $x_1=x_2+d_2-3$. Thus d_2=0,1,2 and x_2=0,1,2,3,4.

$f_2(x_2)$ = min.{$180d_2+30x_2-90+f_1(x_1)$; d_2=0,1,2, x_2=0,1,2,3,4.

Note x=0 is not feasible since 0 plus either 1 or 2 is not up to 3 and $x_2 + d_2 \ge 3$ is not satisfied.

$x_2 = 1$, $d_2 = 2$ and $x_1 = 1+2-3 = 0$.

$$f_2(1) = \min.\{180\times 2 + 30\times 1 - 90 + f_1(0) = 360 + 30 - 90 + 600 = 900,\ d_2 = 2\ .$$

$f_2(1) = 900$. Thus $d^*_2 = 2$.

$x_2 = 2$, $d_2 = 1$, $d_2 = 2$ and $x_1 = 2+1-3 = 0$, $x_1 = 1+2-3 = 0$ respectively.

$$f_2(2) = \min.\begin{cases} 180\times 1 + 30\times 2 - 90 + f_1(0) = 180 + 60 - 90 + 600 = 750,\ d_2 = 1. \\ 180\times 2 + 60 - 90 + f_1(1) = 360 + 60 - 90 + 400 = 730,\ d_2 = 2. \end{cases}$$

$f_2(2) = 730$. Thus $d^*_2 = 2$.

$x_2 = 0$, $x_2 = 1$, $x_2 = 2$ and $d_2 = 0$, $d_2 = 1$,$d_2 = 2$.

Table 2.3 below contains x_2 and d_1 which take on values 0,1,2, the values of $f_2(x_2)$ and the values of (d^*_2,f^*_2). M is used to represents no feasible solution.

Table 2.3: Summary of results for February

		d_2			
		0	1	2	(d^*_2, f^*_2)
x_2	0	M	M	M	-
	1	M	M	900	(2,900)
	2	M	750	730	(2,730)

(iii) Computations for stage 3(January)

$f_3(x_3)$= Min.{ $r_3(x_3,d_3)+f_2(x_2)=175d_3+30(x_3+d_3-2)+f_2(x_2)=205d_3+30x_3-60+f_2(x_2)$

subject to $x_3+d_3\leq 4$, $d_3\leq 3$ i.e $d_3=1,2,3$. $x_3+d_3\geq 2$. with $x_1=1$ already by the beginning inventory level and $x_2=x_3+d_3-2$.

$f_3(x_3)=\min.\{205d_3+30x_3-60+f_2(x_2)$.

$x_3=1$, $d_3=1,2,3$ and $x_2=1+1-2=0$, $x_2=1+2-2=1$, $x_2=1+3-2=2$ respectively.

$$f_3(1)=\min.\begin{cases}205\times 1+30\times 1-60+f_2(0)=175+M,\ d_3=1.\\ 205\times 2+30-60+f_2(1)=380+900=1280,\ d_3=2.\\ 205\times 3+30-60+f_2(2)=615-30+730=1315,\ d_3=3.\end{cases}$$

Result $f_3(1)=1280$. Thus $d^*_3=2$.

Where $f_3(0)$ is not feasible and is denoted by M.

Table 2.4 below contains x_3 and d_3 which take on values 0,1,2,3, the values of $f_3(x_3)$ and the values of (d^*_3,f^*_3). M is used to represents no feasible solution.

Table 2.4: Summary of results for January

		d_3				
		0	1	2	3	(d_3,f_3)
x_3	1	M	M	1280	1315	(2,1280)

Thus, we find that the total cost assumed with the optimal production and inventory policy is 1280. The optimal solution is $f_3(1)=380+f_2(1)=380+900=1280$.

Note $f_2(1)$ is obtained from table 2.3 as $f_2(1)=300+f_1(0)=300+600=900$ where $f_1(0)=600$ is also from table 2.2.

<u>Step 4: Optimal solution from the computer results.</u>

. To find the optimal decisions and inventory levels for each period, we may trace back through each stage and identify x_n and d^*_n as we go.

The company should produce two (2) units of cars with a beginning inventory one (1) in January of a production and inventory holding cost of $380. Moreover, the company should produce two units of cars with a beginning inventory one(1) in February of a production and inventory holding cost of $300 and three units of cars with a beginning inventory zero(0) in March of a production and inventory holding cost of $600.

Table 2.5 summarizes the optimal production and inventory policy. In column one is the month, column two the beginning inventory, column three production cost, column four ending inventory, column five the holding cost and the last column total monthly cost.

Table 2.5: Summary of results for the optimal solution

Month	Beginning inventory	Production capacity(P_n)	Production cost ($C_n d_n$)	Ending inventory	Holding cost ($H_n x_{n-1}$)	Total monthly cost
January	1	2	$ 350	1	$30	$380
February	1	2	$300	0	0	$300
March	0	3	$600	0	0	$600
Total			$1250		$30	$1280

2.6.2 THE STAGECOACH PROBLEM (NETWORK PROBLEM)

The stagecoach problem is a problem specially constructed to illustrate the features and the terminology of dynamic programming. It concerns Mr. Ebenezer in Missouri who decided to go west to join the gold rush in California during the mid-19th century. The journey requires traveling by stagecoach through unsettled country where there was serious danger of attack by marauders. Although the starting point (Missouri) and destination (California) are fixed, he had considerable choice as to which states to travel through en route. The possible routes are shown in figure 2, where each state is represented by a circled capital letter and direction of travel is always from left to right in the diagram. Thus, four stages (stagecoach runs) were required to travel from his point of embarkation in state A (Missouri) to his destination in state J (California). [Hillier et al, 2005]

He was a prudent man who was quite concerned about his safety. After some thought, he came up with a rather clever way of determining the safest route. Life insurance policies were offered to stagecoach passengers. Because the cost of the policy for taking any given stagecoach run was based on a careful evaluation of the safety of that run, the safest route should be the one with the cheapest total life policy. The main problem of the traveler is how to find the safest routes and the cheapest cost of insurance policy in order to minimize cost.

We minimize the cost of insurance from state (A) to state (J) subject to the safety of the route.

Current cost= immediate cost (stage n) + minimum future cost (stage n+1).

<u>Step 0. Variables definitions and datas.</u>

The figure 2 below shows the cost on the edges.

Level 1 2 3 4 5

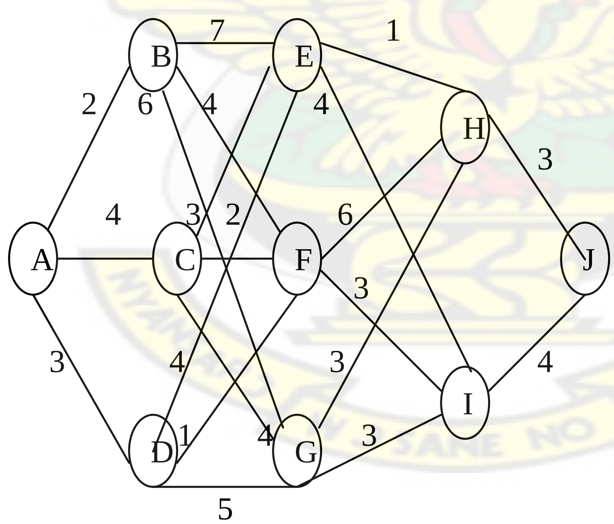


Figure 2: The road system and costs for the stagecoach problem

Table 2.6 shows the cost of insurance for moving from one state to another state and dash(-) is used to represent where there is no cost.

Table 2.6: The road system and costs for the stagecoach problem

	A	B	C	D	E	F	G	H	I	J
A	-	2	4	3	-	-	-	-	-	-
B	-	-	-	-	7	4	6	-	-	-
C	-	-	-	-	3	2	4	-	-	-
D	-	-	-	-	4	1	5	-	-	-
E	-	-	-	-	-	-	-	1	4	-
F	-	-	-	-	-	-	-	6	3	-
G	-	-	-	-	-	-	-	3	3	
H	-	-	-	-	-	-	-	-	-	3
I	-	-	-	-	-	-	-	-	-	4
J	-	-	-	-	-	-	-	-	-	-

Let us consider the cheapest possible ways to get from starting point(A) through to the last point(J).

Let $C_{x_{n,i},x_{n+1,j}}$ denote the cost of insurance from stage n to stage n+1.

Let i be the point it is at the stage n and j the route it should take.

Let $x_{n,i}$ be the state variable.

Let $f_n(x_{n,i})$ denote the minimum cost of the objective function from any city $x_{n,i}$ to the final destination J.

Step 1: The structure of an optimal solution.

The first step of dynamic programming is to characterize the structure of an optimal solution.

Let divide the problem into five stages as follows;

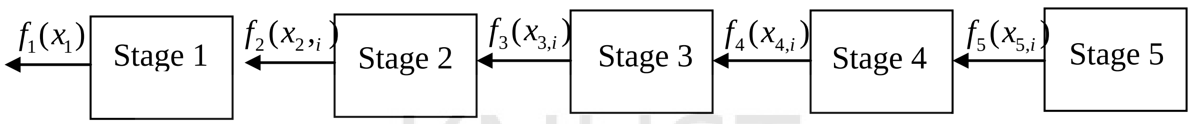


Figure 3: Schematic representation of the stagecoach problem as a five-stage dynamic programming problem

Stage 1: Consists of city A.

The return function is $f_1(x_1)$= immediate cost (stage 1) + minimum future cost (stage 2).

Stage 2: Consists of cities B, C and D.

The return function is $f_2(x_2)$= immediate cost (stage 2) + minimum future cost (stage 3) .

Stage 3: Consists of cities E,F and G.

The return function is $f_3(x_3)$= immediate cost (stage 3) + minimum future cost (stage 4) .

Stage 4: Consists of cities H and I.

The return function is $f_4(x_4)$= immediate cost (stage 4) + minimum future cost (stage 5) .

Stage 5: Consists of city J.

This is the final stage. Therefore $f_5(x_5) = 0$.

For the stagecoach problem, we start with the smaller problem where Mr. Ebenezer has nearly completed his journey and has only one more stage (stagecoach run) to go. The obvious optimal solution for this smaller problem is to go from his current state (whatever it is) to his ultimate destination (state J). At each subsequent iteration, the problem is enlarged by increasing by 1 the number of stages left to go to complete the journey.

<u>Step 2: A recursive Solution</u>

Let $f_n\ (x_{n,i})$ denote the optimal value of the objective function from any city $x_{n,i}$ to the final destination J. Hence the optimum is $f_1(x_1)$ the minimum of the sum of cost of insurance from A to J.

Thus $f_n(x_{n,i}) = \min.[C_{x_{n,i},x_{n+1,j}} + f_{n+1}(x_{n+1,j})]$ subject to $C_{x_{n,i},x_{n+1,j}}$ ≥0 and an integer.

(i) For stage 1: n=1, x_1=A

We minimize the cost of insurance from stage 1(A) to stage 5(J).

$f_1(x_{1,1}) = \min.[C_{x_{1,1},x_{2,1}} + f_2(x_{2,1}),\ C_{x_{1,2},x_{2,2}} + f_2(x_{2,2})]$, .i=1, and j=1, 2, 3

(ii) For stage 2: n=2

We minimize the cost of insurance from stage 2(B, C, D) to stage 5(J).

$f_2(x_{2,i}) = \min.[C_{x_{2,i},x_{3,j}} + f_3(x_{3,j})]$, i=1,2,3 and j=1,2,3.

(iii) For stage 3: n=3

We minimize the cost of insurance from stage 3(E,F,G) to stage 5(J).

$f_3(x_{3,i}) = \min.[C_{x_{3,i},x_{4,j}} + f_4(x_{4,j})]$ i=1,2,3 and j=1,2,3.

(iv) For stage 4: n=4

We minimize the cost of insurance from stage 4(H,I) to stage 5(J).

$f_4(x_{4,i}) = \min.[C_{x_{4,i},x_{5,j}} + f_5(x_{5,j})]$ i =1, j=1,2.

(iv) For stage 5: n=5, x_5=J.

Since the ultimate destination (state J=x_5) is reached at the end of stage 5, $f_5(x_5)=0$.

Let $f_n(x_{n,i})$ be the optimal value of the objective function from any stage n to the final destination J.

Step 3: Computing the stage wise optimal cost.

(i) Computations for stage 5(x_5=J): n=5.

Since J is the final stage there is no cost after J, then $f_5(x_5)=0$

(ii) Computations for stage 4($x_{4,1}$=H, $x_{4,2}$=I): n=4, i=1,2.

