İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

DYNAMIC PROGRAMMING APPLICATIONS IN INVESTMENT ANALYSIS

Master Thesis by
Özlem İNCE, Ind. Eng.
507041118

Date of Submission : 03 July 2006
Date of Defence Examination : 12 June 2006

Supervisor : Prof. Dr. M. Nahit SERARSLAN
Members of the Examining Committee: Prof. Dr. Demet BAYRAKTAR
Yrd. Doç. Dr. Mehmet Mutlu YENİSEY

JUNE 2006

PREFACE

Investment decisions of firms depend on a large number of factors, one of which may be the extent of uncertainty about future events. Uncertainty may well be an especially relevant factor in environments in which investors have difficulties in making predictions about the future, since the environment may be highly volatile information which is difficult to obtain. In this light, uncertainty may be a particularly relevant factor determining investment in developing and transition economies. In this thesis, therefore, investment decisions under uncertainty are selected as an area of interest.

I am very grateful to my supervisor Prof. Dr. M. Nahit SERARSLAN for his help during my work. He also supervised further development of the project and contributed to final form of the thesis. It was a real honour working with him.

I am very indebted to Assist. Prof. Dr. Ufuk KULA who provided helpful advice at various stages of the project. He provided most of the resources of my project and supported to overcome my anxiety about my thesis.

Moreover, I want to thank to Yalçın EYİGÜN who is the manager of the Department of Railway System Presidency of İstanbul Büyükşehir Municipality, Mustafa MURTAZA, Dursun BALCIOĞLU who is the manager of Department of Technical Processes and the authorities of İstanbul Ulaşım A.Ş. for their help and opinions. They also provided application data for my thesis. In addition, I want to thank to Cumhur COŞGUN who provided the communication with İstanbul Büyükşehir Municipality.

May, 2006 Özlem İNCE

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ÖZET

Dünyanın en büyük metropollerinden biri olan İstanbul'un en önemli problemlerinden biri trafik ve ulaşım problemidir. Bu problemin çözümü toplu taşımayı cazip hale getirmektir. Toplu taşımayı cazip hale getirme sunulan taşıma sistemlerinin rahat, hızlı, güvenilir ve ucuz olmasına dayanır. Bu yüzden uzun dönemde tamamlanması planlanan raylı sistemlerin yatırımları en önemli projelerdir. Ayrıca toplu taşımayı zaman, ücret ve fiziksel yapısı açısından bir bütün olarak ele alırsak, raylı sistemler en büyük öneme sahiptir.

Bu çalışmada, yapım aşamasında olan 4.64 km uzunluğunda üç istasyona sahip 4.Levent-Maslak Sanayi Metro hattı incelenmiştir. Projenin iki amacı vardır: Birincisi yatırımın fizibilitesini araştırmak ikinciside en iyi yatırım politikasını belirlemek. Bu analizlerde belirli ve olasılıklı dinamik programlama kullanılmıştır.

Proje hattında genel ulaşım şu anda otobüs ve minibüslerle sağlanmaktadır. Otobüs ve minibüsler toplam araç trafiğinin 12% sini oluşturmakta ve trafiğin yoğun olduğu saatlerde de toplam yolcuların 65% ini taşımaktadırlar.

Bu hattın 2005 – 2009 yılları arasındaki 5 yıllık yapım aşamasından sonra 2010 yılında faaliyete geçmesi planlanmaktadır. Bu yapım aşamasında, yatırımın en iyi zamanlaması araştırılmıştır. Daha sonra 2010 yılından 2034 yılına kadar olan dönemde de en iyi yatırım politikasına, yani her yıl alınması gereken tren sayısına, karar verilmiştir.

Her yıl olasılıklı olarak yolcu sayısını gösteren 3 duruma ve her durum da satın alınması gereken tren sayısından oluşan bir karar kümesine sahiptir. En fazla kar sağlayan tren sayısı o durumun en iyi kararını oluşturmaktadır. Projenin maliyetleri metro hattının yapım maliyetleri, tren yatırım maliyetleri ve bunların çalışma ve bakım maliyetlerinden oluşmaktadır. Projenin gelirleri ise, otobüs, minibüs ve özel taşıtlardan sağlanan maliyet tasarruflarıdır. Metro projesinin faaliyete geçmesiyle trafiğin bir kısmının metroya geçmesi, hem yeni ve ihtiyaç duyulacak, ek otobüs ve minibüs yatırımlarında bir azalma, hem de bunların çalıştırılmasından kaynaklanan maliyetlerde bir azalma sağlayacaktır. Ayrıca seyehat süresinde de büyük bir azalma olacağından bundan sağlanan maliyet tasarruflarının büyük bir katkısı olacaktır. Nüfus artarken trafik de artacağından, metro projesi hızlı, güvenilir ve ucuz olmasından dolayı toplu taşımada en önemli yeri alacaktır.

2010 – 2034 yılları arasındaki en iyi yatırım politikasını bulmak için Microsoft Visual Studio C++ 6.0 kullanılmıştır. Bu bilgisayar programından alınan sonuçlara göre 2029'dan 2034 yılına kadar 1 tren satın alınmasına karar verilmiş ve toplam $355,079,716 kar sağlanmıştır. Sonuçları elde ettikten sonra bazı parametrelerin etkilerini görebilmek için duyarlılık analizi yapılmıştır. Bu parametreler nüfustaki artış, faiz oranındaki azalma, geçiş olasılıklarındaki değişiklikler, tren yatırım maliyetinin düşmesi ve bunların farklı kombinasyonları olarak ele alınmıştır.

Projenin iki amacı birbiriyle ilişkilidir: Eğer en iyi yatırım politikası kar sağlamıyorsa, projeye yatırım yapılmayacaktır. Proje kar sağladığından, yatırım yapılmaya karar verilmiş ve 5 yıllık yapım aşamasında yapılan yatırımların en iyi zamanlamaları bulunmuştur. Bu dönemde, her yıl iki seçenekten oluşmaktadır: Ya o yıl yatırım yapılacak ya da o yıl bekleyip gelecek yıl yatırım yapılacaktır. Yine dinamik programlama kullanarak, yatırımın en iyi zamanlamasına karar verilmiştir.

Tezin birinci bölümünde giriş kısmı yer almaktadır. Bu kısımda yatırım tanımı, özellikleri verilmiş, klasik teori olarak bilinen Orthodox Teorisi'nden ve yatırım analizlerinde yeni bir görüş olan Opsiyon Yaklaşımı'ndan bahsedilmiştir. Ayrıca geleneksel "net şimdiki değer" kuralının yanlış sonuçlar verebildiği gösterilmiştir.

İkinci ve üçüncü bölümlerde, Dinamik Programlama ve Markov Karar Prosesleri' nin tekniklerine yer verilmiştir. Dördüncü bölümde Stokastik Proseslere giriş yapılmış ve süreklilik gösteren stokastik prosesler üzerinde durulmuştur. Beşinci ve altıncı bölümler, yatırım fırsatları ve yatırımların en iyi zamanlamaları hakkında detaylı bilgiler içermektedir.

Yedinci bölümde, dinamik programlama uygulamaları hakkında yapılan literatür araştırmalarına geniş bir biçimde yer verilmiş, sekizinci bölümde de 4.Levent-Maslak Sanayi Metro hattında yapılan uygulama kısmı yer almıştır. Tez, sonuçlar kısmıyla sonlanmıştır.

SUMMARY

One of the most important problems of İstanbul, which is one of the largest cities in the world, is transportation and traffic. The solution to this problem is attracting of mass transportation. Attracting mass transportation is based on the following characteristics: comfort, speed, reliability and cheapness. Therefore, the investments of railway systems that are planned for completion in a long time are the most important projects. Moreover, when we take into consideration the integration of the time, fee and physical structure of the mass transportation, railway systems have the most significance.

In this thesis, 4.Levent – Maslak Sanayi Subway Project was analyzed, which consists of 4.64 km involving three stations. The objective of the thesis is twofold: Investigate the feasibility of the investment and decide on the optimal investment policy by using probabilistic dynamic programming.

In this project line, public transport is provided by buses and minibuses. Buses and minibuses, including those used by companies and schools, account for only 12% of total vehicular traffic and carry about 65% of total passengers in the peak hour.

The subway project is planned to be in operation in 2010 right after a construction period of 5 years between 2005 and 2009. In this part, the feasibility of the project was investigated. Then, from year 2010 to year 2034, the optimal investment policy, the required number of trains to be purchased in each investment epoch, was determined. In each year, we have 3 states that represent the number of passengers of Yenikapı-Maslak Sanayi Line which is the extension of our project line. Moreover, each state has a decision set which is formed from the number of trains to be purchased and the number of trains which provides the maximum profit is our decision in that state. The costs of the project are construction costs of the subway line, investment costs of the trains and their operating and maintenance costs. And the benefits of the project are the costs savings provided from the buses, minibuses and cars and revenue of fee provided from the trains decided to be bought. The Subway project will reduce the investment required for new and additional buses and minibuses that would be required to accommodate the peak hour traffic demand. Therefore, this project will reduce the vehicle operating costs of buses, minibuses and cars, and provide cost savings of road maintenance and accident costs, because some traffic will be diverted from road to rail. Furthermore, the travel time to be saved by the transport users with the construction of project line constitutes the most important part of the cost savings. Since the traffic will increase when the population increases, the subway project will take the more important place in the mass transportation because of speed, reliability, comfort and cheapness.

To find the optimal investment rule, we used probabilistic dynamic programming since the states are not known with deterministically. The Microsoft Visual Studio C++ 6.0 was used for developing computer program. According to the computer program, it was decided to purchase 1 train from the year 2029 to year 2034 with total profit of $355,079,716. After we got the results, a sensitivity analysis was conducted to determine the significance of effects of possible scenarios with respect to some key parameters. These included increases in population, decreases in interest rate, changes of transition probabilities, etc. and their various combinations.

The aims of the project are interrelated: If the optimal investment policy is not profitable, then do not invest in the project. Then, since the project was found as profitable, we decided to invest in the project and tried to determine the optimal timing of the investments made in the construction period. In the 5 years construction period, we have two choices in each year: The first is investing in that year and the second is waiting in that year and investing in the next year. Again by using dynamic programming, we determined the optimal timing of the investments.

The first chapter of the thesis is an introduction part. Here, the definition of the investment was given and the option approach that is the new view in the investment analysis and the orthodox theory that is known as the neoclassical theory were described. Moreover, it was shown that the traditional "net present value" rule can give wrong answers. The reason is that this rule ignores the irreversibility and the option of delaying investment. In the second and third chapters, the techniques of the Dynamic Programming and Markov Decision Processes were studied. The fourth chapter is an introduction to stochastic processes. In this part, continuous-time stochastic processes were explained. In the fifth and sixth chapters, the techniques and detailed information were studied about the investment opportunities and the optimal timing of the investments. In the seventh chapter, the literature research about dynamic programming applications took part and then my application was described in detailed. Finally, the thesis was ended with main conclusions.

1. INTRODUCTION

1.1 A new view of investment

In economics investment is defined as the act of incurring an immediate cost in the exception of future rewards. Firms that construct plants and install equipment, merchants who lay in a stock of goods for sale, and persons who spend time on vocational education are all investors in this sense. Somewhat less obviously, a firm that shuts down a loss-making plant is also "investing": the payments it must make to extract itself from contractual commitments, including severance payments to labor, are the initial expenditure, and the prospective reward is the reduction in future losses.

Viewed from this perspective, investment decisions are ubiquitous. Purchase of any book about investment decisions is an investment. The reward will be an improved understanding of investment decisions if you are an economist, and an improved ability to make such decisions in the course of your future carrier if you are a business school student.

Most investment decisions share three important characteristics in varying degrees. First, the investment is partially or completely *irreversible*. In other words, the initial cost of investment is at least partially sunk; you can not recover it even if you change your mind. Second, there is *uncertainty* over the future rewards from the investment. The best you can do is to assess the probabilities of the alternative outcomes that can mean greater or smaller profit (or loss) for your venture. Third, you have some leeway about the *timing* of your investment. You can postpone action to get more information (but never, of course, complete certainty) about the future.

These three characteristics interact to determine the optimal decisions of investors. The orthodox theory of investment has not recognized the important qualitative and quantitative implications of the interaction between irreversibility, uncertainty, and the choice of timing. Compared to the predictions of most of the earlier models, real world investment seems much less sensitive to interest rate changes and tax policy changes and much more sensitive to volatility and uncertainty over the economic environment. The new view resolves these anormalies, and in the process offers some guidance for designing more effective public policies concerning investment.

1.2 The Orthodox Theory

How should a firm, facing uncertainty over future market conditions, decide whether to invest in a new factory? Most economics and business school students are taught a simple rule to apply to problems of this sort. First, calculate the present value of the expected stream of profits that this factory will generate. Second, calculate the present value of the stream of expenditures required to build the factory. Finally, determine whether the difference between the two – the *net present value (NPV)* of the investment – is greater than zero. If it is, go ahead and invest.

Of course, there are issues that arise in calculating this net present value. Just how should the expected stream of profits from a new factory be estimated? How should inflation be treated? And what discount rate (or rates) should be used in calculating the net present values? Resolving issues like these are important topics in courses in corporate finance, and especially capital budgeting, but the basic principle is fairly simple – calculate the NPV of an investment project and see whether it is positive.

The net present value is also the basis for the nonclassical theory of investment as taught to undergraduate and graduate students of economics. In the new view the rule expressed using the standard incremental or marginal approach of the economist is found: invest until the value of an incremental unit of capital is just equal to its cost. Again, issues arise in determining the value of an incremental unit of capital, and in determining its cost. For example, what production structure should be posited? How should taxes and depreciation be treated?

Much of the theoretical and empirical literature on the economics of investment deals with issues of this sort. There are two essentially equivalent approaches are mentioned. One, following Jorgenson (1963), compares the per-period value of an incremental unit of capital (its marginal product) and an “equivalent per-period rental cost” or “user cost” that can be computed from the purchase price, the interest and depreciation rates, and applicable taxes. The firm’s desired stock of capital is found by equating the marginal product and the user cost. The actual stock is assumed to adjust to the ideal, either as an ad hoc lag process, or as the optimal response to an explicit cost of adjustment.

The other formulation, due to Tobin (1969), compares the capitalized value of the marginal investment to its purchase cost. The value can be observed directly if the ownership of the investment can be traded in a secondary market; otherwise it is an imputed value computed as the expected present value of the stream of profits it would yield. The ratio of this to the purchase price (replacement cost) of the unit, called Tobin's *q*, governs the investment decision. Investment should be undertaken or expanded if *q* exceeds 1, it should not be undertaken, and existing capital should be reduced, if $q < 1$. The optimal rate of expansion or contraction is found by equating the marginal cost of adjustment to its benefit, which depends on the difference between *q* and 1. Tax rules can alter this somewhat, but the basic principle is similar. In all, the underlying principle is the basic net present value rule.

1.3 The Option Approach

The net present value rule, however, is based on some implicit assumptions that are often overlooked. Most important, it assumes that either the investment is reversible, that is, it can somehow be undone and the expenditures recovered should market conditions turn out to be worse than anticipated, or, if the investment is irreversible, it is now or never proposition, that is, if the firm does not undertake the investment now, it will not be able to in the future.

Although some investments meet these conditions, most do not. Irreversibility and the possibility of delay are very important characteristics of most investments in reality. As a rapidly growing literature has show, the ability to delay irreversible investment expenditure can profoundly affect the decision to invest. It also undermines the simple net present value rule, and hence the theoretical foundation of standard neoclassical investment models. The reason is that a firm with an opportunity to invest is holding an "option" analogous to a financial call option – it has the right but not the obligation to buy an asset at some future time of its choosing. When a firm makes irreversible investment expenditure, it exercises or "kills" its option to invest. It gives up the possibility of waiting of new information to arrive that market conditions change adversely. This lost option value is an opportunity cost that must be included as part of the cost of the investment. As a result, NPV rule "invest when the value of a unit of capital is at least as large as its purchase and installation cost" must be modified. The value of the unit must exceed the purchase and installation cost, by an amount equal to the value of keeping the investment option alive.

Recent studies have shown that this opportunity cost of investing can be large, and investment rules that ignore it can be grossly in error. Also, this opportunity cost is highly sensitive to uncertainty over the future value of the project, so that changing economic conditions that affect the perceived riskness of future cash flows can have a large impact on investment spending, larger than, say, a change in interest rates. This may help to explain why neoclassical investment theory has so far failed to provide good empirical models of investment behavior, and has led to overly optimistic forecasts of effectiveness of interest rate and tax policies in stimulating investment.

The option insight also helps explain why the actual investment behavior of firms differs from the received wisdom taught investment behavior of firms differs from the received wisdom taught in business schools. Firms invest in policies that are expected to yield a return in excess of a required. Firms do not invest until price rises substantially above long-run average cost. On the downside, firms stay in business for lengthy periods while absorbing operating losses, and price can fall substantially below average variable cost without inducing disinvestment or exit. This also seems to conflict with standard theory, but it can be explained once irreversibility and option value are accounted for.

Of course, one can always redefine NPV by subtracting from the conventional calculation the opportunity cost of exercising the option to invest, and then say that the rule "invest if NPV is positive" holds once this correction has been made. However, to do so is to accept the criticism. To highlight the importance of option values, keep them separate from the conventional NPV, if others prefer to continue to include all relevant option values in their definition of NPV. [1, 5]

1.4 Irreversibility and the Ability to Wait

What makes an investment expenditure a sunk cost and thus irreversible? Investment expenditures are sunk costs when they are firm or industry specific. For example, most investments in marketing and advertising are firm specific and can not be recovered. Hence they are clearly sunk costs. A steel plant, on the other hand, is industry specific – it can only be used to produce steel. One might think that because in principle the plant could be sold to another steel company, the investment expenditure is recoverable and is not a sunk cost. This is incorrect. If the industry is reasonably competitive, the value of the plant will be about the same for all firms in the industry, so there would be little to gain from selling it. For example, if the price of steel falls so that a plant turns out, exposes, to have been a "bad" investment for the firm that built it, it will also be viewed as a bad investment by other steel companies, and the ability to sell the plant (or any other industry-specific capital) should be viewed as largely a sunk cost.

Even investments that are not firm or industry specific are often partly irreversible because buyers in markets for used machines, unable to evaluate the quality of an item, will offer a price that corresponds to the average quality in the market. Sellers, who know the quality of the item they are selling, will be reluctant to sell an above-average item. This will lower the market average quality, and therefore the market price. For example, office equipment, cars, trucks, and computers are not industry specific, and although they can be sold to companies in other industries, their resale value will be well below their purchase cost, even if they are almost new.

Irreversibility can also arise because of government regulations or institutional arrangements. For example, capital controls may make it impossible for foreign (or domestic) investors to sell assets and reallocate their funds and investments in new workers may be partly irreversible because of high costs of hiring, training, and firing. Hence most major capital investments are in large part irreversible.

Of course, firms do not always have the opportunity to delay their investments. For example, there can be occasions in which strategic considerations make it imperative for a firm to invest quickly and thereby preempt investment by existing or potential competitors. However, in most cases, delay is at least feasible. There may be a cost to delay – the risk of entry by other firms or simply foregone cash flows – but this cost must be weighed against the benefits of waiting for new information. Those benefits are often large.

An irreversible investment opportunity is much like a financial call option. A call option gives the holder the right, for some specified amount of time, to pay an exercise price and in return receive an asset (e.g. a share of stock) that has some value. Exercising the option is irreversible; although the asset can be sold to another investor, one can not retrieve the option or the money that was paid to exercise it. A firm with an investment opportunity likewise has the option to spend money (“exercise price”), now or in the future, in return for an asset (e.g. a project) of some value. Again, the asset can be sold to another firm, but the investment is irreversible. As with the financial call option, this option to invest is valuable in part because the future value of the asset obtained by investing is uncertain. If the asset rises in value, the net payoff from investing rises. If it falls in value, the firm need not invest, and will only lose what it spent to obtain the investment opportunity.

Finally one might ask how firms obtain their investment opportunities, that is, options to invest, in the first place. Sometimes investment opportunities result from patents, or ownership of land or natural resources. More generally, they arise from a firm’s managerial resources, technological knowledge, reputation, market position, and possible scale, all of which may have been built over time, and which enable the firm to productively undertake investments that individuals or other firms can not undertake. Most important, these options to invest are valuable. Indeed, for most firms, a substantial part of their market value is attributable to their options to invest and grow in the future, as opposed to the capital they already have in place. Most of the economic and financial theory of investment has focused on how firms should (and do) exercise their options to invest. To better understand investment behavior it may be just as important to develop better models of how firms obtain investment opportunities. [1, 5, 18]

2. DYNAMIC PROGRAMMING

Dynamic Programming is a useful mathematical technique for making a sequence of interrelated decisions. It provides a systematic procedure for determining the optimal combination of decisions.

