KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

INSTITUTE OF DISTANCE LEARNING

MSc. INDUSTRIAL MATHEMATICS PROGRAMME

OPTIMAL PORTFOLIO SELECTION – A CASE STUDY OF SIX FINANCIAL INSTITUTIONS IN GHANA

BY

FELICIA ASARE KYEI

A THESIS SUBMITTED TO THE INSTITUTE OF DISTANCE LEARNING, KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE MASTER OF SCIENCE IN INDUSTRIAL MATHEMATICS DEGREE

MAY, 2013

DECLARATION

I hereby certify that this thesis submitted to the Institute of Distance Learning, Kwame Nkrumah University of Science and Technology is my own work towards the MSc. degree and that, except for references to other researchers' work which have duly been acknowledged, this thesis has not been submitted to any other university for the award of a degree.

FELICIA ASARE KYEI	....	3/05/13
PG 6318011	Signature	Date
Certified by:		
Prof. Samuel Kwame Amponsah	....	3/05/13
Supervisor	Signature	Date
Certified by:		
Mr. F.K. Darkwa	....	3/05/13
Head of Department	Signature	Date
Certified by:		
....	....	....
Dean (IDL)	Signature	Date

DEDICATION

I wholeheartedly dedicate this work to God for his protection; my family (my husband – Kwadwo, and children, Jennifer, Claudia, Michael, Annie-Pamela and Valerie).

ACKNOWLEDGEMENT

This piece of work of such magnitude could not have been done single handedly. A lot of inspirations were drawn from many people and many sources. The most thanks goes to the Almighty God for his guidance and protection throughout this course and also giving me hope those hard times. Secondly, I am highly indebted to Prof. Samuel Kwame Amponsah my supervisor who spent his relentless time reading through the draft and making very useful suggestions. Again I wish to acknowledge my husband for his enormous support, guidance and direction throughout my research period. Last but not least I thank all loved ones who in diverse ways help in the success of this project.

ABSTRACT

Many investors and portfolio managers always seek maximum returns with relative low risk or conversely, minimum risk with maximum expected returns. Which model or approach best meets investor's investment decisions and portfolio selection. The dynamic programming model amongst other things seeks to address such dilemma faced by investors. Suppose an investor wants to invest and there are several opportunities available to him/her then there arise a problem of choice/allocation. It would be realized that each opportunity require deposits in financial terms and an expected return. The investor may allocate all the money to just one opportunity or split the money between the alternatives of investments all with the aim of obtaining the maximum returns from the investment made. The problem of determining how much to invest in each investment in order to maximize total returns is the major challenge faced by investors and this can be achieved through a multi-stage decision process by modelling a Mathematical program to find the optimal policy using Dynamic Programming. The general objective of this study is to identify the various types of investments and their returns and use dynamic programming to find the optimal investment portolio. Secondary data will be used in the study. Interest rates and yield functions of various investments will be collected from both Banking and Financial Institutions. The data collected will be collated and analyzed using dynamic programming. The results of the research clearly prove that dynamic programming as very efficient in allocating resources for the optimal investment returns from a portfolio. The study concluded that dynamic programming can be used in allocating resources for the optimal investment returns from a portfolio. It is recommended that investors should not invest too much money in a single investment. One should always divide the resources available in bits to invest in different investments.

CONTENTS

LIST OF TABLES

CHAPTER ONE

1.0 Introduction

Many investors and portfolio managers always seek maximum returns with relative low risk or conversely, minimum risk with maximum expected returns. Which model or approach best meets investor's investment decisions and portfolio selection. The dynamic programming model amongst other models seeks to address such dilemma faced by investors.

1.1 Background of Study

People throughout ages have sought to look for avenues to increase their wealth. The best way to maintain and improve one's wealth is to invest the little that one has. Individuals, entities, organizations and cooperate bodies make investments towards the future. One difficulty faced by all is where, how, when and for how long to invest, in the midst of all opportunities, to obtain the maximum satisfaction from the investment made. Usually investors might have a certain amount of money to invest. The investor may have various options of investment based on the returns.

Any time one makes an investment one must decide on the optimal investment strategy. Two very important strategies are active portfolio management and long term investment. The strategies that one chooses for an optimal strategy will depend on one's investment goals.

Investors invest in mutual funds for numerous reasons. The investment decision may be driven by the need for diversification at a low cost or for the desire of professional portfolio management.

Active portfolio management, in a substantial amount of cases, fails to cover the higher costs it imposes on investors. Some studies, which focus on outlier performance, provide empirical evidence that the top decile of investment management performers display a skill in generating alpha (Kosowski, 2006). Parallel to this strand of literature is the question of whether active portfolio management is more likely to provide alpha in recessionary periods.

1.1.1 Investment

Investment has been defined by many and in many fields. Investing is laying out money or capital in an enterprise with the expectation of profit.

Money is invested with an expectation of profit. Investment is the commitment of something other than money (time, energy, or effort) to a project with the expectation of some worthwhile result. Investment is the commitment of money or capital to purchase financial instruments or other assets in order to gain profitable returns in a form of interest, income, or appreciation of the value of the instrument. It is related to saving or deferring consumption.

Farlex (2012) stated that investment is the act of placing capital into a project or business with the intent of making a profit on the initial placing of the capital. An investment may involve the extension of a loan or line of credit, which entitles one to repayment with interest, or it may involve buying an ownership stake in a business, with the hope that the business will become profitable. Investing may also involve buying a particular asset with the intent to resell it later for a higher price. Many types of investing exist, and each is subject to greater or lesser regulation in the jurisdiction in which it takes place. Legally, investing requires the existence and protection of individual property rights. Investing wisely requires a combination of astuteness, knowledge of the market, and timing.

The concise encyclopedia defines investment as the process of exchanging income for an asset that is expected to produce earnings at a later time. An investor refrains from consumption in the present in hopes of a greater return in the future. Investment may be influenced by rates of interest, with the rate of investment rising as interest rates fall, but other factors more difficult to measure may also be important for example, the business community's expectations about future demand and profit, technical changes in production methods, and expected relative costs of labour and capital. Investment cannot occur without saving, which provides funding. Because investment increases an economy's capacity to produce, it is a factor contributing to economic growth.

An investment involves the choice by an individual or an organization such as a pension fund, after some analysis or thought, to place or lend money in a vehicle, instrument or asset, such as property, commodity, stock, bond, financial derivatives (e.g. futures or options), or the foreign asset denominated in foreign currency, that has certain level of risk and provides the possibility of generating returns over a period of time. When an asset is bought or a given amount of money is invested in the bank, there is anticipation that some return will be received from the investment in the future.

Investment is a term frequently used in the fields of economics, business management and finance. It can mean savings alone, or savings made through delayed consumption. Investment can be divided into different types according to various theories and principles.

According to economic theories, investment is defined as the per-unit production of goods, which have not been consumed, but will however, be used for the purpose of future production. Examples of this type of investments are tangible goods like construction of a factory or bridge and intangible goods like six (6) months of on-the-job training. In terms of national production and income, Gross Domestic Product (GDP) has an essential constituent, known as gross investment.

In an economic sense, an investment is the purchase of goods that are not consumed today but are used in the future to create wealth. In finance, an investment is a monetary asset purchased with the idea that the asset will provide income in the future or appreciate and be sold at a higher price. The building of a factory used to produce goods and the investment one makes by going to college or university are both examples of investments in the economic sense

According to business management theories, investment refers to tangible assets like machinery and equipment and buildings and intangible assets like copyrights or patents and goodwill. The decision for investment is also known as capital budgeting decision, which is regarded as one of the key decision of management.

In finance, investment refers to the purchasing of securities or other financial assets from the capital market. It also means buying money market or real properties with high market liquidity. Some examples are gold, silver, real properties, and precious items. Financial investments are in stocks, bonds, and other types of security investments. Indirect financial investments can also be done with the help of mediators or third parties, such as pension funds, mutual funds, commercial banks, and insurance companies.

According to personal finance theories, an investment is the implementation of money for buying shares, mutual funds.

Usually a combination of any of the above investment possibilities may be considered by an investor or an individual. Any time an investor wishes to make an investment with a certain sum of money, say, he must decide on the optimal investment strategy to adopt. The optimal investment strategy could be long term or short term, active portfolio management or long term investing. The choice of a particular investment should be based on the cumulative returns on all the investments. In the financial sense investments include the purchase of bonds, stocks or real estate property. Investing usually involves the creation of wealth whereas speculating is often a zero-sum game; wealth is not created. Although speculators are often making informed decisions, speculation cannot usually be categorized as traditional investing.

1.2 Statement of the Problem

This is what the Good Book says about investment in Ecclesiastes 11:2

"But divide your investment among many places for you do not know what risks might lie ahead".

Suppose an investor wants to invest and there are several opportunities available to him/her then there arise a problem of choice/allocation. It would be realized that each opportunity require deposits in financial terms and an expected return. The investor may allocate all the money to just one opportunity or split the money between the alternatives of investments all with the aim of obtaining the maximum returns from the investment made.

The problem of determining how much to invest in each investment in order to maximize total returns is the major challenge faced by investors and this can be achieved through a multi-stage decision process by modeling a Mathematical program to find the optimal policy using Dynamic Programming.

1.3 Objectives of the Study

The objectives of this study are to:

(i) identify the various types of investments and their returns and

(ii) use dynamic programming to find the optimal investment portolio.

1.4 Methodology Used in the Study

The Mathematical methods that will be used in the research is Dynamic Programming. Secondary data will be used in the study. Interest rates and yield functions of various investments will be collected from both Banking and Financial Institutions. Dynamic Programming will be used to determine the optimal investment and the appropriate investment allocations to be made to each category of investment.

1.5 Justification of the Study

Many are those who have resources and would like to invest, but are not certain of where, when and how to put their resources in order to accrue the maximum returns. To justify the products in which to invest, we need to look out for the various forms of investments available, the expected returns from each investment and the associated cost. Financial institutions would like to know where to keep their excess cash flows to make the maximum returns. All the above can be modeled as dynamic programming problem. It is known that dynamic programming solves problems in stages and is quicker and less time consuming far less than total enumeration.

1.6 Scope of the Study

The study will be carried out exclusively in Ghana. The scope of the study will be limited to six financial institutions namely; the Government of Ghana's Treasury Bills, Barclays Bank Ghana, Ghana Commercial Bank, Data Bank, Guinness Ghana Limited and Fan Milk Limited, Accra.

1.7 Limitations of the Study

The problem to be considered in this study is the Bellman's Principle of Optimality using Dynamic programming. We considered only six investments with five returns, this is due to time constraints which did not allow the researcher to do total enumeration.

1.8 Organization of the Study

The study is organized into five chapters. Chapter One is the Introduction of Investment in general, background, problem statement, objective, methodology and justification of the study.

In Chapter Two, we shall put forward pertinent literature on Dynamic programming and it's variants.

Chapter Three presents the research methodology of the study, Uncertain Pay-Offs, Equipment Replacement problems using total enumeration and Dynamic Programming. It considers cases where there is total enumeration and compares the time and stages used in solving a problem.

Chapter Four deals with the collection, analysis of the data and interpretation of the results..

Chapter Five, the last chapter of the study presents the conclusion and recommendations of the study.

1.9 Summary

This chapter discussed the background, problem statement, objectives and methodology of the study. The justification, scope and limitations of the study were also put forward. The next chapter presents relevant literature on investment and dynamic programming.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

In this chapter, we shall review existing literature on dynamic programming.

2.1 Review of Existing Literature

Dynamic programming is a method for solving complex problems by breaking them down into simpler sub problems. It is applicable to problems exhibiting the properties of overlapping sub problems and optimal substructure. When applicable, the method takes far less time than naive methods. The key idea behind dynamic programming is quite simple. In general, to solve a given problem, we need to solve different parts of the problem (sub problems), then combine the solutions of the sub problems to reach an overall solution.

Dynamic programming is both a mathematical optimization method, and a computer programming method. In both contexts, it refers to simplifying a complicated problem by breaking it down into simpler sub problems in a recursive manner. While some decision problems cannot be taken apart this way, decisions that span several points in time do often break apart recursively; Bellman called this the "Principle of Optimality". Likewise, in computer science, a problem which can be broken down recursively is said to have optimal substructure.

If sub problems can be nested recursively inside larger problems, so that dynamic programming methods are applicable, then there is a relation between the value of the larger problem and the values of the sub problems. In the optimization literature this relationship is called the Bellman equation.

Dynamic programming is a widely used programming technique in bioinformatics. In sharp contrast to the simplicity of textbook examples, implementing a dynamic programming algorithm for a novel and non-trivial application is a tedious and error prone task. The algebraic dynamic programming approach seeks to alleviate this situation by clearly separating the dynamic programming recurrences and scoring schemes.

Hiller and Lieberman (2003) stated that Dynamic programming is a very useful technique for making a sequence of interrelated decisions. It requires formulating an appropriate recursive relationship for each individual problem. They further state that Dynamic programming provides great computational savings over exhaustive enumeration to find the best combination of decisions, especially for large problems.

Proper investment decision making is key to success for every investor in their efforts to keep pace with the competitive business environment. Mitigation of exposure to risk plays a vital role, since investors are now directly exposed to the uncertain decision environment. The uncertainty (and risk) of an investment is increasing with the increased number of competing investors entering to market. As a result, the expected return on investment (ROI) of a decision quite often carries a high degree of uncertainty.