When he has only one more stage to go(n=4), the best route is determined entirely by his current state(either H or I) and his final destination x_5=J.

Result $f_4(x_{4,1}) = f_4(H) = C_{x_{4,1},x_{5,1}} + f_5(x_{5,1}) = 3+0 = 3.$

Result $f_4(x_{4,2}) = f_4(I) = C_{x_{4,2},x_{5,2}} + f_5(x_5) = 4+0 = 4.$

(ii) Computations for stage 3(x_{31}=E, $x_{3,2}$= F and $x_{3,3}$= G): n=3, i=1,2.

When he has two more stages to go(n=3).

$$f_3(x_{3,1}) = \min\begin{cases} C_{x_{3,1},x_{4,1}} + f_4(x_{4,1}) = 1+3 = 4. \\ C_{x_{3,2},x_{4,2}} + f_4(x_{4,2}) = 4+4 = 8. \end{cases}$$

Result $f_3(x_{3,1}) = 4.$

$$f_3(x_{3,2}) = \min\begin{cases} C_{x_{3,1},x_{4,1}} + f_4(x_{4,1}) = 6+3 = 9. \\ C_{x_{3,2},x_{4,2}} + f_4(x_{4,2}) = 3+4 = 7. \end{cases}$$

Result $f_3(x_{3,2}) = 7.$

$$f_3(x_{3,3}) = \begin{cases} C_{x_{3,1},x_{4,1}} + f_4(x_{4,1}) = 3+3 = 6. \\ C_{x_{3,2},x_{4,2}} + f_4(x_{4,2}) = 3+4 = 7. \end{cases}$$

$Result\ f_3(x_{3,3}) = 6.$

(iii) Computations for stage 2($x_{2,1}$=B, $x_{2,2}$= C and $x_{2,3}$=D,): n=2

The solution for the second-stage problem (n=2), where there are three stages to go.

$$f_2(x_{2,1}) = \min \begin{cases} C_{x_{2,1},x_{3,1}} + f_3(x_{3,1}) = 7+4 = 11. \\ C_{x_{2,2},x_{3,2}} + f_3(x_{3,2}) = 4+7 = 11. \\ C_{x_{2,3},x_{3,3}} + f_3(x_{3,3}) = 6+6 = 12. \end{cases}$$

$Result\ \ f_2(x_{2,1}) = 11$

$$f_2(x_{2,2}) = \min \begin{cases} C_{x_{2,1},x_{3,1}} + f_3(x_{3,1}) = 3+4 = 7. \\ C_{x_{2,2},x_{3,2}} + f_3(x_{3,2}) = 2+7 = 9. \\ C_{x_{2,3},x_{3,3}} + f_3(x_{3,3}) = 4+6 = 10. \end{cases}$$

$Result\ \ f_2(x_{2,2}) = 7.$

$$f_2(x_{2,3}) = \min \begin{cases} C_{x_{2,1},x_{3,1}} + f_3(x_{3,1}) = 4+4 = 8. \\ C_{x_{2,2},x_{3,2}} + f_3(x_{3,2}) = 1+7 = 8. \\ C_{x_{2,3},x_{3,3}} + f_3(x_{3,3}) = 6+6 = 12. \end{cases}$$

$Result\ \ f_2(x_{2,3}) = 8$

(i) Computations for stage 1(x_1= A).

Moving to the first-stage problem (n=1), with all four stages to go.

$$f_1(x_1) = \min \begin{cases} C_{x_{1,1},x_{2,1}} + f_2(x_{2,1}) = 2+11 = 13. \\ C_{x_{1,2},x_{2,2}} + f_2(x_{2,2}) = 4+7 = 11. \\ C_{x_{1,3},x_{2,3}} + f_2(x_{2,3}) = 3+8 = 11. \end{cases}$$

$\mathrm{Re}\,sult\ \ f_1(x_1) = 11.$

Since 11 is the minimum cost, f_1(A)=11 and $x_{2,2}$=C or $x_{2,3}$= D.

Step:4 <u>Optimal solution from the computed results</u>

An optimal solution for entire problem can now be identified from the results above. Results for the n=1 problem indicate that Mr. Ebenezer should go initially to either state C or state D. Suppose that he chooses $x_{2,2}$=C. For n=2, the results for $x_{2,2}$=C is $x_{3,1}$=E. This result leads to the n=3, which gives $x_{4,1,}$=H for $x_{3,1}$=E, and the n=4 yields x_5=J for $x_{4,1}$=H. Hence, one optimal route is A→C→E→H→J. Choosing $x_{2,3}$=D leads to the other two optimal routes A→D→E→H→Jand A→D→F→I→J. They all yield a total cost of f_1 (A) =11.

These results of dynamic programming analysis also are summarized in the diagram below. (Hillier et al, 2005.)

Graphical display of the dynamic programming solution of the stagecoach problem. Each arrow shows an optimal policy decision (the best immediate destination) from that state, where the number by the resulting cost from there to the end is shown in figure 4.

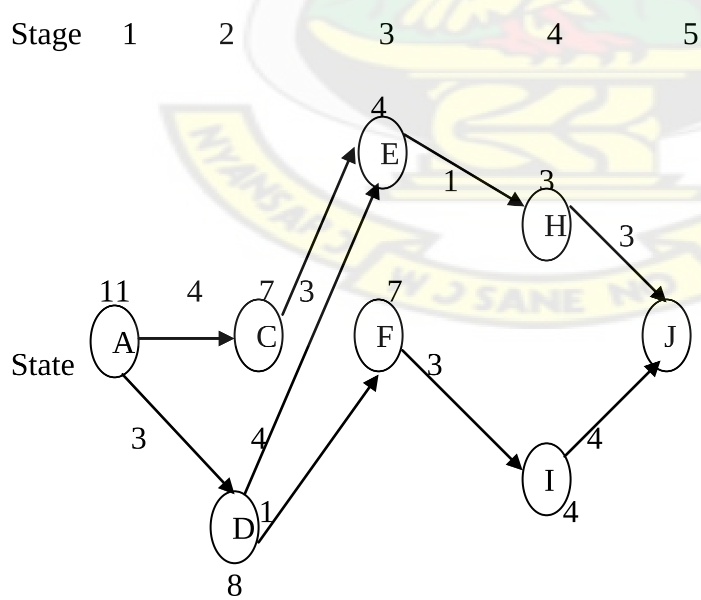


Figure 4: A representation of the optimal solution of the stagecoach problem

2.6.3 THE KNAPSACK PROBLEM

We maximize benefit (total value rating) subject to the number of days available (10) for processing of a job and the number of jobs available. The stage transformation functions are then defined as:

x_{n-1}= the number of days available at stage n – the product of the number days needed to complete one job by the number of jobs to process.

The return functions at each stage are based on the value rating of a job times the number of jobs selected for processing.

The first constraint is that the number of days needed to process a job must be less than or equal to the number days available (10).

The second constraint is that the number of jobs selected must be less than or equal to the number of jobs available.

The Knapsack problem is often encountered in dynamic programming applications. The basic idea is that there are N different types of items that can be put into a knapsack. Each item has a certain weight associated with it as well as a value. The problem is to determine how many units of each item to place in the knapsack in order to maximize total value. A constraint is placed on the maximum weight permissible.

Consider a Manager of a manufacturing operation who has selection of jobs to process during the following 10-day period. A list of the jobs waiting to be processed at the beginning of the current week is presented in table 2.7. The estimated time required for completion and the value rating associated with each category of job are also shown in the table. The main aim of the Manager is to find out how many jobs to choose from each category to process in order to maximize performance value (David, et al 1988).

Step 0 Variables definitions and data

Table 2.7: Job data for the manufacturing operation

Job number n	No. of jobs to be processed (N).	Estimated completion time per job (days) (u_n).	Value rating.
Category 1	4	1	2
Category 2	3	3	8
Category 3	2	4	11
Category 4	2	7	20

The value rating assigned to each job is a subjective score assigned by the supervisor. A scale from 1 to 20 is used to measure the value of each job, where 1 represents jobs of the least value, and 20 represents jobs most value. We would like to make a selection of jobs to process during the next 10-days such that all the jobs selected can be processed in 10 days and that the total performance value of jobs selected is maximized. In knapsack problem terminology we are in essence selecting the best jobs for our 10-day knapsack, where the knapsack has a capacity equal to the 10-day (w) production capacity. We formulate and solve this problem using a dynamic programming solution procedure.

Let d_n denote the number of jobs in category n selected (that is, the decision variable at stage n). The state variable $x_n(x_n \leq w)$ is defined as the number of days of processing time remaining when we reach stage n.

d_n= decision variable, x_n= state variable, u_n=the number of days needed to complete one job.

Step 1: The structure of an optimal solution

This problem can be formulated as a dynamic programming problem involving four stages.

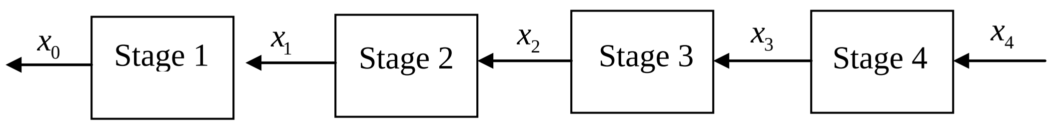


Figure 5: Schematic representation of the knapsack problem as a four-stage dynamic programming problem

Stage 1: Consists of category 1. The number of jobs to be selected for processing in category 1 should be less than or equal to four (4).

The stage transformation functions are then defined as:

$$x_0 = t_1(x_1, d_1) = x_1 - u_1 d_1. \quad d_1 \leq N.$$

The return function is $r_1(x_1, d_1) = 2d_1$, $d_1 \leq 4$

Stage 2: Consists of category 2. The number of jobs to be selected for processing in category 2 should be less than or equal to three (3).

The stage transformation functions are then defined as:

$$x_1 = t_1(x_1, d_1) = x_2 - u_2 d_2. \quad d_2 \leq N_2.$$

The return function is $f_2(x_2) = r_2(x_{2,}d_2) + f_1(x_1) = 8d_2 + f_1(x_1)$, $d_2 \leq 3$

Stage 3: Consists of category 3. The number of jobs to be selected for processing in category 3 should be less than or equal to two (2).

The stage transformation functions are then defined as:

$$x_2 = t_3(x_3, d_3) = x_3 - u_3 d_3. \quad d_3 \leq N_3.$$

The return function is $f_3(x_3) = r_3(x_{3,}\, d_3) + f_2(x_2) = 11d_3 + f_2(x_2)$

Stage 4: Consists of category 4.

The number of jobs to be selected for processing in category 4 should be less than or equal to two (2). The stage transformation functions are then defined as:

$$x_3 = t_4(x_4, d_4) = x_4 - u_4d_4. \quad d_4 \leq N_4.$$

The return function is

$$f_4(x_4) = r_4(x_4,d_4) + f_3(x_3) = 20d_4 + f_3(x_3).$$

At stage 1 we must decide how many jobs from category 1 to process, at stage 2 we must decide how many jobs from category 2 to process, and so on. Thus we let d_n denote the number of jobs in category n selected (that is, the decision variable at stage n). The state variable x_n ($x_n \leq w$) is defined as the number of days of processing time remaining when we reach stage n.

Step 2: A recursive solution

Thus with a 10-day production period, x_4=10 represents the total number of days that are available for processing jobs. The stage transformation functions are then defined as:

$$x_{n-1} = t_n(x_n, d_n) = x_n - u_nd_n. \quad d_n \leq N_n.$$

$f_n(x_n) = r_n(x_n, d_n) + f_{n-1}(x_{n-1})$. The $f_n(x_n)$ is the total return function after decision d_n is made.

(i) Stage 4: n = 4, $x_3 = t_4(x_4, d_4) = x_4 - 7d_4$.

The return at each stage is based on the value rating of jobs and the number of jobs selected at each stage. The return functions are as follows:

$$r_4(x_4, d_4) = 20d_4. \quad d_4 \leq 2$$

$$f_n(x_n) = r_n(x_n, d_n) + f_{n-1}(x_{n-1}).$$

The $f(x_n)$ is the total return function after decision d_n is made.

$f_4(x_4) = r_4(x_4,d_4) + f_3(x_3) = 20d_4 + f_3(x_3)$.

(ii) For Stage 3: n=3

$x_2 = t_3\ (x_3, d_3) = x_3 - 4d_3$.