In contrast to linear programming, there is not a standard mathematical formulation of the dynamic programming problem. Rather, dynamic programming is a general type of approach to problem solving, and the particular equations used must be developed to fit each situation. Therefore, a certain degree of ingenuity and insight into the general structure of dynamic programming problems is required to recognize when and how a problem can be solved by dynamic programming procedures.[4] Moreover, dynamic programming is a branch of applied mathematics rather than as something more specific. The subject's coherence results from the fact that it is pervaded by several themes. We shall see that these themes include the concept of states, the principle of optimality, and functional equations. [25]

2.1 Characteristics of Dynamic Programming Problems

The basic features that characterize dynamic programming problems are;

1. The problem can be divided into stages, with a policy decision required at each stage. Dynamic programming problems require making a sequence of interrelated decisions, where each decision corresponds to one stage of the problem.
2. Each stage has a number of states associated with the beginning of that stage. The number of states may be either finite or infinite.
3. The effect of the policy decision at each stage is to transform the current state to a state associated with the beginning of the next stage (possibly according to the probability distribution). Dynamic programming problems can be interpreted in terms of the networks. Each node would correspond to a state. The network could consist of columns of nodes, with each column corresponding to a stage, so that the flow from anode can go only to a node in the next column to the right. The links from a node to nodes in the next column correspond to the possible policy decisions on which state to go to next. The value assigned to each link usually can be interpreted as the

immediate contribution to the objective function from making that policy decision. In most cases, the objective corresponds to finding either the shortest or the longest path through the network.

4. The solution procedure is designed to find an optimal policy for the overall problem, i.e., a prescription of the optimal policy decision at each stage for each of the possible states. Dynamic programming provides the policy prescription of what to do under every possible circumstance (which is why the actual decision made upon reaching a particular state at a given stage is referred to as a policy decision).
5. Given the current state, an optimal policy for the remaining stages is independent of the policy decisions adopted in previous stages. Therefore, the optimal immediate decision depends on only the current state and not on how you got there. This is the principle of optimality for dynamic programming. Any problem lacking this property cannot be formulated as a dynamic programming problem.
6. The solution procedure begins by finding the optimal policy for the last stage. The optimal policy for the last stage prescribes the optimal policy decision for each of the possible states at that stage.
7. A recursive relationship that identifies the optimal policy for stage *n*, given the optimal policy for stage $n+1$, is available. Therefore finding the optimal policy decision when you start in state s at stage n requires finding the minimizing value of x_n. For a problem, the corresponding minimum cost is achieved by using this value of x_n and then following the optimal policy given when you start in state x_n at stage $n+1$. The precise form of the recursive relationship differs somewhat among dynamic programming problems. [4]

N = number of stages

n = label for current stage $(n = 1, 2, ..., N)$

s_n = current state for stage n

x_n =decision variable for stage n

x_n^* = optimal value of x_n (given s_n)

$f_n(s_n, x_n)$ = Contribution of stages $n, n+1, ..., N$ to the objective function if system starts in state s_n at stage n, immediate decision is x_n, and optimal decisions are made thereafter.

$f_n^* = f_n(s_n, x_n^*)$.

The recursive relationship will always be of the form

$$f_n^* = \max_{x_n}\{f_n(s_n, x_n)\} \quad \text{or} \quad f_n^* = \min_{x_n}\{f_n(s_n, x_n)\},$$

where $f_n(s_n, x_n)$ would be written in terms of $s_n, x_n, f_n^*(s_{n+1})$, and probably some measure of the immediate contribution of x_n to the objective function. It is the inclusion of $f_{n+1}^*(s_{n+1})$ on the right-hand side, so that $f_n^*(s_n)$ is defined in terms of $f_{n+1}^*(s_{n+1})$ that makes the expression for $f_n^*(s_n)$ a recursive relationship.

8. When we use this recursive relationship, the solution procedure starts at the end and moves backward stage by stage - each time finding the optimal policy for that stage - until it finds the optimal policy starting at the initial stage. This optimal policy immediately yields an optimal solution for the entire problem, namely, x_1^* for the initial state s_1, then x_2^* for the resulting state s_2 then x_3^* for the resulting state s_3, and so forth to x_N^* for the resulting s_N. [4]

2.2 Deterministic Dynamic Programming

In deterministic problems, the state at the next stage is completely determined by the state and policy decision at the current stage. The probabilistic case, where there is a probability distribution for what the next state will be, is discussed in the next section.

Deterministic dynamic programming can be described diagrammatically as shown in the following Figure 2.1. Thus, at stage n the process will be in some state s_n. Making policy decision x_n then moves the process to some state s_{n+1} at stage $n+1$.

The contribution thereafter, to the objective function under an optimal policy has been previously calculated to be $f_{n+1}^*(s_{n+1})$. The policy decision x_n also makes some contribution to the objective function. Combining these two quantities in an appropriate way provides $f_n(s_n, x_n)$, the contribution of stages n onward to the objective function. Optimizing with respect to x_n then gives $f_n^*(s_n) = f_n(s_n, x_n^*)$. After x_n^* and $f_n^*(s_n)$ are found for each possible value of s_n, the solution procedure is ready to move back one stage.

One way of categorizing deterministic dynamic programming is by the form of the objective function. For example, the objective might be to minimize the sum of the contributions from the individual stages, or to maximize such a sum, or to minimize a product of such terms, and so on. Another categorization is in terms of the nature of the set of states for the respective stages. In particular, states s_n might be representable by a discrete state variable or by a continuous state variable, or perhaps a state vector (more than one variable) is required. Similarly, the decision variables $(x_1, x_2, ..., x_N)$ also can be either discrete or continuous. [4, 25]

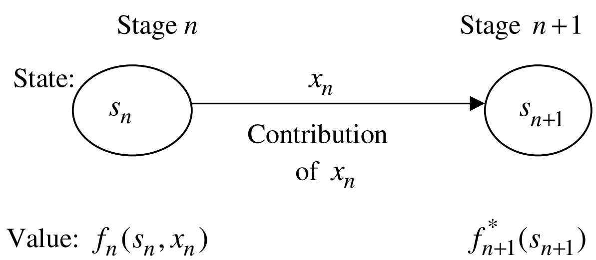

Figure 2.1: The Basic Structure of Deterministic Dynamic Programming

2.3 Probabilistic Dynamic Programming

Probabilistic dynamic programming differs from deterministic dynamic programming in that the state at the next stage is not completely determined by the state and policy decision at the current stage. Rather, there is a probability distribution for what the next state will be. However, this probability distribution still is completely determined by the state and policy decision at the current stage. The resulting dynamic programming is described diagrammatically in Figure 2.2. For the purposes of this diagram, we let S denote the number of possible states at stage $n+1$

and label these states on the right side as $1,2,3,...,S$. The system goes to state i with probability p_i $(i=1,2,...,S)$ given state s_n and decision x_n at stage n . If the system goes to state i, C_i is the contribution of stage n to the objective function.

When Figure 2.2 is expanded to include all the possible states and decisions at all the stages, it is sometimes referred to as a decision tree. If the decision tree is not too large, it provides a useful way of summarizing the various possibilities. Because for the probabilistic structure, the relationship between $f_n(s_n, x_n)$ and the $f^*_{n+1}(s_{n+1})$ necessarily is somewhat more complicated than that for the deterministic dynamic programming. The precise form of this relationship will depend upon the form of the over all objective function. To illustrate, suppose that the objective is to minimize the expected sum of the contributions from the individual stages. In this case, $f_n(s_n, x_n)$ represents the minimum expected sum from stage n onward, given that the state and policy decision at stage n are s_n and x_n respectively. Consequently,

$$f_n(s_n, x_n) = \sum_{i=1}^{S} p_i[C_i + f^*_{n+1}(i)] \quad \text{with} \quad f^*_{n+1}(i) = \min_{x_{n+1}} f_{n+1}(i, x_{n+1})$$

where this minimization is taken over the feasible values of x_{n+1} .

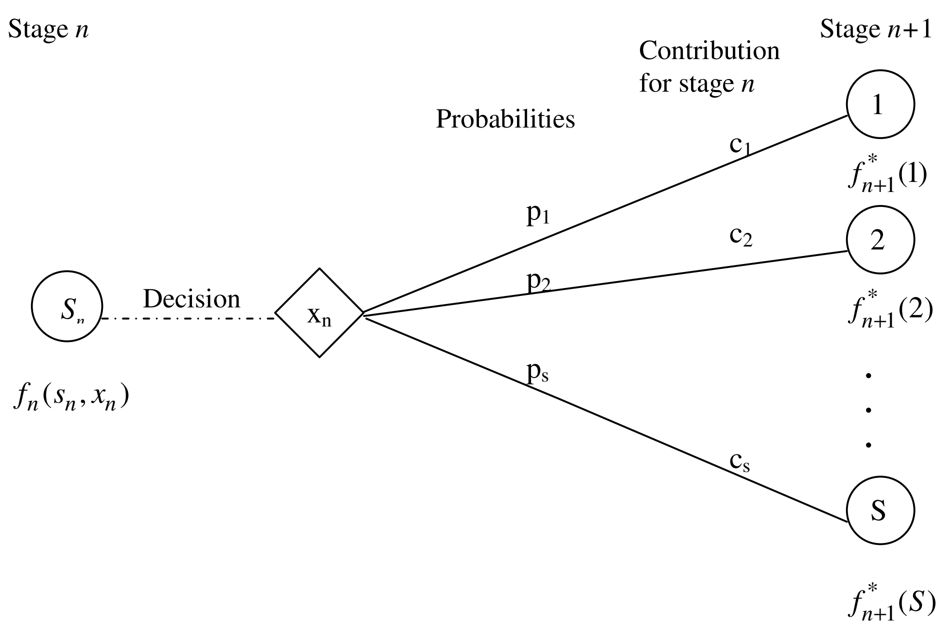

Figure 2.2: The Basic Structure of Probabilistic Dynamic Programming

3. MARKOV DECISION PROCESSES

Markov chains that are observed are only at discrete points in time (e.g., the end of the day) rather than continuously. [4] Each time it is observed, the Markov chain can be in any one of a number of states. Given the current state, a (one-step) transition matrix gives the probabilities of what the state will be next time. Given this transition matrix, steady-state probabilities are found for what state it is in. [4]

Many important systems (e.g., many queuing systems) can be modeled as either a discrete time or continuous time Markov chain. It is useful to describe the behavior of such a system in order to evaluate its performance. However, it may be even more useful to design the operation of the system so as to optimize its performance.

In this part, it is focused on how to design the operation of a discrete time Markov chain so as to optimize its performance. Therefore, rather than passively accepting the design of the Markov chain and the corresponding fixed transition matrix. For each of the possible state of the Markov chain, a decision is made about which one of several alternative actions should be taken in that state. The action chosen affects the transition probabilities as well as both the immediate costs (or rewards) and subsequent costs (or rewards) from operating the system. We want to choose the optimal actions or the respective states when considering both immediate and subsequent costs. The decision process for doing this is referred to as a Markov decision process. [4]

3.1 A Model for Markov Decision Processes

The model for the Markov decision processes can be summarized as follows:

1. The state i of a discrete time Markov chain is observed after each transition $(i = 0,1,...,M)$
2. After each observation, a decision (action) k is chosen from a set of K possible decisions $(k = 1,2,...,K)$. (some of the K decisions may not be relevant for some of the states.)
3. If decision $d_i = k$ is made in state i, an immediate cost is incurred that has an expected value C_{ik} .

4. The decision $d_i = k$ in state *i* determines what the transition probabilities will be for the next transition from state *i*. Denote these transition probabilities by $p_{ij}(k)$, for $j = 0,1,...,M$.
5. A specification of the decisions for the respective states $(d_0, d_1,..., d_M)$ prescribes a policy for the Markov decision process.
6. The objective is to find an optimal policy according to some cost criterion which considers both immediate costs and subsequent costs that result from the future evolution of the process. One common criterion is to minimize the (long run) expected average cost per unit time. (An alternative criterion is considered in [4])

The general model qualifies to be a Markov decision process because it possesses the Markovian property that characterizes any Markov process. In particular, given the current state and decision, any probabilistic statement about the future of the process is completely unaffected by providing any information about the history of the process. This Markovian property holds here since (1) we are dealing with a Markov chain, (2) the new transition probabilities depend on only the current state and decision.

The description of a policy implies two convenient (but unnecessary) properties that they will assume through out this chapter. One property is that a policy is stationary; i.e., whenever the system is in state *i,* the rule for making the decision always is the same regardless of the value of the current time *t*. The second property is that a policy is deterministic, i.e., whenever the system is in state *i*, the rule for making the decision definitely chooses one particular decision. [4]

3.2 Linear Programming and Optimal Policies

The past section described the main kind of policy (called a *stationary, deterministic policy*) that is used by Markov decision processes. It was seen in [4] that any such policy *R* can be viewed as a rule that prescribes decision $d_i(R)$ whenever the system is in state *i*, for each $i = 0,1,...,M$. Thus, *R* is characterized by the values

$\{d_0(R), d_1(R),..., d_M(R)\}$.

Equivalently, R can be characterized by assigning values $D_{ik} = 0$ or 1 in the matrix

$$\text{State } i \quad \begin{matrix} 0 \\ 1 \\ . \\ M \end{matrix} \begin{bmatrix} D_{01} & D_{02} & \dots & D_{0K} \\ D_{11} & D_{12} & \dots & D_{1K} \\ \dots & \dots & \dots & \dots \\ D_{M1} & D_{M2} & \dots & D_{MK} \end{bmatrix}$$

Where each $D_{ik} (i = 0,1,...,M$ and $k = 1,2,...,K)$ is defined as

$$D_{ik} = \begin{Bmatrix} 1 & \text{if decision } k \text{ is to be made in state } i \\ 0 & \text{otherwise} \end{Bmatrix}$$

Therefore, each row in the matrix must contain a single 1 with the rest of the elements 0s.

3.2.1 A Linear Programming Formulation

The convenient decision variables (denoted here by y_{ik}) for a linear programming model are defined as follows [4]. For each $i = 0,1,...,M$ and $k = 1,2,...,K$, let y_{ik} be the steady-state unconditional probability that the system is in state i and decision k is made; i.e.

$$y_{ik} = P \{\text{state} = i \text{ and decision} = k\}.$$

Each y_{ik} is closely related to the corresponding D_{ik} since, from the rules of conditional probability,

$$y_{ik} = \pi_i D_{ik},$$

Where π_i is the steady state probability that the Markov chain is in state i. furthermore,

$$\pi_i = \sum_{k=1}^{K} y_{ik}$$

So that

$$D_{ik} = \frac{y_{ik}}{\pi_i} = \frac{y_{ik}}{\sum_{k=1}^{K} y_{ik}}$$

There exist three sets of constraints on y_{ik}:

1. $\sum_{i=0}^{M} \pi_i = 1$ so that $\sum_{i=0}^{M} \sum_{k=1}^{K} y_{ik} = 1.$

2. From results on steady state probabilities

$$\pi_j = \sum_{i=0}^{M} \pi_i p_{ij}$$

So that

$$\sum_{k=1}^{K} y_{ik} = \sum_{i=0}^{M} \sum_{k=1}^{K} y_{ik} p_{ij}(k), \quad \text{For} \quad j = 1, 2, ..., K.$$

3. $y_{ik} \geq 0$, For $i = 0, 1, ..., M$ and $k = 1, 2, ..., K$.

The long run expected average cost per unit time is given by

$$E(C) = \sum_{i=0}^{M} \sum_{k=1}^{K} \pi_i C_{ik} D_{ik} = \sum_{i=0}^{M} \sum_{k=1}^{K} C_{ik} y_{ik}.$$

Hence the linear programming model is to choose the y_{ik} so as to

Minimize $Z = \sum_{i=0}^{M} \sum_{k=1}^{K} C_{ik} y_{ik}$

Subject to the constraints

1. $\sum_{i=0}^{M} \sum_{k=1}^{K} y_{ik} = 1.$

2. $\sum_{k=1}^{K} y_{ik} - \sum_{i=0}^{M} \sum_{k=1}^{K} y_{ik} p_{ij}(k) = 0,$ for $j = 0, 1, ..., M$

3. $y_{ik} \geq 0$, For $i = 0, 1, ..., M$ and $k = 1, 2, ..., K$.

Thus this model has $M + 2$ functional constraints and $K(M + 1)$ decision variables. [Actually (2) provides one redundant constraint, so any one of these $M + 1$ constraints can be deleted.]

Because this is a linear programming model, it can be solved by the simplex method. Once the y_{ik} values are obtained, each D_{ik} is found from

$$D_{ik} = \frac{y_{ik}}{\pi_i} = \frac{y_{ik}}{\sum_{k=1}^{K} y_{ik}}$$

The key conclusion is that the optimal policy found by the simplex method is deterministic rather than randomized. Thus, allowing policies to be randomized does not help at all in improving the final policy. However, it serves an extremely useful role in this formulation by converting integer variables (the D_{ik}) to continuous variables so that linear programming (LP) can be used. [4]

3.3 Policy Improvement Algorithm for Finding Optimal Policies

Linear programming can be used to solve vastly larger problems, and software packages for the simplex method are ver widely available. [4]

The second popular method is namely a policy improvement algorithm. The key advantage of this method is that it tends to be very efficient, because it usually reaches an optimal policy in a relatively small number of iterations.

If the current state *i* of the system and the decision $d_i(R) = k$ when operating under policy *R*, two things occur. An (expected) cost C_{ik} is incurred that depends upon only the observed state of the system and the decision made. The system moves to state *j* at the next observed time period, with transition probability given by $p_{ij}(k)$. If; in fact, state *j* influences the cost that has been incurred, and then C_{ik} is calculated as follows. Let

$q_{ij}(k) =$ Expected cost incurred when the system is in state *i*, decision *k* is made and the system evolves to state *j* at the next observed time period.

Then, $$C_{ik} = \sum_{j=0}^{M} q_{ij}(k) p_{ij}(k).$$

Preliminaries

Referring to the description and notation for Markov decision processes, for any given policy R, it was shown that there exist values $g(R), v_0(R), v_1(R), ..., v_M(R)$ that satisfy

$$g(R) + v_i(R) = C_{ik} + \sum_{j=0}^{M} p_{ij}(k) v_j(R), \qquad i = 0, 1, 2, ..., M$$

$v_i^n(R)$ was denoted by the total expected cost of a system starting in state i (beginning the first observed time period) evolving for *n* time periods. It was shown that by the following recursive function

$$v_i^n(R) = C_{ik} + \sum_{j=0}^{M} p_{ij}(k) v_j^{n-1}(R), \qquad i = 0, 1, 2, ..., M.$$

where $v_i^1(R) = C_{ik}$ for all *i*. The expected average cost per unit time following any policy *R* can be expressed as, $g(R) = \sum_{i=0}^{M} \pi_i C_{ik}$, which is independent of the starting state *i*. [2, 3, 4]

3.3.1 The Policy Improvement Algorithm

The algorithm begins by choosing an arbitrary policy R_1. It then solves the system of equations to find the values of $g(R_1), v_0(R), v_1(R), ..., v_{M-1}(R)$ [with $v_M(R) = 0$]. This step is called *value determination*. A better policy, denoted by R_2, is then constructed. This step is called *policy improvement*. These two steps constitute an iteration of the algorithm. Using the new policy R_2, iteration is performed. These iterations continue until two successive iterations lead to identical policies, which signify that the optimal policy has been obtained.

3.3.2 Summary of the Policy Improvement Algorithm

Initialization: Choose an arbitrary initial trial policy R_1. Set $n=1$.

Iteration n:

Step 1: Value determination: For policy R_n, use $p_{ij}(k), C_{ik}$ and $v_M(R_n)=0$ to solve the system of $M+1$ equations

$$g(R_n)=C_{ik}+\sum_{j=0}^{M} p_{ij}(k)v_j(R_n)-v_i(R_n), \qquad i=0,1,2...M$$

for all $M+1$ unknown values of $g(R_1), v_0(R), v_1(R),..., v_{M-1}(R)$.

Step 2: Policy Improvement: Using the current values of $v_i(R_n)$ computed for policy R_n, find the alternative policy R_{n+1} such that, for each state *i*, $d_i(R_{n+1})=k$ is the decision that minimizes

$$C_{ik}+\sum_{j=0}^{M} p_{ij}(k)v_j(R_n)-v_i(R_n)$$

For each state *i*,

$$\underset{k=1,2,...k}{\text{Minimize}} \ [C_{ik}+\sum_{j=0}^{M} p_{ij}(k)v_j(R_n)-v_i(R_n)]$$

And then set $d_i(R_{n+1})$ equal to the minimizing value of *k*. This procedure defines a new policy R_{n+1}.