Dynamic programming is a systematic tool based on the simple idea of the principle of optimality (Bertsekas, 2007). If a decision problem can be viewed as multiple stages with multiple states and known state transitions associated with each particular action, then the problem can be systematically solve with deterministic dynamic programming with the help of prominent Bellman equation (Bertsekas,2007).If the state transition is probabilistic, then we can apply stochastic dynamic programming. Apart from this, if the number of stages is infinite (i.e., if we do not want to impose a limit on the number of stages), then the problem becomes an Infinite Horizon Stochastic Dynamic Programming Problem (IHSDP) (Bertsekas, 2007).

At present, investment decision making is a critical task because every investment exhibits at least some amount of risk and uncertainty. These risks and uncertainties are the results of huge business competition and vibrant market economy. As a result, recent research in investment decision making is undergoing a paradigm shift with much integration of new techniques with existing methods to develop robust decision making processes (Heikkinen *et al*., 2009).

Xu-song and Jian-mou (2002) studied the investment decision-making of a project with deterministic dynamic programming. Yan and Bai (2009) formulated a deterministic dynamic programming model to allocate funds between stocks in a portfolio to maximize income. The authors captured the risk issues by incorporating the positive correlation between risks and returns of a stock to a large extent.

Heikkinen and Pietola (2009) studied the use of stochastic programming approaches to make optimal investment decisions by modeling the problem as a Markov decision process. A dynamic uncertainty cost is presented with the modification of the classical expected value of perfect information to a dynamic setting.

Dixit and Pindyck (1994) described the use of a Markov Decision Process (MDP) defined in continuous time and with a continuous state spacefor optimal investment decisions.

Most of the prior research does not consider the inter-related dynamics of the systems that can be encountered by a stochastic dynamic investment model (Botterud *et al*., 2007))

2.2 Investment Portfolio

Given the risk-return trade-off and the risk-reducing effect of diversification, the major component of portfolio performance is investment policy – allocation of the portfolio across different asset classes with different amounts of risk (Brinson et al., 1986). A key study of private pension plans indicates that investment policy accounted for up to 94 per cent of the variation of plan return (Brinson et al., 1986). Other studies also indicate that investment policy is a determinant of investment return for public and/or private plans (Ambachtsheer, 1994; Engebretson, 1995).

2.2.1 The Investment Process

According to Myles (2003), the investment process involves the following steps:

i) Determine Objectives: Investment policy has to be guided by a set of objectives. Before investment can be undertaken, a clear idea of the purpose of the investment must be obtained. The purpose will vary between investors. Some may be concerned only with preserving their current wealth. Others may see investment as a means of enhancing wealth. What primarily drives objectives is the attitude towards taking on risk. Some investors may wish to eliminate risk as much as is possible, while others may be focused almost entirely on return and be willing to accept significant risks.

ii) Choose Value: The second decision concerns the amount to be invested. This decision can be considered a separate one or it can be subsumed in the allocation decision between assets (what is not invested must either be held in some other form which, by definition, is an investment in its own right or else it must be consumed).

iii) Conduct Security Analysis: Security analysis is the study of the returns and risks of securities. This is undertaken to determine in which classes of assets investments will be placed and to determine which particular securities should be purchased within a class. Many investors find it simpler to remain with the more basic assets such as stocks and fixed income securities rather than venture into complex instruments such as derivatives. Once the class of assets has been determined, the next step is to analyze the chosen set of securities to identify relevant characteristics of the assets such as their expected returns and risks. This information will be required for any informed attempt at portfolio construction. Another reason for analyzing securities is to attempt to find those that are currently mispriced. For example, a security that is under-priced for the returns it seems to offer is an attractive asset to purchase. Similarly, one that is overpriced should be sold. Whether there are any assets are underpriced depends on the degree of efficiency of the market. More is said on this issue later.

Such analysis can be undertaken using five alternative approaches:

a) Technical analysis: This is the examination of past prices for predictable trends. Technical analysis employs a variety of methods in an attempt to find patterns of price behavior that repeat through time. If there is such repetition (and this is a disputed issue), then the most beneficial times to buy or sell can be identified.

b) Fundamental analysis: The basis of fundamental analysis is that the true value of a security has to be based on the future returns it will yield. The analysis allows for temporary movements away from this relationship but requires it to hold in the longrum. Fundamental analysts study the details of company activities to makes predictions of future profitability since this determines dividends and hence returns.

c) Portfolio Construction: Portfolio construction follows from security analysis. It is the determination of the precise quantity to purchase of each of the chosen securities. A factor that is important to consider is the extent of diversification. Diversifying a portfolio across many assets may reduce risk but it involves increased transactions costs and increases the effort required to manage the portfolio.

d) Evaluation. Portfolio evaluation involves the assessment of the performance of the chosen portfolio. To do this it is necessary to have some yardstick for comparison since a meaningful comparison is only achieved by comparing the return on the portfolio with that on other portfolios with similar risk characteristics.

e) Revision. Portfolio revision involves the application of all the previous steps. Objectives may change, as may the level of funds available for investment. Further analysis of assets may alter the assessment of risks and returns and new assets may become available. Portfolio revision is therefore the continuing reapplication of the steps in the investment process.

2.3 Dynamic Programming

Based on the programming style, Steffen et al, (2005) introduced a generic product operation of scoring schemes. This lead to a remarkable variety of applications, allowing us to achieve optimizations under multiple objective functions, alternative solutions and back tracing, holistic search space analysis, ambiguity checking, and more, without additional programming effort. The authors demonstrated the method on several applications for

RNA secondary structure prediction. The product operation as introduced here adds a significant amount of flexibility to dynamic programming. It provides a versatile text bed for the development of new algorithmic ideas which can immediately be put to practice. Institutional fund managers generally rebalance using ad hoc methods such as calendar basis or tolerance band triggers.

Sun *et al*., (2005) proposed a different framework that quantifies the cost of a rebalancing strategy in terms of risk-adjusted returns net of transaction costs. The authors then developed an optimal rebalancing strategy that actively seeks to minimize that cost. They used certainty equivalents and the transaction costs associated with a policy to define a cost-to-go function, and they minimized this expected cost-to-go using dynamic programming. The authors applied Monte Carlo simulations to demonstrate that their method outperforms traditional rebalancing. They also showed the robustness of our method to model error by performing sensitivity analyses.

The existence of an optimum and dynamic programming techniques was derived from abstract assumptions based on primitive utility function U and its W and M primitive aggregators. A non-positive-valued utility function U that is derived from a W dynamic aggregator and an M stochastic aggregator was constructed. The resulting examples exhibit mean growth without the distribution of unbounded support due to the few growth restrictions of non-positive objective.

Differential dynamic programming is a technique, based on dynamic programming rather than the calculus of variations, for determining the optimal control function of a nonlinear system. Unlike conventional dynamic programming where the optimal cost function is considered globally, differential dynamic programming applies the principle of optimality in the neighborhood of a nominal, possibly no optimal, trajectory. This allowed the coefficients of a linear or quadratic expansion of the cost function to be computed in reverse time along the trajectory: these coefficients may then to be used to yield a new improved trajectory (i.e. the algorithms are of the "successive sweep" type). A class of nonlinear control problems, linear in the control variables, is studied using differential dynamic programming. It is shown that for the free-end-point problem, the first partial derivatives of the optimal cost function are continuous throughout the state space, and the second partial derivatives experience jumps at switch points of the control function. A control problem that has an analytic solution is used to illustrate these points. The fixed-end-point problem is converted into an equivalent free-end-point problem by adjoining the end-point constraints to the cost functional using Lagrange multipliers: a useful interpretation for Pontryagin's adjoin variables for this type of problem emerges from this treatment. The above results are used to devise new second- and first-order algorithms for determining the optimal bang-bang control by successively improving a nominal guessed control function. The usefulness of the proposed algorithms is illustrated by the computation of a number of control problem examples (Jacobson, 2003).

Dynamic programming solutions for optimal portfolios in which the solution for the portfolio vector of risky assets is constant were solved by Merton in continuous time and by Hakansson and others in discrete time. There is no case with a closed form solution where this vector of risky asset holdings changes dynamically. Tenney (1995) derived such solutions for the first time, and is thus a dynamic dynamic-programming solution as opposed to a static dynamic-programming solution for this vector. The solution is valid when there is a set of basis assets whose excess expected return is linear in the state vector, whose variance-covariance matrix is time-dependent and for which the interest rate is a quadratic function of the state vector.

Battocchio (2002) considered a stochastic model for a defined-contribution pension fund in continuous time. In particular, they, focused on the portfolio problem of a fund manager who wants to maximize the expected utility of his terminal wealth in a complete financial market with stochastic interest rate. The fund manager must cope with a set of stochastic investment opportunities and two background risks: the salary risk and the inflation risk. We used the stochastic dynamic programming approach. He showed that the presence of the inflation risk can solve some problems linked to the use of the stochastic dynamic programming technique, and namely to the stochastic partial differential equation deriving from it. The author found a closed form solution to the asset allocation problem, without specifying any functional form for the coefficients of the diffusion processes involved in the problem. Finally, the derivation of a closed form solution allows us to analyse in detail the behavior of the optimal portfolio with respect to salary and inflation.

Guangliang *et al.*, (1999) solved the problem of constructing an optimal portfolio consisting of many risky assets to maximize the long-term growth rate of a representative agent's expected utility, subject to a set of general linear constraints on the portfolio weight vector as well as a constraint to prevent wealth drawdown's below a dynamic floor.

The dynamic floor is defined as the time-decayed historical all-time high. Our results generalize those achieved by earlier authors, including Grossman and Zhou (1993) and

Cvitannic and Karatzas (1994). The authors solved a special case of the authors' problem by focusing on single risky assets without portfolio weight constraints.

Cvitannic and Karatzas (1994) solved a problem involving many risky assets but that ignored portfolio weight constraints and the time decay on the dynamic floor. To illustrate the usefulness of our method, the authors presented several numerical examples based on both actual and simulated (Monte Carlo) returns. Finally, the authors suggested applications of their results to various practical investment management problems, including the management of hedge fund portfolios and 'principal-protected' investment strategies.

Herman *et al.*, (2009) developed a multi-period investment portfolio model that includes risky farmland, risky and risk-free nonfarm assets, and debt financing on farmland in the presence of transaction costs and credit constraints. The model is formulated as a stochastic continuous-state dynamic programming problem, and is solved numerically for South-western Minnesota, USA. Result show that optimal investment decisions are dynamic and take into account the future decisions due to uncertainty, partial irreversibility, and the option to wait. The optimal policy includes ranges of inaction, states where the optimal policy in the current year is to wait. The risk-averse farmer makes a lower investment in risky farmland reflecting risk-avoiding behaviour. The authors found that, in addition to risk aversion, the length of the planning horizon affects risk-avoiding behavior in investment decisions. The authors found that higher debt financing on farmland is optimal when risky nonfarm assets can be included in the optimal investment portfolio and that the probability of exiting farming increases with the risky nonfarm investment.

Ghezzi (1997) considered an immunization problem, in which a bond portfolio is to be periodically rebalanced. Max-min optimal control is applied to the problem. The target is to maximize the final portfolio value under the worst possible evolution of interest rates. The optimal control law, obtained by means of dynamic programming, turns out to be different from any duration-based immunization policy.

Vila *et al*., (1991) used stochastic dynamic programming to study the inter-temporal consumption and portfolio choice of an infinitely lived agent who faces a constant opportunity set and a borrowing constraint. The authors showed that, under general assumptions on the agent's utility function, optimal policies exist and can be expressed as feedback functions of current wealth. They described these policies in detail, when the agent's utility function exhibits constant relative risk aversion.

Optimal asset allocation deals with how to divide the investor's wealth across some asset-classes in order to maximize the investor's gain. Pola *et al*., (2006) put forward the optimal asset allocation in a multi-period investment setting: optimal dynamic asset allocation provides the (optimal) re-balancing policy to accomplish some investment's criteria. Given a sequence of target sets, which represent the portfolio specifications at each re-balancing time, an optimal portfolio allocation is synthesized for maximizing the joint probability for the portfolio to fulfill the target sets requirements. The approach pursued is based on dynamic programming. The optimal solution is shown to conditionally depend on the portfolio realization, thus providing a practical scheme for the dynamic portfolio rebalancing. Finally some case studies are given to show the proposed methodology.

Rudoy *et al*., (2008) studied the problem of optimal portfolio construction when the trading horizon consists of two consecutive decision intervals and rebalancing is permitted. It is assumed that the log-prices of the underlying assets are non-stationary, and specifically follow a discrete-time co integrated vector autoregressive model. The authors extended the classical Markowitz mean-variance optimization approach to a multi-period setting, in which the new objective is to maximize the total expected return, subject to a constraint on the total allowable risk. In contrast to traditional approaches, they adopted a definition for risk which takes into account the non-zero correlations between the inter-stage returns. This portfolio optimization problem amounts to not only determining the relative proportions of the assets to hold during each stage, but also requires one to determine the degree of portfolio leverage to assume. Due to a fixed constraint on the standard deviation of the total return, the leverage decision is equivalent to deciding how to optimally partition the allowed variance, and thus variance can be viewed as a shared resource between the stages. The authors derived the optimal portfolio weights and variance scheduling scheme for a trading strategy based on a dynamic programming approach, which is utilized in order to make the problem computationally tractable. The performance of this method is compared to other trading strategies using both Monte Carlo simulations and real data, and promising results are obtained.

Ye (2007) modeled a continuous-time optimal life insurance, consumption and portfolio is examined by dynamic programming technique. The Hamilton-Jacobi- Bellman (HJB in short) equation with the absorbing boundary condition is derived. Then explicit solutions for Constant Relative Risk Aversion (CRRA) utilities with subsistence levels are obtained. Asymptotic analysis was used to analyze the model.