$r_3(x_3, d_3) = 11d_3$. $d_3 \leq 2$.

$f_3(x_3) = r_3(x_{3,}d_3) + f_2(x_2) = 11d_3 + f_2(x_2)$

(iii)For Stage 2: n=2,

$x_1 = t_2(x_2, d_2) = x_2 - 3d_2$. $r_2(x_2,d_2) = 8d_2$.

$f_2(x_2) = r_2(x_{2,}d_2) + f_1(x_1) = 8d_2 + f_1(x_1)$, $d_2 \leq 3$.

(iv)For Stage 1: n = 1

$x_0 = t_1(x_1, d_1) = x_1 - 1d_2$. $r_1(x_1, d_1) = 2d_1$, $d_1 \leq 4$.

<u>Step 3: Computing the cheapest cost at each category</u>

We will apply a backward solution procedure; that is, we will begin by considering the stage 1 decision.

(i) Computations for Stage 1(filling with item category 1 only): n=1.

Note that the input to stage 1, x_1, which is the number of days of processing time available at stage 1, is unknown because we have not yet identified the decisions at the previous stages. Therefore in our analysis at stage 1 we will have to consider all possible values x_1 and identify the best decision d_1 for each case; $f_1(x_1)$ will be the total return after decision d_1 is made. $f_1(x_1) = r_1(x_1,d_1) = 2d_1$. $d_1 \leq 4$. Also we are to consider all possible values of d_1 (that is,0,1,2,3, or 4). Results are shown in Appendix A.

The number of category 1 jobs selected will depend upon the processing time available but cannot exceed 4.

Recall that $f_1(x_1)$ represents the value of the optimal total return from stage 1 and all remaining stages, given an input of x_1 to stage 1. Let us move on to stage 2 and carry out the optimization at that stage.

(ii) For Stage 2(filling with items 1 and 2): n=2.

Since the input to stage 2, x_2, is unknown, we have to consider all possible values from 0 to 10. Also we to consider all possible values of d_2 (that is,0,1,2, or 3).

$f_2(x_2) = 8d_2 + f_1(x_1)$, $d_2 \leq 3$. Results are shown in Appendix B.

Note that some combinations of x_2 and d_2 are not feasible. For example with x_2=2 days, d_2=1 is infeasible (i e not possible) because category 2 jobs each require 3 days to process.

(iii) For Stage 3(filling 1, 2 and 3 items): n = 3.

$f(x_3) = 11d_3 + f_2(x_2)$, $d_3 \leq 2$.Results are shown in Appendix C

(iv) Computations for stage 4(filling with 1,2,3 and 4): n=4.

$f_4=20d_4+f_3(x_3)$, $d_4 \leq 2$. Results are shown in Appendix D

The optimal solution is $f_4(10)= 20+f_3(3)=28$. Note that $f_3(3)=0+f_2(3)=8$ i.e $f_2(3)=8\times1=8$.

<u>Step 4: An optimal solution from the computed results</u>

The optimal decision, given x_4=10, is d^*_4=1. In order to identify the overall optimal solution, we must now trace back through the tables beginning at stage 4. The optimal decision at stage 4 is d^*_4=1. Thus x_3=10-7=3, and we enter stage 3 with 3 days available for processing.

With x_3=3 we see that the best decision at stage 3 is d^*_3=0. Thus, we enter stage 2 with x_2=3. The optimal decision at stage 2 with x_2=3 is d^*_2=1, resulting in x_1=0. Finally the decision at stage 1 must be d^*_1=0.

The table 2.8 below is the optimal solution of the number of jobs to be processed from each category.

Table 2.8: Summary of the optimal solution of the knapsack problem

Decision	Return
d^*_1=0	0
d^*_2=1	8
d^*_3=0	0
d^*_4=1	20
Total return	28

We should schedule one job from category 2 and one job from category 4 for processing over the next 10-day planning period

2.6.4. AN EQUIPMENT REPLACEMENT MODEL

A company needs to determine the optimal replacement policy for a current 3-year old machine over the next 4 years (n=4). The company requires that a 6-year old machine be replaced. The cost of a new machine is $100,000.00.

<u>Step 0 Variables definitions and data</u>

The table 2.9 below gives the age (yr), revenue r(t), operating cost c(t) and the salvage value s(t) of a machine.

Table 2.9 Data for the operation of the machine

Age, t(yr)	Revenue, r(t) ($)	Operating cost, c(t) ($)	Salvage value, s(t) ($)
0	20,000	200	-
1	19,000	600	80000
2	18,500	1200	60000
3	17,500	1500	50000
4	15,500	1700	30000
5	14,000	1800	10000
6	12,200	2200	5000

At the start of year 1, we have a 3-year-old machine. We can either replace it (R) or keep it (K) for another year. At the start of year 2, if replacement had occurred in year 1 the new machine will be 1 year old; otherwise, the old machine will be 4 years old. The logic applies at the start of years 3 to 4. If a 1-year-old machine is replaced at start of year 2, 3 or 4, its replacement will be 1 year old at the start of the following year.

In addition, at the start of year 4, a 6-year-old machine must be replaced, and at the end of year 4 (end of the horizon), we salvage (S) the machines. Figure 9 summarizes the network representing the problem.

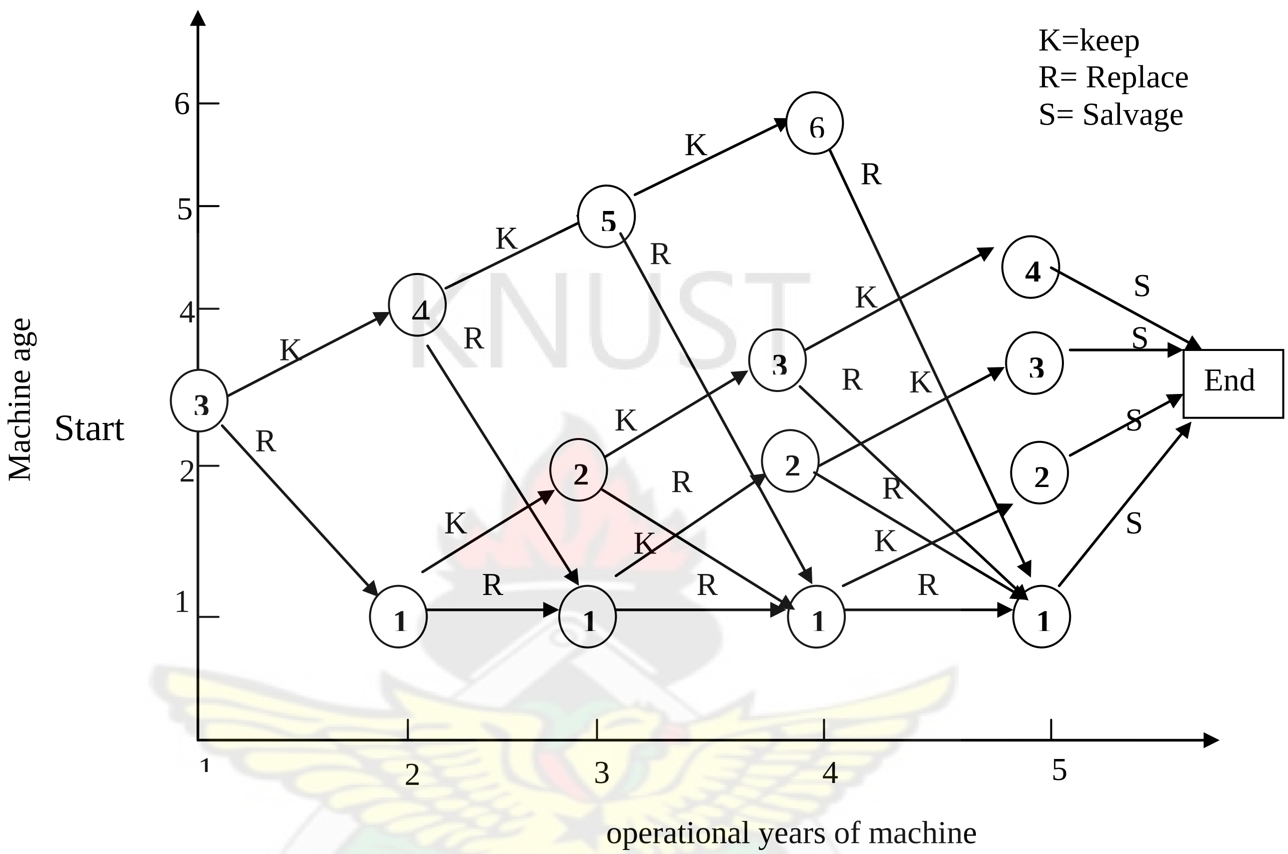


Figure 6: Representation of machine age as a function of operational year.

The network shows that at the start of year 2, the possible ages of the machine are 1 for a new machine and 4 years for the old machine. For the start of year 3, the possible ages are 1,2 and 5 years, and for the start of year 4, the possible ages are 1,2,3 and 6 years.

Let r(t), c(t), and s(t) represent the yearly revenue, operating cost, and salvage value of a t-year-old machine respectively and t the age of machine. The cost of acquiring a new machine in any year is I.

Let the stage variable i represents the year of operation of machine i, i=1, 2,…,n where i is the operational year of machine.

The alternatives at stage (year) i call for either keeping or replacing the machine at the start of year i.

The state at stage i is the age of the machine at the start of year i.

Give that the machine is t years old at the start of operational year i, define

$f_i((t)$= maximum net income for years i, i + 1,…and n.

The recursive equation is derived as

$f_i(t)$= revenue-operating cost+ previous maximum profit ie

$$f_i(t) = \max \begin{cases} r(t) - c(t) + f_{i+1}(t+1), & \text{if KEEP} \\ r(0) + s(t) - I - c(0) + f_{i+1}(1), & \text{if REPLACE} \end{cases}$$

$$f_{n+1}(.) \equiv 0$$

Step 1. The structure of an optimal solution.

Let divide the problem into four stages as detailed in figure 7 below.

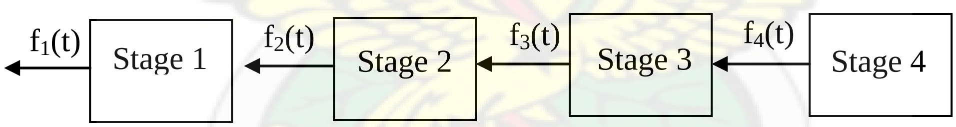


Figure 7: Schematic representation of the equipment replacement problem as a four-stage dynamic programming problem

Stage 1: For the start of year 1; t= 3

The return function is the maximum net income(profit) for year 1; t=3

$$f_1(t) = \max \begin{cases} r(t) - c(t) + f_2(t+1), & \text{if KEEP} \\ r(0) + s(t) - I - c(0) + f_2(1), & \text{if REPLACE} \end{cases}$$

Stage 2: For the start of year 2; t=1, 4

The return function is the maximum net income(profit) for year 2; t=1, 4

$$f_2(t) = \max\begin{cases} r(t) - c(t) + f_3(t+1), \text{ if KEEP} \\ r(0) + s(t) - I - c(0) + f_3(1), \text{ if REPLACE} \end{cases}$$

Stage 3: For the start of year 3; t=1, 2, 5

The return function is the maximum net income(profit) for year 3; t= 1, 2, 5

$$f_3(t) = \max\begin{cases} r(t) - c(t) + f_4(t+1), \text{ if KEEP} \\ r(0) + s(t) - I - c(0) + f_4(1), \text{ if REPLACE} \end{cases}$$

Stage 4: For the start of year 4; t=1, 2, 3, 6.

The return function is the maximum net income(profit) for year 4; t=1,2,3,6

$$f_4(t) = \max\begin{cases} r(t) + s(t+1) - c(t), \text{ if KEEP} \\ r(0) + s(t) + s(1) - I - c(0), \text{ if REPLACE} \end{cases}$$

Step 2: A recursive solution

The problem is solved in four stages in backwards recursion.

The return function is

$$f_i(t) = \max\begin{cases} r(t) - c(t) + f_{i+1}(t+1), \quad \text{if KEEP} \\ r(0) + s(t) - I - c(0) + f_{i+1}(1), \text{ if REPLACE} \end{cases}$$

$$f_{n+1}(.) \equiv 0$$

$f_{n+1}(.)=0$.