Optimality test: The current policy R_{n+1} is optimal if this policy is identical to policy R_n. If it is, stop. Otherwise, reset $n=n+1$ and perform iteration. [4,25]

Two key properties of this algorithm are

1. $g(R_{n+1}) \le g(R_n), \qquad n=1,2,...$
2. The algorithm terminates with an optimal policy in a finite number of iterations.

3.4 Discounted Cost Criterion

This measure uses a discount factor α, where $0 < \alpha < 1$. The discount factor α can be interpreted as equal to $1/(1+i)$, where i is the current interest rate per period. Thus, α is the present value of one unit of cost m periods in the future.

This discounted cost criterion becomes preferable to thee average cost criterion when the time periods for the Markov chain are sufficiently long that the time value of money should be taken into account in adding costs in future periods to the cost in the current period. Another advantage is that the discounted cost criterion can readily be adapted to dealing with a finite-period Markov decision process where the Markov chain will terminate after a certain number of periods.[2,3,4]

Both the policy improvement technique and the linear programming approach can be applied with minor adjustments as follows: For example in the policy improvement algorithm:

Let $V_i^n(R)$ be the expected total discounted cost;

$$V_i^n(R) = C_{ik} + \alpha \sum_{j=0}^{M} p_{ij}(k) V_j^{n-1}(R),$$

with $V_i^1(R) = C_{ik}$. As n approaches infinity, this recursive function converges to

$$V_i(R) = C_{ik} + \alpha \sum_{j=0}^{M} p_{ij}(k) V_j(R), \qquad i = 0,1,2,...,M$$

Where $V_i(R)$ can now be interpreted as the expected total discounted cost when the process starts in state i and continues indefinitely. The detailed formulations are found in [4,25].

3.4.1 Finite-Period Markov Decision Processes and the Method of Successive Approximations

This method is for quickly finding at least an approximation to an optimal policy. We have assumed that Markov Decision process will be indefinitely, and we have sought an optimal policy for such a process. The basic idea of the method of successive approximations is to instead find an optimal policy for the decisions to make in the first period when the process has only n time periods to go before termination, starting with $n=1$, then $n=2$ and so on. As n grows large, the corresponding optimal policies will converge to an optimal policy for the infinite period problem of interest. Thus, the policies obtained for $n=1,2,3....$ provide successive approximations that lead to the desired optimal policy.

In particular, for $i=0,1,...,M$, let

V_i^n = Expected total discounted cost of following an optimal policy given that process starts in state *i* and has only *n* periods to go.

V_i^n is obtained from the recursive relationship by the *principle of optimality* for dynamic programming,

$$V_i^n = \min_k \left\{ C_{ik} + \alpha \sum_{j=0}^{M} p_{ij}(k) V_j^{n-1} \right\}, \qquad i = 0,1,2,...,M$$

$$V_i^1 = \min_k \{C_{ik}\}, \qquad i = 0,1,2,...,M.$$

To minimizing value of *k* provides the optimal decision to make in the first period when the process starts in state *i*.

Although the method of successive approximations may not lead to an optimal policy for the infinite-period problem after only a few iterations, it has one distinct advantage over the policy improvement algorithm and linear programming techniques. It never requires solving a system of simultaneous equations, so each iteration can be performed simply and quickly.

Furthermore, if the Markov decision process actually does have just *n* periods to go, *n* iterations of this method definitely will lead to an optimal policy. (For an *n*–period problem, it is permissible to set $\alpha = 1$, that is, no discounting, in which case the objective is to minimize the total expected cost over *n* periods. [4])

4. STOCHASTIC PROCESSES AND ITO'S LEMMA

4.1 Stochastic Processes

A stochastic process is a variable that evolves over time in a way that is at least in part random. The temperature in downtown Boston is an example; its variation through time is partly deterministic (rising during the day and falling at night, and rising towards summer and falling towards Winter), and partly random and unpredictable. The price of IBM stock is another example; it fluctuates randomly, but over the long haul has had a positive expected rate of growth that compensated investors for risk in holding the stock.

Somewhat more formally, a stochastic process is defined by a probability law for the evolution x_t of a variable x over time t. Thus, for given times $t_1 < t_2 < t_3$, etc., the probability that the corresponding values x_1, x_2, x_3, etc. can be calculated, lie in some specified range, for example

$$\text{prob}\,(a_1 < x_1 \le b_1, a_2 < x_2 \le b_2, ...)$$

When time t_1 arrives and we observe the actual value x_1, we can condition the probability of future events on this information.

In continuous-time stochastic process, the time index t is a continuous variable. (Even though we might only measure the temperature or stock price at particular points in time, these variables vary continuously through time.) However, the variables in discrete-time processes can change only at discrete points in time. One of the simplest examples of a stochastic process is the discrete-time discrete-state random walk. Here, x_t is a random variable that begins at a known value x_0, and at times $t = 1, 2, 3, ...,$ takes a jump of size 1 either up or down, each with probability ½. Since the jumps independent of each other, we can describe the dynamics of x_t with the following equation:

$$x_t = x_{t-1} + \in_t ,$$

where $\in_t$ is a random variable with probability distribution

$$\text{prob}(\in_t = 1) = \text{prob}(\in_t = -1) = \frac{1}{2} \quad (t = 1, 2, 3, \ldots)$$

We call x_t a discrete-state process because it can only take on discrete values. For example, set $x_0 = 0$. Then for odd values of *t*, possible values of x_t are $(-t, \ldots, -1, 1, \ldots, t)$, and for even values of *t*, possible values of x_t are $(-t, \ldots, -2, 0, 2, \ldots, t)$. The probability distribution for x_t is found from the binomial distribution. For *t* steps, the probability that there are *n* downward jumps and $t - n$ upward jumps is

$$\binom{t}{n} 2^{-t} \quad \textbf{(4.1)}$$

Therefore, the probability that x_t will take on the value *t-2n* at time *t* is

$$\text{prob}(x_t = t - 2n) = \binom{t}{n} 2^{-t} \quad \textbf{(4.2)}$$

We will use this probability distribution in the Wiener process as the continuous limit of the discrete-time random walk. At this point, however, note that the range of possible values that x_t can take on increases with *t*, as does the variance of x_t. Hence x_t is a nonstationary process.

Because the probability of an upward or downward jump is ½, at time $t = 0$ the expected value of x_t is 0 for all *t*. (Likewise, at time *t*, the expected value of x_t for $T > t$ is x_t.) One way to generalize this process is by changing the probabilities for an upward or downward jump. Let *p* be the probability of an upward jump and $q = (1 - p)$ the probability of a downward jump, with $p < q$. Now we have a random walk with drift; at time $t = 0$, the expected value of x_t for $t > 0$ is greater than zero, and is increasing with *t*.

The random walk (with discrete or continuous states, and with drift or without) satisfies the *Markov property*, and is therefore called *Markov process*. This property is that the probability distribution for x_{t+1} depends only on x_t, and not depend on what happened before time *t*. For example, in the case of the simple random walk given by equation (1), if $x_t = 6$, then x_{t+1} can equal 5 or 7, each with probability ½. The values of x_{t-1}, x_{t-2}, etc. are irrelevant once we know x_t. The *Markov property* is important because it can greatly simplify the analysis of a stochastic process. [1]

4.2 The Wiener Process

A Wiener process – also called a Brownian motion – is a continuous-time stochastic process with three important properties. *First*, it is a *Markov process*. This means that the probability distribution for all future values of the process depends only on its current value, and is unaffected by past values of he process or by any other current information. As a result, the current value of the process is all one needs to make a best forecast of its future value. *Second*, the Wiener process has *independent increments*. This means that the probability distribution for the change in the process over any time interval is independent of any other (no overlapping) time interval. *Third*, changes in the process over any finite interval of time are *normally distributed*, with a variance that increases linearly with the time interval.

The *Markov property* is particularly important. Again, it implies that only current information is useful for forecasting the future path of the process. Stock prices are often modeled as Markov processes, on the grounds that public information is quickly incorporated in the current price of the stock, so that the past pattern of prices has no forecasting value. (This is called the weak form of market efficiency. If it didn't hold, investors could in principle "beat the market" through technical analysis, that is, by using the past pattern of prices to forecast the future.) The fact that a Wiener process has independent increments means that we think of it as a continuous-time version of a random walk.

The three conditions discussed above the Markov property, independent increments, and changes that are normally distributed – may seem quite restrictive, and might suggest that there are very few real-world variables that can be realistically modeled with Wiener processes. For example, while it probably seems reasonable that stock prices satisfy the Markov property and have independent increments, it is not reasonable to assume that price changes are normally distributed; after all, we know that the price of a stock can never fall below zero. It is more reasonable to assume that changes in stock prices are log normally distributed, that is, that changes in the logarithm of the price as a Wiener process, rather than the price itself. Through the use of suitable transformations, the Wiener process can be used as a building block to model an extremely broad range of variables that vary continuously (or almost continuously) and stochastically through time.

It is useful to rotate the properties of a Wiener process somewhat more formally. If $z(t)$ is a Wiener process, then any change in $z, \Delta z,$ corresponding to a time interval Δt, satisfies the following conditions:

1. The relationship between Δz and Δt is given by

   $$\Delta z = \in_t \sqrt{\Delta t}\,,$$

   Where $\in_t$ is a normally distributed random variable with a mean of zero and a standard deviation of 1.
2. The random variable $\in_t$ is serially uncorrelated, that is, $\varepsilon\left[\in_t \in_s\right] = 0$ for $t \neq s.$ Thus the values of Δz for any two different intervals of time are independent. [Thus $z(t)$ follows a Markov process with independent increments.]

Let examine what these two conditions imply for the change in z over some finite interval of time T. We can break this interval up into n units of length Δt each, with $n = T / \Delta t$. Then the changes in z over this interval is given by

$$z(s+T) - z(s) = \sum_{i=1}^{n} \in_t \sqrt{\Delta t} \qquad \textbf{(4.3)}$$

The $\in_i{}'s$ are independent of each other. Therefore we can apply the Central Limit Theorem to their sum, and say that the change $z(s+T) - z(s)$ is normally distributed with mean zero and variance $n\Delta t = T$. This last point, which follows from the fact that Δz depends on $\sqrt{\Delta t}$ and not on $\Delta t,$ is particularly important; the variance of the change in a Wiener process grows linearly with the item horizon.

Also note that the Wiener process is nonstationary. Over the long run its variance will go to infinity. By letting Δt become infinitesimally small, we can represent the increment of a Wiener process, *dz*, in continuous time as

$$dz = \in_t \sqrt{dt} \quad \textbf{(4.4)}$$

Since $\in_t$ has zero mean and unit standard deviation, $\varepsilon(dz) = 0$, and $V(dz) = \varepsilon\left[(dz)^2\right] = dt$. Note, however, that a Wiener process has no time derivative in a conventional sense; $\Delta z / \Delta t = \varepsilon_t (\Delta t)^{-1/2}$ which becomes infinite as Δt approaches zero.

At times we may want to work with two or more Wiener processes, and we will be interested in their covariances. Suppose that $z_1(t)$ and $z_2(t)$ are Wiener processes. Then we can write $\varepsilon(dz_1\, dz_2) = \rho_{12}\, dt$, where ρ_{12} is the coefficient of correlation between the two processes. Because a Wiener process has a variance and standard deviation per unit of time equal to 1 $(\varepsilon\left[(dz)^2\right] / dt = 1)$, ρ_{12} is also the covariance per unit of time for the two processes. [1]

4.2.1 Brownian motion with Drift

The Wiener process can easily be generalized into more complex processes. The simplest generalization of equation (4.4) is the Brownian motion with drift:

$$dx = \alpha\, dt + \sigma\, dz \quad \textbf{(4.5)}$$

where *dz* is the increment of a Wiener process as defined above. In equation (4.5), α is called the drift parameter, and σ the variance parameter. Note that over any time interval Δt, the change in *x*, denoted by Δx, is normally distributed, and has expected value $\varepsilon(\Delta x) = \alpha\, \Delta t$ and variance $V(\Delta x) = \sigma^2 \Delta t$. [1]

4.2.1.1 Random Walk Representation of Brownian motion

In this part it is shown how equation (4.5) can be derived as the continuous limit of a discrete-time random walk. To do this, we will divide time up into discrete periods of length Δt, and we will assume that in each period the variable *x* either moves up or down by an amount Δh. Let the probability that it moves up be *p,* and the probability that it moves down be $q = 1 - p$. Figure 4.1 shows the possible values of *x* in each of three periods, assuming it begins at the point x_0. For each possible combination of *t* and *x*, the probability of it being reached is also shown. Note that from each period to the next, Δx is a random variable that can take on the values $\pm\Delta h$. Also note that x follows a Markov process with independent increments – the probability distribution for its future value depends only on where it is now, and the probability that it will move up or down in each period is independent of what happened in previous periods.

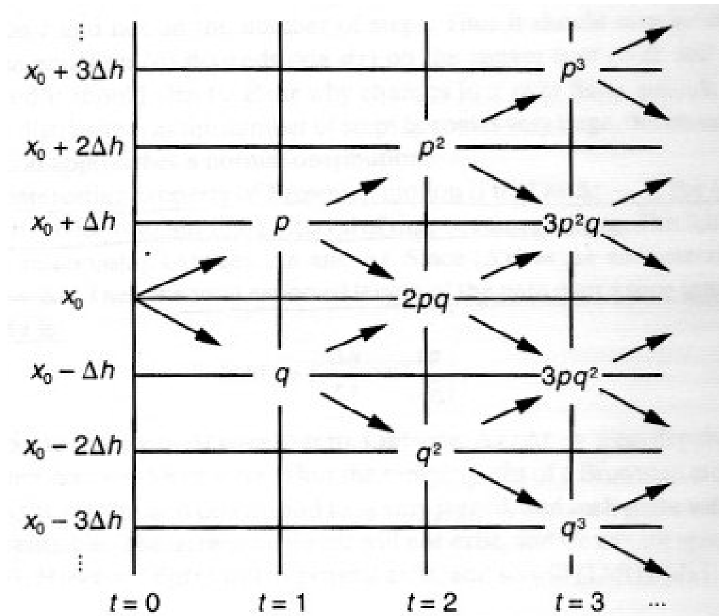

Figure 4.1: Random Walk Representation of Brownian motion (From [1], page 68)

Let us examine the distribution for future values of x. First, observe that the mean of Δx is $\varepsilon[\Delta x] = (p-q)\Delta h.$ Thus the variance of Δx is

$$V[\Delta x] = \varepsilon\left[(\Delta x)^2\right] - (\varepsilon[\Delta x]^2) = \left[1-(p-q)^2\right](\Delta h)^2 = 4pq(\Delta h)^2 \qquad \textbf{(4.6)}$$

A time interval of length t has $n = t/\Delta t$ discrete steps. Since the successive steps of the random walk are independent, the cumulated change $(x_t - x_0)$ is a binomial random variable with mean

$$n(p-q)\Delta h = t(p-q)\Delta h/\Delta t,$$

and the variance

$$n\left[1-(p-q)^2\right](\Delta h)^2 = 4pqt(\Delta h)^2/\Delta t.$$

So far the probabilities p and q and the increments Δh and Δt have been chosen arbitrarily, and shortly we will want to let Δt go to zero. As it does, we would like the mean and variance of $(x_t - x_0)$ to remain unchanged and to be independent of the particular choice of p, q, Δh and Δt. In addition, we would like to reach equation (4.5) in the limit. We can ensure that this will indeed be the case by setting

$$\Delta h = \sigma \sqrt{\Delta t} \qquad \textbf{(4.7)}$$

And

$$p = \frac{1}{2}\left[1 + (\alpha / \sigma)\sqrt{\Delta t}\right], \qquad q = \frac{1}{2}\left[1 - (\alpha / \sigma)\sqrt{\Delta t}\right] \qquad \textbf{(4.8)}$$

then $$p - q = \frac{\alpha}{\sigma}\sqrt{\Delta t} = \frac{\alpha}{\sigma^2}\Delta h$$

Substitute these expressions for Δh and $p - q$ into the formulas above, when the number of step, n, goes to infinity, and the binomial distribution converges to a normal distribution, with mean

$$t\frac{\alpha}{\sigma^2}\Delta h \ (\Delta h / \Delta t) = \alpha t$$

and variance

$$t\left[1 - (\alpha / \sigma^2)\Delta t\right](\sigma^2 \Delta t)/\Delta t \to \sigma^2 t .$$

These are exactly the values we need for Brownian motion; α is the drift; and σ^2 the variance, per unit time. In the limit as $\Delta t \to 0$, both the mean and variance of $(x_t - x_0)$ are independent of Δh and Δt.

We see, then, that Brownian motion is the limit of random walk, when the time interval and step length go to zero together while preserving the relationship of equation (4.7).

An interesting property of Brownian motion is that as $\Delta t \to 0$, the total distance traveled over any finite interval of time becomes infinite. [1]

4.3 Generalized Brownian motion – Ito Processes

The Wiener process can serve as a building block to model a broad range of stochastic variables. A number of examples will be examined, all of which are special cases of the following generalization of the simple Brownian motion with drift that are studied in the previous section:

$$dx = a(x,t)\,dt + b(x,t)\,dz \qquad \textbf{(4.9)}$$

where, again, dz is the increment of a Wiener process, and $a(x,t)$ and $b(x,t)$ are known (nonrandom) functions. The new feature is that the drift and variance coefficients are functions of the current state and time. The continuous-time stochastic process $x(t)$ represented by equation (4.9) is called an *Ito's process*.[1]

4.4 Ito's Lemma

We have seen that in the previous section the Ito process of equation (4.9) is continuous in time, but is not differentiable. However, we will often need to work with functions of Ito processes, and we will want to the differentials of such functions. For example, we might describe the value of an option to invest in a copper mine as a function of the price of copper, which in turn might be represented by a geometric Brownian motion. In this case, we would want to determine the stochastic process that the value of the option values. To do this, and in general to differentiate or integrate functions of Ito processes, we will need to make use of *Ito's Lemma.*

Ito's Lemma is easiest to understand as a Taylor series expansion. Suppose that $x(t)$ follows the process of equation (4.9), and consider a function $F(x,t)$ that is at least twice differentiable in *x* and once in *t*. We would like to find the total differential of this function, dF. The usual rules of calculus define this differential in terms of first-order changes in *x* and *t*: [1]

$$dF = (\frac{\partial F}{\partial x})dx + (\frac{\partial F}{\partial t})dt \,.$$

5. DYNAMIC OPTIMIZATION UNDER UNCERTAINTY

Time plays a particularly important role for investment decisions. The payoffs to a firm's investment made today accrue as a stream over the future, and are affected by uncertainty as well as by other decisions that the firm or its rivals will make later. The firm must look ahead to all these developments when making its current decision. One aspect of this future is an opportunity to make the same decision later; therefore the option of postponement should be included in today's menu of choices. The mathematical techniques to model investment decisions must be capable of handling all these considerations.

In this section dynamic programming technique will be developed. Dynamic programming is a very general tool for dynamic optimization, and is particularly useful in treating uncertainty. It breaks a whole sequence of decisions into just two components; the immediate decision, and a valuation function that encapsulates the consequences of all subsequent decisions, starting with the position that results from the immediate decision. If the planning horizon is finite, the very last decision at its end has nothing following it, and can therefore be found using standard static optimization methods. This solution then provides the valuation function appropriate to the penultimate decision. That, in turn, serves for the decision two stages from the end, and so on. One can work backwards all the way to the initial condition. This sequence of computations might seem difficult, but advances in computing hardware and software have made it quite feasible. If the planning horizon is infinite, what might seem like an even more difficult calculation is simplified by its recursive nature: each decision leads to another problem that looks exactly like the original one. This not only facilitates numerical computation, but also often makes it possible to obtain a theoretical characterization of the solution, and sometimes an analytical solution itself. [1,25]

5.1 Dynamic Programming

The two period example

Dynamic programming is in essence a systematic method of making comparisons such as the present value that result from the immediate investment and from waiting for more general dynamic decisions.

Consider a firm that is trying to decide whether to invest in a widget factory. The investment is completely irreversible – the factory can only be used to make widgets, or should the market for widgets evaporate, the firm can not *"uninvest"* and recover its expenditure.

Let I denote the sunk cost of investment in the factory that then produces one widget per period forever, and r be the interest rate. Suppose the price of a widget in the current period 0 is P_0. From period 1 onward, it will be $(1+u)P_0$ with probability q, and $(1-d)P_0$ with probability $(1-q)$.