Papi *et al*., (2006) devoted a paper to the analysis of a discrete-time dynamic programming algorithm for the numerical solution of an optimal asset-liability management model with transaction costs and in presence of constraints. By exploiting the financial properties of the model, the authors proposed an approximation method based on the classical dynamic programming algorithm, which reduces significantly the computational and storage requirements of the algorithm and avoids any artificial boundary condition. The regularity of the value function is used to estimate the global error introduced by the numerical procedure and to prove a convergence result.

Dijkhuizen *et al*., (1993) used a personal computer-based stochastic Dynamic Programming (DP) model for the determination of the optimal replacement policy in swine breeding is evaluated. The model provides the maximal expected annual net returns of current herd sows and subsequent replacements over time. The DP-based system was seen to be viable in modeling such factors as biological variations, but are limited by hardware requirements. Result accuracy is effected by the number of DP runs achieved.

Lubbecke *et al*., (2005) used column generation method to solve linear programs with a large number of variables. Dynamic program algorithms are used for column generation and a simple technique is used to reduce the state space of these algorithms.

Dynamic Programming has been applied to a number of digital signal processing problems. Rader *et al*., (2003) discussed a well known application of determining the optimum order of sections in a digital filter realization. The authors showed that the method is quite insensitive to the specific details of the problem; it is applicable over a wide range of possible optimality criteria, various kinds of arithmetic, scaling options, etc. This is characteristic of the application of dynamic programming to many signal processing problems. Also, since a problem, to be solved by dynamic programming, must be represented as the traversal of a directed graph, we usually discover unsuspected structure in the problem when we attempt to solve it using dynamic programming. Quite often it is necessary to recognize this structure in order to solve the problem efficiently. In the case of ordering of filter section the structure leads to an efficient utilization of memory.

A system and method are disclosed for capturing the full dynamic and multi-dimensional nature of the asset allocation problem through applications of stochastic dynamic programming and stochastic programming techniques. The system and method provide a novel approach to asset allocation and based on stochastic dynamic programming and Monte Carlo sampling that permit one to consider many rebalancing periods, many asset classes, dynamic cash flows, and a general representation of investor risk preference. The system and method further provide a novel approach of representing utility by directly modeling risk aversion as a function of wealth, and thus provide a general framework for representing investor preference. The system and method demonstrate how the optimal asset allocation depends on the investment horizon, wealth, and the investors risk preference and how optimal asset allocation therefore changes over time depending on cash flow and the returns achieved and how dynamic asset allocation leads to superior results compared to static or myopic techniques. Examples of dynamic strategies for various typical risk preferences and multiple asset classes are described.

The dramatic growth in institutionally managed assets, coupled with the advent of Internet trading and electronic brokerage for retail investors, has led to a surge in the size and volume of trading. At the same time, competition in the asset-management industry has increased to where fractions of a percent in performance can separate the top funds from those in the next tier. In this environment, portfolio managers have begun to explore active management of trading to boost returns. Controlling execution cost can be viewed as a stochastic dynamic optimization problem because trading takes time, stock prices exhibit random fluctuations, and execution prices depend on trade size, order flow, and market conditions. In this article, the authors apply stochastic dynamic programming to derive trading strategies that minimize the expected cost of executing a portfolio of securities over a fixed time period. That is, they derive the optimal sequence of trade as a function of prices, quantities, and other market conditions. To illustrate the practical relevance of these methods, Bertsimas et al., (1999) applied them to a hypothetical portfolio of 25 stocks. The authors estimated the methods' price-impact functions using 1996 trade data and derive the optimal execution strategies. The authors also perform several Monte Carlo simulations to compare the optimal strategy's performance to that of several alternatives.

Battocchio *et al*., (2002) considered a stochastic model for a defined-contribution pension fund in continuous time. In particular, we focus on the portfolio problem of a fund manager who wants to maximize the expected utility of his terminal wealth in a complete financial market with stochastic interest rate. The fund manager must cope with a setoff stochastic investment opportunities and two background risks: the salary risk and the inflation risk. We use the stochastic dynamic programming approach. We show that the presence of the inflation risk can solve some problems linked to the use of the stochastic dynamic programming technique, and namely to the stochastic partial differential equation deriving from it. The technique, and namely to the stochastic partial differential equation deriving from it.The authors a closed form solution to the asset allocation problem, without specifying any functional form for the coefficients of the diffusion processes involved in the problem. Finally, the derivation of a closed form solution allows us to analyze in detail the behavior of the optimal portfolio with respect to salary and inflation. The authors also solved the problem of constructing an optimal consisting of many risky assets to maximize the long-term growth rate of a representative agent's expected utility, subject to a set of general linear constraints on the portfolio weight vector as well as a constraint to prevent wealth drawdown below a dynamic floor. The dynamic floor is defined as the time-decayed historical all-time high. Our results generalize those achieved by earlier authors, including Grossman and Zhou (1993) and Cvitannic and Karatzas (1994).

Grossman and Zhou (1993) solved a special case of our problem by focusing on a single risky asset without portfolio weight constraints. Karatzas (1994) solved a problem involving many risky assets but that ignored portfolio weight constraints and the time decay on the dynamic floor. To illustrate the usefulness of our method, we present several numerical examples based on both actual and simulated (Monte Carlo) returns. The authors suggested applications of our results to various practical investment management problems, including the management of hedge fund portfolios and 'principal-protected' investment strategies.

Bouzaher *et al*., (1990) used dynamic programming algorithm design to analyze soil movement, to ensure water quality and reduce the costs of water treatment by facilitating the control of agricultural sediment pollution in surface waters. The algorithm models analyze the spatial characteristics of soil movement though a watershed and the impact of soil movement on reservoirs and water channels. The model solves this class of pollution control problems by generating sediment abatement cost frontiers. This information is valuable to watershed management and planning because it devises control strategies to reduce sediment deposition in water courses and can be used to identify special problem areas.

Greco (1990) put forward that dynamic programming is a general technique for solving optimization problems. It is based on the division of problems into simpler sub problems that can be computed separately. In this paper, they showed that Datalog with aggregates and other non-monotonic constructs can express classical dynamic programming optimization problems in a natural fashion, and then we discuss the important classes of queries and applications that benefit from these techniques.

The second interest of DP approach, and the main avenue for future research, is that it allows RM to incorporate consumer choice within the optimization process. El Haber and El Taha, 2004 formulated a dynamic programming model to solve the seat inventory control problem for a two-leg airline with realistic elements of consumer behaviour. Ahead of the Origin and Destination formulation, they consider cancellation, no shows and overbooking.

Ryzin's work, Van Ryzin and Vulcano(2011)considered a revenue management, network capacity control problem in a setting where heterogeneous customers choose among the various products offered by a firm (for example, different fight times, fare classes and/or routings). Customers may therefore substitute if their preferred products are not offered, even buy up. Their choice model is very general, simply specifying the probability of purchase for each fare product as a function of the set of fare products offered. Overall, the value of this paper is to facilitate the understanding of more complex, and probably more realistic, models of revenue management.

Dynamic programming addresses how to make optimal decisions over time under uncertain conditions and to control a system. Most RM situations can be analyzed assuming a discrete-state and a discrete time over a finite-horizon modeling.

Four criteria of research analyzed:

i) A paper could consider a single product (at various prices) or multiple products (depending on purchase restrictions or independent demands for example);

ii) A paper could consider a static policy (assuming a strict order of booking arrivals) or allow for a dynamic policy (not assuming the early birds hypothesis);

iii) A paper could consider various forms of demand process;

iv) A paper could consider either a single resource for 1 to *n* products or multiple resources (such as an airline network of hubs and spokes).

2.4 Summary

In this chapter, relevant literature on investment and dynamic programming were put forward. In the next chapter, we shall discuss the research methodology of the study.

CHAPTER TRHEE

METHODOLOGY

3.0 Introduction

This chapter discusses the research methodology and design strategy to be adopted for this study. The chapter will address data collection instruments, methods, research technique, the population, sampling procedure, sampling type, sampling technique and data analysis to be used for this study. It will provide detailed explanations to each of the methods employed and how the methods adopted were used to address the aims and objectives of the study.

3.1 Data Collection Methods

Secondary data were used for the study. They are already compiled data for statistical analysis. They are not collected especially for the investigation under consideration but have been collected for some other purpose(s). Secondary data are cheaper and easier to obtain.

Extraction from Administrative Records: This method is solely used to collect secondary data from published sources such as administrative files, libraries, print/electronic media, internet etc. Information on interest rates in Ghana could be obtained from Government of Ghana and the Ghana Stock Exchange. selected for the study. In our study secondary data was obtained from some ~~selected~~ financial institutions.

3.2 Dynamic Programming

Dynamic Programming is a technique that can be used to solve many optimization problems. In most applications, dynamic programming obtains solutions by working backward from the end of a problem toward the beginning, thus breaking up a large, unwieldy problem into a series of smaller, more tractable problems.

In mathematics and computer science, dynamic programming is a method for solving complex problems by breaking them down into simpler sub problems. It is applicable to problems exhibiting the properties of overlapping sub problems which are only slightly smaller and optimal substructure (described below). When applicable, the method takes far less time that naïve methods.

The key idea behind dynamic programming is quite simple. In general, to solve a given problem, we need to solve different parts of the problem (sub problems), then combine the solution of the sub problems to reach an overall solution. Often, many of these sub problems are really the same. The dynamic programming approach seeks to solve each sub problem only once, thus saving a lot of computation. This is especially useful when the number of repeating sub problems is exponentially large.

Top-down dynamic programming simply means storing the results of certain calculations, which are later used again since the completed calculation is a sub-problem of a larger calculation. Bottom-up dynamic programming involves formulating a complex calculation as a recursive series of simpler calculations.

The term dynamic programming was originally was originally used in the 1940s by Richard Bellman to describe the process of solving problems where one needs to find the best decisions one after another. By 1953, he refined this to the modern meaning, referring specifically to nesting smaller decision problems inside larger decisions, and the field was there after recognized by the IEEE as a systems analysis and engineering topic. Bellman's contribution is remembered in the name of a systems Bellman equation, a central result of dynamic programming which restates an optimization problem in recursive form.

The word dynamic was chosen by Bellman because it sounded impressive, not because it described how the method worked. The word programming referred to the use of the method to find an optimal program, in the sense of a military schedule for training or logistics. This usage is the same as that in the phrases linear programming and mathematical programming a synonym for optimization.

Finding the shortest path in a graph using optimal substructure; a straight line indicates a single edge; a wavy line indicates a shortest path between the two vertices it connects (other nodes on these paths are not shown); the bold line is the overall shortest path from start to goal.

Dynamic programming is both a mathematical optimization method and a computer programming method. In both contexts it refers to simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner. While some decision problems cannot be taken apart this way, decisions that span several points in time do often break apart recursively; Bellman called this the "Principle of Optimality". Likewise, in computer science, a problem which can be broken down recursively is said to have optimal substructure.

If sub-problems can be bested recursively inside larger problems, so that dynamic programming methods are applicable, then there is a relation between the value of the larger problem and the values of the sub-problems (Bellman, 2011) In the optimization literature this relationship is called the Bellman equation.

3.2.1 Principle of Optimality

Dynamic programming is an approach for optimizing multistage decision processes. It is based on Bellman's principle of optimality.

An optimal policy has the property that, regardless the decisions taken to enter a particular state in a particular stage, the remaining decisions must constitute an optimal policy for leaving that state. To implement this principle, begin with the last stage of an n-stage process and determine for each stage the best policy for leaving that state and completing the process, assuming that all proceeding stages have been completed. Then move backwards through the process, stage by stage. At each stage, determine the best policy for leaving each state and completing the process, assuming that all proceeding stages have been completed and making use of the results already obtained for the succeeding stage. In doing so, the entries of table one (1) will be calculated, where:

U =the state variable, whose values specify the states

Mj (u) = optimum return from completing the process beginning at stage J in state U

Dj(u) = decision taken at stage j that achieves mj (u)

Table 3.1*:* **A Multistage decision processes table**

μ

	0	**1**	**2**	**3**	
$m_n(u)$					*Stage n*
$d_n(u)$					
$m_{n-1}(u)$					*Stage n-1*
$d_{n-1}(u)$					
…………….....	………..	…			
$m_1(u)$					*Stage one*
$d_1(u)$					

The entries corresponding to the last stage of the process, m_n (u) and d_n (u), are generally straight forward to compare. The remaining entries are obtained recursively that is the entries for the Jth stage (j= 1, 2,…… , (n – 1)) are determined as functions of the entries for the (j+1) stage. The recursion formula is problem dependent, and must be obtained anew for each different type of multi stage process.

Note: the values of m_n (u) for U = 0,1,……,b are given by the formula

$$Mn(u) = \underset{0 \leq x < \mu}{\text{optimum}} [f_n(x)] \quad \ldots\ldots (1)$$

The recursion formula is $$mj(u) = \underset{0 \leq x \leq u}{\text{optimal}} [f_j(x)+m_{j+1}(u-x)] \quad \ldots\ldots (2)$$

For j = n-1, n-2,..,1

3.2.2 Policy Tables

For process in which randomness exits in the states associated with the decisions, a policy in particular , an optimal policy – may be exhibited as a policy table here, $d_j(a_h)$ (j=1,2,…..,n, h= 1, 2, ……r) denotes the decision at stage j if the process finds itself in state a_h.