Stage 4: For the start of year 4; t=1,2,3,6

$$f_4(t) = \max\begin{cases} r(t) + s(t+1) - c(t), \text{ if KEEP} \\ r(0) + s(t) + s(1) - I - c(0), \text{ if REPLACE} \end{cases}$$

Stage 3: For the start of year 3; t= 1,2, 5.

$$f_3(t) = \max \begin{cases} r(t) - c(t) + f_4(t+1), \text{ if KEEP} \\ r(0) + s(t) - I - c(0) + f_4(1), \text{ if REPLACE} \end{cases}$$

Stage 2: For start of year 2: t= 1, 4

$$f_2(t) = \max \begin{cases} r(t) - c(t) + f_3(t+1), \text{ if KEEP} \\ r(0) + s(t) - I - c(0) + f_3(1), \text{ if REPLACE} \end{cases}$$

Stage 1: For the start of year 1; t = 3.

$$f_1(t) = \max \begin{cases} r(t) - c(t) + f_2(t+1), \text{ if KEEP} \\ r(0) + s(t) - I - c(0) + f_2(1), \text{ if REPLACE} \end{cases}$$

Step 3 Computation of optimal solution

The solution of the network in figure 4 is finding the maximum profit from the start of year 1 to the end of year 4. All values are in thousands of dollars. Note that if a machine is replaced in year 4(i.e., end of the planning horizon), its revenue will include the salvage value s(t), of the replaced machine and the salvage value, s(1), of the replacement machine.

Stage 4 Computation for operational year 4.

$$f_4(t) = \max \begin{cases} r(t) + s(t+1) - c(t), \text{ if KEEP} \\ r(0) + s(t) + s(1) - I - c(0), \text{ if REPLACE} \end{cases}$$

For i=4, t=1, r(1)= 19,s(2)=60, c(1)=0.6, r(0)=20, s(1)=80, I=100.

$$f_4(1) = \max \begin{cases} 19 + 60 - 0.6 = 78. \\ 20 + 80 + 80 - 100 - 0.2 = 79.8. \end{cases}$$

Result $f_4(1) = 79.8$ replace

For i=4, t=2, r(2)=18.5, s(3)=50, c(2)=1.2, r(0)=20, s(2)=60, s(1)=80, c(0)=0.2, I=100.

$$f_4(2) = \max \begin{cases} 18.5 + 50 - 1.2 = 67.3, \text{ if KEEP} \\ 20 + 60 + 80 - 100 - 0.2 = 59.8, \text{ if REPLACE} \end{cases}$$

Result $f_4(2) = 67.3$ keep

From table 2.9 above, for i=4, t=3, r(3)=17.2, s(4)=30, c(2)=1.5, r(0)=20, s(3)=50, s(1)=80, c(0)=0.2, I=100

$$f_4(3) = \max \begin{cases} 17.2 + 30 - 1.5 = 45.7, \text{ if KEEP} \\ 20 + 50 + 80 - 100 - 0.2 = 49.8, \text{ if REPLACE} \end{cases}$$

Result $f_4(3) = 49.8$ replace

For i=4, t=6 , r(0)=20, s(3)=50, s(1)=80, c(0)=0.2, I=100,

$$f_4(6) = \max \begin{cases} \textit{must} \text{ replace} \\ \text{20+5+80-0.2-100=4.8 if replace} \end{cases}$$

$f_4(6) = 4.8$ replace

In table 2.9.1 below, column 1 is the age of the vehicle and column 2 the revenue that will be generated when the bus is kept, column 3 the revenue when the bus is replaced and the last column is the optimal decision to either keep or replace the bus.

Table 2.9.1: Summary of results for operational year 4

	K	R	Optimum solution	
t	r(t)+s(t+1)- c(t)	r(0)+s(t)+s(1)-c(0)-I	f_4(t)	Decision
1	19.0+60-0.6=78.40	20+80+80-0.2-100=79.8	79.80	R
2	18.50+50-1.2=67.3	20+60+80-0.2-100= 59.8	67.3	K
3	17.2 +30-1.5=45.7	20+50+80-0.2-100= 49.8	49.8	R
6	(must replace)	20+5+80-0.2-100=4.8	4.8	R

Stage 3 Computation for operational year 3.

$$f_i(t) = \max \begin{cases} r(t) - c(t) + f_{i+1}(t+1), & \text{if KEEP} \\ r(0) + s(t) - I - c(0) + f_{i+1}(1), & \text{if REPLACE} \end{cases}$$

$$f_{n+1}(.) \equiv 0$$

From table 2.9 above, we have t=1,2,5, r(1)=19, c(1)=0.6, $f_4(2)$=67.3, r(0)=20, c(0)=0.2, s(1)=80, I=100, $f_4(1)$=79.8, r(2)=18.5, c(2)=1.2, $f_4(3)$)=49.8, s(2)=60, r(5)=14, c(5)=1.8, $f_4(6)$=4.8, s(5)=10.

In table 2.9.2 below, column 1 is the age of the vehicle and column 2 the revenue that will be generated when the bus is kept, column 3 the revenue when the bus is replaced and the last column is the optimal decision to either keep or replace the bus.

Table 2.9.2. Summary of results for operational year 3

	K	R	Optimum solution	
t	r(t)-c(t)+f_4(t+1)	r(0)+s(t)-c(0)-I+ f_4(1)	f_3(t)	Decision
1	19.0-0.6 + 67.3=85.7	20+80-0.2-100+79.8=79.6	85.7	K
2	18.5-1.2+49.8=67.1	20+60-0.2-100+79.8=59.6	67.1	K
5	14.0-1.8+4.8=17	20+10-0.2-100+79.8=19.6	19.6	R

Stage 2: Computation for operational year 2

From table 2.9 above, we have t= 1,4, r(1)=19, c(1)=0.6, f_3(2)=67.1, r(0)=20, s(1)=80, c(0)=0.2, I=100, f_3(1)=85.7, r(4)=15.5, c(4)=1.7, f_3(5)=19.6, s(4)=30.

In table 2.9.3. below, column 1 is the age of the vehicle and column 2 the revenue that will be generated when the bus is kept, column 3 the revenue when the bus is replaced and the last column is the optimal decision to either keep or replace the bus.

Table 2.9.3. Summary of results for operational year 2

	K	R	Optimum solution
t	$r(t)-c(t)+f_3(t+1)$	$r(0)+s(t)-c(0)-I + f_3(1)$	$f_2(t)$ Decision
1	19.0-0.6+67.1=85.5	20 + 80-0.2-100+85.7=85.5	85.5 K or R
4	15.5-1.7+19.6=33.4	20+30-.2-100+85.7=35.5	35.5 R

Stage 1: Computation for operational year 1

From table 2.9 above, we have r(3)=17.2,c(3)=1.5, $f_2(4)$=35.5, r(0)=20, s(3)=50, c(0)=0.2, I=100, $f_2(1)$=85.5.

In table 2.9.4 below, column 1 is the age of the vehicle and column 2 the revenue that will be generated when the bus is kept, column 3 the revenue when the bus is replaced and the last column is the optimal decision to either keep or replace the bus.

Table 2.9.4: Summary of results for operational year 1

	K	R	Optimum solution
t	$r(t)-c(t)+f_2(t+1)$	$r(0)+s(t)-c(0)-I+f_2(1)$	$f_1(t)$ Decision
3	17.2-1.5+35.5=51.2	20+50-0.2-100+85.5=55.3	55.3 R

Step 4: An optimal solution from the computed results.

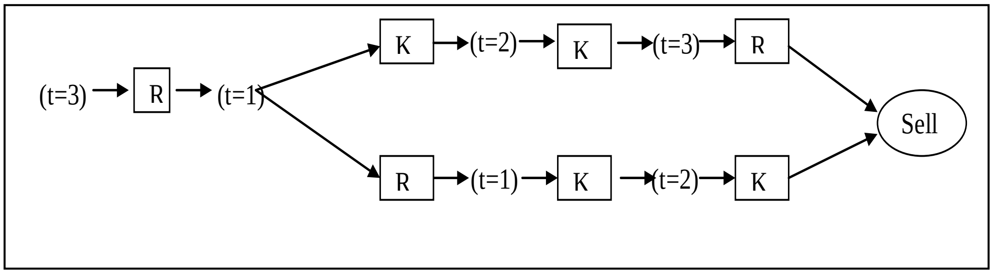


Figure 8: Schematic representation of the optimal solution of the replacement policy

Figure 8 summarizes the optimal solution. At the start of year 1, given t=3, the optimal decision is to replace the machine. Thus, the new machine will be 1 year old at the start of year 2, and t=1 at the start of year 2 calls for keeping or replacing the machine. If it is replaced, the new machine will be 1 year old at the start of operational year 3; otherwise, the kept machine will be 2 years old. The process is continued in this manner until operational year 4 is reached.

The alternatives optimal policies starting in year 1 are (R,K,K,R) and (R, R ,K, K). The optimal profit is $55.30. From the two scenarios above, it can be deduced that the machine should always be replaced when it is three years old, in order to maximize profit. (Hamdy, 2007)

CHAPTER THREE

REPLACEMENT MODELS

3.1. HISTORICAL BACKGROUND OF THE THEORY OF REPLACEMENT:

Equipment replacement policy origin from two early papers by Taylor(1923) and Hotelling (1925).

Taylor developed by means of a discrete period analysis, a formula relating the average unit cost of the output of a machine over L years to the cost of the new machine, the scrap value of the machine after L periods of service, the operating costs of the machine in each period of service up to the L period, the output of the machine in each period, and the rate of interest.

Taylor then proceeded further to show how one determines L (the years of machine life) such that the unit cost of production will be minimum.

Hotelling, two years later, gave a different dimension to Taylor's proposition by raising the following questions;

'Does the manufacturer desire to makes his unit cost a minimum? Or may be that considerations of profit lead him to scrap the machine at some different point in time from that which makes the unit cost a minimum'.

He then advances the viewpoint that the owner of the machine wishes to maximize the present value of machine's output minus its operating costs.

The 'fundamental formula' which he ultimately puts forth with the assumption of continuous discounting at a constant rate of interest can be stated in the following form:

$$V(T) = \int_{T}^{L} [\, R(t) - E(t)]\, e^{-r(t-T)}\, dt + S(L)\, e^{-r(L-T)} \,,$$

where V(T) is the value of the equipment at time T; S(L) is a function giving the scrap value at time L; R(t) is the machine's revenue at time t; E(t) is the machine's operating cost at time t; and r is the continous rate of interest.

Preinreich (1940), showed that the economic life of a single machine could not be determined in isolation from the economic life of other machines in the chain of future replacements extending as far into the future as the firm's profit horizon.

Hence, he argues that the firm should maximize the present value of the 'aggregate goodwill' of all replacement, where the goodwill is present value of earnings of the future machine, replacements minus the present value of costs all such machines. That is according to Preinreich, one must maximize the expression

$$V(T) = \sum_{k=0}^{\infty} e^{-rkL} \left[\int_T^L (R(t) - E(t))\, e^{-rt}\, dt + S(L)\, e^{-rt} - P\right]$$

where p is the cost of each new machine in the replacements.

These early contributors on the theory of replacement models did not take into consideration the effect of technological changes or obsolescence. The assumption was that machine was to be replaced by the exact 'replica'. That is to say the only factor which eventually made replacement economically justifiable was the effect of age in causing the net earning of a machine to decline.

The idea of replacement models that takes into consideration technological change was given birth by Terborgh(1949). He contended that the replacement of all existing facility should be conditioned by comparison of its performance with that of the latest new comparable facility which is available.

3.2 CATEGORIZATION OF REPLACEMENT MODELS

Replacement models have been divided into five categories.

These are :

3.2.1 Replacement models which are stochastic in nature.

Rifas(1957), developed a replacement model that compares alternatives policies for replacement on the basis of discounted value of all future costs associated with each policy. For periods of equal duration, i = 1,2,….operating costs, C_1, C_2, …. acquisition cost, A and a replacement interval of n periods, he defines k_n, the present value of all future costs as given below;

$$K_n = \frac{A + \sum_{i=1}^{n}[\frac{C_i}{(1 + r)^{i - 1}}]}{1 - [\frac{1}{(1+r)^n}]} \quad .$$

The above equation assumes

(i) an infinite chain of equipment purchases

(ii) the costs are incurred at the beginning each period.

3.2.2 Replacement models wherein more refined and newer technique from statistics and / or mathematics are used and are more theoretical in nature and do not have necessarily any practical problem in view:

Smith(1961), put forth an economic –equipment replacement model that is based on maximizing the present value of all future returns V for an 'optimal replacement policy as applied to one important kind of industrial equipment, viz line-haul-tractor power units. The salient points, in his work are;

(i) the empirical work done at an industry level on obsolesce.