First suppose that the investment opportunity is available only in period 0; if the firm decides not to invest in period 0, it can not change its mind in period 1. Let V_0 denote the expected present value of the revenues the firm gets if it invests. Weighting the two alternative possibilities for widget prices by their respective probabilities, discounting, and adding, we have

$$
\begin{aligned}
V_0 &= P_0 + \left[q(1+u)P_0 + (1-q)(1-d)P_0\right]\left[\frac{1}{1+r} + \frac{1}{(1+r)^2} + \ldots\right] \\
&= P_0 + \left[1 + q(u+d) - d\right]P_0\left(\frac{1/(1+r)}{1-1/(1+r)}\right) \\
&= P_0\left[1 + r + q(u+d) - d\right]/r
\end{aligned}
$$

(Note that we need $r > 0$ for convergence of the sum.) If $V_0 > I$, the investment is made and the firm gets $V_0 - I$; if $V_0 < I$, the investment is not made and the firm gets 0; if $V_0 = I$, the firm is indifferent between investing and not investing and gets zero in either case. Let Ω_0 denote the net payoff of the project to the firm, if it is forced in period 0 to decide whether to invest, on a now-or-never basis. Thus we have shown that

$$\Omega_0 = \max\left[V_0 - I, 0\right] \qquad \textbf{(5.1)}$$

Now consider the actual situation, where the investment opportunity remains available in future periods. Here the period 0 decision involves a different trade-off; invest now, or wait and do what is best when period 1 arrives. To asses this, the firm must look ahead to its own actions in different future eventualities. From period 1 onward the conditions will not change, so there is no point postponing any profitable projects beyond period 1. Hence we need look ahead only as far as period 1.

Suppose the firm does not invest in period 0, but instead waits. In period 1 the price will be

$$P_1 = \begin{cases} (1+u)P_0 & \text{with probability } q \\ (1-d)P_0 & \text{with probability } 1-q \end{cases}$$

It will stay at this level for periods 2, 3…. The present value of this stream of revenues, discounted back to period 1, is

$$\begin{aligned} V_1 &= P_1 + P_1/(1+r) + P_1/(1+r)^2 + ... \\ &= P_1(1+r)/r \end{aligned}$$

For each of the two possibilities (the price going up or down between periods 0 and 1), the firm will invest if $V_1 > I$, realizing a net payoff

$$F_1 = \max\left[V_1 - I, 0\right]$$

This outcome of the future decisions is sometimes called the *continuation value*. From the perspective of period 0, the period 1 price P_1, and therefore the values V_1 and F_1, are all random variables. Let ε_0 denote the expectation (probability-weighted average) calculated using the information available at period 0. Then we have

$$\varepsilon_0\left[F_1\right] = q \max\left[(1+u)P_0\frac{1+r}{r} - I, 0\right] + (1-q)\max\left[(1-d)P_0\frac{1+r}{r} - I, 0\right] \qquad \textbf{(5.2)}$$

This could be called the expected continuation value, or just the continuation value, with the expectation being understood.

Now return to the decision at period 0. The firm has two choices. If it invests immediately, it gets the expected present value of the revenues minus the cost of the investment, $V_0 - I$. If it does not, it gets the continuation value $\varepsilon_0\left[F_1\right]$ derived above, but that starts in period 1 and must be discounted by the factor $1/(1+r)$ to express it in period 0 units. The optimal choice is obviously the one that yields the larger value. Therefore the net present value of the whole investment opportunity optimally deployed, which we denote by F_0, is

$$F_0 = \max\left\{V_0 - I, \frac{1}{1+r}\varepsilon_0\left[F_1\right]\right\} \qquad \textbf{(5.3)}$$

The firm's optimal decision is the one that maximizes this net present value.

This captures the essential idea of dynamic programming. We split the whole sequence of decisions into two parts: the immediate choice, and the remaining decisions, all of whose effects are summarized in the continuation value. To find the optimal sequence of decisions we work backward. At the last relevant decision point we can make the best choice and thereby find the continuation value (F_1). Then at the decision point before that one, we know the expected continuation value and therefore can optimize the current choice. In this example there were just two periods. When there are more than two periods, the same procedure applies repeatedly.

The decision where the investment opportunity remains available at period 1 is less constrained than the one where it must be made on a now-or-never basis in period 0. Equation (5.1) shows the net payoff Ω_0 for this latter case; since that situation terminates the decision process at time 0, let us call it the *termination value* at time 0. Now we have the net worth F_0 of the less constrained decision problem from equation (5.3). The difference $(F_0 - \Omega_0)$ is just the value of the extra freedom, namely the *option to postpone* the decision.

To get a better idea of the factors that affect the value of the option to postpone, let us examine more closely the sources of the differences between F_0 and Ω_0. First, by postponing the decision the firm gives up the period 0 revenue P_0. This difference favors immediate action. Second, postponing the decision also means postponing the cost of investment; this favors waiting since the interest rate is positive. (More generally, the cost of investment could itself be changing over time, and that would bring new considerations; for example, if the firm expects capital equipment to get cheaper over time, this is an additional reason for waiting.) Third, and most important, waiting allows a separate optimization in each of the contingencies of a price rise and a price fall, whereas immediate action must be based on the average of the two. This ability to tailor action to contingency, specifically to refrain from investment if the price goes down, gives value to the extra freedom to wait. [1]

Many periods

In this subsection the theory of dynamic programming is developed in a setting where uncertainty is modeled using discrete-time Markov processes. Some general properties are easier to demonstrate in this format. Diffusion properties are Markov processes can be regarded as limits of random walks in discrete time as the length of each time period and of each step of the walk both become small in a suitable way.

With our application to investment in mind, we will refer to the decisions of a firm, but the theory is of course perfectly general. The firm's current status as it affects its operation and expansion opportunities is described by a state variable *x.* For simplicity of exposition we take this to be a scalar (real number), but the theory extends readily to vector states of any dimension. At any date or period *t*, the current value of this variable x_t is known, but future values $x_{t+1}, x_{t+2}, \ldots$ are random variables. We supposes that the process is Markov, that is all the information relevant to the determination of the probability distribution of future values is summarized in the current state x_t.

At each period *t*, some choices are available to the firm, and we represent them by the control variables *u*. In the above example where the only choice was whether to invest at once or wait, we could let *u* be a scalar binary variable, whose value 0 represents waiting and 1 represents investing at once. In other applications, for example, if the scale of investment is a matter of choice, *u* can be a continuous variable. If the firm has choices in addition to those bearing on investment, for example, hiring labor at time *t*, then *u* can be a vector. The value u_t of the control at time *t* must be chosen using only the information that is available at that time, namely x_t.

The state and the control at time *t* affect the firm's immediate profit flow, which is denoted by $\pi_t(x_t, u_t)$. Here the relevant control variable u_t might be the quantity of labor hired or raw materials purchased. The x_t and u_t of period *t* also affect the probability distribution of future states. Here u_t can be the amount of investment or

R&D, or even a decision to abandon the enterprise. Let $\Phi_t(x_{t+1} \mid x_t, u_t)$ denote the cumulative probability distribution function of the state next period conditional upon the current information (state and control variables).

The discount factor between any two periods is $1/(1+\rho)$ where ρ is the discount factor. The aim is to choose the sequence of controls $\{u_t\}$ over time so as to maximize the expected net present value of the payoffs. Sometimes we will force the decision process to end at some period *T*, with a final payoff that depends on the state reached; this termination payoff function is denoted by $\Omega_t(x_t)$.

Now the basic dynamic programming technique can be applied. Remember the idea is to split the decision sequence into two parts, the immediate period and the whole continuation beyond that. Suppose the current date is *t* and the state is x_t. Let the outcome is denoted by $F_t(x_t)$ – the expected net present value of the firm's cash flows – when the firm makes all decisions optimally from this point onwards.

When the firm chooses the control variables u_t, it gets an immediate profit flow $\pi_t(x_t, u_t)$. At the next period $(t+1)$, the state will be x_{t+1}. Optimal decisions thereafter will yield, in the notation it is $F_{t+1}(x_{t+1})$. This is random from the perspective of period *t*, so the expected value must be taken, $\varepsilon_t\left[F_{t+1}(x_{t+1})\right]$. That is called the continuation value. Discounting back to period *t,* the sum of the immediate payoff and the continuation value is

$$\pi_t(x_t, u_t) + \frac{1}{1+\rho}\varepsilon_t\left[F_{t+1}(x_{t+1})\right] .$$

The firm will choose u_t to maximize this, and the result will be just the value $F_t(x_t)$. Thus

$$F_t(x_t) = \max_{u_t}\left\{\pi_t(x_t, u_t) + \frac{1}{1+\rho}\varepsilon_t\left[F_{t+1}(x_{t+1})\right]\right\} \qquad \textbf{(5.4)}$$

The idea behind this decomposition is formally stated in *Bellman's Principle of optimality*: "*An optimal policy has the property that, whatever the initial action, the remaining choices constitute an optimal policy with respect to the subproblem* *starting at the state that results from the initial actions"*. Here the optimality of the remaining choices $u_{t+1}, u_{t+2}, \ldots$ etc., is subsumed in the continuation value, so only the immediate control u_t remains to be chosen optimally.

The result of this decomposition, namely equation (5.4), is called the *Bellman Equation*, or the fundamental equation of optimality. To reiterate, the first term on the right-hand side is the immediate profit, the second term constitutes the continuation value, and the optimum action this period is the one that maximizes the sum of these two components.

In the two period example, immediate investment gave $V_0 - I$, waiting had no period-0 payout but only a discounted continuation value $\varepsilon_0 \left[F_1 \right] / (1+r)$, and the optimal binary choice between these alternatives yielded the larger of these two. Thus the earlier equation (5.3) is a special case of the general *Bellman Equation* (5.4).

If the many-period problem has a fixed finite time horizon *T*, we can start at the end and work backward similarly. At the end of the horizon the firm gets a termination payoff $\Omega_t(x_t)$. Then the period before,

$$F_{T-1}(x_{T-1}) = \max_{u_{T-1}} \left\{ \pi(x_{T-1}, u_{T-1}) + \frac{1}{1+\rho} \varepsilon_{T-1} \left[F_T(x_T) \right] \right\}$$

Thus we know the value function at $T-1$. That in turn allows us to solve the maximization problem for u_{T-2}, leading to the value function $F_{T-2}(x_{T-2})$, and so on. At one time this was thought to be too complex a procedure to be practicable, and all kinds of indirect methods were devised. However, advances in computing have made the backward calculation remarkably usable. [1]

Infinite horizon

If there is no fixed time horizon for the decision problem, there is no known final value function from which we can work backward. Instead, the problem gets a recursive structure that facilitates theoretical analysis as well as numerical computation. The crucial simplification that an infinite horizon brings to equation (5.4) is independence from time *t* as much. Of course the current state x_t matters, but the calendar date t by itself has no effect. This works provided the flow profit function π, the transition probability distribution function Φ, and the discount rate ρ are themselves all independent of the actual label of the date, a condition that is satisfied or assumed in many economic applications.

In this setting, the problem one period hence looks exactly like the problem now, except of course for the new starting state. Therefore the value function is common to all periods, although of course it will be evaluated at different points x_t. Therefore the function $F(x_t)$ is written without any time label on the function symbol. The Bellman equation for any t becomes

$$F(x_t) = \max_{u_t} \left\{ \pi(x_t, u_t) + \frac{1}{1+\rho} \varepsilon_t [F(x_{t+1})] \right\}$$

Since x_t and x_{t+1} could be any of the possible states, they are written in general form as x and x'. Then for all x we get

$$F(x) = \max_{u} \left\{ \pi(x, u) + \frac{1}{1+\rho} \varepsilon [F(x') \mid x, u] \right\} \qquad \textbf{(5.5)}$$

where the expectation has been denoted as conditioned on the knowledge of the current period's x and u. This is the Bellman equation for the infinitely repeating, or recursive, dynamic programming problem.

Now that there is no fixed terminal date which to work backward, they are seemed to have lost an explicit or constructive way to find the value function F, we cannot find the optimal control u by solving the maximization problem on the right-hand side of the Bellman equation. Thus we need assurance that a solution actually exists, and a way to find it.

The recursive Bellman equation (5.5) can be thought as a whole list of equations, one for each possible value of x , with a whole list of unknowns, namely all the values $F(x)$. If x took on only a finite number of discrete values x_i, this would be a simultaneous system with exactly as many equations as the number of unknowns $F(x_i)$. More generally, we can regard (5.5) as a functional equation, with the whole function F as its unknown.

Despite superficial appearances, this equation is not linear. The optimal choice of u depends on all the values $F(x')$ that appear, weighted by the appropriate probabilities, in the expectation on the right-hand side. When this optimal control is substituted back, the result can be nonlinear in the $F(x')$ values.

This takes the form of an iterative procedure. Start with any guess for the true value function, say $F^{(1)}(x)$. Use it on the right-hand side of equation (5.5) and find the optimal choice rule u^1 , which can now be expressed as a function of x alone. Substituting it back, the right hand side becomes value function of x; call it $F^{(2)}(x)$. Now use it as the next guess of the true value function, and repeat the procedure. Then the successive guesses $F^{(3)}(x)$, $F^{(4)}(x)$, etc., will converge to the true function. Convergence is guaranteed no matter how bad the initial guess, but of course with a good initial guess the procedure will reach the desired accuracy of the approximation in fewer steps.

The key lies in the factor $1/(1+\rho)$ on the right-hand side. This being less than 1, it scales down, or contracts, any errors in the guess from one step to the next. As long as the profit flows are bounded, any errors in the choice of u cannot blow up. Gradually, only the correct solution is left. This method is increasingly used in many applications, and even in econometric work. [1]

6. A FIRM'S DECISIONS

6.1 Investment Opportunities and Investment Timing

In this chapter, investment decisions under uncertainty will be analyzed. The main concern with investment expenditures has two very important characteristics. First, the expenditures are at least partly irreversible; in other words, sunk costs that cannot be recovered. Second, these investments can be delayed, so that the firm has the opportunity to wait for new information to arrive about prices, costs, and other market conditions before it commits resources.

The ability to delay an irreversible investment expenditure can profoundly affect the decision to invest. In particular, it invalidates the simple net present value rule as it is commonly taught to students in business schools: "Invest in a project when the present value of its expected cash flows is at least as large as its cost." This rule is incorrect because it ignores the opportunity cost of making a commitment now, and thereby giving up the option of waiting for new information. As I mentioned before, that opportunity cost must be included as part of the total cost of investing. In this section, this opportunity cost and its implications for investment at a greater level of generality will be examined.

Firstly the most basic continuous-time models of irreversible investment will be set. In this model, which was originally developed by McDonald and Siegel (1986), a firm must decide when to invest in a single project. The cost of the investment, I, is known and fixed, but the value of the project, V, follows a geometric Brownian motion. The simple net present value rule is to invest as long as $V > I$, but as McDonald and Siegel demonstrated, this is incorrect. Because future values of V are unknown, there is an opportunity cost to investing today. Hence the optimal investment rule is to invest when V is at least as large as a critical value V^* that exceeds I. After describing the basic model in detail, they showed that how the optimal investment rule (that is, the critical value V^*) can be found by dynamic programming. Finally they extended the model by considering alternative stochastic processes for the value of the project, V. [1]

6.1.1 The Basic Model

McDonald and Siegel (1986) considered the following problem: At what point is it optimal to pay a sunk cost I in return for a project whose value is V, given that V evolves according to the following geometric Brownian motion:

$$dV = \alpha V\, dt + \sigma V\, dz \qquad \textbf{(6.1)}$$

where dz is the increment of a Wiener process. Equation (6.1) implies that the current value of the project is known, but future values are lognormally distributed with a variance that grows linearly with the time horizon. Thus although information arrives over time (the firm observes V changing), the future value of the project is always uncertain.

Equation (6.1) is clearly an abstraction from most real projects. For example, suppose the project is a widget factory with some capacity. If variable costs are positive and managers have the option to shut down the factory temporarily when the price of output is below variable cost, and or the option to abandon the project completely, V will not follow a geometric Brownian motion even if the price of widgets does. (Models in which the output price follows a geometric Brownian motion will be developed and the project can be temporarily shut down or abandoned.) If variable cost is positive and the managers do not have the option to shut down (perhaps because of regulatory constraints), V can become negative, which is again in conflict with the assumption of lognormality. In addition, one might believe that a competitive product market will prevent the price from wandering too far from long-run industry-wide marginal cost, or that stochastic changes in price are likely to be infrequent but large, so that V should follow a mean-reverting or jump process.

Note that the firm's investment opportunity is equivalent to a perpetual call option – the right but not the obligation to buy a share of stock at a prespecified price. Therefore the decision to invest is equivalent to deciding when to exercise such an option. Thus, the investment decision can be viewed as a problem of option valuation. Alternatively, it can be viewed as a problem in dynamic programming. They derived the optimal investment rule in two ways, first using dynamic programming, and then using option pricing (contingent claims) methods.

The value of the investment opportunity (that is, the value of the option to invest) will be denoted by $F(V)$. A rule is wanted that maximizes this value. Since the payoff from investing at time t is $V_t - I$, we want to maximize its expected present value:

$$F(V) = \max \varepsilon\left[\left(V_T - I\right)e^{-\rho T}\right] \qquad \textbf{(6.2)}$$

where ε denotes the expectation, T is the (unknown) future time that the investment is made, ρ is a discount rate, and the maximization is subject to equation (6.1) for V. For this problem to make sense, $\alpha < \rho$ must be assumed; otherwise the integral in equation (6.1) could be made indefinitely larger by choosing a larger T. Thus waiting longer would always be a better policy, and the optimum would not exist. The difference $\rho - \alpha$ will be denoted as δ; thus $\delta > 0$ are being assumed. [5]

6.1.1.1 The Deterministic Case

Although the investment decision is affected by uncertainty, it is useful to first examine the case in which there is no uncertainty, that is, σ in equation (6.1) is zero. So, there can still be a value to waiting.

If $\sigma = 0$, $V(t) = V_0 e^{\alpha t}$, where $V_0 = V(0)$. Thus given a current V, the value of the investment opportunity assuming we invest at some arbitrary future time T is

$$F(V) = (V e^{\alpha T} - I) e^{-\rho T} \qquad \textbf{(6.3)}$$

Suppose $\alpha \le 0$. Then $V(t)$ will remain constant or fall over time, so it is clearly optimal to invest immediately if $V > I$, and never invest otherwise. Hence, $F(V) = \max\left[V - I, 0\right]$.

What if $0 < \alpha < \rho$? Then $F(V) > 0$ even if currently $V < I$, because eventually V will exceed I. Also, even if V now exceeds I, it may be better to wait rather than to invest now. To see this, maximize $F(V)$ in equation (6.3) with respect to T. Then

$$T^* = \max\left\{\frac{1}{\alpha}\log\left[\frac{\rho I}{(\rho - \alpha)V}, 0\right]\right\} \qquad \textbf{(6.4)}$$

Note that if V is not too much larger than I, we will have $T^* > 0$. The reason for delaying the investment in this case is that in present value terms, the cost of the investment decreases by a factor of $e^{-(\rho-\alpha)T}$.

For what values of V is it optimal to invest immediately? By setting $T^* = 0$, we see that one should invest immediately if $V \geq V^*$ where

$$V^* = \frac{\rho}{\rho - \alpha} I > I \qquad \textbf{(6.5)}$$

Finally, by substituting expression (6.4) into equation (6.3), the following solution for $F(V)$ was obtained:

$$F(V) = \begin{cases} [\alpha I / (\rho - \alpha)][(\rho - \alpha) V / \rho I]^{\rho/\alpha} & \text{for } V \leq V^*, \\ V - I & \text{for } V > V^* \end{cases} \qquad \textbf{(6.6)}$$

Figure 6.1 shows $F(V)$ as a function of V for $I = 1$, $\rho = 0.10$ and $\alpha = 0, 0.03, 0.06$. Note that $F(V)$ increases when α increases, as does the critical value V^*. Growth in V creates a value to waiting, and increases the value of the investment opportunity. [1]

6.1.1.2 The Stochastic Case

Now, the case $\sigma > 0$ will be studied. The problem is to determine the point at which it is optimal to invest I in return for an asset worth V. Since V evolves stochastically, a time T as it was did before will not be able to determined. Instead, the investment rule will take the form of a critical value V^* such that it is optimal to invest once $V \geq V^*$.As we will see, a higher value of σ will result in a higher V^*, that is, a greater value to waiting. It is important to keep in mind, however, that in general both growth $(\alpha > 0)$ and uncertainty $(\sigma > 0)$ can create to waiting and thereby affect investment timing.