Table 3.2 Optimal Policy table

		States			
		a_1	a_2	……………	a_r
	1	$d_1(a_1)$	$d_1(a_2)$	……………	$d_1(a_r)$
Stages	2	$d_2(a_1)$	$d_2(a_2)$	……………	$d_2(a_r)$
	……………	……………	……………	……………	
	N	$d_n(a_1)$	$d_n(a_2)$	……………	$d_n(a_r)$

3.3 Dynamic Programming in Mathematical Optimization

In terms of mathematical optimization, dynamic programming usually refers to simplifying a decision by breaking it down into a sequence of decision steps over time. This is done by defining sequence of value functions V_1, V_2,...V_n, with an argument y representing the state of the system at times I from 1 to n. the definition of $V_n(y)$ is the value obtained in state y at the last time n. they values V_i at earlier times i=n-1,...,2,1 can be found by working backwards, using a recursive relationship called the Bellman equation. For i=2,...n, V_{i-i} for the those states. Finally, V_1 at the initial state of the system is the value of the optimal solution. The optimal values of the decision variables can be recovered, one by one, tracking back the calculations already performed.

3.4 Dynamic Programming in Computer Programming

There are two key attributes that a problem must have in order for dynamic programming to be applicable: optimal substructure and overlapping sub problems which are only slightly smaller.

When the overlapping problems are, say, half the size of the original problem the strategy is called "divide and conquer" rather than "dynamic programming". This why merge sort, quick sort, and finding all matches of a regular expression are not classified as dynamic programming problems.

Optimal substructure means that the solution to a given optimization problem can be obtained by the combination of optimal solutions to its sub problems. Consequently, the first step towards devising a dynamic programming solution is to check whether the problem exhibits such optimal substructure. Such optimal substructures are usually described by means of recursion. For example, given a graph G = (V, E), the shortest path p from a vertex u to a vertex v exhibits optimal substructure: take any intermediate vertex won this shortest path p. If p is truly the shortest path, then the path p_1 from u to w and p_2 from w to v are indeed the shortest paths between the corresponding vertices (by the simple cut-and-paste a argument described in CLRS). Hence, one can easily formulate the solution for finding shortest paths in a recursive manner, which is what the Bellman-Ford algorithm does.

Overlapping sub problems means that the space of sub problems must be small, that is, any recursive algorithm solving the problem should solve the same sub problems over and over, rather than generating new sub problems. For example, consider the recursive formulation for generating the Fibonacci series: $F_i = F_{i-1} + F_{i-2}$, with base case $F_1=F_2=1$. Then $F_{43} = F_{42} + F_{41}$, and $F_{42} = F_{41} + F_{40}$. Now F_{41} is being solved in the recursive sub trees if both F_{43} as well as F_{42}. Even though the total number of sub problems is actually small (only 43 of them), we end up solving the same problems over and over if we adopt a naïve recursive solution such as this. Dynamic programming takes account of this fact and solves each sub-problem only once.

This can be a achieved in either of two ways

- Top-down approach:

  This is the direct fall-out of the recursive formulation of any problem. If the solution to any problem can be formulated recursively using the solution to its sub problems, and if its sub problems are overlapping, then one can easily memorize or store the solutions to the sub problems in a table. Whenever we attempt to solve a new sub problem, we first check the table to see if it is already solved. If a solution has been recorded we can use it directly, otherwise we solve the sub problem and add its solution to the table.

- Bottom-up approach:

  This is the more interesting case. Once we formulate the solution to a problem recursively as in terms of its sub problems, we can try reformulating the problem in a bottom-up fashion: try solving the sub problems first and use their solutions to build-on and arrive at solutions to bigger sub problems. This is also usually done in a tabular form by iteratively generating solutions to bigger and bigger sub problems by using the solutions to small sub problems. For example, if we already know the values of F_{41} and F_{40}, we can directly calculate the value of F_{42}.

3.5 Characteristics of Dynamic Programming Applications

There are a number of characteristics that are common to all problems and all dynamic programming problems.

1. The problem can be divided into stage with a decision required at each stage. In capital budgeting problem the stages were the allocations to a single plant and the decision was how much to spend.
2. Each stage has a number of states associated with it. The states for a capital budgeting problem correspond to the amount spent at that point in time. In the shortest path problem the states were the node reached.
3. The decision at one stage transforms on state into a state in another stage. The decision of much to spend gave a total amount spent for the next stage. The decision of where to go next defined where you arrived in the next stage.

4. Given the current state, the optimal decision for each of the remaining state does not depend on the previous states or decisions. In the budgeting problem, it is not necessary to know how the money was spent in previous stages, only how much was spent.

   In the path problem, it is not necessary to know how you got to a node, only that you did.

5. There exist a recursive relationship that identifies the optimal decision for stage j, given that stage$(j + 1)$ has already been solved.
6. The final stage must be solvable by itself. The last time properties are tied up in the recursive relationship given above.

3.6 Computational Efficiency of Dynamic Programming

In smaller networks it would be a matter of determining the shortest path form on point to another by enumerating all the possible paths (after all there are only a few path).

In larger networks however, compute enumeration is practically impossible and the use of dynamic programming is much more efficient for determine a shortest path.

In a network where there five stage with: stage1 – 1 state, stage 2 – 3 state, stage 3 – 3 state, stage 4 – 2 state and stage 5 – 1 state. Total enumerate will result in 1(3)(3)(2)(1) = 18 paths while DP with result in 1(3)(3)(2)(1) = 18 path.

If in another network there are seven stages with 5 states each. The total enumeration gives $5(5^5)$ paths.

However DP require 4(25) + 5 = 105 addition = DP requires 105 = 0.00 7 times as many additions is explicit enumeration.

3.7 Deterministic Dynamic Programming Versus Stochastic Dynamic Programming

There is one major difference between Stochastic Dynamic Programming and Deterministic Dynamic Programming. In Deterministic Dynamic Programming the complete decision path is known. In Stochastic Dynamic Programming the actual decision path will depend on the way the random aspects play out. Because of this solving a Stochastic Dynamic Programming problem involves giving a decision rule for every possible state, not just along an optimal path.

A multi-stage decision process is Stochastic if the return associated with at least one decision in the process is random. This randomness generally enters in one of two ways; either the states are uniquely determined by the decisions but the returns associated with one or more states are uncertain or the returns are uniquely determined by the states arising from one or more decision are uncertain.

If the probability distributions governing the events are known and if the number of stages are finite, then deterministic dynamic programming approach is useful for optimizing a stochastic multistage decision process. The general procedure is to optimize the expected value of the return. In those cases where the randomness occurs exclusively in the returns associated with the states arising from the decision, this procedure has the effect of transforming a stochastic process into a deterministic one. For processes in which randomness exists in the states associated with the decisions, a policy may be exhibited as a policy table.

3.8 Integer Programming

An Integer Programming problem (IP) is a Linear Programming (LP) problem in which some or all the variables are required to be nonnegative integers. An Integer programming in which all variables are required to be integers is a Pure Integer Programming problem.

Many problems can be modeled as an Integer Programming problem. The model is;

For a maximization problem

Maximize $\quad Z = \sum_{i=1}^{n} r_i(ax_i + b)$

Subject to $\quad \sum_{i=1}^{n} x_i \leq c_i$

$$x_i \geq 0, 1, 2, 3, 4, \dots, N$$

3.9 Applications of Dynamic Programming

Dynamic programming can be used many types of Integer Programming-Consumption and savings problems, shortest path problem, The Knapsack Problem, Network Problems, Inventory Problems, Equipment replacement problems, Resource Allocation Problems etc

3.10 Optimal Consumption and Saving Problems

A mathematical optimization problem that is often used in dynamic programming to economists concerns a consumer who lives over the periodst = 0,1,2,...,T and must decide how much to consume and how much to save in each period.

Let c_t be consumption in period t, and assume consumption yields utility$u(c_t) = \ln(c_t)$ as long as the consumer lives. Assume the consumer is impatient, so that he discounts future utility by a factor b each period, where 0 <b< 1. Let k_t be capital in period t. Assume initial capital is a given amount k_0> 0, and suppose that this period's capital and consumption determine next period's capital as $k_t + 1 = Ak_t^a - c_t$, where A is a positive constant and $0 < a < 1$. Assume capital cannot be negative. Then the consumer's decision problem can be written as follows:

Maximize $\quad Z = \sum_{t=0}^{T} b^t In\,(c_t)$

Subject to $\quad k_{t+1} = Ak_t^a - c_t \geq 0$

for all t = 0,1,2,...,T

Written this way, the problem looks complicated, because it involves solving for all the choice variables $c_0, c_1, c_2, \ldots, c_T$ and $k_1, k_2, \ldots, k_{T+1}$ simultaneously. (Note that k_0 is not a choice variable—the consumer's initial capital is taken as given.)

The dynamic programming approach to solving this problem involves breaking it apart into a sequence of smaller decisions. To do so, we define a sequence of value functions $V_T(k)$, for $t = 0, 1, 2, \ldots, T, T+1$ which represent the value of having any amount of capital k at each time t. Note that $V_{T+1}(k) = 0$,that is, there is (by assumption) no utility from having capital after death.

The value of any quantity of capital at any previous time can be calculated by backward induction using the Bellman equation. In this problem, for each $t = 0, 1, 2, \ldots, T$ the Bellman equation is $\quad V_t(k_t) = \max(\ln(c_t) + bV_{t+1}(k_{t+1})$

subject to $\quad Ak_t^a - c_t \geq 0$

This problem is much simpler than the one we wrote down before, because it involves only two decision variables, c_t and k_{t+1}. Intuitively, instead of choosing his whole lifetime plan at birth, the consumer can take things one step at a time. At time t, his current capital k_t is given, and he only needs to choose current consumption c_t and savingk_{t+1}.

To actually solve this problem, we work backwards. For simplicity, the current level of capital is denoted as k. $V_{T+1}(k)$ is already known, so using the Bellman equation once we can calculate $V_T(k)$, and so on until we get to $V_0(k)$, which is the value of the initial decision problem for the whole lifetime. In other words, once we know $V_{T-j+1}(k)$, we can calculate $V_{T-j}(k)$, which is the maximum of $\ln(c_{T-j}) + bV_{T-j+1}(Ak^a - c_{T-j})$, where c_{T-j} is the choice variable and working backwards, it can be shown that the value function at time t = T − j iswhere each V_{T-j} is a constant, and the optimal amount to consume at time t = T − j iswhich can be simplified to

$c_T(k)$= $A\mathrm{k}^a$, and $c_{T-1}(k) = (1+x)^n = \frac{Ak^a}{1+ab}$, and $c_T - 2(k) = \frac{Ak^a}{1+ab+a^2b^2}$

We see that it is optimal to consume a larger fraction of current wealth as one gets older, finally consuming all remaining wealth in period T, the last period of life.

3.11 Dijkstra's Algorithm for the Shortest Path Problem

From a dynamic programming point of view, Dijkstra's the shortest path problem is a successive approximation scheme that solves the dynamic programming functional equation for the shortest path problem by the Reaching method.

In fact, Dijkstra's explanation of the logic behind the algorithm is to find the path of minimum total length between two given nodes P and Q. We use the fact that, if R is a node on the minimal path from P to Q, knowledge of the latter implies the knowledge of the minimal path from P to R. This is a paraphrasing of Bellman's famous Principle of Optimality in the context of the shortest path problem.

Dynamic programming may look somewhat familiar. Both the shortest path algorithm and our method for CPM project scheduling have a lot in common with it.

Let's look at a particular type of shortest path problem. Suppose we wish to get from A to J in the road network of Figure 3.1