(ii) the inclusion of a factor in the general equipment replacement model to take account of differences in performance of successive members of the chain of machines, and the rationale and statistical justification for the factor.

(iii) Delayed replacement as a cheap source of capital for expanding firm

He considered the following factors;

A=initial cost of the equipment.

S(n) = salvage value of the equipment after n years of usage.

r = rate of interest on capital investment

$Q(K_n, t)$ = annual rate of earnings as a function of purchase time K_n and age t.

The present value of net earnings is then

$$V = e^{-rK_n} \left[\int_0^{r-t} Q(K_n, t) e \, dt - A + S(n) e^r \right]$$

where the net earning could be written as

$$Q(K_n, t) = R(K_n, t) - E(K_n, t) \quad R = \text{gross – revenue rate.}$$

E = operating cost – expense rate.

(3) Replacement models which are dominated by the consideration of the concepts from the theory of reliability renewal theory and estimation of failure rates.

Rifas(1957), developed and solved the group replacement problem where the failure rates and costs of unit replacement after failure and group replacement are known. Let

C_1 = unit cost group replacement.

C_2 = unit cost individual after failure.

f(x) = expected number of failures in the xth period .

N= number of units in the group.

Then the total cost of group replacement after t periods is given by

$$K(t) = NC_1 + C_2 \sum_{x=1}^{t=1} f(x)$$

3.2.3. Replacement models in which replacement is viewed as an investment decision so that the dominance of capital Budgeting considerations is seen explicitly in these models. Orenstein (1956) developed a theory that the economic life is independent of rate of return for an annual-cost-minimization model in the event of equal depreciation payment. He considers three costs:

(i) Acquisition cost, A,

(ii) annual rate of return on capital, r,

(iii) Linear operating cost (Ft dt)

He defines the economic life as one which minimizes the average annual sum of the above three constraints:

$$E = \frac{A}{t} + \frac{A}{2} + [\text{Ftdt} + \frac{t(t+1)}{2}\ \text{dt}]\ \frac{1}{t}$$

$$\Rightarrow \frac{dE}{at} = \frac{A}{t^2} + \frac{d}{2} = 0$$

$$or\ t = \sqrt{\frac{2A}{d}}$$

The economic life is then the value of t_0 such that

$\frac{dt_0}{2} = \frac{A}{t_0}$ or the average increase in operating cost is equal to the annual depreciation value of the equipment.

3.2.4. Replacement models which are put forth by considering concrete replacement problems in view.

Pennycick's(1956) developed a model about replacement – timing which does not consider costs but determines the number of replacements for a specified repair rule. For this purpose he has analyzed data pertaining to the radios used by British army division. He observed that in general the repair rates were increasing suggesting wear-out as a major cause. It was not possible to determine the statistical distribution of periods between repairs for either new or current radios.

However, knowing the number of radios-sets sent to repair in each four-week period, he chose the measure of performance as the percentage increase in unserviceability between periods one year apart. The desired repair rates were specified to be the rates prevailing for the previous year. The problem to be tacked was to determine the number and timing of units of each type of equipment to be replaced necessary to satisfy the repair-rate criteria.

For a given class of equipment he defines

m = mean time to repair all existing sets.

m_1 = mean time to repair the worst r_n percent, n = 100w.

m_2 = mean time torepair the stock after replacing worst r_n percent with/ new units and

$$r_n = \frac{m_1}{m}.$$

He made the following assumptions;

(i) the life of new sets is comparable to the life of the old sets.

(ii) the mean life of the n percent removed is not less than n percent of the mean life of the existing stock.

(iii) the mean life of the best 100(1- w) percent equals $[(1 - wr_n)/(1 – w)]m$.

Injection of the new radio-sets decreases the number of sets sent to repair each period by factor (1 – w). So that

$$\frac{m_2}{m} = \frac{1 - wr_n}{(1 - w)^2}.$$

He observed that $0 < r_n \leq 1$ and for $r_n = 1$, injection of new has the least effect. Conversely the greatest effect occurs for $r_n = 0$ (ie all existing sets are useless).

Normally r_n increases with n and w never exceeds r_n.

3.2.5. Replacement problems models that deal with the situations that arise when some items such as men, machine, and electric light bulbs e.t.c. need replacement due to their decreased efficiency, failure or breakdown.

Some of the situations, which demand the replacement of certain items, are:

(i) The old item has become inefficiency or requires expensive maintenance.

(ii) The old item has failed due to accident or otherwise and does not work at all, or the old item is expected to fail shortly.

(iii) A better design of equipment has been developed or due to obsolescence.

The problem of replacement is to decide the best policy in determining the time at which the replacement is most economical instead of continuing at a increase cost. The main objective of replacement is to direct the organization for maximizing its profit (or minimizing the cost).

The replacement situations may be divided into four categories.

(i) Replacement of capital equipment that suffers heavy depreciation in the course of the time. Examples are machines tools, buses in a transport organization, planes etc.

(ii) Group replacement of items that fail completely. For examples light bulbs, radio tubes

(iii) Problems of mortality and staffing, and

(iv) Miscellaneous problems.

3. 3: REPLACEMENT OF ITEMS THAT DETERIORATE WITH TIME.

Generally, the cost of maintenance and repair of certain items increases with time and at one stage these costs become so high that it is more economical to replace the items by new one. At this point, a replacement is justified.

3.3.1. CASE I: REPLACEMENT WHEN VALUE OF MONEY DOES NOT CHANGE WITH TIME

The aim is to determine the optimum replacement age of an equipment or item whose running or maintenance cost increases with time and the value of money remains the same during the period.

Let C= capital cost of equipment

S=scrap value or resale value of equipment.

N= Number of years the equipment would be in use.

F(t)=maintenance total annual cost function, where time t is a continuous variable.

A(n)= Average total annual cost ,

If the equipment is used for n years then the cost incurred during this period is given by

$$Tc = capital \text{ cost - scrap value + maintenance cost}$$

$$= c - s + \int^{n} f(t)dt$$

Average annuel total cost $A(n) = \frac{1}{n} Tc$

$$= \frac{\text{c-s}}{\text{n}} + \frac{1}{\text{n}} \int_a^n f(t) dt.$$

For minimum cost, we have $\frac{\text{d}}{\text{d}_\text{n}} (A(n)) = 0$

or $\text{f(n)} = \frac{\text{c-s}}{\text{n}} + \int_\text{a}^\text{n} \text{f(t)dt} = \text{A(n)}.$

Clearly $\frac{\text{d}^2}{dn^2} (A(n)) \geq 0$ at f(n)= A(n).

This suggests that the equipment should be replaced when the maintenance cost equals the average annual total cost.

For discrete value of t, *n* is optimal at the average annual cost.

This suggests the optimal replacement policy. According to which:

$$F(n) = \frac{1}{\text{n}} \left[\text{c-s(t)} + \sum_\text{a}^\text{n} \text{f(t)} \right]$$

(a) Replace the equipment at the end of n years, if the maintenance cost in the $(n+1)^{th}$ year, is more than the average total cost in the nth year.

(b) Do not replace the equipment if the current year's maintenance cost is less than the previous year's total cost.(Kalavathy, 2008)

3.3.2: CASE II: REPLACEMENT WHEN VALUE OF MONEY CHANGES WITH TIME.

When the time value of money is taken into consideration, we shall assume that

(i) the equipment in question has no salvage value

(ii) the maintenance cost are incurred in the beginning of various periods.

To find the optimal replacement Policy, we first find the present worth factor (PWF) and the discount rate.

(iii) Present worth factor (PWF): If r is the rate of interest then it is called the present worth factor . The expression $(1+r)^{-n}$ is known as the payment compound amount factor (caf) in n years of operation.

(iv) Discount rate: The present worth factor of a unit amount to be spent after one year is given by $V=(1+r)^{-1}$, where r is called the rate of interest and V is called discount rate.

Let the initial cost of the equipment be C and let R_n be the operation cost in year n. Let r be the rate of interest in such a way that $V= (1+r)^n$ is the discount rate. We find the weighted average cost of all the previous n years with weights, 1, V, $V^2, \ldots V^{n-1}$ respectively.

The expression weighted average cost is given by

$$W(n) = \frac{C + R_0 + VR_1 + V^2R_2 + \ldots + V^{n-1}R_{n-1}}{1 + V + V^2 + \ldots + V^{n-1}}$$

The optimal replacement policy of the equipment after n period is

(i) Do not replace the equipment if the next period's cost is less than the weighted average of previous costs.

(ii) Replace the equipment if the next period's cost is greater than the weighted average of previous costs.

<u>Selection of the best equipments</u>

We use the average cost equipment above to find an economically best item amongst the available equipments. The following procedure used;

Step 1: Considering the case of two equipments, A, B, we first find the replacement age for both the equipments by making use of $R_{n-1} < w(n) < R_n$.

Let n_1, n_2, be the replacement age for the equipments A and B.

Step 2: Next,we find the weighted average cost for each equipment .

Substitute n = n_1 for equipment A and n = n_2 for equipment B.

Step 3: (i) If $w(n_1) < w(n_2)$, choose equipment A

(ii) If $w(n_1) > w(n_2)$, choose equipment B.

(iii) If $w(n_2) = w(n_1)$, both equipment are equally good.

Note : if the salvage is not negligible the expression for weighted average cost is given by

$$W(n) = \frac{C + \sum_{n=1} R_n V^{n-1} - S_n V^n}{\sum_{n=1} V^{n-1}} \qquad \text{(Kalavathy, 2008).}$$

3.3.3: REPLACEMENT OF EQUIPMENT THAT FAILS SUDDENLY.

It is difficult to predict that a particular equipment will fail at a particular time. This difficulty can be overcome by determining the probability distribution of failures. Assuming that the failures occur only at the end of period say t, the objective is to find the value of t which minimize the total cost involved for the replacement.

We shall consider the following two types of replacement policies:

(i) individual replacement policy. Under this policy an item is replaced immediately after it fails.

(ii)Group replacement policy . Under this policy, we take decisions as to when all the items must be replaced irrespective of the fact that an item has failed or has not failed with a provision that if any item fails before the optimal time it may be individually replaced.

Hence, we have the following optimal policy.

Let all the items in a system be replaced after a time interval ‘t’ with provisions that individual replacement can be replaced if and when any item fails during this time period.

Group replacement must be made at the end of r^{th} period if the cost of individual replacements for the r^{th} period is greater than the average cost per period through the end of r periods.

Group replacement is not possible at the end of period t if the cost of individual replacement at the t-1 end period is less than the average cost per period through the end of t period.

Example

The following mortality rates have been observed for a certain type of bulbs.

Table 3.1: Data of percentage of failure of bulbs

Week	1	2	3	4	5
Percent failing by the end of week.	10	25	50	80	100

There are 100 bulbs in use and it costs R_s = \$2.00 to replace an individual bulb which has burnt out. If all the bulbs were replaced simultaneously, it would cost 0.50 per bulb. It is proposed to replace all bulbs at fixed intervals whether or not they have burnt out and to continue replacing burnt out bulbs as they fail. At what intervals should all the bulbs be replaced?

Solution

Step 1: Let pi be the probability that a bulb which was new when placed in position for use, fails during the ith week of its life.

$P_1 = 0.1$

$P_2 = 0.25 – 0.1= 0.15$

$P_3 = 0.5 – 0.25 = 0.25$

$P_4 = 0.8 – 0.5 = 0.3$

$P_5 = 1.0 – 0.8 = 0.2.$

Since the sum of probabilities is 1, all the probabilities beyond P_5 will be take as zero.

Step 2: Let N_i be the number of replacement at the end of the i^{th} week.

N_0 = Number of items in the beginning.= 1000.

$N_1 = N_0P_1 = 1000 (0.1) = 100.$

$N_2 = N_0P_2 + N_1P_1 = 150 + 10 =160.$

$N_3 = N_0P_3 + N_1P_2 + N_2P_1 = 1000(0.25) + 100(0.15) + 160(0.1) = 281$

$N_4 = N_0P_4 + N_1P_3 + N_2P_2 + N_4P_1 = 1000(0.3) + 100(0.25) + 160(0.15) + 281(0.1) = 379.$

$N_5 = N_0P_5 + N_1P_4 + N_2P_3 + N_4P_2 + N_5P_1 = 1000(0.2) + 100(0.3) + 160(0.25) + 281(0.15) + 379(0.1) = 350.$

From the above caiculations, we observe that the expected number of bulbs failing each week increases till 4^{th} week and then starts decreasing. Thus , Ni will oscillate till the system acquires a steady state.