In the next part, this problem is solved by the dynamic programming.

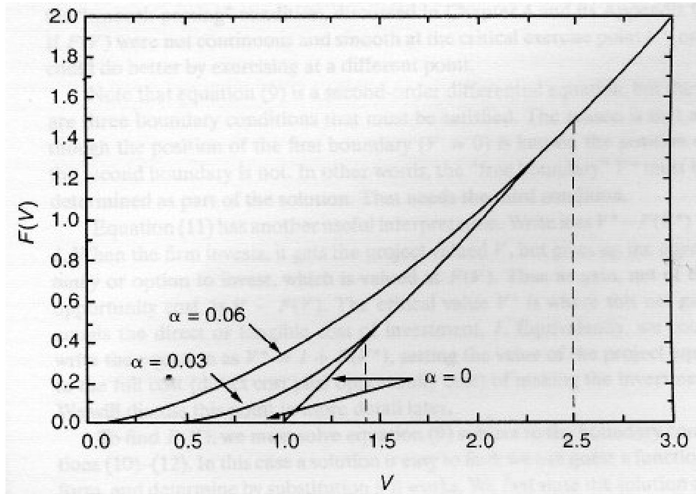

Figure 6.1: Value of Investment Opportunity, F(V), for $\sigma = 0, \rho = 0.1$(From [1], page 139)

6.1.2 Solution by Dynamic Programming

Because of the investment opportunity, $F(V)$, yields no cash flows up to time T that the investment is undertaken, the only return from holding it is its capital appreciation. Hence, in the continuation region (values of V for which it is not optimal to invest) the Bellman equation,

$$\rho F\, dt = \varepsilon(dF) \quad \textbf{(6.7)}$$

Equation (6.7) says that over a time interval *dt*, the total expected return on the investment opportunity, $\rho F\, dt$, is equal to its expected rate of capital appreciation.

When *dF* was expanded using Ito's Lemma, then the Bellman equation becomes (after dividing through by *dt*):

$$\frac{1}{2}\sigma^2 V^2 F''(V) + \alpha V\, F'(V) - \rho F = 0\ . \quad \textbf{(6.8)}$$

To ensure existence of an optimum (for reasons already explained in connection with the deterministic case), it was assume that $\alpha < \rho$ or $\delta > 0$. With this notation, the Bellman equation becomes the following equation that must be satisfied by $F(V)$:

$$\frac{1}{2}\sigma^2 V^2 F''(V) + (\rho - \delta) V\, F'(V) - \rho F = 0 \quad \textbf{(6.9)}$$

In addition, $F(V)$ must satisfy the following boundary conditions:

$$F(0)=0, \quad \textbf{(6.10)}$$

$$F(V^*)=V^*-I, \quad \textbf{(6.11)}$$

$$F'(V^*)=1. \quad \textbf{(6.12)}$$

Condition (6.10) arises from the observation that if *V* goes to zero, it will stay at zero. Therefore the option to invest will be of no value when $V=0$. The other two conditions come from consideration of optimal investment. V^* is the price at which it is optimal to invest, or the free boundary of the continuation region. Then (6.11) is the value-matching condition; it just says that upon investing, the firm receives a net payoff V^*-I. Finally, condition (6.12) is the smooth-pasting condition. If $F(V)$ were not continuous and smooth at the critical exercise point V^*, one could do better by exercising at a different point.

Equation (6.11) has another useful interpretation. Write it as $V^*-F(V)=I$. When the firm invests, it gets the project valued *V*, but gives up the opportunity or option to invest, which is valued at $F(V)$. Thus, its gain, net of the opportunity cost, is $V-F(V)$. The critical value V^* is where this net gain equals the direct or tangible cost of investment, *I*. Equivalently, the equation could be written as $V^*=I+F(V^*)$, setting the value of the project equal to the full cost (direct cost plus the opportunity cost) of making the investment.

To find $F(V)$, equation (6.9) must be solved subject to the boundary conditions (6.10)-(6.12). To satisfy the boundary condition (6.10), the solution must take the form

$$F(V)=A\,V^{\beta_1} \quad \textbf{(6.13)}$$

where *A* is a constant that is yet to be determined, and $\beta_1>1$ is a known constant whose value depends on the parameters σ,ρ and δ of the differential equation. Then we get

$$V^*=\frac{\beta_1}{\beta_1-1}I \quad \textbf{(6.14)}$$

and

$$A = (V^* - I)/(V^*)^{\beta_1} = (\beta_1 - 1)^{\beta_1 - 1} / \left[(\beta_1)^{\beta_1} I^{\beta_1 - 1} \right] \qquad \textbf{(6.15)}$$

Equations (6.13)-(6.15) give the value of the investment opportunity and the optimal investment rule, that is, the critical value V^* at which it is optimal to invest. For the time being, the most important point is that since $\beta_1 > 1$, we have $\beta_1/(\beta_1 - 1) > 1$ and $V^* > I$. Thus the simple NPV rule is incorrect; uncertainty and irreversibility drive a wedge between the critical value V^* and *I*. The size of the wedge is the factor $\beta_1/(\beta_1 - 1)$, and it becomes important to examine its magnitude for realistic values of the underlying parameters and its response to changes in these parameters.[1,5]

6.1.3 Characteristics of the Optimal Investment Rule

Since all parameters σ, δ, r are interdependent, it is important to be careful while the analysis is making. As mentioned in the past section, $\delta = \mu - \alpha = r + \phi\sigma\rho_{xm} - \alpha$. Hence, for example, an increase in the risk-free rate, *r*, is likely to result in the risk-adjusted expected return, μ, which, if the drift rate α is constant, implies an increase in δ. Likewise, an increase in σ is likely to be accompanied by an increase in μ, which again implies an increase in δ if α is constant. These interdependencies should be kept in mind when analyzing how a change in a market driven parameter (such as *r*) will affect the value of the investment opportunity and the optimal investment rule.

Another issue that should be kept in mind when performing comparative static experiments is that our model assumes that the parameters α, σ, etc. are fixed numbers. If α and σ are changing over time or in response to changes in the state variable *V* (either deterministically or stochastically) and the firm knows this, it should take into account when determining the optimal investment rule. For example, it may be that α and σ in equation (6.1) should be replaced with functions $\alpha(V,t)$ and $\sigma(V,t)$. This will complicate the problem considerably. [1]

6.2 The value of a project and the decision to invest

The basic model of irreversible investment demonstrated a close analogy between a firm's option and a financial call option. In the case of a call option, the price of the stock underlying the option was assumed to follow a stochastic process, usually a geometric Brownian motion. In the real investment model, the corresponding state variable was the value of the project, *V*, for which they stipulated a stochastic process.

At the above, *V* followed a stochastic process, and particularly a geometric Brownian motion, it is an abstraction from reality. First, if the project is a factory and there are variable costs of operation, *V* will not follow a geometric Brownian motion. Second and more important, the value of a project depends on future prices of outputs and inputs, interest rates, etc. These in turn can be explained in terms of the underlying demand and technology conditions in various markets. Hence fluctuations in *V* can be traced back to uncertainty in these more basic variables. To understand a firm's behavior, we might be satisfied to work with an exogenous process for the output and input prices. At the industry level, we must make the output price endogenous. At an even more general equilibrium level, the input prices must also be determined simultaneously by considering all industries' factor demands. [1, 5]

6.2.1 The Simplest Case: No Operating Costs

The firm's investment project, once completed, will produce a fixed flow of output forever. For convenience, the units so that the quantity of output from the project was chosen as equal to one unit per year. The inverse demand function giving price in terms of quantity Q was supposed as $P = Y\,D(Q)$, where Y is a stochastic shift variable. Here, since the variable cost of production was assumed to be zero, the firm's profit flow is just $P = Y\,D(1)$. Hence, without further loss of generality, P itself can be taken as the stochastic variable.

The simplest stochastic process for P was assumed as the geometric Brownian motion:

$$dP = \alpha\,P\,dt + \sigma\,P\,dz \qquad \textbf{(6.16)}$$

The profit flow is *P* in perpetuity, and its expected value grows at the trend rate α. If future values are discounted at the rate μ , then the expected present value *V* of the project when the current price is *P* is just given by $V = P/(\mu - \alpha)$. In this case, V, being a multiple of *P*, also follows a geometric Brownian motion with same parameters α and σ. Hence the investment problem reduced to the same in the past section. Here the work will be directly in terms of *P* to set the stage for the generalizations to come. [1]

Valuing the Project

If the project is a contingent or derivative asset, whose payoffs depend on the value of the more basic asset *P*, then the value of the project as a function $V(P)$ of the price of the basic asset can be derived.

A portfolio was constructed at time *t* that contains one unit of the project, and a short position of *n* units of output, where *n* was chosen to make the portfolio riskless. The holder of the project will get the revenue or profit flow $P\,dt$ over the time interval of length dt. Also, a holder of each unit of the short position must pay to the holder of the corresponding long position an amount equal to the dividend or convergence yield that the latter would have earned, namely, $\delta P\,dt$. Thus holding the portfolio yields a net dividend $(P - n\delta P)\,dt$.

It also yields a (stochastic) capital gain, which is equal to

$$dV - ndP = \left\{\alpha(P)\,P\left[V'(P) - n\right] + \frac{1}{2}\sigma(P)^2 P^2 V''(P)\right\}dt$$
$$+ P\left[V'(p) - n\right]\sigma(P)dz$$

When $n = V'(P)$ is chosen and the required processes are done, we get the fundamental component of value as:

$$V(P) = P/\delta\,. \qquad \textbf{(6.17)}$$

Valuing the Option to Invest

The value *V* of an installed project is a function of the current price *P* as we know. So, we can obtain the diffusion process of *V* from that of *P* by using Ito's lemma. To find the value of the option to invest as a function of the price, $F(P)$, using the above solution for $V(P)$ as the boundary condition that holds at the optimal exercise threshold. When we looked the behavior of $F(P)$ at P^*, at this threshold it becomes optimal to exercise the option. According to the value matching condition and smooth pasting condition,

$$P^* = \frac{\beta_1}{\beta_1 - 1}\delta I \quad \textbf{(6.18)}$$

Using the relation (6.17) the price threshold equivalently can be expressed in terms of a value threshold [1]

$$V^* = \frac{\beta_1}{\beta_1 - 1} I \, .$$

Dynamic Programming

If the risk in P cannot be spanned by existing assets, then we cannot construct a riskless portfolio and use it to obtain a differential equation for $V(P)$. As explained before, we can use instead dynamic programming with an exogenously specified discount rate ρ, although we will not be able to relate this discount rate to the riskless rate and the market price of risk using CAPM. [1]

The value of the project at time t can be expressed as the sum of the operating profit over the interval $(t,\, t+dt)$ and the continuous value beyond $t+dt$ .Thus

$$V(P) = P\,dt + \varepsilon\left[V(P+dP)\,e^{-\rho\,dt}\right].$$

Expanding the right-hand side using Ito's lemma, and when we take the limit as $dt \to 0$, we get the differential equation, $\frac{1}{2}\sigma^2 P^2 V''(P) + \alpha P V'(P) - \rho V(P) + P = 0$. When the equation solved, it is obtained $V(P) = P/(\rho - \alpha)$. For this, to make economic sense we need $\rho > \alpha$.

Then the option to invest can be analyzed similarly. Start with a P in the range $(0,\, P^*)$, where the option continues to be held. Finally when we use the value matching and smooth pasting between $F(P)$ and $V(P)$ at P^* to complete the project; the result is

$$P^* = \frac{\beta_1}{\beta_1 - 1}(\rho - \alpha) I.$$

6.2.2 Operating Costs and Temporary Suspension

Suppose once again that the output price follows the geometric Brownian motion of equation (6.1). Then α, σ, μ, and $\delta \equiv \mu - \alpha$ are all constants. If the option of investing in the project is ever going to be exercised, we need $\mu > \alpha$, or $\delta > 0$, and we will assume that this is indeed the case. We will also assume that operation of the project entails a flow cost *C*, but that the operation can be temporarily and costlessly suspended when *P* falls below *C*, and costlessly resumed later if *P* rises above *C*. Therefore, at any instant the profit flow from this project is given by

$$\pi(P) = \max\left[P - C, 0\right]. \quad \textbf{(6.19)}$$

McDonalds and Siegel pointed out another useful way to look at such a project. It gives the owner an infinite set of options. The option at time *t*, if exercised, means paying *C* to receive the *P* that prevails at that instant. Since each option can only be exercised at its specified instant, these are European call options. They also showed that the project can be valued by valuing each of these options and then summing these values by integrating over *t*. [1,5]

7. DYNAMIC PROGRAMMING APPLICATIONS

7.1 Investment Planning

In investment planning problems, the aim is usually to determine the optimal timing and sequence of capital investments.

In [6], they presented a method for determining the optimal timing and economic feasibility of a new 1945 km railway linking North and South Brazil. Possible new railway links, together with the existing road, rail and water transport system are modeled as a network. Shippers route their traffic over the network to minimize their cost, and the railway investor selects the sequence and timing of new links (if any) that maximize the present value of benefits to the investor. The problem was formulated as a mixed integer programming problem but then they showed that the problem can be formulated as nested dynamic programming models that can be easily implemented in a spreadsheet. The traffic assignment problem was implemented as a recursive model that was used to calculate the benefits for each possible system state. A second dynamic programming problem calculates the optimal expansion path for the system.

The investment model determined the optimal sequence and timing of capital investments in new rail lines in the North-Central corridor of Brazil, to maximize the present value of benefits. The benefit in each year is the annual revenue from traffic on the new rail lines minus the sum of the annual long run operating costs and the cost of any capital investments in new rail made in that year.

For each origin to destination traffic flow, the cheapest (shortest) path was found from origin to destination on the existing system. When new rail links were added to the system, there were cost savings for any particular origin to destination shipment if its shortest path included some of the new links. The new total OD cost across all shipments was lower, and the cost reduction compared to the existing system was the benefit that resulted from the new links. The total benefit was then simply this cost saving compared to the existing system considering all OD shipments.

In the first dynamic programming model, they selected the investment decisions in new rail links to maximize the present value of the net benefits. They had three decisions as do not build the link, build next link from the south and build next link from the north. And then an additional dynamic programming algorithm was used to calculate the shortest (minimum cost) path for each state and OD pair.

Moreover, the paper [7] extended the real options literature by discussing an investment problem, where a firm has to determine optimal investment timing and optimal capacity choice at the same time under conditions of irreversible investment expenditures and uncertainty in future demand. After the project is installed with a certain maximum capacity, this capacity is fixed as an upper boundary to the output and can not be adjusted later on. And for all decision, uncertainty in future demand leads to an increase in optimal installed capacity. But on the other hand, it causes investment to be delayed to an extent that even small uncertainty makes waiting and accumulation of further information the optimal decision for large ranges of demand. Limiting the capacity which may be installed weakens this extreme effect of uncertainty.

When a firm has the opportunity to invest in a project, its interest is to find the optimal investment strategy using the full freedom of choice that is restricted by many constraints. But there may be uncertainty in future values of several input quantities concerning this decision and in most cases the investment time is not fixed. Investment timing is one of the main instruments to optimize the firm's strategy.

When the firm decides to exercise its option to invest, it has to fix the capacity which will be installed. There is no possibility to adjust the capacity when uncertain parameters – like prices or demand – have changed to unexpected values. Firms face problems of this kind when capacity-adjustment of a ready built production facility is not possible and the installation of an additional project is out of discussion. Because you had the unique chance to commit the use of some natural resource, for example to build a hydrostorage plant. When the plant is completed and the water is dammed up to a certain level, the resource is committed for the lifetime of the plant. Another example is planning a hotel in the center of the city. The capacity choice corresponds to the determination of the number of rooms which shall be installed, so the maximum capacity is fixed. There is no possibility to add the capacity.

The standard problem of investment timing is: As long as the option to invest is alive, the firm has to decide either to keep it alive and wait or to exercise the option, that is to pay the investment cost and establish the project. But now the firm has to fix the size of the project and this will cause modification in the investment strategy. This paper discussed that uncertainty leads to higher values and higher marginal values of the project. And therefore increasing uncertainty will cause increasing size of the project. But the threshold up to which opportunity costs are positive increases fast with increasing uncertainty, so waiting becomes more valuable and investment is delayed to an extent which is not seen in the standard timing model.

In this paper, he considered a firm that has the option to invest in a production facility with maximum capacity *m*, where *m* has to be fixed during the conception of the project. The demand function is

$$P = \theta(t) - \delta q \qquad \textbf{(7.1)}$$

Where *q* is the output of the firm, *P* is the price which can be achieved for one unit of the output and $\delta = -dP/dq$ describes the dependence of the price on the output. $\theta(t)$ is the demand shift parameter which follows a stochastic process of the form

$$d\theta = \alpha\theta\, dt + \sigma\theta\, dz$$

$$\theta(0) = \theta_0 \geq 0, \text{geometric Brownian motion} \qquad \textbf{(7.2)}$$

where dz is the increment to a wiener process, α the expected relative drift of θ per unit of time and σ^2 the relative variance per unit of time. That means, the current value of the demand shift parameter is known but the future values are log-normally distributed and the variance is increasing. This stochastic process induces uncertainty and thus risk into the investment problem. He assumed the marginal production costs c' to be a function of the project's size but constant with respect to the output $(c' = c'(m))$. The profit flow π is,

$$\pi(\theta, m, q) = \left[P(\theta, q) - c'(m)\right]q, \qquad 0 \leq q \leq m \qquad \textbf{(7.3)}$$

The investment costs I that had to be paid for installation of a production facility are a function of the facility's maximum capacity $I = I(m)$. They were assumed to be sunk costs and

$$I(m) = bm^{\gamma}, \qquad \gamma \leq 1 \qquad \textbf{(7.4)}$$

So the marginal investment costs are decreasing with increasing installed capacity. The firm's task was then to observe the system and decide either to wait or to invest and fix the size of the project.

When the firm decides to exercise the option to invest and to install a project with capacity *m* it has to pay $I(m)$ and receives a project worth *V*. For this, it is necessary to know the value of the project which is a function of θ and installed capacity *m*. θ's current value is known and the future values are defined by (7.2). So the derivation of *V* follows the standard approach using dynamic programming. The firm was assumed to choose its output *q* to maximize *V* and the optimal *q* is found by maximizing the profit flow π at every point when the level of demand shift parameter θ has changed.

$$V(\theta, m) = \varepsilon \max \int \pi(\theta(t+\tau), m, q(t+\tau))\, e^{-r\tau} d\tau \qquad \textbf{(7.5)}$$

Where *r* is the discount rate $(r > \alpha > 0)$. After he got the boundary values of θ, he formed the ranges for θ and *V* was substituted by the following equation and the calculation of *V* would be done by the use of dynamic programming:

$$V(\theta, m) = \pi(\theta, m)\, dt + e^{-r\,dt} \varepsilon (V(\theta, m) + dV(\theta, m)) \qquad \textbf{(7.6)}$$

This leads (applying Ito's lemma) to the nonhomogeneous differential equation and according to this equation the particle solutions of the differential equations $(\bar{V}_j)$ were found for the ranges and the solution was written as follows:

$$V(\theta, m) \mid \theta \in R_j = V_j(\theta, m) = A_{j,1}(m)\, \theta\, \beta$$

After the firm decided to invest at a level θ of the demand shift parameter, the question that must be asked is which capacity should be installed. Because of this, the marginal value of the project was determined. By using different variances the figure of marginal value and θ was drawn and according to the figure , it was seen that the marginal value is increasing with increasing uncertainty and the marginal project is only utilized when $\theta > \theta_2$ (the second boundary value for θ, which is $c' + 2\delta m$).

In another paper [15], the weaknesses of the net present value criterion was considered. An attempt was made to modify the NPV criterion by incorporating the real opitons approach and itas application was demonstrated in a greenhouse construction investment plan. Moreover, in the paper [16], the optimal investment strategy of an investor was solved with preferences for wealth at some future date - the investment horizon and in paper [17] a mathematical model of sequential investment behaviour was presented under conditions of uncertainty. The model adresses the problem of an investor with access to a limited pool of capital. They evaluated how the optimal investment behaviour should change when changes occurred in the environment. And in paper [19], the model calculates an optimal investment plan for a highway corridor subject to budget constraints and the dynamic programming model was used to solve for the optimal expansion path for each link in the highway system.