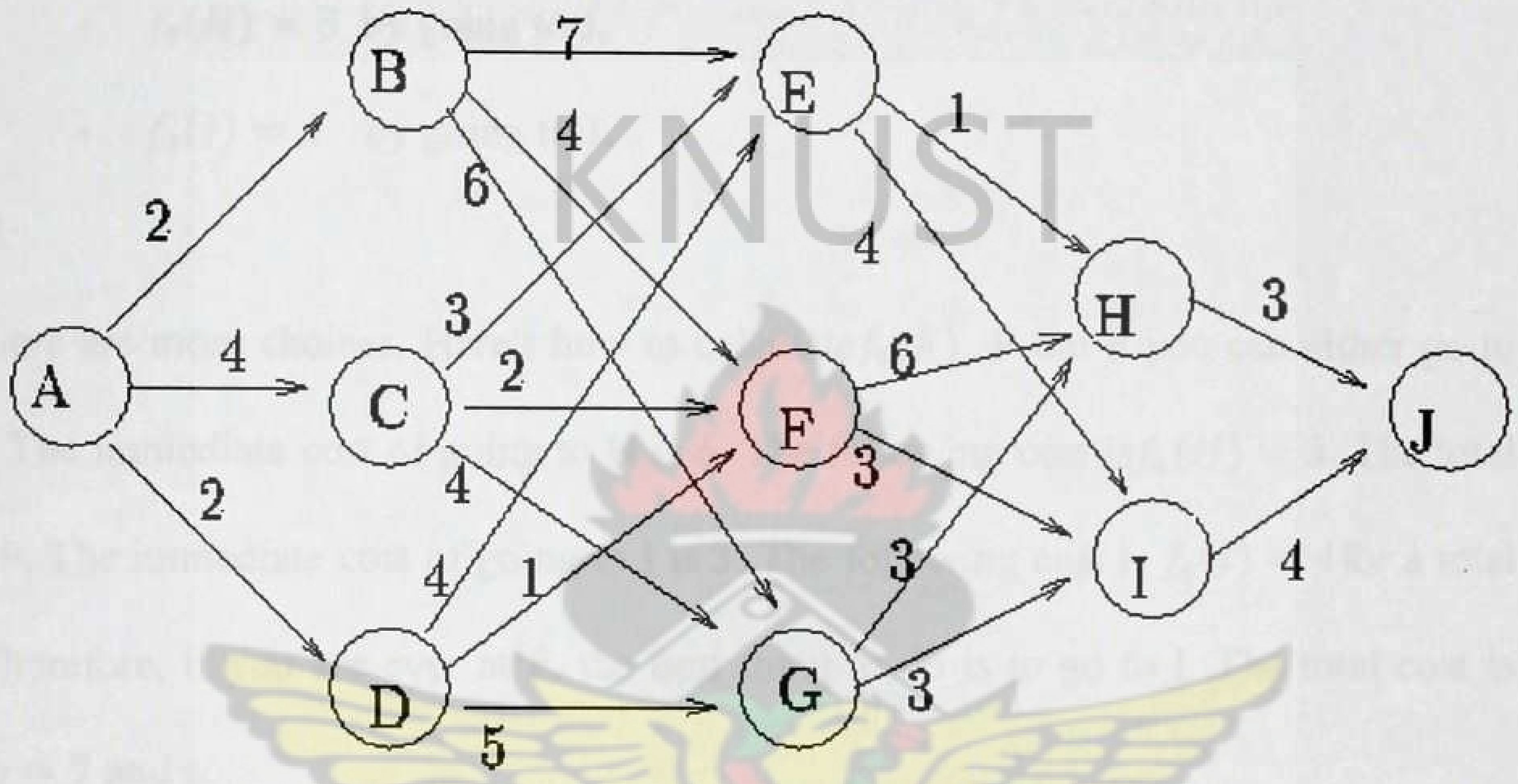

Figure 3.1: Road Network

The numbers on the arcs represent distances. Due to the special structure of this problem, we can break it up into stages. Stage 1 contains node A, stage 2 contains nodes B, C, and D, stage 3 contains node E, F, and G, stage 4 contains H and I, and stage 5 contains J. The states in each stage correspond just to the node names, so stage 3 contains states E, F, and G.

If we let S denote a node in stage j and let $f_j(S)$ be the shortest distance from node S to the destination J, we can write

$$f_j(S) = \min_{\text{nodes Z in stage } j+1} \{c_{sz} + f_{j+1}(Z)\}$$

where c_{sz} denotes the length of arc SZ. This gives the recursion needed to solve this problem. We begin by setting $f_5(J) = 0$ and follow with the rest of the calculations:

Stage 4:

During stage 4, there are no real decisions to make: you simply go to your destination *J*. So you get:

- $f_4(H) = 3$ by going to J,
- $f_4(I) = 4$ by going to J.

Stage 3:

Here there are more choices. Here's how to calculate $f_4(F)$. From F you can either go to H or I. The immediate cost of going to H is 6. The following cost is $f_4(H) = 3$. The total cost is 9. The immediate cost of going to I is 3. The following cost is $f_4(I) = 4$ for a total of 7. Therefore, if you are ever at F, the best thing to do is to go to I. The total cost is 7, $f_4(F) = 7$ and so.

Table 3.3 gives all the calculations:

Table 3.3 Dijkstra's Algorithm Table for Road network Stage 1

S_3	$CS_3\,Z_3 + F_4(Z_3)$		$F_3(S_3)$	Decision
	H	I		
E	4	8	4	H
F	9	7	7	I
G	6	7	6	H

You now continue working back through the stages one by one, each time completely computing a stage before continuing to the preceding one. The results are:

Table 3.4 Dijkstra's Algorithm Table for Road network Stage 2

S_2	$CS_2\,Z_2 + F_3(Z_3)$			$F_3(S_3)$	Decision Go to
	E	F	G		
B	11	11	12	11	E or F
C	7	9	10	7	E
D	8	8	11	8	E or F

Table 3.5 Dijkstra's Algorithm Table for Road network Stage 1

S_1	$CS_1\,Z_1 + F_2(Z_1)$			$F_2(S_1)$	**Decision**
	B	**C**	**D**		**Go to**
A	**13**	**11**	**11**	**11**	**C or D**

There is another formulation for the knapsack problem. This illustrates how arbitrary our definitions of stages, states, and decisions are. It also points out that there is some flexibility on the rules for dynamic programming. Our definitions required a decision at a stage to take us to the next stage (which we would already have calculated through backwards recursion). In fact, it could take us to any stage we have already calculated. This gives us a bit more flexibility in our calculations.

The recursion I am about to present is a forward recursion. For a knapsack problem, let the stages be indexed by *w*, the weight filled. The decision is to determine the last item added to bring the weight to *w*. There is just one state per stage. Let $g(w)$ be the maximum benefit that can be gained from a *w* pound knapsack. Continuing to use b_j and w_j as the weight and benefit, respectively, for item *j*, the following relates $g(w)$ to previously calculated *g* values:

$$g(w) = \max_j \{b_j + g(w - w_j)\}$$

Intuitively, to fill a w pound knapsack, we must end off by adding some item. If we add item j, we end up with a knapsack of size $w - w_j$ to fill. To illustrate on the above example:

- $g(0) = 0$
- $g(1) = 30$ add item 3.
- g(2)= $\max\{65 + g(0) = 65, 30 + g(1) - 60\} = 65$add item 1.
- g(2)= $\max\{65 + g(1) = 95, 80 + g(0) = 80, 30 + g(2) = 95\} = 95$add item 1 or 3.
- g(2)= $\max\{65 + g(2) = 130, 80 + g(1) = 110, 30 + g(3) = 125\} = 130$ add item 1.
- g(2)=$\max\{65 + g(3) = 160, 80 + g(2) = 145, 30 + g(4) = 160\} = 160$add item 1 or3.

This gives a maximum of 160, which is gained by adding 2 of item 1 and 1 of item 3.

3.12 The Knapsack Problem

Imagine we have a homework assignment with different parts labeled A through G. Each part has a "value" (in points) and a "size" (time in hours to complete). For example, say the values and times for our assignment are:

Table 3.6: The Knapsack Problem Table

	A	B	C	D	E	F	G
Value	7	9	5	12	14	6	12
Time	3	4	2	6	7	3	5

Say we have a total of 15 hours: which parts should we do? If there was partial credit that was proportional to the amount of work done (e.g., one hour spent on problem C earns you 2.5 points) then the best approach is to work on problems in order of points/hour. But, what if there is no partial credit? In that case, which parts should you do, and what is the best total value possible? The above is an instance of the knapsack problem, formally defined as follows:

In this case, the optimal strategy is to do parts A, B, F, and G for a total of 34 points. We notice that this doesn't include doing part C which has the most points/hour!

In the knapsack problem we are given a set of n items, where each item $i$isspecified by a size s_i and a valuev_i. We are also given a size bound S (the size of our knapsack).The goal is to find the subset of items of maximum total value such that sum of their sizes is atmost S (they all fit into the knapsack).We can solve the knapsack problem in exponential time by trying all possible subsets. With Dynamic Programming, we can reduce this to time O(nS).Let's do this top down by starting with a simple recursive solution and then trying to memorize.

it. Let's start by just computing the best possible total value, and we afterwards can see how to actually extract the items needed.

3.13 Network Problems

In many applications, dynamic programming reduces to finding the shortest (or longest) path joining two points in a given network. Dynamic programming (working backward) can be used to find the shortest path in a network.

3.13.1 Equipment Replacement Problems

Suppose a shop needs to have a certain machine over the next five year period. Each new machine cost \$100o. the cost of maintaining the machine during its with year of operation is as follows: $c_1 = \$60$, c_2 =\$80, and $c_3 = \$120$. A machine may be kept up to three years before being traded in. the trade in value after i years is $s_1 = \$800, s_2 = \$600$ as $s_3 = \$500$. How can the shop minimize cost over the five year period?

Let the stages corresponds to each year. The state is the age of the machine for that year. The decision are whether to keep the machine or trade it in for a new machine.

Let $f_t(x)$ be the minimum cost incurred from time t to time 5, given the machine is x years old in time t.

Since we have to trade in at time 5 $f_5(x) = -s_x$

Now we consider the time periods.

If you have three year old machine in time t, you must trade in, so

$$f_t(3) = -500 + 1000 + 60 + f_{t+1}(1)$$

If you have a two year old machine you can either trade or keep.

- Trade will not cost $-600 + 1000 + 60 + f_{t+1}(1)$
- Keep will cost $120 + f_{t+1}(3)$

So the best thing to do with a two year old machine is the minimum of the two

$f_t(2) = \min\{-600 + 1000 + 60 + f_{t+1}(1), 120f_{t+1}(3)\}$

For a one year old machine trade will cost $-800 + 100 + 60 + f_{t+1}(1)$

Keep will cost $80 + f_{t+1}(2)$

$f_t(1) = \min\{-800 + 1000 + 60 + f_t(1), 80 + f_t(2)\}$

For a zero year old machine ~~we have~~ to buy $1000 + 60 + f_t(1)$

$f_0(0) = 1000 + 60 + f_1(1)$

$f_o(0) = 1060 + f_1(1)$

$f_t(1) = \min(260 + f_{t+1}(1), 80 + f_{t=1}(2)$

$f_t(2) = \min(460 + f_{t+1}(1), 120 + f_{t+1}(3))$

$f_t(3) = \min(560 + f_{t+1}(1))$

$$f_5(x) = -5_x$$

$f_5(1) = -S_1 = -800$

$f_5(2) = -S_2 = -600$

$f_5(3) = -S_3 = -500$

This is solved with backwards recursion as follows:

Table 3.7 Decision table for Equipment Replacement Problem at Stage 5

Age x	$f_5(x)$
1	-800
2	-600
3	-500

Table 3.8 Decision table for Equipment Replacement Problem at Stage 4

Age	Trade	Keep	$f_4(2)$	Decision
1	-540	-520	-540	Trade
2	-340	-380	-380	Keep
3	-240	–	-240	Trade

Table 3.9 Decision table for Equipment Replacement Problem at Stage 3

Age	Trade	Keep	$f_3(2)$	Decision
1	-280	-300	-300	Keep
2	-80	-120	-120	Keep
3	20		20	Trade

Table 3.10 Decision table for Equipment Replacement Problem at Stage 2

Age	Trade	Keep	$f_1(2)$	Decision
1	220	220	220	Trade or Keep

Table 3.11 Decision table for Equipment Replacement Problem at Stage 1

Age	Trade	Keep	$f_0(2)$	Decision
0	–	1280	1280	Keep

So the cost is 1280 and one solution is to trade in years; 1 and 2. There are other optimal solutions.

3.13.2 An Inventory Problem

In this section, we illustrate how dynamic programming can be used to solve an inventory problem with the following characteristics:

1. Time is broken up into periods, the present period being period 1, the next period 2, and the final period T. At the beginning of period 1, the demand during each period is known.
2. At the beginning of each period, the firm must determine how many units should be produced. Production capacity during each period is limited.
3. Each period's demand must be met on time from inventory or current production. During any period in which production takes place, a fixed cost of production as well as a variable per-unit cost is incurred.
4. The firm has limited storage capacity. This is reflected by a limit on end-of- period inventory. A per-unit holding cost is incurred on each periods ending inventory.

5. The firm's goal is to minimize the total cost of meeting on time the demands for periods 1, 2,…,T.

3.13.3 Resource Allocation Problems

Resource allocation problems, in which limited resources must be allocated among several activities, are often solved by dynamic programming. Recall that we have solved such problems by linear programming. To use linear programming to do resource allocation three assumptions must be made:

Assumption 1

The amount of a resource assigned to an activity may be any nonnegative number.

Assumption 2

The benefit obtained from each activity is proportional to the amount of the resource assigned to the activity.

Assumption 3

The benefit obtained from more than one activity is the sum of the benefits obtained from the individual activities.

Even if assumptions 1 and 2 do not hold, dynamic programming can be used to solve resource allocation problems efficiently when assumption 3 is valid and when the amount of the resource allocated to each activity is a member of a finite set.

3.14 Summary

In this chapter, we considered the research methodology of the study. The next chapter is devoted for data collection and analysis of the study.

CHAPTER FOUR

4.0 Data Collection, Methodology and Analysis

In this chapter Four, investment features are considered in order to test the performance of the trading strategy and returns. We collect data on the six investments –Government of Ghana's Treasury Bills, Barclays Bank Ghana, Ghana Commercial Bank, Data Bank, Guinness Ghana Limited and Fan Milk Limited, Accra. Sample period ranges from 2003 to 2011 and we normalize the price series such that each commodity's price changes have annualized volatility of 10%. Each commodity characteristic is its past returns at various time horizons. As such, in order to predict the 1-year return factors for the commodities, pooled panel regression on the data set is run to obtain the annual returns. Dynamic programming is then used to determine the optimal investment returns and the corresponding investments to be made.

The companies used in the research are Government of Ghana's Treasury Bills, Barclays Bank Ghana, Ghana Commercial Bank, Data Bank, Ghana Guinness Limited and Fan Milk Limited, Accra.

We make the following notations:

Return from investment 1 - Government of Ghana's Treasury Bills

Return from investment 2 - Barclays Bank Ghana, Accra

Return from investment 3 - Ghana Commercial Bank

Return from investment 4 - Data Bank, Accra

Return from investment 5 - Ghana Guinness Limited, Accra

Return from investment 6 – Fan Milk Limited, Accra

We state that the amount invested are restricted to be integral multiples of ¢100. There is a maximum of ¢900 available and six investment options. The table shows the amount invested and the return at the end of one year period.