Step 3: we calculate the expected life of each bulb

$$= \sum_{i=1}^{5} iP_i$$

$$= 1(0.1) + 2(0.15) + 3(0.2) = 3.35.$$

Average number of failure per week $= \dfrac{1000}{3.35} = 299.$

Step 4: The cost of individual replacement = 299(2) = R_s = \$598.

Now, since the replacement of all the 1000 bulbs simultaneously cost \$0.50 per bulb and the replacementof an individual bulb on failure cost Rs = \$2.00, the average for different group replacement policies is given below.

Table 3.2: Summary of the optimal policy of replacing the bulbs

End of week	Individual replacement	Total cost R_s (individual + group)	Average cost
1	100	100(2) + 1000(0.5) = 700	700
2	100 + 160 = 260	260(2) + 1000(0.5) = 1020	510 ●
3	260 + 281 = 541	541(2) + 1000(0.5) = 1582	527.33
4	541 + 379 = 920	920(2) + 1000(0.5) = 2340	585
5	920 + 350 = 3040	3040(2) + 1000(0.5) = 3040	608

Since the average cost is minimum in the 2nd week, the optimal replacement period to have a group replacement is after every 2nd week.

Since the average cost is less than R_s = \$598.00 for individual, the group replacement policy is preferable (Kalavathy, 2008).

3.4: SUMMARY

This chapter has looked at optimization techniques, recursive algorithm, dynamic programming- literature survey and application to various problems, and the literature survey of equipment replacement models.

In the next chapter we apply the dynamic programming algorithm to a equipment replacement model of State Transport Company (STC).

CHAPTER FOUR

DATA COLLECTION, ANALYSIS AND RESULTS

4.1 DATA FROM THE COMPANY

The data obtained was the cost of a new bus, age (yr), operating cost, and salvage value of a bus after one, two, three, four, five and six years of usage. Data was obtained from STC workshop office in Oforikrom (Kumasi).

Since none of their buses exceed six years before being disposed of and the current Higher buses are in their third year. Therefore, year three is the starting point of the machine year as shown in the figure below.

The main aim is to determine an optimal replacement policy for the State Transport Company buses over the next 4 years.

The current higher buses are in the third year. The cost of a new higher bus (higher) is GH¢180,000.00.

The table 4.1. below gives the age (yr), revenue, r(t), operating cost, c(t) and the salvage values s(t) of STC Higher bus.

Table 4.1: Data obtained from STC

Age, t(yr)	Revenue, r(t) GH¢	Operating cost, c(t) GH¢	Salvage value, s(t) GH¢
0	144000	30000	-
1	143000	31000	160000
2	138000	32500	150000
3	132000	34000	145000
4	125000	36000	115000
5	115000	38000	100000
6	105000	40000	70000

In figure 9 below, at the start of operational year 1, we have a 3-year-old bus. We can either replace it (R) or keep it (K) for another year. At the start of operational year 2, if replacement occurs, the new bus will be 1 year old; otherwise, the old bus will be 4 years old. The logic applies for all the years until the bus is six years old.

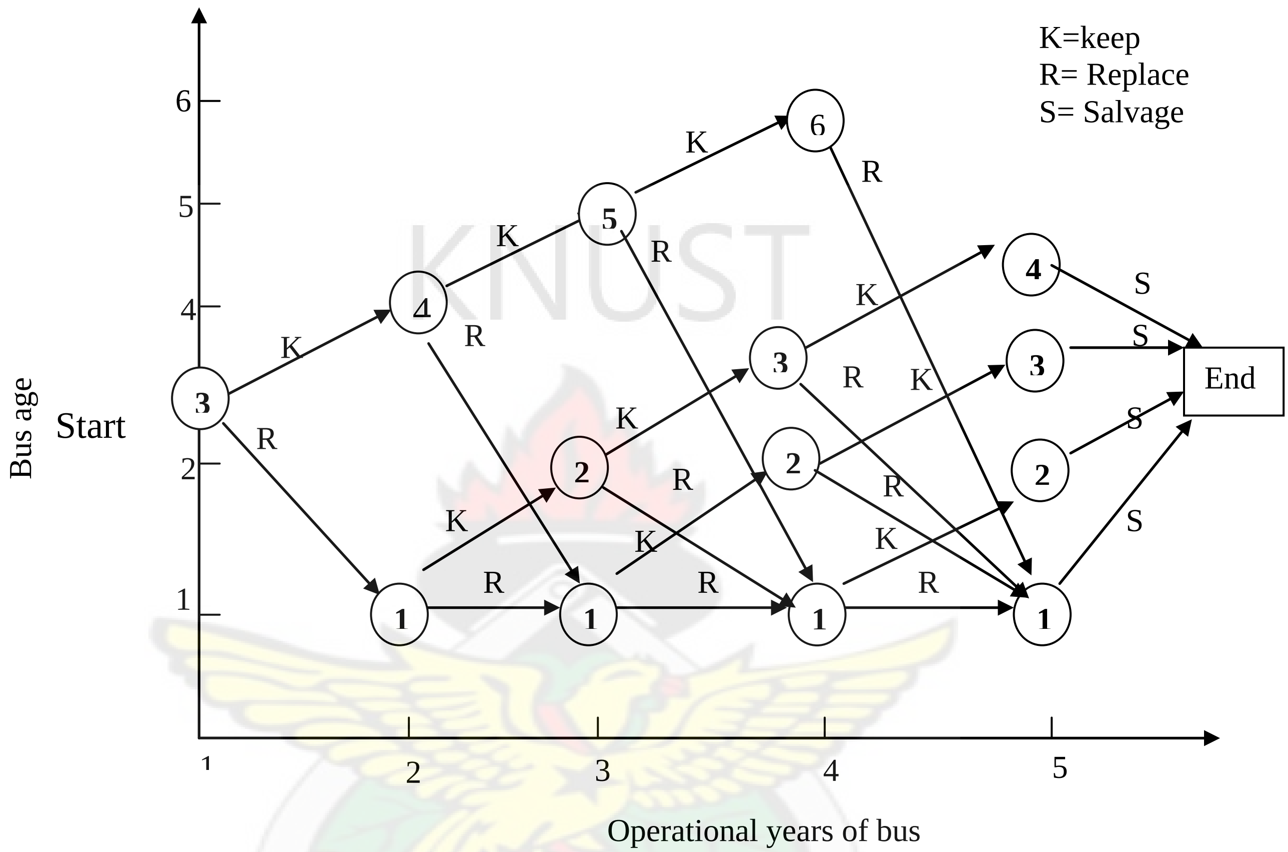


Figure 9: A representation of the machine age as a function of operational year

4.1.1: MODEL FORMULATION

The formulation for the problem is

$$f_i(t) = \max \begin{cases} r(t) - c(t) + f_{i+1}(t+1), & \text{if KEEP} \\ r(0) + s(t) - I - c(0) + f_{i+1}(1), & \text{if REPLACE} \end{cases} \quad \text{i=1,2,3,4 subject to}$$

$f_{n+1}(.) \equiv 0$

The problem is solved for each value of i

(i) when i=4, t=1,2,3,6

(ii) when i=3, t=1,2,5

(iii) when i=2, t=1,4

(iv) when i=1, t=3.

The values of c(t), r(t) and s(t) are found in table, r(0)=144000, c(0)=30000, I=180000

4.1.2: STRUCTURE OF THE PROBLEM

Let divide the problem into four stages as detailed in figure 10:

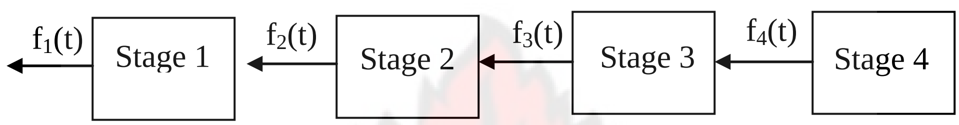


Figure 10: Schematic representation of the equipment replacement problem as a four-stage dynamic programming problem.

Stage 1: For the start of year 1; t= 3

The return function is the maximum net income (profit) for year 1; t=3

$$f_1(t) = \max \begin{cases} r(t) - c(t) + f_2(t+1), \text{ if KEEP} \\ r(0) + s(t) - I - c(0) + f_2(1), \text{ if REPLACE} \end{cases}$$

Stage 2: For the start of year 2; t=1, 4

The return function is the maximum net income (profit) for year 2; t=1, 4

$$f_2(t) = \max \begin{cases} r(t) - c(t) + f_3(t+1), \text{ if KEEP} \\ r(0) + s(t) - I - c(0) + f_3(1), \text{ if REPLACE} \end{cases}$$

Stage 3: For the start of year 3; t=1, 2, 5

The return function is the maximum net income (profit) for year 3; t= 1, 2, 5

$$f_3(t) = \max \begin{cases} r(t) - c(t) + f_4(t+1), \text{ if KEEP} \\ r(0) + s(t) - I - c(0) + f_4(1), \text{ if REPLACE} \end{cases}$$

Stage 4: For the start of year 4; t=1, 2, 3, 6.

The return function is the maximum net income (profit) for year 4; t=1,2,3,6

$$f_4(t) = \max \begin{cases} r(t) + s(t+1) - c(t), \text{ if KEEP} \\ r(0) + s(t) + s(1) - I - c(0), \text{ if REPLACE} \end{cases}$$

4.1.3 COMPUTATIONAL RESULTS

The solution of the network in figure 4 is finding the maximum profit from the start of operational year 1 to the end of operational year 4. All values are in thousands of Ghana cedis. Note that if a bus is replaced in operational year 4(i.e., end of the planning horizon), its revenue will include the salvage value, s(t), of the replaced bus and the salvage value, s(1), of the replacement bus.

Stage 4 Computation for operational year 4.

$$f_4(t) = \max \begin{cases} r(t) + s(t+1) - c(t), \text{ if KEEP} \\ r(0) + s(t) + s(1) - I - c(0), \text{ if REPLACE} \end{cases}$$

For i=4, t=1, r(1)= 143,s(2)=150, c(1)=31, r(0)=144, s(1)=160, I=180.

$$f_4(1) = \max \begin{cases} 143 + 150 - 31 = 262 \text{ if KEEP.} \\ 144 + 160 + 160 - 30 - 180 = 254 \text{ if REPLACE.} \end{cases}$$

$f_4(1) = 262$ keep

For i=4, t=2, r(2)=138, s(3)=145, c(2)=32.5, r(0)=144, s(2)=150, s(1)=160, c(0)=30, I=180. We picked from table 3.0.

$$f_4(2) = \max \begin{cases} 138 + 145 - 32.5 = 250.5, \text{ if KEEP} \\ 144 + 150 + 160 - 30 - 180 = 244, \text{ if REPLACE} \end{cases}$$

$f_4(2) = 250.5$ keep

For i=4, t=3, r(3)=132, s(4)=115, c(2)=34, r(0)=144, s(3)=145, s(1)=160, c(0)=30, I=180. We picked from table 3.0.

$$f_4(3) = \max \begin{cases} 132 + 115 - 34 = 213, \text{ if KEEP} \\ 144 + 145 + 160 - 30 - 180 = 239, \text{ if REPLACE} \end{cases}$$

$f_4(3) = 239$ replace

For i=4, t=6 , r(0)=144, s(3)=70, s(1)=160, c(0)=30, I=180,

$$f_4(6) = \max \begin{cases} \textit{must} \text{ replace} \\ \text{144+70+160-30-180=164 if replace} \end{cases}$$

$$f_4(t) = \max \begin{cases} r(t) + s(t+1) - c(t), \text{ if KEEP} \\ r(0) + s(t) + s(1) - I - c(0), \text{ if REPLACE} \end{cases}$$

In table 4.2. below, column 1 is the age of the vehicle and column 2 the revenue that will be generated when the bus is kept, column 3 the revenue when the bus is replaced and the last column is the optimal decision to either keep or replace the bus.