7.2 Optimal Allocation

When given a resource, x, it is divided into two parts, y and $x-y$. From y, it is obtained a return of $g(y)$; from $x-y$ a return of $h(x-y)$. In so doing, they expended a certain amount of the original quantity and left with a new quantity, $a\,y+b(x-y)$, where $0<a,b<1$. This process was then continued. How does one allocate at each stage, so as to maximize the total return obtained over a finite or unbounded number of stages? [8]

If there was only one stage, the total return was $R_1(x,y)=g(y)+h(x-y)$ and if there was *N*-stage, the total return due to an initial allocation of *y* was described as follows:

Let $f_N(x) =$ total return obtained from *N*-stage process given an initial amount *x* and employing an optimal policy.

$$R_N(x,y)=g(y)+h(x-y)+f_{N-1}[ay+b(x-y)]$$

By definition;

$$f_N(x) = \max_{0 \le y \le x} R_N(x, y)$$
$$= \max_{0 \le y \le x} \{g(y) + h(x - y) + f_{N-1}[ay + b(x - y)]\}.$$

In paper [22], the method of dynamic programming was used for solving problems where several different types of limited resources are to be divided optimally between a great number of projects.

7.3 Exploitation of Natural Resources

It is well known that a number of optimality problems in investment analysis can be phrased in a dynamic programming framework, for example, optimal stopping problems, portfolio selection, and the exploitation of natural resources. In the paper [9], the last topic, the exploitation of natural resources, forms the main idea of the interest. Generally, such dynamic programming problems can be modeled in discrete time under uncertainty and with a finite horizon consisting of *N* steps. To be more specific, stationary and that the state space process was a k-dimensional Markov process would be assumed where the states were numbered by $i = 1, 2, 3, ..., k$. V_n was the optimal value function at stage *n*. Using the dynamic programming principle, V_n satisfied the recursive equation

$$V_n(i) = \max_d \left[C(i,d) + \beta \sum_{j=1}^{k} P_{ij}(d) V_{n+1}(j) \right] \quad \textbf{(7.7)}$$

Where the recursion goes from $n = N - 1, ..., 1$. Here $C(i,d)$ is the expected immediate return in state *i* using decision *d*, $P_{ij}(d)$ is the transition probability from state *i* to state *j*, and β is a discount factor. At stage *N*, they assumed

$$V_n(i) = \max_d \{0,\ C(i,d)\}$$

now they assumed that the optimal policy can be determined from (7.7) using standard algorithms. But there were a number of problems when it came to practical applications to investment analysis. One problem was that the state space may be continuous, but a manageable state space could be obtained as an approximation by discretization. A more difficult practical problem was the specification of the transition probability matrix $\{P_{ij}(d),\ i,\ j=1,2,...,k\}$. Usually there would not be enough empirical data so that these quantities can be estimated reasonably well using Markov chain estimates even for moderate values of *k*. Some of these problems have been eliminated by postulating a continuous time model for the state. In much of the work of practical interest, the simple model

$$dS(t)/S(t)=\alpha\,dt+\sigma\,dW(t) \tag{7.8}$$

has been used. Here the state space $S(t)$ was described by a stochastic differential equation and $W(t)$ was a Wiener process and σ and α were unknown parameters.

The main idea of this article was to return to a discrete time framework where (7.8) was replaced by a more general type difference equation for $S(t)$ in such a way that the transition probabilities $P_{ij}(d)$ entering in (7.7) can be generated from the difference equation that in turn contains a few parameters that can be estimated from the data or determined by choice of scenarios. The dynamic programming algorithm was then used to pinpoint the optimal policy.

As I said before, the main idea of this article was to return to a discrete time framework. Discretization means that a continuous state problem

$$V_n'(x)=\max_d\left[C(x,d)+\beta\int(P(x,y);d)V_{n+1}'(y)dy\right] \tag{7.9}$$

is replaced by the discrete version (7.7). To have confidence in their model they thought that it was important to have an idea of how accurate this approximation was. As the discretization was made finer, the right hand side of (7.7) converged to the right hand side of (7.9) for a given discretization point $x_i=\left(x_{i1},x_{i2},...,x_{ip}\right)$, representing a state i, $i=1,2,...,k$. Thus if a fine enough grid was chosen, $|V_n'(x_i)-V_n(x_i)|$ could be made arbitrarily small. To apply the dynamic programming approach, the price had to be discretized. So they are worth with levels of discretization ranging with S_t going in equidistant steps. They assumed that the probability transition matrix $\left[P_{ij}(d)\right]=(P_{ij})$ was independent of the decision d made. They used standard backward recursion to solve dynamic programming problem. It was also possible to solve this problem with $N = \infty$ using a policy improvement algorithm or linear programming but they didn't because of their computational complexity. In this example, there were 32 different price sates, and in 16 of these it found to be optimal to open the mine but in Brennan and Schwartz (1985) paper they found 4 of these price states as optimal and this was of course an important theoretical and practical distinction between the two models.

In the paper [8], it was mentioned that this problem possessed an additional feature of difficulty due to presence of chance mechanisms. In this problem, they had two gold mines, Amerconda and Bonanza, the first with an amount x of gold, and the second contains an amount of y. In addition they had a rather delicate gold-mining machine which has the property that if used to mine gold in Amerconda, there was a probability p_1 that it mined a fraction r_1 of the gold there and a probability $(1 - p_1)$ that it mined no gold and be damaged beyond repair. Similarly bonanza associated the probabilities p_2 and $(1 - p_2)$ and the fraction r_2. They began by using the machine in either the Amerconda (*A*) or Bonanza (*B*) mine. If the machine undamaged, they again made a choice of using the machine in either of the two mines, and continued in this way, making a choice before each mining operation, until the machine is damaged. Then they tried to decide what sequence of choices maximizes the amount of gold mined before the machine is damaged. Since they dealed with a stochastic process, they use the average of the possible returns (expected value). After they listed all the policies, to determine an optimal policy, computed the expected returns and compared them. The functional equation of this problem is:

$f(x, y) =$ Expected amount of gold mined before the machine is damaged when *A* has *x*, and *B* has *y*, and an optimal policy is employed.

If an *A* operation is used first, the expected amount of gold mined was denoted by $f_a(x, y)$ and similarly for *B* operation, it was denoted by $f_b(x, y)$.

$$f_a(x, y) = p_1\{r_1 x + f[(1 - r_1)x,\ y]\}$$
$$f_b(x, y) = p_2\{r_2 x + f[x,\ (1 - r_2)y]\}$$

Since A or B must be chosen so as to maximize the over-all expected return, the following equation yielded this:

$$f(x, y) = \max \begin{Bmatrix} A: & p_1\{r_1 x + f[(1-r_1)x,\ y]\} \\ B: & p_2\{r_2 x + f[x,\ (1-r_2)y]\} \end{Bmatrix}$$

Then the solution equation was shown as follows:

If $p_1 r_1 x/(1-p_1) > p_2 r_2 y/(1-p_2)$, take the A choice

If $p_1 r_1 x/(1-p_1) < p_2 r_2 y/(1-p_2)$, take the B choice

If $p_1 r_1 x/(1-p_1) = p_2 r_2 y/(1-p_2)$, either of the choice is optimal

7.4 Feasibility Problems

In the paper [10], they considered the problem of routing trains through railway stations. It was motivated by the project DONS that was carried out under the supervision of the Dutch Organization Railned and Netherlands Railways. The Decision Support System that is the objective of the project DONS contained two modules. One was CADANS that assisted the planners in generating the cyclic hourly timetables. The second was the STATIONS that assisted the planners in checking whether a timetable generated by the other module was feasible with respect to the routing of trains through railway stations.

In this paper they tried to solve the feasibility problem. Given the layout of a railway station, the arrival and departure times, as well as the arrival and departure directions of a number of trains, is it possible to route these trains through the railway station such that no pair of trains is conflicting, such that trains can be coupled or uncoupled if necessary, and such that a number of service constraints are satisfied? A railway station can be entered by a train from a number of entering points and left through a number of leaving points. Each of these points corresponds to the direction of travel. The railway network outside these points was not relevant for the feasibility problem. A railway consists of platforms and many track sections. A complete route for a train is a sequence of sections connecting an entering point to a leaving point, by passing the platforms. So, there will be many different routes between a given pair of entering and leaving points and even several different routes that use the same platform.

The routing of one train depends on the routing of others. Since any track section can only be reserved by one train at a time, no section of the inbound route may be reserved by another train before the section is released again. This will happen as soon as the train leaves the section. Moreover, a complete outbound route leading from a platform towards a leaving point is reserved for each train. If a train doesn't stop at a platform, a complete route that consists the inbound route and outbound, is reserved for the train.

While they assign the trains to their route, they tried to prevent conflicting of the trains and thus the reservation of a common section within the route for a train doesn't conflict with the reservation of this section within the other route of the other train. So, the assignment of the train $t \rightarrow r(t)$ is feasible when intersection with the other train's assignments is empty set. Furthermore, they modeled the other constraints such as coupling, that means the following train coupled onto the leading train, and other service constraints according to these safety rules. Safety rule was represented by $F_{t,t'}$. Its mean is safety rule for the train t and train t'. Based on the information contained in the sets $F_{t,t'}$ some of the allowed routes for a certain train may be excluded from further consideration, since they are dominated by other routes. In particular route r may be eliminated from the route set if there exists another route r' that leaves at least the same routing possibilities for all other trains. Therefore, the set of relevant sections of r' is a subset of the set of relevant sections of route r.

The feasibility problem is firstly formulated by integer linear program. The objective is to maximize the number of trains that can be routed through the railway station. $X_{t,r}$ was assumed the value 1 if train t is assigned to route r and the value 0 otherwise.

Objective function

$$\max \sum_{t \in T} \sum_{r \in R_t} X_{t,r} \qquad \textbf{(7.10)}$$

Subject to

$$X_{t,r} + X_{t',r'} \leq 1 \qquad \forall t, t' \qquad \textbf{(7.11)}$$

$$\sum_{r \in R_t} X_{t,r} \leq 1 \qquad \forall t \qquad \textbf{(7.12)}$$

$$X_{t,r} \in \{0,1\} \qquad \forall t \qquad \textbf{(7.13)}$$

Where constraint (7.11) guarantees that only allowed train-route combinations are selected. Constraint (7.12) ensures that each train is assigned to at most one route. After they modeled the problem by the integer linear program, they showed that the problem can be interpreted as a Node Packing Problem (NPP) which deduces a number of valid inequalities that tighten the integer programming formulation, and thus make the LP-relaxation more accurate. Then they showed that, if the layout of the railway station is fixed, then the safety optimization problem can be solved by a dynamic programming approach in an amount of time that is polynomial in the number of trains.

7.5 Capacity Allocation Problem

In the paper [11], they faced the problem of finding the optimal trade-off between the number of arrivals and departures in order to reduce a delay function of all the flights, using a more realistic representation of the airport capacity, i.e. the capacity envelope. For general airport capacity envelopes, they proposed a dynamic programming formulation with a corresponding backward solution algorithm, which is robust, easy to implement and has a linear computational complexity. The algorithm performances were evaluated on different realistic scenarios, and the optimal solutions were compared with the greedy algorithm which can be seen as the approximation of the current decision procedures.

They presented the capacity allocation problem. The main difference between this model and the ground-holding model, which has acquired greatest interest in the research community, was the explicit consideration of interdependent arrivals and departures capacities. Since many interactions between the arrivals and departures, due to the safety rules and airport layout, they represented the airport capacity by an arrival-departure capacity curve or envelope. Airport arrivals and departures capacities were given by interdependent variables whose values depend on the arrival/departure rate of the total airport operations in a specific time unit. So, airport capacity was defined as the number of aircraft allowed to land and take-off in a unit of time. All the combinations of arrivals and departures which saturate airport capacity defined the capacity envelope. Therefore, they represented the airport capacity by a set of linear constraints indexed by $i \in I_t$ at time t of the type

$$\alpha_{i,t}(\text{departures at time } t) + \beta_{i,t}(\text{arrivals at time } t) \le \gamma_{i,t} \quad \forall i \in I_t$$

Where $\alpha_{i,t}$, $\beta_{i,t}$, and $\gamma_{i,t}$ were given constants. The envelope provided a better and more accurate representation of the real airport capacity especially in those airports where either there is just one runway or there are intersections among runways. For instance, in the case of Rome Fiumincino airport with good weather conditions, if all the capacity was allocated to arrivals then 56 flights could arrive and if all the capacity was allocated to departures then 44 flights could depart in one hour. The case of mixed operations, they told that they might have 34 arrivals and 32 departures.

The airport capacity allocation problem was defined as the problem of finding an optimal policy that uses the airport capacity in an efficient way, i.e., that reduces the total number of delays (of landing and airborne aircraft).

The aim of the paper was, given the airport traffic flow demands, to find for each time period t of the time horizon $T = \{1, 2, ..., T\}$, the optimal capacity allocation in order to satisfy the airlines' demands as much as possible, minimizing a function of the flight delays. Each airport had requested demands of arrivals and departures called demand point. If the demand point is inside the airport capacity, both the demand of arrivals and departures were satisfied. But if the demand point was outside the airport capacity, then some flights must be delayed or cancelled. Hence, they had to decide the number of arrivals and departures to serve in order to match the corresponding requested demands. The objective function was the minimization of total delay costs (costs means delays). The constraints were formed from two things: capacity constraint and assignment constraints. For each flight, the assignment constraints fixed the unique departure and arrival times. Moreover, once the optimal airport working point, which is the optimal combinations of arrivals and departures, was established for each time period, the FIFO rule was the optimal policy for the airport capacity allocation problem.

Let them denote the requested number of departures and arrivals by $\bar{D}_t$ and $\bar{A}_t$ respectively, for each time period t. The decision variables of the problem are

$d_t \rightarrow$ Number of delayed departures at time t

$a_t \rightarrow$ Number of delayed arrivals at time t

The objective function was the minimization of cumulative weighted delay, and $\Delta_t(a_t)$ and $\Delta_t(d_t)$ represented the marginal delay increments at time t when decisions a_t and d_t were taken.

$$\min \sum \left[\Delta_t(a_t) + \Delta_t(d_t)\right]$$

Subject to

$$\begin{aligned}
&\alpha_{i,t}(d_{t-1} + \bar{D}_t - d_t) + \beta_{i,t}(a_{t-1} + \bar{A}_t - a_t) \le \gamma_{i,t} \\
&d_{t+1} \le d_t + \bar{D}_t \\
&a_{t+1} \le a_t + \bar{A}_t \\
&d_t, a_t \in Z^+
\end{aligned}$$

Second and third constraints defined the transition of the system from one stage to the following one, with initial condition $d_0 = a_0 = 0$.

At each stage t (time period), the state of the system z_t was defined by the total number of requested arrivals and departures, that was given by the sum of those delayed in the previous time period and the current demand. The state of the system depends on the decision taken in the previous stage. At any stage the decision consisted in establishing the number of delayed departures and arrivals which is equivalent to fixing the airport working point AWP. The resulting problem has a dynamic programming formulation. It is a discrete-time dynamic system whose state evolves according to a decision or control, and the objective function accumulates additively over time and depends on the states visited and the controls chosen. This algorithm iteratively constructed the optimal policies, i.e. the sequence $\pi = \{m_1, m_2, ...\}$ where m_k is a function mapping states into decisions.

The transition of the system from one stage to next one was given by

$$z_{t+1} = (a_{t+1}, d_{t+1}) = (a_t, d_t) + (\bar{A}_t, \bar{D}_t) - (u_a(t), u_d(t))$$

Where $u_t = (u_a(t), u_d(t))$ is the decision taken at stage *t*. And the problem was stated as below

$$\min_{u_1,\ldots,u_{t-1}} \left[\sum_{t \in \{1,\ldots,T-1\}} r_t(z_t, u_t) + r_T^*(z_T) \right]$$

Where $r_t(z_t, u_t) = \Delta_t(a_t) + \Delta_t(d_t),$ which is the immediate return if the system is in state z_t at stage *t* and the decision u_t is taken and $z_t = (a_t, d_t)$ is the state of the system at stage *t*. According to these, the problem could be solved using a backward dynamic programming algorithm. This procedure reduces the possible sequence of decisions to that of finding the optimal decision in all states for all stages.

In the following sections, the computational analysis and evaluation of the algorithm performances were carried out. The algorithm performances were evaluated on different realistic scenarios and also with respect to commonly used greedy decision policy. On this subject, optimal solutions were more balanced than corresponding greedy solutions as evidenced by both average and maximum values of the percentage deviation of the greedy solutions from optimal solutions. As a result, the dynamic programming algorithm gave better results than the other.

7.6 Capacity Expansion Problems

In the paper [12], the problem of optimally meeting a stochastically growing demand for capacity over an infinite horizon was considered. Under the assumption that demand for product follows either a nonlinear Brownian motion or a non-Markovian birth and death process, they showed that stochastic problem can be transformed into an equivalent deterministic problem. The equivalent problem was formed by replacing the stochastic demand by its deterministic trend and discounting all costs by a new interest rate that is smaller than the original, in approximate proportion to the uncertainty in the demand.

Within the modeling environment a stochastic demand is often replaced by a forecasted demand that grows deterministically over time. But some of the papers have established that simply replacing demand by their forecasted means is not satisfactory. Their assumptions are from Manne’s (1961) paper. The key transformation is the replacement of interest rate by an interest rate that is a decreasing function of the variance of the demand process.

They assumed that the available capacity must meet or exceed the demand for product or service. The decisions include the timing, sizing, and type of capacity to be installed over time.

At each transition epoch, the future states of a semi-Markov process are conditionally independent of the past states when given the present state. By the assumptions, thus, the deterministic variable installed capacity captures all information regarding the historical evolution of the problem that is relevant to the optimal determination of the next facility. Therefore, the problem was stated as a dynamic program with the state variable representing installed capacity at expansion epochs. Let $T(x)$ be the time at which a total capacity x is exhausted and let $f(x)$ denoted the minimum expected cost discounted to time $T(x)$ of expanding capacity to meet future demand given that the accumulated capacity is currently exhausted.

When an equivalent interest rate exists, it is unique and given by

$$r^* = |\ln \phi_{T(x)}(r)| / E[T(x)] \qquad \textbf{(7.14)}$$

where $\phi_{T(x)}(r)$ is the laplace transform of $T(x)$. When equivalent interest rate r^* exists, the original stochastic expansion problem may be solved via a deterministic problem formulation in which the random expansion epochs $T(x)$ are replaced by their expected values and the original interest rate is replaced by its equivalent r^*.

In the following, they gave two definitions and according to them, they formed the equivalent interest rate r^*.

In the continuous model the r^* was given by

$$r^* = (p/\sigma)^2 (\sqrt{1 + 2r(\sigma/p)^2} - 1$$

In the discrete model, the r^* was given by

$$r^* = \ln(\phi(r)^{1/\phi'(0)}) \text{ where } \phi'(0) = d\phi(r)/dr\,|_{r=0} .$$

In either case, they showed that every optimal capacity expansion sequence for the deterministic problem with demand $P^*(.)$ in which all costs were continuously discounted using the interest rate r^* was optimal for the stochastic problem.

As a result, working within the framework of the general model, they provided conditions for the existence of a deterministic equivalent problem P^* and an equivalent interest rate r^*. The equivalent interest rate enables the effects of the variability of the demand process to be completely summarized in a single number r^*. Finally, they told that the transformation to the equivalent problem required that only a reduction of the interest rate from r to r^*.

In another paper [14], the problem of choosing an optimal initial capacity expansion was considered. They discussed an algorithm about how long a horizon is sufficient to reach stability in the first facility choice and they showed that this first choice was the optimal choice for the infinite horizon problem. In paper [20], two approaches were developed for capacity planning for large multilocation systems and in the paper [21], dynamic programming approaches for the multilocation problem adapted to solve the specialization problem in which a growing demand must be met.

7.7 Replacement Analysis

Replacement analysis is concerned with determining the optimal time to remove a current asset (defender) from service and selection of another asset (challenger) to take its place. The economic life of an asset is dependent on a variety of factors, including deterioration and obsolescence. While obsolescence is generally a result of external to the asset, such as technological change, deterioration is generally a result of how the asset is utilized over its lifetime. If multiple assets are available to meet demand and the assets must not continually operate at maximum capacity, then a decision maker may have some control over asset utilization patterns by allocating workload. These utilization patterns directly impact operating costs and salvage values and thus have a strong influence on the optimal replacement time of the assets. In the paper [13], they examined asset replacement decisions, based on age and cumulative utilization, under various costs and demand assumptions. They provided an efficient optimal solution procedure through the use of stochastic dynamic programming. They provided a method to easily examine solutions for the two asset case.

In the case of a single asset utilization is generally not a controllable variable as its usage must be a reaction to the demand environment. However in the case of multiple assets, one has the ability to set utilization patterns by allocating demand among the available assets, assuming all assets must not continually operate at their maximum capacity. This allocation decision has a direct impact on each asset's operating costs and salvage values and thus the optimal replacement schedule.