We start the algorithm by first identifying the appropriate state, stage and decision. We define stage such that when one stage is remains, the problem will be trivial to solve. That is if we arrange the investment in no specific other as shown in the table above.

At Investment 6:

Investment 6 Returns = Investment 5 Returns + Investment 6 inputs

At Investment 1:

Investment 1Returns = Investment 2 Returns + Investment 1 inputs

Hence Investment is the stage.

Define the state of the stage that is at each stage (investment) the investor will have to decide how much money he will have to invest. To do this we need to know only the amount of money left at the beginning of the investment (stage). Hence State is the amount of money left to be invested.

Letting $f_i(x)(i = 1, 2, 3, 4, 5, 6)$ denotes the return in (¢) from investment i when x units of money are invested in it. The above return table has been written as shown below

Table 4.0 Returns from Investment table

	AMOUNT INVESTED									
RETURN FROM INVESTMENT	0	100	200	300	400	500	600	700	800	900
$f_1(x)$	0	10	20	25	33	35	40	45	55	60
$f_2(x)$	0	15	30	40	45	50	55	70	80	90
$f_3(x)$	0	12	22	32	35	40	50	60	65	70
$f_4(x)$	0	18	40	50	55	65	70	70	85	95
$f_5(x)$	0	20	28	38	50	75	75	80	80	85
$f_6(x)$	0	19	25	35	40	50	60	70	75	80

Return values of various units as the number of units of money invested in investment$i$.

Define $x_i(i = 100, 200, 300, 400, 500, 600, 700, 800, 900)$ as the number of units of money invested in investment$i$.

Define $M_j(i)$ = the best return beginning in stage j and state i.

$d_j(i)$ = Decisions taken at state that achieves $M_j(i)$

We note that $M_j(0) = 0$ and $d_j(0) = 0$

The model for solving the above is :

Maximize: $Z = f_1(x) + f_2(x) + f_3(x) + f_4(x) + f_5(x) + f_6(x)$

Subject to$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 \leq 900$

SOLUTION

We begin the solution by considering the last stage of the process, stage 6. We assume that the previous stages have been completed and we are to complete the allocation of the money to the investment 6. Since we do not know how much was allocated to the previous investment (investment 5), we do not know how many units are available for investment 6. Thus we consider all possibilities.

After the first five investments have been made there will be either ¢0, ¢1, ¢2, ¢3, ¢4, ¢5, ¢6, ¢7, ¢8, ¢9. It is clear from the definition of $f_6(x)$ that the best way to complete the process is to allocate all available units to investments 6.

From investment 6:

$$M_6(9) = \max[f_6(0), f_6(1), f_6(2), f_6(3), f_6(4), f_6(5), f_6(6), f_6(7), f_6(8), f_6(9)]$$

$= \max[0, 19, 25, 35, 40, 50, 60, 70, 75, 80]$ $\quad M_6(9) = 80 \text{with} d_6(9) = 9$

$$M_6(8) = \max[f_6(0), f_6(1), f_6(2), f_6(3), f_6(4), f_6(5), f_6(6), f_6(7), f_6(8)]$$

$= \max[0, 19, 25, 35, 40, 50, 60, 70, 75]$ $\quad M_6(8) = 75 \text{with} d_6(8) = 8$

$$M_6(7) = \max[f_6(0), f_6(1), f_6(2), f_6(3), f_6(4), f_6(5), f_6(6), f_6(7)]$$

$= \max[0, 19, 25, 35, 40, 50, 60, 70]$ $\quad M_6(7) = 70 \text{with} d_6(7) = 7$

$$M_6(6) = \max[f_6(0), f_6(1), f_6(2), f_6(3), f_6(4), f_6(5), f_6(6)]$$

$= \max[0, 19, 25, 35, 40, 50, 60]$ $\quad M_6(6) = 60 \text{with} d_6(6) = 7$

$$M_6(5) = \max[f_6(0), f_6(1), f_6(2), f_6(3), f_6(4), f_6(5)]$$

$= \max[0, 19, 25, 35, 40, 50]$ $\quad M_6(5) = 50 \text{with} d_6(5) = 5$

$$M_6(4) = \max[f_6(0), f_6(1), f_6(2), f_6(3), f_6(4)]$$

$= \max[0, 19, 25, 35, 40]$ $\quad M_6(4) = 40 \quad \text{with} d_6(4) = 4$

$$M_6(3) = \max[f_6(0), f_6(1), f_6(2), f_6(3)]$$

$= \max[0, 19, 25, 35]$ $\quad M_6(3) = 35 \quad \text{with} d_6(3) = 3$

$$M_6(2) = \max[f_6(0), f_6(1), f_6(2)]$$

$= \max[0, 19, 25]$

$M_6(2) = 25 \text{with} d_6(2) = 2$

$$M_6(1) = \max[f_6(0), f_6(1)]$$

$= \max[0, 19]$

$M_6(1) = 19 \text{with} d_6(1) = 1$

$$M_6(0) = \max[f_6(0)]$$

$= \max[0]$ $\quad M_6(0) = 0 \text{with} d_6(0) = 0$

From investment 5:

$$M_5(9) = \max[f_5(0) + M_6(9), \quad f_5(1) + M_6(8),$$
$$f_5(2) + M_6(7), \quad f_5(3) + M_6(6), \quad f_5(4) + M_6(5),$$
$$f_5(5) + M_6(4), \quad f_5(6) + M_6(3), \quad f_5(7) + M_6(2),$$
$$f_5(8) + M_6(1), \quad f_5(9) + M_6(0)]$$

$$M_5(9) = \max[0 + 80, \quad 20 + 75, \quad 28 + 70, \quad 38 + 60, \quad 50 + 50, \quad 75 + 40,$$
$$75 + 35, \quad 80 + 25, \quad 80 + 19, \quad 85 + 0]$$

$$M_5(9) = \max[80, \ 95, \ 98, \ 98, \ 100, \ 115, \ 110, \ 105, \ 99, \ 85]$$

$$M_5(9) = 115 \text{ with } \quad d_5(9) = 5$$

$$M_5(8) = \max[f_5(0) + M_6(8), \quad f_5(1) + M_6(7),$$
$$f_5(2) + M_6(6), \quad f_5(3) + M_6(5), \quad f_5(4) + M_6(4),$$
$$f_5(5) + M_6(3), \quad f_5(6) + M_6(2), \quad f_5(7) + M_6(1),$$
$$f_5(8) + M_6(0)]$$

$$M_5(8) = \max[0 + 75, \quad 20 + 70, \quad 28 + 60, \quad 38 + 50, \quad 50 + 40, \quad 75 + 35,$$
$$75 + 25, \quad 80 + 19, \quad 80 + 0]$$

$$M_5(8) = \max[75, \ 90, \ 88, \ 88, \ 90, \ 110, \ 100, \ 99, \ 80]$$

$$M_5(8) = 110 \text{ with } \quad d_5(8) = 5$$

$$M_5(7) = \max[f_5(0) + M_6(7), \quad f_5(1) + M_6(6),$$

$$f_5(2) + M_6(2), \quad f_5(3) + M_6(4), \quad f_5(4) + M_6(3),$$

$$f_5(5) + M_6(2), \quad f_5(6) + M_6(1), \quad f_5(7) + M_6(0)]$$

$$M_5(7) = \max[0 + 70, \quad 20 + 60, \quad 28 + 50, \quad 38 + 40, \quad 50 + 35, \quad 75 + 25,$$

$$75 + 19, \quad 80 + 0,]$$

$$M_5(7) = \max[70, \quad 80, \quad 78, \quad 78, \quad 85, \quad 100, \quad 92, \quad 80]$$

$M_5(8) = 100$ with $\quad d_5(7) = 5$

$$M_5(6) = \max[f_5(0) + M_6(6), \quad f_5(1) + M_6(5),$$

$$f_5(2) + M_6(4), \quad f_5(3) + M_6(3), \quad f_5(4) + M_6(2),$$

$$f_5(5) + M_6(1), \quad f_5(6) + M_6(0)]$$

$$M_5(6) = \max[0 + 60, \quad 20 + 50, \quad 28 + 40, \quad 38 + 35, \quad 50 + 25, \quad 75 + 19,$$

$$75 + 0]$$

$$M_5(6) = \max[60, \quad 70, \quad 68, \quad 73, \quad 75, \quad 94, \quad 75]$$

$M_5(6) = 94$with $\quad d_5(6) = 5$

$$M_5(5) = \max[f_5(0) + M_6(5), \quad f_5(1) + M_6(4),$$

$$f_5(2) + M_6(3), \quad f_5(3) + M_6(2), \quad f_5(4) + M_6(1),$$

$$f_5(5) + M_6(0)]$$

$$M_5(5) = \max[0 + 50, \quad 20 + 40, \quad 28 + 35, \quad 38 + 25, \quad 50 + 19, \quad 75 + 0]$$

$$M_5(5) = \max[50, \quad 60, \quad 63, \quad 63, \quad 69, \quad 75]$$

$M_5(5) = 75 \quad$ with $\quad d_5(5) = 5$

$$M_5(4) = \max[f_5(0) + M_6(4), \quad f_5(1) + M_6(3),$$

$$f_5(2) + M_6(2), \quad f_5(3) + M_6(1), \quad f_5(4) + M_6(0)]$$

$$M_5(4) = \max[0 + 40, \quad 20 + 35, \quad 28 + 25, \quad 38 + 19, \quad 50 + 0]$$

$$M_5(4) = \max[40, \quad 55, \quad 53, \quad 57, \quad 50]$$

$M_5(4) = 57$ with $d_5(4) = 4$

$$M_5(3) = \max[f_5(0) + M_6(3), \quad f_5(1) + M_6(2),$$

$$f_5(2) + M_6(1), \quad f_5(3) + M_6(0)]$$

$$M_5(3) = \max[0 + 35, \quad 20 + 25, \quad 28 + 19, \quad 38 + 0]$$

$$M_5(3) = \max[35, \quad 45, \quad 47, \quad 38]$$

$M_5(3) = 47$ with $d_5(3) = 2$

$$M_5(2) = \max[f_5(0) + M_6(2), \quad f_5(1) + M_6(1), \quad f_5(2) + M_6(0)]$$

$$M_5(2) = \max[0 + 25, \quad 20 + 19, \quad 28 + 0]$$

$$M_5(2) = \max[25, \quad 39, \quad 28]$$

$M_5(2) = 39$ with $d_5(2) = 1$

$$M_5(1) = \max[f_5(0) + M_6(1), \quad f_5(1) + M_6(0)]$$

$$M_5(1) = \max[0 + 19, \quad 20 + 0]$$

$$M_5(1) = \max[19, \quad 20]$$

$M_5(1) = 20$ with $d_5(1) = 1$

$$M_5(0) = \max[f_5(0) + M_6(0)]$$

$$M_5(0) = \max[0 + 0]$$

$$M_5(0) = \max[0]$$

$M_5(0) = 0$ with $d_5(0) = 0$

From investment 4:

$M_4(9) = \max[f_4(0) + M_5(9), \quad f_4(1) + M_5(8), \quad f_4(2) + M_5(7), \quad f_4(3) + M_5(6),$

$f_4(4) + M_5(5), \quad f_4(5) + M_5(4), \quad f_4(6) + M_5(3),$

$f_4(7) + M_5(2), \quad f_4(8) + M_5(1), \quad f_4(9) + M_5(0)]$

$M_4(9) = \max[0 + 115, \quad 18 + 110, \quad 40 + 100, \quad 50 + 94, \quad 55 + 75,$

$65 + 57, \quad 70 + 47, \quad 70 + 39, \quad 85 + 20, \quad 95 + 0]$

$M_4(9) = \max[115, \quad 128, \quad 140, \quad 144, \quad 130, \quad 122, \quad 117, \quad 109, \quad 105, \quad 95]$

$M_4(9) = 144$ with $d_4(9) = 3$

$M_4(8) = \max[f_4(0) + M_5(8), \quad f_4(1) + M_5(7), \quad f_4(2) + M_5(6), \quad f_4(3) + M_5(5),$

$f_4(4) + M_5(4), \quad f_4(5) + M_5(3), \quad f_4(6) + M_5(2),$

$f_4(7) + M_5(1), \quad f_4(8) + M_5(0)]$

$M_4(8) = \max[0 + 110, \quad 18 + 100, \quad 40 + 94, \quad 50 + 75, \quad 55 + 57,$

$65 + 47, \quad 70 + 39, \quad 70 + 20, \quad 85 + 0]$

$M_4(8) = \max[110, \quad 118, \quad 134, \quad 125, \quad 112, \quad 112, \quad 112, \quad 90, \quad 85]$

$M_4(8) = 134$ with $d_4(8) = 2$

$M_4(7) = \max[f_4(0) + M_5(7), \quad f_4(1) + M_5(6), \quad f_4(2) + M_5(5), \quad f_4(3) + M_5(4),$

$f_4(4) + M_5(3), \quad f_4(5) + M_5(2), \quad f_4(6) + M_5(1),$

$f_4(7) + M_5(0)]$

$M_4(7) = \max[0 + 100, \quad 18 + 94, \quad 40 + 75, \quad 50 + 57, \quad 55 + 47,$

$65 + 39, \quad 70 + 20, \quad 70 + 0]$

$M_4(7) = \max[100, \quad 112, \quad 115, \quad 107, \quad 102, \quad 104, \quad 90, \quad 70]$

$M_4(7) = 115$ with $d_4(7) = 2$

$M_4(6) = \max[f_4(0) + M_5(6), \quad f_4(1) + M_5(5), \quad f_4(2) + M_5(4), \quad f_4(3) + M_5(3),$