Table 4.2: summary of results for the computation of operational year 4

	K	R	Optimum solution	
t	r(t)+s(t+1)- c(t)	r(0)+s(t)+s(1)-c(0)-I	$f_4(t)$	Decision
1	143+150-31=262	144+160+160-30-180=254	262	K *
2	138+145-32.5=250.5	144+150+160-30-180=244	250.5	K
3	132+115-34=213	144+145+160-30-180=239	239	R
6	(must replace)	144+70+160-30-180=164	164	R

Stage 3: Computation for operational year i= 3, t=1,2,5

$$f_i(t) = \max \begin{cases} r(t) - c(t) + f_{i+1}(t+1), & \text{if KEEP} \\ r(0) + s(t) - I - c(0) + f_{i+1}(1), & \text{if REPLACE} \end{cases}$$

$$f_{n+1}(.) \equiv 0$$

From table 4.1 above, we have t=1,2,5, r(1)=143, c(1)=31, $f_4(2)$=250.5, r(0)=144, c(0)=30, s(1)=160, I=180, $f_4(1)$=262, r(2)=138, c(2)=32.5, $f_4(3)$)=239, s(2)=150, r(5)=100, c(5)=38, $f_4(6)$=164, s(5)=100.

In table 4.3 below, column 1 is the age of the vehicle and column 2 the revenue that will be generated when the bus is kept, column 3 the revenue when the bus is replaced and the last column is the optimal decision to either keep or replace the bus.

Table 4.3: summary of results for the computation of operational year 3

	K	R	Optimum solution
t	r(t)-c(t)+f_4(t+1)	r(0)+s(t)-c(0)-I+ f_4(1)	f_3(t) Decision
1	143-31+250.5=362.5	144+160-30-180+262=356	362.5 K
2	138-32.5+239=344.5	144+150-30-180+262=346	346 R *
5	100-38+164=226	144+100-30-180+262=296	296 R

Stage 2: Computation for operational year i=2, t=1,4.

From table 4.1, we have t= 1,4, r(1)=143, c(1)=31, $f_3(2)$=346, r(0)=144, s(1)=160, c(0)=30, I=180, $f_3(1)$=362, r(4)=120, c(4)=36, $f_3(5)$=296, s(4)=115.

In table 4.4 below, column 1 is the age of the vehicle and column 2 the revenue that will be generated when the bus is kept, column 3 the revenue when the bus is replaced and the last column is the optimal decision to either keep or replace the bus.

Table 4.4: summary of results for the computation of operational year 2

	K	R	Optimum solution	
t	$r(t)-c(t)+f_3(t+1)$	$r(0)+s(t)-c(0)-I + f_3(1)$	$f_2(t)$	Decision
1	143-31+346=458	144+160-30-180+362.5=456.5	458	K *
4	120-36+296=380	144+115-30-180+362.5=411.5	411.5	R

Stage 1: Computation for operational year i=1, t=3.

From table 4.1 above, we have r(3)=132, c(3)=34, $f_2(4)$=411.5, r(0)=144, s(3)=145, c(0)=30, I=180, $f_2(1)$ = 458.

In table 4.5 below, column 1 is the age of the vehicle and column 2 the revenue that will be generated when the bus is kept, column 3 the revenue when the bus is replaced and the last column is the optimal decision to either keep or replace the bus.

Table 4.5: summary of results for the computation of operational year 1.

	K	R	Optimum solution	
t	$r(t)-c(t)+f_2(t+1)$	$r(0)+s(t)-c(0)-I+ f_2(1)$	$f_1(t)$	Decision
3	132-34+411.5=509.5	144+145-30-180+458=533	553	R *

Figure 11 summarizes the optimal solution. At the start of year 1, given t=3, the optimal decision is to replace the machine. Thus, the new machine will be 1 year old at the start of year 2, and t=1 at the start of year 2 calls for keeping bus. The kept machine will be 2 years old at the start of year 3 and should be replaced. In addition, at the start of year 4 the bus will be one year old and should be kept.

(t=3)→ $\boxed{R}$ →(t=1)→ $\boxed{K}$ →(t=2)→ $\boxed{R}$ →(t=1) → $\boxed{K}$ → sell.

Figure 11 Schematic representation of the optimal solution of the replacement policy.

The optimal policies starting in operational year 1 are (R, K, R, K).

The optimal profit is $f_1(3) = 537$ when the bus is replaced ie ffrom table 4.5, $f_2(1) = 458$ when the bus is kept ie from table 4.4, $f_3(2) = 346$ when the bus replaced ie from table 4.3, $f_4(1) = 262$ when the bus is kept ie from table 4.2.

4.2. DISCUSSION

From the computation (analysis) above, in order for the company to maximize profit, they should keep the bus (higher) when it is one old. This will yield them a profit of GH¢8,000.00(262000 – 254000 = 8000) i.e. from table 4.2 row 3.

However, the bus should be replaced when it is two years old, else the company will be running at a lost of GH¢1500 (34600 – 344500 = 1500) for keeping the old one i.e. from table 4.3 row 4.

If the two year old bus is not replaced, in its third year, the company will be losing GH¢23,500(533000 – 509500 = 23500) i.e. from table 4.5 row 3.

In addition, in the fourth year of the bus the company will be running at a lost of GH¢21500(411500 – 380000 = 21500) for keeping the old bus ie from table 4.4 row 4.

Again, when the bus is not replaced, in its fifth year the company stands to loss as much as GH¢70000(296000 – 226000 = 70000) for keeping it i.e. from table 4.3 row 5.

The formulation of the problem

$$f_i(t) = \max \begin{cases} r(t) - c(t) + f_{i+1}(t+1), & \text{if KEEP} \\ r(0) + s(t) - I - c(0) + f_{i+1}(1), & \text{if REPLACE} \end{cases}$$

$$f_{n+1}(.) \equiv 0$$

by Hamdy (2007) was used in finding the policy.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1. CONCLUSION

Considering the optimal profit for the four-year policy, we see that the maximum profit is obtained, if the higher buses are replaced when they are two years old.

5.2. RECOMMENDATIONS

With the results above, we recommend that the State Transport Company should always dispose off its buses after two years of usage.

Students may also do further research into the applications of dynamic programming in other areas.

REFERENCES

Adda, Jerome, Cooper, Russel (2003). An accessible introduction to dynamic programming in economics, MIT press, Cambridge.

Alexander, S.(1998). Theory of Linear Programming and Integer Programming. JohnWiley and Sons, New York.

Amponsah, S. K. (2006). Optimization techniques 1, University Printing Press, KNUST, Kumasi.

Beaumont, N. (2007). Using dynamic programming to determine an optimal strategy in a contract bridge tournament. Journal of operational research society advance online publication 22/04/09.

Bellman, (1957). Dynamic programming, Princeton university press, New Jersey.

Bellman, R. and Kalaba, R.(1963); An inverse problem in dynamic programming and automatic control. Journal of mathematical analysis and applications, volume 7, pp 322-325.

Bellman, R. E. and Dreyfus, S. E. (1962). Applied dynamic programming. Princeton University Press, New Jersey.

Bush W. R., Pincus D. J. and Sielaff D. J. A static analyzer for finding dynamic programming errors, journal willey interscience, volume 30, issue 7, pp 775-802.

Cornuejols, G., Nemhauser, G. L. and Wolsey, L. A. (1990). The uncapacitated Facility Location Problem in discrete Location Theory(P. B Mirchandani and R. L. Francis, eds), Wiley, New York, chapter 3, pp 119-171.

Daniel, P. W., Ellis (2007), Beat Tracking by Dynamic Programming, Journal of new music research, volume, 36, issue 1, pp 51-60.

David R. A., Sweeney J. D., Thomas A. W. (1988). An introduction to management science, West Publishing Company, San Francisco.

Hamdy, A. T. (2007). Operations research, an introduction 8th edition. Pearson Education Inc., Singapore.

Hillier F. S. and Gerald J. L. (2005). Introduction to operations research. McGraw-Hill, New York.

Intercity STC (2004), BusinessGhana, Wikipedia, free encyclopedia.

Lay, Maxwell G (1992), Ways of the History of the World's Roads and of the Vehicles that used them, Rutgers University press.

Mahmut, P. (2000), Interactive Operations Research with Maple: Methods and Models. Birkhauser, Boston.

Nicole, E. K. and Marie-Claire Quenez (1995), Dynamic programming of contingent claims incomplete market: SIAM J control and optimization volume 33, issue 1, pp 29-66.

Normal J M and Clarke S. R.(2004), dynamic programming in cricket: optimizing batting order for a sticky wicket, journal of the operational Research society, volume 48, pp 1678-1682.

Orenstein, R.B., (1956). Topic on the MAPI, Journal of industrial Engineering, volume 7, (pp. 283-294)

Pennycuick, K.,(1956). A problem of replacement, Operational Research Quarterly. Preinreich, G. D.,(1940). The economic life of industrial equipment, econometrical.

Quansong R; Steele A. J., Schwalbach M. S., Fuhrman J. A. and Fengzhu S.(2006). A dynamic programming for binning microbial community profiles. Journal on dynamic programming in Bioinformatics, volume 22, pp 1508-1514.

Rifas, E. B.,(1957). An introduction to operational research, John Wiley and Sons press, New York.

Slater L. J. (1964). A dynamic programming process, journal on dynamic programming in the computer , volume 7, pp 36-39.

Smith K. D, Pass M, (1997), Dynamic programming: an introduction, MIT press, Cambridge.

Stopford, Martin (1997), Maritime Economies. Routledge, London.

Taylor, J. S., (1923). A statistical theory of depreciation, Journal of the American Statistical Association, McGraw-Hill book Company, New York.

Terborgh, G., (1949). Dynamic equipment policy, McGraw-Hill book Company, New York.

Thomas H. C., Leiserson, Charles. E., Rivest, R. L., and Stein, C. (2002). Introduction algorithms (2nd edition) McGraw-Hill, New York.

Totelling, H., (1925). A general mathematical theory of depreciation, Journal of the American Statistical Association, McGraw-Hill book Company, New York.

Varaiya, P., (1970). Lecture notes on optimization.

Zeqing L. and Shin M. K.(2006), Properties of solutions for certain functional equations arising in dynamic programming, journal of global optimization, volume 34, issue 2, pp 273-292.

APPENDIX A

Computations for Stage 1(filling with item category 1only): n = 1, of Job data for the manufacturing operation (page 49)

$$f_1(10) = \max.\begin{cases} 2\times 0 = 0,\ d_1 = 0. \\ 2\times 1 = 2,\ d_1 = 1. \\ 2\times 2 = 4,\ d_1 = 2. \\ 2\times 3 = 6,\ d_1 = 3. \\ 2\times 4 = 8,\ d_1 = 4. \end{cases} \qquad x_1 = 10$$

Result $f_1(10) = 8,\ d^*_1 = 4$.

$$f_1(9) = \max.\begin{cases} 2\times 0 = 0,\ d_1 = 0. \\ 2\times 1 = 2,\ d_1 = 1. \\ 2\times 2 = 4,\ d_1 = 2. \\ 2\times 3 = 6,\ d_1 = 3. \\ 2\times 4 = 8,\ d_1 = 4. \end{cases} \qquad x_1 = 9$$

Result $f_1(9) = 8,\ d^*_1 = 4.$

$$f_1(8) = \max.\begin{cases} 2\times 0 = 0,\ d_1 = 0. \\ 2\times 1 = 2,\ d_1 = 1. \\ 2\times 2 = 4,\ d_1 = 2. \\ 2\times 3 = 6,\ d_1 = 3. \\ 2\times 4 = 8,\ d_1 = 4. \end{cases} \qquad x_1 = 8$$

Result $f_1(8) = 8,\ d^*_1 = 4.$

$$f_1(7) = \max.\begin{cases} 2\times 0 = 0,\ d_1 = 0. \\ 2\times 1 = 2,\ d_1 = 1. \\ 2\times 2 = 4,\ d_1 = 2. \\ 2\times 3 = 6,\ d_1 = 3. \\ 2\times 4 = 8,\ d_1 = 4. \end{cases} \qquad x_1 = 7$$

Result $f_1(7) = 8,\ d^*_1 = 4.$

$$f_1(6) = \max.\begin{cases} 2\times 0 = 0,\ d_1 = 0. \\ 2\times 1 = 2,\ d_1 = 1. \\ 2\times 2 = 4,\ d_1 = 2. \\ 2\times 3 = 6,\ d_1 = 3. \\ 2\times 4 = 8,\ d_1 = 4. \end{cases} \qquad x_1 = 6$$

Result $f_1(6) = 8,\ d^*_1 = 4.$

$$f_1(5) = \max.\begin{cases} 2\times 0 = 0,\ d_1 = 0. \\ 2\times 1 = 2,\ d_1 = 1. \\ 2\times 2 = 4,\ d_1 = 2. \\ 2\times 3 = 6,\ d_1 = 3. \\ 2\times 4 = 8,\ d_1 = 4. \end{cases} \qquad x_1 = 5$$