In this paper, they were concerned with examining the optimal replacement and utilization schedules for a number of assets over a finite horizon with stochastic demand. Both age and cumulative utilization were formed the state variables for replacement decisions. These are assumed to be the same for similar assets (same age and technology).

In the single asset replacement problem, the decision is whether to keep or replace the asset at the end of each period. In the two asset case, the decisions are whether to keep both assets (KK), replace both assets (RR) , or keep one and replace the other (RK, KR), totaling four possibilities. After this decision made, the allocation of demand to the two assets must be made. This two stage decision process repeated at each period over the decision horizon. The states of dynamic programming referred to the state of each asset, defined by their age *i*, and cumulative utilization *j*. Once an asset reached its maximum service life age *N* or cumulative utilization *M*, it must be replaced.

$f_t(i_1, j_1, i_2, j_2) =$ Minimum expected net present value of costs when starting with two assets of age i_1 and i_2 and cumulative utilization levels of j_1 and j_2 at time *t* and choosing optimal decisions through time *T*.

$P_t(i, j) =$ Purchase cost of ith period old asset with cumulative utilization j at time t

$S_t(i, j) =$ Salvage value of ith period old asset with cumulative utilization j at time t

$C_t(u, i, j) =$ Operating and maintenance cost of an ith period old asset with cumulative utilization j utilized at level u at time t

$K_t =$ Fixed cost charge if an asset is purchased at time t

$d_{m,t} =$ Demand level m in period t

$p(d_{m,t}) =$ Probability of demand $d_{m,t}$ in period t

$$
f_t(i_1, j_1, i_2, j_2) = \min \left\{ \begin{array}{l}
KK : \alpha \sum_{m=1}^{D} p(d_{m,t}) \left[\min_{u_1, u_2} \left\{ \begin{array}{l} C_t(u_1, i_1, j_1) \\ + C_t(u_2, i_2, j_2) \\ + f_t(i_1 + 1, j_1 + u_1, i_2 + 1, j_2 + u_2) \end{array} \right\} \right], \\
RK : K_t + P_t - S_t(i_1, j_1) + \\
\qquad \alpha \sum_{m=1}^{D} p(d_{m,t}) \left[\min_{u_1, u_2} \left\{ \begin{array}{l} C_t(u_1, 0, 0) \\ + C_t(u_2, i_2, j_2) \\ + f_t(1, u_1, i_2 + 1, j_2 + u_2) \end{array} \right\} \right], \\
KR : K_t + P_t - S_t(i_2, j_2) + \\
\qquad \alpha \sum_{m=1}^{D} p(d_{m,t}) \left[\min_{u_1, u_2} \left\{ \begin{array}{l} C_t(u_1, i_1, j_1) \\ + C_t(u_2, 0, 0) \\ + f_t(i_1 + 1, j_1 + u_1, 1, u_2) \end{array} \right\} \right], \\
RR : K_t + 2P_t - S_t(i_1, j_1) - S_t(i_2, j_2) + \\
\qquad \alpha \sum_{m=1}^{D} p(d_{m,t}) \left[\min_{u_1, u_2} \left\{ \begin{array}{l} C_t(u_1, 0, 0) \\ + C_t(u_2, 0, 0) \\ + f_t(1, u_1, 1, u_2) \end{array} \right\} \right]
\end{array} \right\}
$$

8. APPLICATION

8.1 Problem Definition

In this project, 4.Levent – Maslak Sanayi Subway Project was analyzed (see Figure 8.1) which consists of 4.64 kilometers with 3 stations, by using probabilistic dynamic programming.

The objective of the project is twofold:

- Investigate the feasibility of the investment
- Decide on the optimal investment policy (required number of trains to be purchased in each investment epoch).

These two decisions are interrelated: If the optimal investment policy is not profitable, then do not invest in the project. Therefore, our main objective is to find the optimal investment policy, i.e., the required number of trains to be bought in each investment epoch. For example, how many trains to buy at the beginning of year, say, 2012.

The main factor affecting the required number of trains to buy is the traffic intensity during the peak hours. The distribution of subway peak hours is given in Figure 8.2 and Table 8.1 shows the estimated daily passenger traffic that constitutes approximately 53% of the daily traffic in peak hour. [24]

Table 8.1: Peak and Off-Peak Hour Traffic on Subway (*) (Source: [24], page 33)

	Hours	Percentage of Daily Traffic
Peak Hours (8:00 – 10:00, 16:00 – 20:00)	6	53.0
Off Peak Hours (10:00 – 16:00, 20:00 – 21:00)	7	35.0
Night Hours (21:00 – 01:00, 08:00 – 07:00)	8	12.0
Total	19	100.0
(*) Source: Ulaşım A.Ş., October 2004		

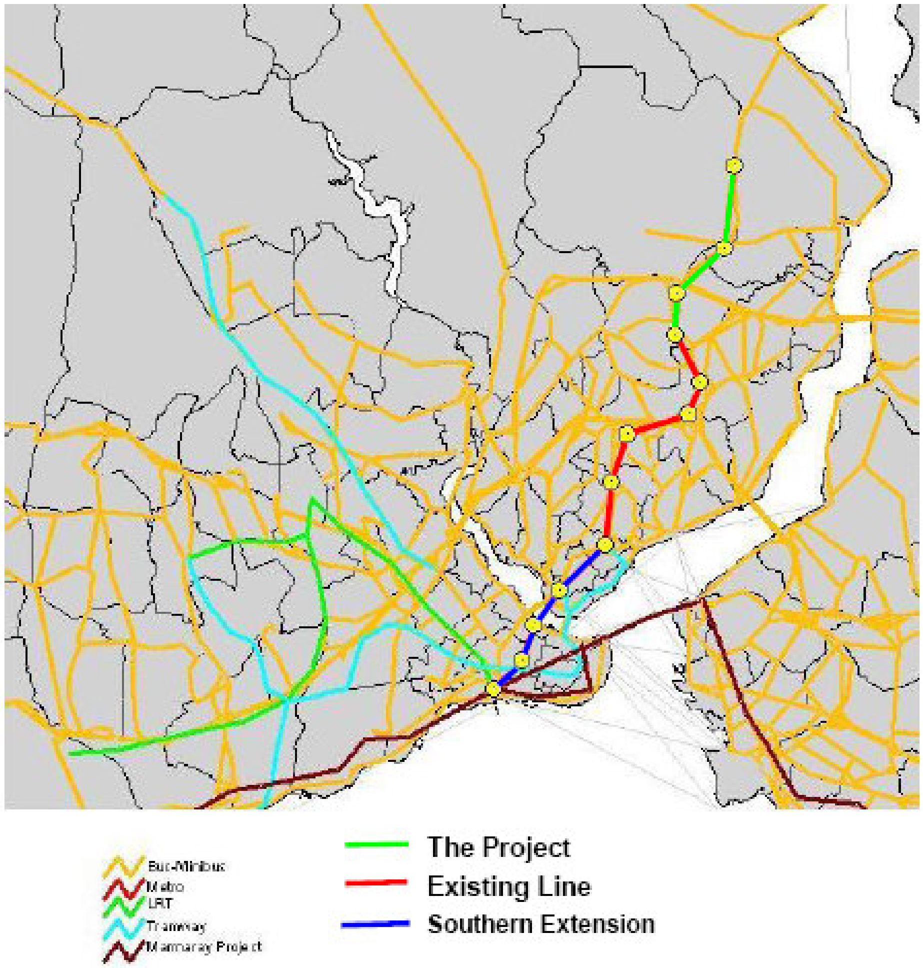

Figure 8.1: Present Subway Line and Its Extensions (Source: [24], page 8)

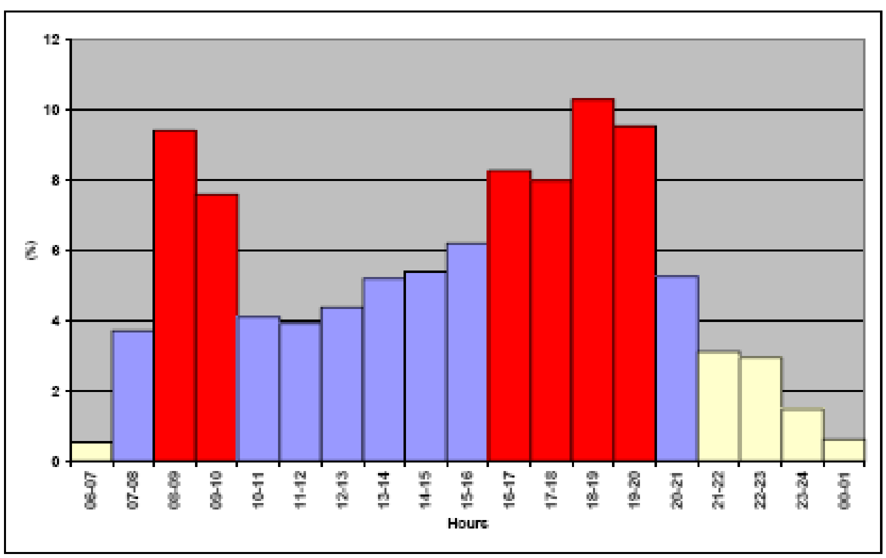

Figure 8.2: Distribution of Subway Traffic by Hours (Source: Ulaşım A.Ş. 2004)

Table 8.2 shows the estimated annual traffic. Our approach is instead of taking a single value for yearly traffic, giving probabilities to yearly traffic values for each year. For example, for year 2011 we have 3 states namely 160,000,000; 165,000,000; 170,000,000 that denote the yearly passenger number of the Yenikapı – Maslak Sanayi Line with different probabilities.

At each stage we have 3 states representing the number of passengers per year. We will choose these states as 3% above the estimated average number of passengers per year and 3% below that.

Table 8.2: Annual Traffic of the Subway Project (Source: [24], page 36)

Years	Annual Demand		
	With Project Case (Yenikapı - Maslak Oto Sanayi)	Without Project Case (Yenikapı - 4. Levent)	Difference
2010	156,816,742	130,991,461	25,825,281
2011	160,819,239	133,876,180	26,943,060
2012	164,821,737	136,760,899	28,060,838
2013	168,824,235	139,645,618	29,178,617
2014	172,826,733	142,530,337	30,296,396
2015	176,829,231	145,415,056	31,414,175
2016	180,831,729	148,299,775	32,531,953
2017	184,834,226	151,184,494	33,649,732
2018	188,836,724	154,069,213	34,767,511
2019	192,839,222	156,953,933	35,885,290
2020	196,841,720	159,838,652	37,003,068
2021	200,844,218	162,723,371	38,120,847
2022	204,846,716	165,608,090	39,238,626
2023	208,849,213	168,492,809	40,356,404
2024	213,524,271	172,264,495	41,259,776
2025	218,303,979	176,120,609	42,183,370
2026	223,190,681	180,063,042	43,127,639
2027	228,186,770	184,093,726	44,093,044
2028	233,294,696	188,214,636	45,080,060
2029	238,516,962	192,427,792	46,089,170
2030	243,856,128	196,735,258	47,120,869
2031	249,314,810	201,139,147	48,175,663
2032	254,895,684	205,641,616	49,254,068
2033	260,601,485	210,244,872	50,356,613
2034	266,435,009	214,951,171	51,483,838

As we see from the Figure 8.3, the population increases linearly according to the year like as employment and number of students. To meet their demand, the number of buses, minibuses or train numbers should be increased. But they have different profits and costs. The trains carry more passengers than the buses and minibuses so when trains are decided to buy, the traffic of the buses and minibuses will diverted from road to railway. Therefore we will get cost savings from the buses and minibuses. As a result, when we decide to buy train, we should consider all these considerations.

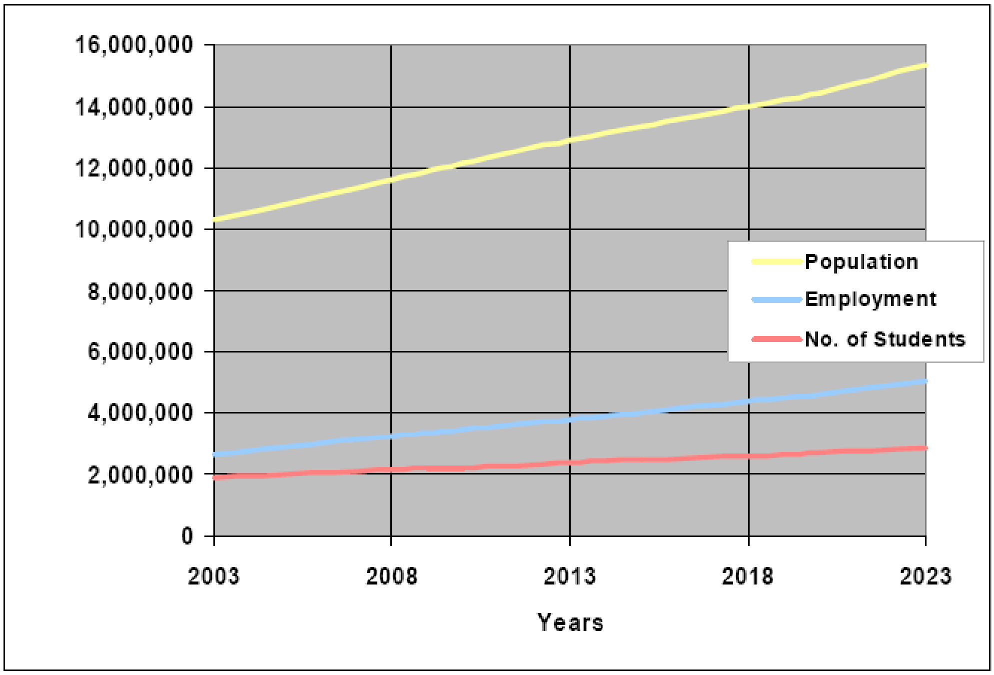

Figure 8.3: Population, Employment and Number of Students Estimates (Source: [24], page 19)

8.1.1 Model Formulation

In this study, our objective is to decide on whether to buy train or not, if we decide on to buy train, how many trains to add each year. In each state, we have 5 decisions: we can buy no train, we can buy one train, we can buy two trains, etc. i.e., $d_i = \{0, 1, 2, 3, 4\}$.

The policy π is formed from the decisions of the years, $\pi = \{d_1, d_2, ..., d_{25}\}$.

Let,

$f_n(i) =$ The maximum expected discounted reward that can be earned during n periods if the state at the beginning of the current period is i

$r_n(i, d) =$ The revenue of the state i under decision d at stage n

$c_n(i, d) =$ The cost of the state i under decision d at stage n

$p(j \mid i,d) =$ The probability of the next period's state j given the current state is i with decision d

$r =$ the interest rate, $1/(1+r)$ is the discount factor between two periods.

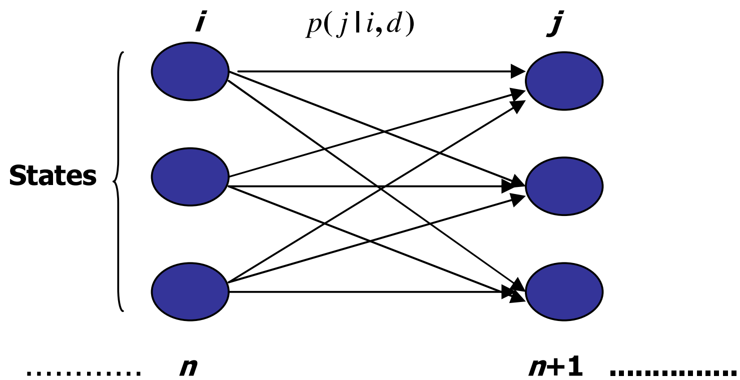

Figure 8.4: Diagram for the Optimality Equation

Then the optimality equation becomes

$$f_n(i) = \max_{d \in d_i} \left\{ r_n(i,d) - c_n(i,d) + \frac{1}{1+r} \sum_{j=1}^{3} p(j \mid i,d)\, f_{n+1}(j) \right\}.$$

8.2 Planning Horizon

The period of calculation for the analysis is assumed as follows: 5 years of construction period beginning in 2005 and 25 years of operations period from 2010 to 2034. In the operations period, we found the optimal investment policy, how many trains should be added to meet demand each year, by using probabilistic dynamic programming. We worked with backward dynamic programming. In the construction period, we investigated the feasibility of the investment; optimal timing of the investments.

8.3 The Project Costs and Revenues

8.3.1 Economic Costs

1. Construction costs:

Table 8.3: Construction Costs of the Project ($1,000) (Source: [24], page 53)

		Year 1 2005	Year 2 2006	Year 3 2007	Year 4 2008	Year 5 2009	Total
Civil Works	Capital	50,000	50,000	40,000	0	0	140,000
E-M Works & SubwayVehicles	Capital	0	0	50,000	50,000	98,400	198,400
Engineering & Consulting	Capital	1,500	1,500	2,000	2,000	2,000	9,000
Financing Costs	Capital	1,636.8	148.8	148.8	111.3	73.8	2,119.5
Miscellaneous & Others	Capital	7,000	7,000	8,000	7,000	7,000	36,000
General Total		60,136.8	58,648.8	100,148.8	59,111.3	107,473.8	385,519.5

2. **Investment Costs of Subway Vehicles:** Each train has 8 vehicles and the cost of each vehicle is $1,600,000.

3. **Operating and Maintenance Costs:**

Table 8.4: Unit Operations and Maintenance Costs (Source: [24], page 39)

Energy (USD / Vehicle-km)	0.52
Track & Facilities Maintenance (USD / km)	10,275
Repair & Maintenance of Rolling Stock (USD / Vehicle-km)	0.04
Managerial Personnel (USD / Vehicle-km)	0.096
Other Expenditures (*) (USD / km)	513,135
(*) Rents and general expenditures	
Source: Ulaşım A.Ş.(İstanbul Transportation Company), 2004	

8.3.2 Economic Benefits

1. **Vehicle Operating Cost Savings of Buses and Minibuses** :

The Subway Project will reduce the vehicle operating costs of buses and minibuses, because some traffic will be diverted from road to rail. Vehicle operating cost savings are $1.16 per vehicle-km for buses and $0.29 per vehicle-km for minibuses. The modal split of the subway traffic for the "without project" case was estimated as follows:

Private cars 15 %

Buses 43 %

Minibuses 43 %

2. Vehicle Operating Cost Savings of Private Cars:

Operating cost of cars, including, fuel, oil and tyres consumption as well as maintenance and depreciation costs, was estimated as $0.39 per vehicle-km in 2003. The value of economic operating costs was calculated excluding duties and taxes. Shadow price coefficient of 0.70 (conversion factor) was used to convert operating costs of cars to economic costs. [24]

3. Capital Cost Savings of Buses and Minibuses:

The Subway Project will considerably reduce the investment required for new and additional buses and minibuses that would be required to accommodate the peak hour traffic demand and to replace the old buses and minibuses. It was assumed that, under normal conditions, a bus and a minibus could operate 180 km daily. Service availability factors are assumed to be 0.85 and 0.90 for buses and minibuses, respectively. Capitals are $200,000 per bus and $63,000 per minibus. [24]

4. Road Maintenance Cost Savings:

The Subway Project is also expected to provide savings of road maintenance costs because of the traffic to be diverted from cars, public transport buses and minibuses. It is clear that heavy vehicles (trucks and buses) cause more damage on roads than light vehicles such as cars. The damage is assumed to be a non-linear function of the axle weight of the road vehicle. It has been calculated that a minibus causes 16 times and a bus causes 4000 time as much damage as a typical car. Based on the statistics published by the General Directorate of Highways (K.G.M.), road maintenance cost per 1 million vehicle-km was estimated as $16.16 for cars, $258.00 for minibuses and $65,000 for buses. [24]

5. Road Accident Cost Savings :

Economic costs of road accidents are very difficult to quantify. A considerable portion of road accidents on the city roads are not reported. Road accident costs cover the cost of the material damaged in accident, hospital and police costs and economic losses caused by fatalities and injuries. Based on the accident data published by the General Directorate of Highways (K.G.M.) for Istanbul, road accident cost per 1 million vehicle-km was estimated as $36,208. [24]

6. Travel Time Cost Savings :

The travel time to be saved by transport users was estimated as the difference in travel time (total passenger-hours) between the “without” and the “with” project cases. Average travel time to be saved per trip with the project was estimated to increase from 12.2 minutes in 2010 to 15.5 minutes in 2023. The GDP data to calculate the value of time were obtained from the Household Income Distribution Study of State Institute of Statistics (D.I.E.) and shown in Table 8.5 and value of time savings of the project is shown in Table 8.6.