$f_4(4) + M_5(2), \quad f_4(5) + M_5(1), \quad f_4(6) + M_5(0)]$

$M_4(6) = \max[0 + 94, \quad 18 + 75, \quad 40 + 57, \quad 50 + 47, \quad 55 + 39, \quad 65 + 20,$

$70 + 0]$

$M_4(6) = \max[94, \ 92, \ 97, \ 97, \ 94, \ 84, \ 70]$

$M_4(6) = 97$ with $d_4(6) = 2$

$M_4(5) = \max[f_4(0) + M_5(5), \quad f_4(1) + M_5(4), \quad f_4(2) + M_5(3), \quad f_4(3) + M_5(2),$

$f_4(4) + M_5(1), \quad f_4(5) + M_5(0)]$

$M_4(5) = \max[0 + 75, \quad 18 + 57, \quad 40 + 47, \quad 50 + 39, \quad 55 + 20, \quad 65 + 0]$

$M_4(5) = \max[75, \ 75, \ 87, \ 89, \ 75, \ 65]$

$M_4(5) = 89$ with $d_4(5) = 3$

$M_4(4) = \max[f_4(0) + M_5(4), \quad f_4(1) + M_5(3), \quad f_4(2) + M_5(2), \quad f_4(3) + M_5(1),$

$f_4(4) + M_5(0)]$

$M_4(4) = \max[0 + 57, \quad 18 + 47, \quad 40 + 39, \quad 50 + 20, \quad 55 + 0]$

$M_4(4) = \max[57, \ 65, \ 79, \ 70, \ 55]$

$M_4(4) = 79$ with $d_4(4) = 2$

$M_4(3) = \max[f_4(0) + M_5(3), \quad f_4(1) + M_5(2),$

$f_4(2) + M_5(1), \quad f_4(3) + M_5(0)]$

$M_4(3) = \max[0 + 47, \quad 18 + 39, \quad 40 + 20, \quad 50 + 0]$

$M_4(3) = \max[47, \ 57, \ 60, \ 50]$

$M_4(3) = 60$ with $d_4(3) = 2$

$M_4(2) = \max[f_4(0) + M_5(2), \quad f_4(1) + M_5(1), \quad f_4(2) + M_5(0)]$

$M_4(2) = \max[0 + 39, \ 18 + 20, \quad 40 + 0]$

$M_4(2) = \max[39, \quad 38, \quad 40]$

$M_4(2) = 40$ with $d_4(2) = 2$

$$M_4(1) = \max[f_4(0) + M_5(1), \quad f_4(1) + M_5(0)]$$

$$M_4(1) = \max[0 + 20, \quad 18 + 0]$$

$$M_4(1) = \max[20, \quad 18]$$

$M_4(1) = 20$ with $d_4(1) = 0$

$$M_4(0) = \max[f_4(0) + M_5(0) f_4(0) + M_5(0)]$$

$$M_4(0) = \max[0, \quad 0]$$

$M_4(0) = 0$ with $d_4(0) = 0$

From investment 3:

$$M_3(9) = \max[f_3(0) + M_4(9), \quad f_3(1) + M_4(8), \quad f_3(2) + M_4(7), \quad f_3(3) + M_4(6), \quad f_3(4) + M_4(5), \quad f_3(5) + M_4(4), \quad f_3(6) + M_4(3), \quad f_3(7) + M_4(2), \quad f_3(8) + M_4(1), \quad f_3(9) + M_4(0)]$$

$$M_3(9) = \max[0 + 144, \quad 12 + 134, \quad 22 + 115, \quad 32 + 97, \quad 35 + 89, \quad 40 + 79, \quad 50 + 60, \quad 60 + 40, \quad 65 + 20, \quad 70 + 0]$$

$$M_3(9) = \max[144, \quad 146, \quad 137, \quad 129, \quad 124, \quad 119, \quad 107, \quad 100, \quad 85, \quad 70]$$

$M_3(9) = 146$ with $d_3(9) = 1$

$$M_3(8) = \max[f_3(0) + M_4(8), \quad f_3(1) + M_4(7), \quad f_3(2) + M_4(6), \quad f_3(3) + M_4(5), \quad f_3(4) + M_4(4), \quad f_3(5) + M_4(3), \quad f_3(6) + M_4(2), \quad f_3(7) + M_4(1), \quad f_3(8) + M_4(0)]$$

$$M_3(8) = \max[134, \quad 127, \quad 119, \quad 121, \quad 114, \quad 100, \quad 90, \quad 80, \quad 65]$$

$M_3(8) = 134$ with $d_3(8) = 0$

$$M_3(7) = \max[f_3(0) + M_4(7), \quad f_3(1) + M_4(6),$$

$$f_3(2) + M_4(5), \quad f_3(3) + M_4(4), \quad f_3(4) + M_4(3),$$

$$f_3(5) + M_4(2), \quad f_3(6) + M_4(1), \quad f_3(7) + M_4(0)]$$

$$M_3(7) = \max[0 + 115, \quad 12 + 97, \quad 22 + 89, \quad 32 + 79, \quad 35 + 57,$$

$$40 + 40, \quad 50 + 20, \quad 60 + 0]$$

$$M_3(7) = \max[115, \quad 109, \quad 111, \quad 111, \quad 92, \quad 80, \quad 70, \quad 60]$$

$$M_3(7) = 115 \text{ with } \quad d_3(7) = 0$$

$$M_3(6) = \max[f_3(0) + M_4(6), \quad f_3(1) + M_4(5),$$

$$f_3(2) + M_4(4), \quad f_3(3) + M_4(3), \quad f_3(4) + M_4(2),$$

$$f_3(5) + M_4(1), \quad f_3(6) + M_4(0)]$$

$$M_3(6) = \max[0 + 97, \quad 12 + 89, \quad 22 + 79, \quad 32 + 57, \quad 35 + 40, \quad 40 + 20,$$

$$50 + 0]$$

$$M_3(6) = \max[97, \quad 101, \quad 101, \quad 89, \quad 75, \quad 60, \quad 50]$$

$$M_3(6) = 101 \text{ with } \quad d_3(6) = 1$$

$$M_3(5) = \max[f_3(0) + M_4(5), \quad f_3(1) + M_4(4),$$

$$f_3(2) + M_4(3), \quad f_3(3) + M_4(2),$$

$$f_3(4) + M_4(1), \quad f_3(5) + M_4(0)]$$

$$M_3(5) = \max[0 + 89, \quad 12 + 79, \quad 22 + 57, \quad 32 + 40, \quad 35 + 20, \quad 40 + 0]$$

$$M_3(5) = \max[89, \quad 91, \quad 79, \quad 72, \quad 55, \quad 40]$$

$$M_3(5) = 91 \quad \text{with} \quad d_3(5) = 1$$

$$M_3(4) = \max[f_3(0) + M_4(4), \quad f_3(1) + M_4(3),$$

$$f_3(2) + M_4(2), \quad f_3(3) + M_4(1), \quad f_3(4) + M_4(0)]$$

$$M_3(4) = \max[0 + 79, \quad 12 + 57, \quad 22 + 40, \quad 32 + 20, \quad 35 + 0]$$

$$M_3(4) = \max[79, \quad 69, \quad 65, \quad 62, \quad 35]$$

$M_3(4) = 79$ with $d_3(4) = 0$

$$M_3(3) = \max[f_3(0) + M_4(3), \quad f_3(1) + M_4(2),$$

$$f_3(2) + M_4(1), \quad f_3(3) + M_4(0)]$$

$$M_3(3) = \max[0 + 60, \quad 12 + 40, \quad 22 + 20, \quad 32 + 0]$$

$$M_3(3) = \max[60, \quad 52, \quad 42, \quad 32]$$

$M_3(3) = 60$ with $d_3(3) = 0$

$$M_3(2) = \max[f_3(0) + M_4(2), \quad f_3(1) + M_4(1), \quad f_3(2) + M_4(0)]$$

$$M_3(2) = \max[0 + 40, \quad 12 + 20, \quad 22 + 0]$$

$$M_3(2) = \max[40, \quad 32, \quad 22]$$

$M_3(2) = 40$ with $d_3(2) = 0$

$$M_3(1) = \max[f_3(0) + M_4(1), \quad f_3(1) + M_4(0)]$$

$$M_3(1) = \max[0 + 20, \quad 12 + 0]$$

$$M_3(1) = \max[20, \quad 12]$$

$M_3(1) = 20$ with $d_3(1) = 0$

$$M_3(0) = \max[f_3(0) + M_4(0)]$$

$$M_3(0) = \max[0 + 0]$$

$$M_3(0) = \max[0]$$

$M_3(0) = 0$ with $d_3(1) = 0$

From investment 2:

$$M_2(9) = \max[f_2(0) + M_3(9), \quad f_2(1) + M_3(8),$$

$$f_2(2) + M_3(7), \quad f_2(3) + M_3(6), \quad f_2(4) + M_3(5),$$

$$f_2(5) + M_3(4), \quad f_2(6) + M_3(3), \quad f_2(7) + M_3(2),$$

$$2(8) + M_3(1), \quad f_2(9) + M_3(0)]$$

$$M_2(9) = \max[0 + 146, \quad 15 + 134, \quad 30 + 115, \quad 40 + 101, \quad 45 + 91,$$

$$50 + 79, \quad 55 + 60, \quad 70 + 40, \quad 80 + 20, \quad 90 + 0]$$

$$M_2(9) = \max[146, \quad 149, \quad 145, \quad 141, \quad 136, \quad 129, \quad 115, \quad 111, \quad 120, \quad 90]$$

$$M_2(9) = 149 \text{ with } \quad d_2(9) = 1$$

$$M_2(8) = \max[f_2(0) + M_3(8), \quad f_2(1) + M_3(7),$$

$$f_2(2) + M_3(6), \quad f_2(3) + M_3(5), \quad f_2(4) + M_3(4),$$

$$f_2(5) + M_3(3), \quad f_2(6) + M_3(2), \quad f_2(7) + M_3(1),$$

$$2(8) + M_3(0)]$$

$$M_2(8) = \max[0 + 134, \quad 15 + 115, \quad 30 + 101, \quad 40 + 91, \quad 45 + 79,$$

$$50 + 60, \quad 55 + 40, \quad 70 + 20, \quad 80 + 0]$$

$$M_2(8) = \max[134, \quad 130, \quad 131, \quad 131, \quad 124, \quad 107, \quad 95, \quad 90, \quad 80]$$

$$M_2(8) = 134 \text{ with } \quad d_2(8) = 0$$

$$M_2(7) = \max[f_2(0) + M_3(7), \quad f_2(1) + M_3(6),$$

$$f_2(2) + M_3(5), \quad f_2(3) + M_3(4), \quad f_2(4) + M_3(3),$$

$$f_2(5) + M_3(2), \quad f_2(6) + M_3(1), \quad f_2(7) + M_3(0)]$$

$$M_2(7) = \max[0 + 115, \quad 15 + 101, \quad 30 + 91, \quad 40 + 79, \quad 45 + 60,$$

$$50 + 40, \quad 55 + 20, \quad 70 + 0]$$

$$M_2(7) = \max[115, \quad 116, \quad 121, \quad 119, \quad 105, \quad 90, \quad 75, \quad 70]$$

$$M_2(7) = 121 \text{ with } \quad d_2(7) = 2$$

$$M_2(6) = \max[f_2(0) + M_3(6), \quad f_2(1) + M_3(5),$$

$$f_2(2) + M_3(4), \quad f_2(3) + M_3(3), \quad f_2(4) + M_3(2),$$

$$f_2(5) + M_3(1), \quad f_2(6) + M_3(0)]$$

$$M_2(6) = \max[0 + 101, \quad 15 + 91, \quad 30 + 79, \quad 40 + 60, \quad 45 + 40,$$

$$50 + 20, \quad 55 + 0]$$

$$M_2(6) = \max[101, \quad 106, \quad 109, \quad 97, \quad 100, \quad 95, \quad 70]$$