Result $f_1(5) = 8,\ d^*_1 = 4.$

$$f_1(4) = \max.\begin{cases} 2\times 0 = 0,\ d_1 = 0. \\ 2\times 1 = 2,\ d_1 = 1. \\ 2\times 2 = 4,\ d_1 = 2. \\ 2\times 3 = 6,\ d_1 = 3. \\ 2\times 4 = 8,\ d_1 = 4. \end{cases} \qquad x_1 = 4$$

Result $f_1(4) = 8,\ d^*_1 = 4.$

$$f_1(3) = \max.\left\{\begin{matrix} 2\times 0 = 0,\ d_1 = 0 \\ 2\times 1 = 2,\ d_1 = 2 \\ 2\times 2 = 4,\ d_1 = 2 \\ 2\times 3 = 6,\ d_1 = 3 \end{matrix}\right\} \qquad x_1 = 3$$

Result $f_1(3) = 6,\ d^*_1 = 3.$

$$f_1(2) = \max.\left\{\begin{matrix} 2\times 0 = 0,\ d_1 = 0 \\ 2\times 1 = 2,\ d_1 = 1 \\ 2\times 2 = 4,\ d_1 = 2 \end{matrix}\right\} \qquad x_1 = 2.$$

Result $f_1(2) = 4,\ d^*_1 = 2.$

$$f_1(1) = \max.\begin{Bmatrix} 2\times 0 = 0,\ d_1 = 0 \\ 2\times 1 = 2,\ d_1 = 1 \end{Bmatrix} \qquad x_1 = 1$$

Result $f_1(1) = 2,\ d^*_1 = 1.$

$$f_1(0) = \max\{2\times 0 = 0,\ d_1 = 0.\ \} \qquad x_1 = 0.$$

Result $f_1(0) = 0,\ d^*_1 = 0.$

APPENDIX B

Computations for Stage 2 (filling with items 1 and 2): n=2, of Job data for the manufacturing operation (page 50)

$$f_2(10) = \max.\begin{cases} 8\times 0 + f_1(10) = 0+8 = 8,\ \mathrm{d}_2 = 0. \\ 8\times 1 + f_1(7) = 8+8 = 16,\ \mathrm{d}_2 = 1. \\ 8\times 2 + f_1(4) = 16+8 = 24,\ \mathrm{d}_2 = 2. \\ 8\times 3 + f_1(1) = 24+2 = 26,\ \mathrm{d}_2 = 3 \end{cases} \qquad x_2=10.$$

Result $f_2(10) = 26$, $d^*_2=3$.

$$f_2(9) = \max.\begin{cases} 8\times 0 + f_1(9) = 0+8 = 8,\ \mathrm{d}_2 = 0. \\ 8\times 1 + f_1(6) = 8+8 = 16,\ \mathrm{d}_2 = 1. \\ 8\times 2 + f_1(3) = 16+6 = 22,\ \mathrm{d}_2 = 2. \\ 8\times 3 + f_1(0) = 24+0 = 24,\ \mathrm{d}_2 = 3 \end{cases} \qquad x_2=9.$$

Result $f_2(9) = 24$, $d^*_2=3$.

$$f_2(8) = \max.\begin{cases} 8\times 0 + f_1(8) = 0+8 = 8,\ \mathrm{d}_2 = 0. \\ 8\times 1 + f_1(5) = 8+8 = 16,\ \mathrm{d}_2 = 1. \\ 8\times 2 + f_1(2) = 16+4 = 20,\ \mathrm{d}_2 = 2. \end{cases} \qquad x_2 = 8.$$

Result $f_2(8) = 20$, $d^*_2=2$.

$$f_2(7) = \max.\begin{cases} 8\times 0 + f_1(7) = 0+8 = 8,\ \mathrm{d}_2 = 0. \\ 8\times 1 + f_1(4) = 8+8 = 16,\ \mathrm{d}_2 = 1. \\ 8\times 2 + f_1(1) = 16+2 = 18,\ \mathrm{d}_2 = 2. \end{cases} \qquad x_2 = 7.$$

Result $f_2(7) = 18$, $d^*_2=2$.

$$f_2(6) = \max.\begin{cases} 8\times 0 + f_1(6) = 0+8 = 8,\ \mathrm{d}_2 = 0. \\ 8\times 1 + f_1(3) = 8+6 = 14,\ \mathrm{d}_2 = 1. \\ 8\times 2 + f_1(0) = 16+0 = 16,\ \mathrm{d}_2 = 2. \end{cases} \qquad x_2 = 6.$$

Result $f_2(6) = 16$, $d^*_2=2$.

$$f_2(5) = \max.\begin{cases} 8\times 0 + f_1(5) = 0 + 8 = 8,\ d_2 = 0. \\ 8\times 1 + f_1(2) = 8 + 4 = 12,\ d_2 = 1. \end{cases}$$ $x_2 = 5.$

Result $f_2(5) = 12,\ d^*_2 = 1.$

$$f_2(4) = \max.\begin{cases} 8\times 0 + f_1(4) = 0 + 8 = 8,\ d_2 = 0. \\ 8\times 1 + f_1(1) = 8 + 2 = 10,\ d_2 = 1. \end{cases}$$ $x_2 = 4$

Result $f_2(4) = 10,\ d^*_2 = 1.$

$$f_2(3) = \max.\begin{cases} 8\times 0 + f_1(3) = 0 + 6 = 6,\ d_2 = 0. \\ 8\times 1 + f_1(0) = 8 + 0 = 8,\ d_2 = 1. \end{cases}$$ $x_2 = 3.$

Result $f_2(3) = 8,\ d^*_2 = 1.$

$$f_2(2) = \max\{8\times 0 + f_1(2) = 0 + 4 = 4,\ d_2 = 0\}$$ $x_2 = 2.$

Result $f_2(2) = 4,\ d^*_2 = 0.$

$f_2(1) = \max.\{8\times 0 + f_1(1) = 0 + 2 = 2,\ d_2 = 0\}.$ $x_2 = 1.$

Result $f_2(1) = 2,\ d^*_2 = 0.$

$f_2(0) = \max.\{8\times 0 + f_1(0) = 0 + 0 = 0,\ d_2 = 0\}.$ $x_2 = 0.$

Result $f_2(0) = 0,\ d^*_2 = 0.$

APPENDIX C

Computations for Stage 3, (filling with items 1, 2 and 3): n=3, of Job data for the manufacturing operation (page 50)

$$f_3(10) = \max.\begin{cases} 11\times 0 + f_2(10) = 0 + 26 = 26,\ \mathrm{d}_3 = 0. \\ 11\times 1 + f_2(6) = 11 + 16 = 27,\ \mathrm{d}_3 = 1. \\ 11\times 2 + f_2(2) = 22 + 4 = 24,\ \mathrm{d}_3 = 2. \end{cases} \qquad x_3=10\ .$$

Result $f_3(10) = 27,\ \mathrm{d}^*{}_3 = 1.$

$$f_3(9) = \max.\begin{cases} 11\times 0 + f_2(9) = 0 + 24 = 24,\ \mathrm{d}_3 = 0. \\ 11\times 1 + f_2(5) = 11 + 12 = 23,\ \mathrm{d}_3 = 1. \\ 11\times 2 + f_2(1) = 22 + 2 = 24,\ \mathrm{d}_3 = 2. \end{cases} \qquad x_3=9.$$

Result $f_3(9) = 24,\ \mathrm{d}^*{}_3=0,\ \mathrm{d}^*{}_3 = 2.$

$$f_3(8) = \max.\begin{cases} 11\times 0 + f_2(8) = 0 + 20 = 20,\ \mathrm{d}_3 = 0. \\ 11\times 1 + f_2(4) = 11 + 10 = 21,\ \mathrm{d}_3 = 1. \\ 11\times 2 + f_2(0) = 22 + 0 = 22,\ \mathrm{d}_3 = 2. \end{cases} \qquad x_3=8$$

Result $f_3(8) = 22,\ \mathrm{d}^*{}_3 = 2.$

$$f_3(7) = \max.\begin{cases} 11\times 0 + f_2(7) = 0 + 18 = 18,\ \mathrm{d}_3 = 0. \\ 11\times 1 + f_2(3) = 11 + 8 = 19,\ \mathrm{d}_3 = 1. \end{cases} \qquad x_3=7$$

Result $f_3(7) = 19,\ \mathrm{d}^*{}_3 = 1.$

$$f_3(6) = \max.\begin{cases} 11\times 0 + f_2(6) = 0 + 16 = 16,\ \mathrm{d}_3 = 0. \\ 11\times 1 + f_2(2) = 11 + 4 = 15,\ \mathrm{d}_3 = 1. \end{cases} \qquad x_3=6$$

Result $f_3(6) = 16,\ \mathrm{d}^*{}_3 = 0.$

$$f_3(5) = \max.\begin{cases} 11\times 0 + f_2(5) = 0 + 12 = 12,\ \mathrm{d}_3 = 0. \\ 11\times 1 + f_2(1) = 11 + 2 = 13,\ \mathrm{d}_3 = 1. \end{cases} \qquad x_3=6$$

Result $f_3(5)=13,\ \mathrm{d}^*{}_3=1.$

$$f_3(4)=\max.\begin{cases}11\times 0+f_2(4)=0+10=10,\ \mathrm{d}_3=0.\\ 11\times 1+f_2(0)=11+0=11,\ \mathrm{d}_3=1.\end{cases}$$ x_3=4

Result $f_3(4)=11,\ \mathrm{d}^*{}_3=1.$

$f_3(3)=\max\{11\times 0+f_2(3)=0+8=8,\ \mathrm{d}_3=0\ \}.$

Result $f_3(3)=8,\ \mathrm{d}_3=0.$ x_3=3.

$f_3(2)=\max\{11\times 0+f_2(2)=0+4=4,\ \mathrm{d}_3=0\ \}.$

Result $f_3(2)=4,\ \mathrm{d}^*{}_3=0.$ x_3=2.

$f_3(1)=\max\{11\times 0+f_2(1)=0+2=2,\ \mathrm{d}_3=0\ \}.$

Result $f_3(1)=2,\ \mathrm{d}^*{}_3=0.$ x_3=1.

$f_3(0)=\max\{11\times 0+f_2(0)=0+0=0,\ \mathrm{d}_3=0\ \}.$

Result $f_3(0)=0,\ \mathrm{d}^*{}_3=0.$ x_3=0,

APPENDIX D

Computations for Stage 4, (filling with items 1, 2, 3 and 4): n=4, of Job data for the manufacturing operation (page 50)

$$f_4(10) = \max.\begin{cases} 20\times 0 + f_2(10) = 0 + 27 = 27,\ \mathrm{d}_4 = 0. \\ 20\times 1 + f_2(3) = 20 + 8 = 28,\ \mathrm{d}_4 = 1. \end{cases} \qquad \mathrm{x}_4\text{=}10$$

Result $f_4(10) = 28,\ \mathrm{d}^*{}_4 = 1.$

$$f_4(9) = \max.\begin{cases} 20\times 0 + f_2(9) = 0 + 24 = 24,\ \mathrm{d}_4 = 0. \\ 20\times 1 + f_2(2) = 20 + 4 = 24,\ \mathrm{d}_4 = 1. \end{cases} \qquad \mathrm{x}_4\text{=}9$$

Result $f_4(9) = 24,\ \mathrm{d}^*{}_4\text{=}0,\ \mathrm{d}^*{}_4 = 1.$

$$f_4(8) = \max.\begin{cases} 20\times 0 + f_3(8) = 0 + 21 = 21,\ \mathrm{d}_4 = 0. \\ 20\times 1 + f_3(1) = 20 + 2 = 22,\ \mathrm{d}_4 = 1. \end{cases} \qquad \mathrm{x}_4\text{=}8$$

Result $f_4(8) = 22,\ \mathrm{d}^*{}_4 = 1.$

$$f_4(7) = \max.\begin{cases} 20\times 0 + f_3(7) = 0 + 19 = 19,\ \mathrm{d}_4 = 0. \\ 20\times 1 + f_3(0) = 20 + 0 = 20,\ \mathrm{d}_4 = 1. \end{cases} \qquad \mathrm{x}_4\text{=}7$$

Result $f_4(7) = 20,\ \mathrm{d}^*{}_4 = 1.$