$M_2(6) = 109$ with $d_2(6) = 2$

$$M_2(5) = \max[f_2(0) + M_3(5), \quad f_2(1) + M_3(4),$$

$$f_2(2) + M_3(3), \quad f_2(3) + M_3(2), \quad f_2(4) + M_3(1),$$

$$f_2(5) + M_3(0)]$$

$$M_2(5) = \max[0 + 91, \quad 15 + 79, \quad 30 + 60, \quad 40 + 40, \quad 45 + 20, \quad 50 + 0]$$

$$M_2(5) = \max[91, \quad 94, \quad 90, \quad 80, \quad 65, \quad 50]$$

$M_2(5) = 94$ with $d_2(5) = 1$

$$M_2(4) = \max[f_2(0) + M_3(4), \quad f_2(1) + M_3(3),$$

$$f_2(2) + M_3(2), \quad f_2(3) + M_3(1), \quad f_2(4) + M_3(0)]$$

$$M_2(4) = \max[0 + 79, \quad 15 + 60, \quad 30 + 40, \quad 40 + 20, \quad 45 + 0]$$

$$M_2(4) = \max[79, \quad 75, \quad 70, \quad 60, \quad 45]$$

$M_2(4) = 79$ with $d_2(4) = 0$

$$M_2(3) = \max[f_2(0) + M_3(3), \quad f_2(1) + M_3(2),$$

$$f_2(2) + M_3(1), \quad f_2(3) + M_3(0)]$$

$$M_2(3) = \max[0 + 60, \quad 15 + 40, \quad 30 + 20, \quad 40 + 0]$$

$$M_2(3) = \max[60, \quad 55, \quad 50, \quad 40]$$

$M_2(3) = 60$ with $d_2(3) = 0$

$$M_2(2) = \max[f_2(0) + M_3(2), \quad f_2(1) + M_3(1), \quad f_2(2) + M_3(0)]$$

$$M_2(2) = \max[0 + 40, \quad 15 + 20, \quad 30 + 0]$$

$$M_2(2) = \max[40, \quad 35, \quad 30]$$

$M_2(3) = 40$ with $d_2(2) = 0$

$$M_2(1) = \max[f_2(0) + M_3(1), \quad f_2(1) + M_3(0)]$$

$$M_2(1) = \max[0 + 20, \quad 15 + 0]$$

$$M_2(1) = \max[20, \quad 15]$$

$M_2(1) = 20$ with $d_2(1) = 0$

$$M_2(0) = \max[f_2(0) + M_3(0)]$$

$$M_2(0) = \max[0 + 0]$$

$$M_2(0) = \max[0]$$

$M_2(0) = 0$ with $d_2(0) = 0$

From investment 1:

$$M_1(9) = \max[f_1(0) + M_2(9), \quad f_1(1) + M_2(8), \quad f_1(2) + M_2(7), \quad f_1(3) + M_2(6), \quad f_1(4) + M_2(5), \quad f_1(5) + M_2(4), \quad f_1(6) + M_2(3), \quad f_1(7) + M_2(2), \quad f_1(8) + M_2(1), \quad f_1(9) + M_2(0)]$$

$$M_1(9) = \max[0 + 149, \quad \cancel{10 + 134}, \quad 20 + 121, \quad 25 + 109, \quad 33 + 94, \quad 35 + 79, \quad 40 + 60, \quad 45 + 40, \quad 55 + 20, \quad 60 + 0]$$

$$M_1(9) = \max[149, \quad 144, \quad 141, \quad 134, \quad 127, \quad 114, \quad 100, \quad 85, \quad 75, \quad 60]$$

$M_1(9) = 149$ with $d_1(9) = 0$

ALLOCATION

The optimal return from the investment is 149 which we obtained by starting the allocation from stage 1, then to stage 2 up to stage 6 as follows:

i. With 9 units, available allocate to stage 1, $d_1(9) = 0$, leaving 9 – 0 = 9.

ii. With 9 units, available allocate to stage 2, $d_2(9) = 1$, leaving 9 – 1 = 8.

iii. With 8 units, available allocate to stage 3, $d_3(8) = 0$, leaving 8 – 0 = 8.

iv. With 8 units, available allocate to stage 4, $d_4(8) = 2$, leaving 8 – 2 = 6.

v. With 6 units, available allocate to stage 5, $d_5(6) = 5$, leaving 6 – 5 = 1.

vi. With 1 units, available allocate to stage 6, $d_6(1) = 1$, leaving 1 – 1 = 0.

Table 4.1 Optimal allocations and returns from investment table

	AMOUNT INVESTED									
RETURN FROM INVESTMENT	40	100	200	300	400	500	600	700	800	900
$f_1(x)$	0 *	10	20	25	33	35	40	45	55	60
$f_2(x)$	0	15 *	30	40	45	50	55	70	80	90
$f_3(x)$	0 *	12	22	32	35	40	50	60	65	70
$f_4(x)$	0	18	40 *	50	55	65	70	70	85	95
$f_5(x)$	0	20	28	38	50	75	75 *	80	80	85
$f_6(x)$	0	19 *	25	35	40	50	60	70	75	80

The table shows that with ¢900 available for investment and given the corresponding annual returns from the various financial institutions we should not invest in Government of Ghana's Treasury Bills and Ghana Commercial Bank. However, we should invest ¢100 in Barclays Bank to get ¢15, ¢200 in Data Bank, Accra to get ¢40, ¢500 in Ghana

Guinness Limited, Accra for a return of ¢75 and ¢100 in Fan Milk Limited, Accra for a return of ¢19. This gives the optimal returns of ¢149.

4.1 Summary

In this chapter, we undertook collection and analysis of the study of the study. The next chapter discusses the summary, conclusions and recommendations of the study.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.0 Introduction

This concluding chapter provides insights into the major findings of the study. It further provides key recommendations to policy implications and also outlines recommendations for further research.

5.1 Summary

The use of Dynamic Programming in Investment Portfolios helps to decide whether to accept or reject an investment with more realism. There are two main points to note. One is the proposition that DP in Investment Portfolios allows to relax the low-before-high fare order of arrival bookings. The problem can be solved from any direction, without any particular arrangement of the investments. However, the optimal solution is obtained by a backward substitution. In practice, the Dynamic Programming provides the optimal policy for the Portfolio problem, by evaluating the whole tree of possibilities and making at each point in time the decision that would imply higher future expected revenues, processing backward recursion. The dark side is the increase in the computation difficulties according to the dimension of the problem.

5.2 Conclusion

Based on the above literature review and the analysis of the data it could be concluded that dynamic programming can be used in allocating resources for the optimal investment returns from a portfolio. The findings from the research optimally allocate the ¢900 to obtain a maximum return of ¢145. The allocation was done first by a forward substitution, identifying the optimal at each stage and then a backward substitution for the allocations to obtain the optimal returns. The maximum return that was realized from the six stage programme beginning from stage 6 with ¢'900 was achieved as follows allocating $d_1(9) = 0$ to investment 1 leaving $9 - 0 = 9$ units for investment 2. But $d_2(9) = 2$ indicating that we allocate 2 units to investment 2, leaving 9 – 2 = 7 for investment three. Then allocate nothing to investment 3 since $d_3(7) = 0$. With ¢7, available allocate to stage 4, $d_4(7) = 2$, leaving $7 - 2 = 5$. With ¢5, available allocate to investment 5, $d_5(5) = 5$ and nothing for investment 6.The results of the research clearly prove that Dynamic Programming as very efficient in allocating resources for the optimal investment returns from a portfolio.

5.3 Recommendation

It is recommended that investors should not invest too much money in a single investment. One should always divide the resources available in bit to invest in different investments. An important aspect of investment that was assumed to be equal is risk. The research did not consider the risk associated with the portfolios. When the element of risk is considered then we can apply a Stochastic Dynamic Programming. For instance, we have the option to buy Guinness Ghana Limited share at ¢0.70. We can exercise this option at any time in the next ten days. The current price of Guinness Ghana Limited is ¢0.50. Assuming a model of Guinness Ghana Limited share movement that predicts the following: on each day the share will go up by ¢0.02 with probability 0.4, stay the same with probability 0.1 and go down by ¢0.02 with probability 0.4. the value of the option if we exercise it at price x is $x - 0.70$. Then we can formulate this as a Stochastic Dynamic Programming as follows:

We will have stage i for each day i, just before the exercise or keep decision.

Let $f_i(x)$ be the expected value of the option on day i given that share price is x. Then the optimal decision is given by

$f_i(x)$ =max$\{x - 0.70, 0.4f_{i+1}(x + 0.02) + 0.1f_{i+1}f(x) + 0.5f_{i+1}(x - 0.02)\}$, and

$f_{10}(x)$ =max $\{0,\ x - 0.70\}$.

The above problem can be solved with the help of a spread sheet.

5.3.1 Recommendations for future Research

The study can be strengthened by increasing the sample size and including participants in other geographical areas. With an increased sample size, a more detailed analysis among the available investment portfolios can be undertaken and be reported in a future study.

Another approach to develop this research to a higher level might be carrying out similar research in different countries in the West Africa sub-Region.

REFERENCES

Ambachtscheer, K. P. (1994). The Economics of Pension Fund Management. Financial Analysts Journal,50:6 pp21–31.

Battochio, P. (2002). Optimal Portfolio Strategies with Stochastic Wage Income and Inflation, The Case of a defined Pension Plan.

Bertimas, D., A. W. Lo, and Hummel P. (1999). Optimal control of execution costs for portfolios. Computing in Science & Engineering 1.6, pp.40-53.

Bertsekas, D. P. (2007). Dynamic Programming and Suboptimal Control: A Survey from ADP to MPC, in Fundamental Issues in Control. pp. 911-916.

Bertsekas, D. P. (2007).Dynamic Programming and optimal Control. Third Edition, Volume II., Athena Scientific, Belmont, Massachusetts.

Botterud, A., and Korpa's, M. A. (2007).A stochastic dynamic model for optimal timing of investments in new generation capacity in restructured power systems, International Journal of Electrical Power and Energy Systems, vol. 29, pp. 163–174.

Bouzaher, Aziz, John B. Braden, and Gary V. Johnson (1990).A dynamic programming approach to a class of nonpoint source pollution control problems. Management Science 36.1: 1-15.

Brinson, G. P., Hood, L. R. and Beebower, G. L. (1986).Determinants of Portfolio Performance. Financial Analysts Journal, 42:4 pp. 39–44.

Brinson, G., L. Hood and Beebower G (1986).Determinants of Portfolio Performance, Financial Analyst Journal (July/August), pp. 39–48.

Cvitani, C. J. and Ma, J. (1996). Hedging options for a large investor and forward backward

SDE's, Annals of Applied Probability 6, 370-398.

Cvitani,C. J., Pham, H. and Touzi, N. (2000). Super-replication in stochastic volatility models under portfolio constraints, Journal of Applied Probability 36, 523-545.

Dijkhuizen, A.A, Hurne R.B.M, Beck, P. Van, Hendriks, Th. G. B. (1993). European Journal of Operational Research.

Dixit, A. and Pindyck, R. (1994).Investment under Uncertainty, Princeton University Press, Princeton.

Engebretson, K. J. (1995). A Multi-Asset Class Approach to Pension Fund Investments. Government Finance Review. Vol.11, No. 1 (February): 11-14.

Ghezzi, L.L. (1997). Immunization and maximum optimal control. Journal of Optimization

Theory & Applications, 95, pp. 701-711.

Greco, Sergio. Dynamic programming in data log with aggregates. Knowledge and Data Engineering, IEEE Transactions on 11.2 (1999): 265-283.

Grossman, S.J. and Zhou, Z. (1993).Optimal Investment Strategies for Controlling Drawdowns.

Guangliang, H. (1999). Drawdown Controlled Optimal Portfolio Selection with Linear Constraints on Portfolio Weights.

Heikkinen, T. and Pietola, K. (2009).Investment and the dynamic cost of income uncertainty: The case of diminishing expectations in agriculture, European Journal of Operational Research, vol. 192, pp.634–646.

Hillier Frederick S., Gerald J. Lieberman, Mark S. Hillier, (2003). Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets (Unabridged. Edition).

Hillier F, Lieberman G. (1995).Introduction to operations research, 6th ed. New York: McGraw-Hill.

Jacobson, D. (1968). Differential dynamic programming methods for solving bang-bang control problems. Automatic Control, IEEE Transactions on, 13(6), pp. 661-675.

Jinchun, Y. (2007). Optimal life insurance purchase, consumption and portfolio under an uncertain life. Consumption and Portfolio Under an Uncertain Life.

Karatzas, I., Shreve, S.E., (1994). Brownian Motion and Stochastic Calculus. Springer-Verlag, New York.

Kosowski, R. (2006). Do Mutual Funds Perform when it Matters Most to Investors? US Mutual Fund Performance and Risk in Recessions and Expansions.

Kosowski, R. (2006). Lessons for Hedge Funds from the May Meltdown, Hedge Fund Manager Week, Adviser.

Myles,G. D.,(2003).Investment Analysis

Pola, G. (2006). Optimal Dynamic Asset Allocation: A Stochastic Invariance Approach, submitted for publication. Also available online at http://www.diel.univaq.it/people/pola/.

Rudo, M and Rohrs, C. (2008). Optimal portfolio construction in co-integrated vector autoregressive systems, Proceedings of the 2008 American Control Conference (ACC).

Steffen P., and Giegerich, R., (2005).Versatile and declarative dynamic programming using pair algebras.

Sun, Walter, et al, (2006).Using dynamic programming to optimally rebalance portfolios. The Journal of Trading 1.2, pp. 16-27.

Tenney, M.S., (1995). Dynamic Dynamic-Programming Solutions for the Portfolio of Risky Assets.

Tyndale, H. F. (1996). Holy Bible. New Living Translation.

Van Ryzin, G. and Vulcano, G. (2011).An expectation-maximization algorithm to estimate a general class of non-parametric choice models. Revised: March 2012.

Vilaa, J. L, Papi, R.T., and Michael, E., (2001),Use and benefits of tools for project risk management Optimal Consumption and Portfolio Choice with Borrowing Constraints International Journal of Project Management, vol. 19, pp. 9–17.

Villeneuve, D., Desrosiers, J., Lübbecke, M. E. and Soumis, F. (2005). On compact formulations for integer programs solved by column generation. Annals of Operations Research, 139(1), 375-388.

Wang, S.Q., (1998), Evaluation and competitive tendering of power plant project in China. Journal of Construction Engineering and Management, vol. 124, pp. 333–341.

Wayne L. Winston (1994). Operations Research-applications and Algorithms. Duxbury Press.

Xu-song, X., and Jian-mou, W., (2002).A Dynamic Programming Algorithm on Project-Gang Investment Decision-Making, Wuhan University Journal of Natural Sciences vol. 7, no. 4, pp. 103-107.

Yan, F., and Bai, F., (2009), Application of Dynamic Programming Model in Stock Portfolio-under the Background of the Subprime Mortgage Crisis, International Journal of Business and management, vol.9, no 3, pp. 178-182.