7. Environmental Cost Saving :

Similar to road accidents, environmental cost savings resulting from a net reduction in noise levels and motor vehicle emissions are also very difficult to quantify. Accurate valuations relating to air quality will always be difficult owing to problems of obtaining reliable data on emissions, ambient pollutant levels, quantifying the link between emissions and impacts. Environmental damage costs of road vehicles were derived from "*Fuel and Location on the Damage Costs of Transport Emissions, Eyre, N.J. et al., JTEP, January 1997, pp. 5-24*". An average of $0.40 per km for buses and minibuses, $0.15 per km for cars were assumed as the environmental damage cost of urban emissions of vehicles.

Table 8.5: Average Value of Time (*) (Source: [24], page 45)

	Car user (**)	PT User (***)	Average
GDP per capita (USD)	7,066	4,162	6,065
GDP per working Person (USD)	23,553	13,874	20,217
Value of Time per Working Hour (USD / Hour)	11.06	6.51	9.49
Value of Time per Nonworking Hour (USD / Hour)	2.76	1.63	2.37
Average Value of Time (USD / Hour)	6.91	4.07	5.93
(*) As of average of year 2004, $1 = 1,432,144 TL			
(**) The lowest income group of 20% of population was excluded			
(***) The highest income group of 20% of population was excluded			

Table 8.6: Value of Time Savings ($) (Source: [24], page 46)

Year	With Project	Without Project	Value of Time Savings (USD)
2010	25,419,568	49,277,704	23,858,137
2011	26,390,276	51,674,645	25,284,368
2012	27,360,985	54,120,207	26,759,222
2013	28,331,694	56,615,886	28,284,192
2014	29,302,403	59,163,237	29,860,834
2015	30,273,112	61,763,882	31,490,771
2016	31,243,821	64,419,512	33,175,691
2017	32,214,530	67,131,889	34,917,359
2018	33,185,238	69,902,851	36,717,612
2019	34,155,947	72,734,317	38,578,370
2020	35,126,656	75,628,291	40,501,635
2021	36,097,365	78,586,865	42,489,500
2022	37,068,074	81,612,228	44,544,154
2023	38,038,783	84,706,666	46,667,884
2024	39,166,364	87,926,910	48,760,546
2025	40,293,945	91,278,687	50,984,742
2026	41,421,526	94,768,028	53,346,502
2027	42,549,107	98,401,283	55,852,176
2028	43,676,687	102,185,144	58,508,457
2029	44,804,268	106,126,668	61,322,399
2030	45,931,849	110,233,293	64,301,444
2031	47,059,430	114,512,873	67,453,443
2032	48,187,011	118,973,697	70,786,686
2033	49,314,592	123,624,524	74,309,932
2034	50,442,173	128,474,608	78,032,435

8. **Fee (Financial Revenue) :**

The current fare structure for the buses and rail systems has a flat rate irrespective of the distance of the journey. The existing policy of applying reduced for students, elderly and other eligible categories of users is expected to be retained. The existing full fare is 1.10 YTL ($0.81). Considering the passengers traveling at reduced fares, the fare level of $0.70 was applied as an average to estimate the revenues. [24]

8.4 Results of the Model

8.4.1 Part 1: Optimal Investment Policy

In this part, we used Microsoft Visual Studio C++ 6.0 language to solve the 25 years period of investment policy based on dynamic programming. Decisions are about the number of trains required to be purchased in each investment epoch. You can see the computer program code in the Appendix. The output of the code is shown in Table 8.7. At the beginning we have 4 trains and after 19th year (that is, year 2029) we decided to buy 1 train each year for 6 years until the end of the planning horizon. Thus, in 2034 we should have 10 trains in all.

After we got the results, we changed some of the variables and made sensitivity analysis. The results are shown in Table 8.8. As we see from the sensitivity analysis, interest rate is one of the most important variables in determining the decisions for the problems that have long planning horizon.

Table 8.7: Output of the code

```
******************************************************
**********  initial number of trains = 4  ************
******************************************************

Year: 1
                Decision:0      Number of trains:4
Year: 2
                Decision:0      Number of trains:4
Year: 3
                Decision:0      Number of trains:4
Year: 4
                Decision:0      Number of trains:4
Year: 5
                Decision:0      Number of trains:4
Year: 6
                Decision:0      Number of trains:4
Year: 7
                Decision:0      Number of trains:4
Year: 8
                Decision:0      Number of trains:4
Year: 9
                Decision:0      Number of trains:4
Year: 10
                Decision:0      Number of trains:4
Year: 11
                Decision:0      Number of trains:4
Year: 12
                Decision:0      Number of trains:4
Year: 13
                Decision:0      Number of trains:4
Year: 14
                Decision:0      Number of trains:4
Year: 15
                Decision:0      Number of trains:4
Year: 16
                Decision:0      Number of trains:4
Year: 17
                Decision:0      Number of trains:4
Year: 18
                Decision:0      Number of trains:4
Year: 19
                Decision:0      Number of trains:4
Year: 20
                Decision:1      Number of trains:5
Year: 21
                Decision:1      Number of trains:6
Year: 22
                Decision:1      Number of trains:7
Year: 23
                Decision:1      Number of trains:8
Year: 24
                Decision:1      Number of trains:9
Year: 25
                Decision:1      Number of trains:10

  ---------> with profit: 355079716
```

Table 8.8: Results of the Sensitivity Analysis

	Decision Years	Decision (units) → Number of Trains (units)	Profit ($)	
Base case	20 - 25	1 → 10	355,079,716	(*)
Reduction in purchasing cost of train (18.75%)	17 - 25	1 → 13	356,498,280	(**)
Reduction in purchasing cost of train (10%)	19 - 25	1 → 11	355,754,341	(***)
Increase population (2%)	22 - 25	1 → 8	361,591,791	
Increase population (3%)	22 - 25	1 → 8	364,916,092	
Change probabilities	20 - 25	1 → 10	355,913,707	(****)
Change interest rate	20 - 25	1 → 10	413,484,149	(******)
Reduction in purchasing cost (18.75%) and increase population (2%)	19 - 25	1 → 11	362,774,473	
Reduction in purchasing cost (18.75%) and increase population (3%)	20 - 25	1 → 10	365,998,144	
Reduction in purchasing cost (10%) and increase population (3%)	20 - 25	1 → 10	365,446,435	
Increase population (2%) and decrease interest rate to (10%)	22 - 25	1 → 8	420,882,077	
Change probabilities and decrease interest rate to (10%)	20 - 25	1 → 10	414,440,031	(******)

(*) Transition probabilities of the states are:

→ First state: (0.6, 0.3, 0.1)

→ Second state: (0, 0.7, 0.3)

→ Third state: (0, 0.5, 0.5)

and interest rate, r =12%. Decisions were given between the years 20th and 25th and determined as 1 train. At the end, we should have 10 trains with total profit of $355,079,716.

(**) The cost of the train is $1,600,000. Here, its cost was decreased to $1,300,000.

(***) The cost of the train was decreased to $1,440,000.

(****) Transition probabilities were changed as follows: (0.5, 0.3, 0.2), (0, 0.6, 0.4), (0, 0.4, 0.6).

(*****) Interest rate was decreased from 12 % to 10 %.

(******) Transition probabilities were changed as follows: (0.5, 0.3, 0.2), (0, 0.6, 0.4), (0, 0.4, 0.6)

8.4.2 Part 2: Optimal Timing of the Investment

In this part, we investigated the feasibility of the investment. The period is 5 years from 2005 to 2009. We decided to optimal timing of the investments made between these years. In each year we have two decisions:

1. Invest immediately at that year
2. Wait until the next year and invest at that year

Let,

$F_t(x_t) =$ The expected net present value of the cash flows when the firm makes all decisions optimally from this point onwards

$r_t(x_t) =$ Revenue of the state at time t

$\pi_t(x_t, d_t) =$ The profit flow of the current state x under decision d at that time t

$I =$ Sunk cost, investment

$\varepsilon_t[F_{t+1}(x_{t+1})] =$ Expected value of the next state at the next stage t+1

$$F_t(x_t) = \max_{d_t}\left\{\pi_t(x_t, d_t) + \frac{1}{1+r}\varepsilon_t\left[F_{t+1}(x_{t+1})\right]\right\}$$

When we write the equation briefly, we get

$$F_t(x_t) = \max\left\{r_t(x_t) - I,\ \frac{1}{1+r}\varepsilon_t\left[F_{t+1}(x_{t+1})\right]\right\}$$

If we invest immediately, we will get revenue of that state minus the sunk cost, $r_t(x_t)-I$. If we wait and invest next year, we will get the discounted expected value of the next year net present value, $\frac{1}{1+r}\varepsilon_t\left[F_{t+1}(x_{t+1})\right]$. During the construction phase, since we will not get any revenue, the value of $r_t(x_t)$ will be 0. Moreover, at the end we will have only one choice, invest immediately at that year, so we can eliminate the waiting decision. Then we can write the equation again as follows:

$$F_t(x_t)=\min\left\{I;\ \frac{1}{1+r}\varepsilon_t\left[F_{t+1}(x_{t+1})\right]\right\}.$$

$$F_5=I.$$

The solution of investment timing is:

In year 2009:

$$F_5=\{107{,}473{,}800\}$$

→ Invest immediately in that year

In year 2008:

$$\begin{aligned}F_4&=\min\left\{59{,}111{,}300;\ \frac{1}{1+0.12}(59{,}111{,}300+F_5)\right\}\\&=\min\{59{,}111{,}300;\ 148{,}736{,}696\}\\&=59{,}111{,}300\end{aligned}$$

→ Invest immediately in that year

In year 2007:

$$\begin{aligned}F_3&=\min\left\{100{,}148{,}800;\ \frac{1}{1+0.12}(100{,}148{,}800+F_4)\right\}\\&=\min\{100{,}148{,}800;\ 142{,}196{,}517\}\\&=100{,}148{,}800\end{aligned}$$

→ Invest immediately in that year

In year 2006:

$$F_2 = \min\left\{58{,}648{,}800;\ \frac{1}{1+0.12}(58{,}648{,}800 + F_3)\right\}$$
$$= \min\{58{,}648{,}800;\ 141{,}783{,}571\}$$
$$= 58{,}648{,}800$$

→ Invest immediately in that year

In year 2005:

$$F_1 = \min\left\{60{,}136{,}800;\ \frac{1}{1+0.12}(60{,}136{,}800 + F_2)\right\}$$
$$= \min\{60{,}136{,}800;\ 106{,}058{,}571\}$$
$$= 60{,}136{,}800$$

→ Invest immediately in that year

The summary of this analysis is as follows:

- 5th year (2009) → Invest immediately in that year
- 4th year (2008) → Invest immediately in that year
- 3rd year (2007) → Invest immediately in that year
- 2nd year (2006) → Invest immediately in that year
- 1st year (2005) → Invest immediately in that year.

9. CONCLUSIONS

Most investment decisions share three important characteristics: Irreversibility, uncertainty and timing. These three characteristics interact to determine the optimal decisions of investors. In this thesis, I tried to explain investment under uncertainty based on dynamic programming. It was shown that the "net present value" rule can give wrong answers by the Option Approach. Therefore, we should think every time that we will have an option value while deciding on an investment.

The first chapter of the thesis is an introduction part. In this part, the definition of the investment was given and the option approach that is the new view in the investment analysis and the orthodox theory that is known as the neoclassical theory were described. Moreover, it was shown that the traditional "net present value" rule can give wrong answers. The reason is that this rule ignores the irreversibility and the option of delaying investment.

In the second and third chapters, Dynamic Programming techniques and Markov Decision Processes were studied. Dynamic programming was identified as a branch of applied mathematics rather than as something more specific. The subject's coherence results are pervaded by several themes. We saw that these themes include the concept of states, the principles of optimality and functional equations.

Since we used probabilistic dynamic programming in the application part, the fourth chapter focused on stochastic processes. Moreover, continuous-time stochastic processes were tried to explain.

In the fifth and sixth chapters, the techniques about the investment opportunities and the optimal timing of the investments were studied and detailed information was given. It was concerned on two important characteristics of the investment expenditures. First, the expenditures are at least partly irreversible; in other words, sunk costs that cannot be recovered. Second, these investments can be delayed, so that the firm has the opportunity to wait for new information to arrive about prices, costs, and other market conditions before it commits resources.

In the seventh chapter, the literature research about dynamic programming applications was taken part. It was seen that dynamic programming was used in many applications: Investment planning, capacity expansion, optimal allocation of resources, replacement anlaysis, etc.

In the eighth chapter, my application was described in detailed. Here, transportation and traffic problem which are the main problems of İstanbul were considered. By using C++ computer programming language, a programme was developed based on probabilistic dynamic programming. From the code, an optimal policy was found which is formed from the required number of trains to be purchased each year. The decisions given according to the computer program are purchasing 1 train between the years 2029 and 2034 with the total profit of $355,079,716.

After we got the results, a sensitivity analysis was conducted to determine the significance of effects of possible scenarious with respect to some key parameters. The best result was obtained by changing the discount factor according to the sensitivity analysis and when the investment cost of the trains decrease, the required number of trains to be purchased increased. In the 25 years operation period, we used constant discount factor. According to these results, if we change the discount factor each year, we may take more truly results.

Since this optimal investment policy is profitable, we decided that we will invest. Again by using dynamic programming, we determined the optimal timing of the investments. During the construction phase, since we won't get any revenue from the project, the timing of the investments will take the most important place. As a result, we decided to invest immediately in these years.

REFERENCES

[1] **Dixit, A.K. and Pindyck, R.S.,** 1994. Investment under Uncertainty, Princeton University Press, Princeton, N.J.

[2] **Gheorghe, A.V.,** 1990. Decision Processes in Dynamic Probabilistic Systems, Dordrecht; Kluwer Academic, Boston

[3] **Howard, R.A.,** 1971. Dynamic Probabilistic System: Volume II: Semimarkov and Decision Processes, John Wiley and Sons, New York

[4] **Hillier, F.S. and Lieberman, G.J.,** 2005. Introduction to Operations Research, Eighth Edition, McGraw-Hill, New York

[5] **Hubbard, R.G.,** 1994. Investment under uncertainty: Keeping one's options open, *Journal of Economic Literature*, **32**, 1816-1831

[6] **Petersen, E.R. and Taylor, A.J.,** 2000. An investment planning model for a new North-Central railway in Brazil, *Transportation Research Part A*, **35** (9), 847-862.

[7] **Dangl, T.,** 1998. Investment and capacity choice under uncertain demand, *European Journal of Operational Research*, **117**, 415-428.

[8] **Bellman, R.,** 1954. Some applications of the theory of dynamic programming – a review, *Operations Research*, **2**, 275-288

[9] **Stensland, G. and Tjostheim, D.,** 1989. Optimal investments using empirical dynamic programming with application to natural resources, *Journal of Business*, **62**, No. 1

[10] **Kroon L.G., Romeijn, H.E., Zwaneveld, P.J.,** 1997. Routing trains through railway stations: complexity issues, *European Journal of Operational Research*, **98**, 485-498

[11] **Dell'Olmo, P. and Lulli, G.,** 2003. A dynamic programming approach for the airport capacity allocation problem, *IMA Journal of Management Mathematics*, **14**, No. 3, 235-249 (15)

[12] **Bean, J.C., Higle, J.L., Smith, R.L.,** 1992. Capacity expansion under stochastic demands, *Operations Research*, **40**, No. 2

[13] **Hartman, J.C.,** 2004. Multiple asset replacement analysis under variable utilization and stochastic demand, *European Journal of Operational Research*, **159**, 145-165

[14] **Bean, J.C., Smith, R.L.,** 1985. Optimal capacity expansion over an infinite horizon, *Management Science*, **31**, No.12

[15] **Tzouramani, I., Mattas, K.,** 2004. Employing real options methodology in agricultural investments: the case of greenhouse construction, *Applied Economic Letters*, **11**, 355-359

[16] **Sqrensen, C.,** 1999. Dynamic asset allocation and fixed income management, *Journal of Financial and Quantitative Analysis*, **34**, No. 4

[17] **Prastacos, G.P.,** 1983. Optimal sequential investment decisions under conditions of uncertainty, *Management Science,* **29**, No. 1

[18] **Nihn, L.K., Hermes, N., Lanjouw, G.,** 2004. Investment, uncertainty and irreversibility: An empirical study of rice mills in the Mekong River Delta, Vietnam, *Economics of Transition*, **12** (2), 307-332

[19] **Petersen, E.R**., 2002. A highway corridor planning model: QROAD, *Transportation Research Part A*, **36**, 107-125

[20] **Erlenkotter, D.,** 1975. Capacity planning for large multi location systems: Approximate and incomplete dynamic programming approaches, *Management Science*, **22**, No.3

[21] **Erlenkotter, D.,** 1974. A Dynamic Programming Approach to capacity expansion with specialization, *Management Science*, **21**, No. 3, 360-363

[22] **Gulbrandsen, O.,** 2001. Optimal priority rating of resources – Allocation by dynamic programming

[23] **Ünsaler, O.,** 1984. Bazı deterministik dinamik programlama problemlerinin çözümü, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[24] **Gerçek, H., Demir, O.,** 2005. 4. Levent – Ayazağa – Maslak Sanayi Metro Project: Economic and Financial Feasibility Study, *Unpublished Report*, İstanbul

[25] **Denardo, E.V.,** 2001. Dynamic Programming Models and Applications, Prentice-Hall Inc., Englewood Clifts, New Jersey

APPENDIX

In this C++ computer program, it was decided to optimal investment policy (the required number of trains to be purchased in each year). The inputs of the code are the population size, interest rate, cost and profit information of the bus, minibus, car and subway, initial number of trains, transition probabilities, etc. The output of the program is the required number of trains to be purchased in each year and the profit of the project.

The source of the code

```
#include<iostream>
#include<iomanip>
#include<cmath>

using namespace std;

const double interest_rate = 0.12;
const int year = 26;
const int state = 3;
const int ds = 5;
int project_pop[year][state];

int population[year][state] = { { 0 , 0 , 0 } , { 316000000 , 327000000 , 335000000 }
, { 307000000 , 316000000 , 325000000 } , { 298000000 , 310000000, 316000000 }
, { 289000000 , 300000000 , 307000000 } , { 280000000 , 290000000 , 298000000 }
, { 272000000 , 282000000 , 289000000 } , { 264000000 , 275000000 , 280000000 }
, { 257000000 , 265000000 , 272000000 } , { 249000000 , 259000000 , 264000000 }
, { 242000000 , 250000000 , 257000000 } , { 235000000 , 245000000 , 249000000 }
, { 228000000 , 236000000 , 242000000 } , { 221000000 , 230000000 , 235000000 }
, { 215000000 , 223000000 , 228000000 } , { 209000000 , 217000000 , 221000000 }
, { 202000000 , 210000000 , 215000000 } , { 197000000 , 204000000 , 209000000 }
, { 191000000 , 198000000 , 202000000 } , { 185000000 , 192000000 , 197000000 }
, { 180000000 , 186000000 , 191000000 } , { 175000000 , 181000000 , 185000000 }
, { 170000000 , 176000000 , 180000000 } , { 165000000 , 171000000 , 175000000 }
```

```
, { 160000000 , 165000000 , 170000000 } , { 156000000 , 156000000 , 156000000 }
} ;

int car_km[year][state]= { { 0 , 0 , 0 } , { 19679157 , 19679157 , 19679157 } ,
{ 19183650 , 19183650 , 19183650 } , { 18700060 , 18700060 , 18700060 } ,
{ 18228109 , 18228109 , 18228109 } , { 17767525 , 17767525 , 17767525 } ,
{ 17318044 , 17318044 , 17318044 } , { 16879406 , 16879406 , 16879406 } ,
{ 16451357 , 16451357 , 16451357 } , { 16033649 , 16033649 , 16033649 } ,
{ 15626041 , 15626041 , 15626041 } , { 15228297 , 15228297 , 15228297 } ,
{ 14840185 , 14840185 , 14840185 } , { 14461479 , 14461479 , 14461479 } ,
{ 14082774 , 14082774 , 14082774 } , { 13704068 , 13704068 , 13704068 } ,
{ 13325363 , 13325363 , 13325363 } , { 12946657 , 12946657 , 12946657 } ,
{ 12567952 , 12567952 , 12567952 } , { 12189246 , 12189246 , 12189246 } ,
{ 11810540 , 11810540 , 11810540 } , { 11431835 , 11431835 , 11431835 } ,
{ 11053129 , 11053129 , 11053129 } , { 10674424 , 10674424 , 10674424 } ,
{ 10295718 , 10295718 , 10295718 } , { 9917013 , 9917013 , 9917013 } } ;

int decision[ds] = { 0 , 1 , 2 , 3 , 4 } ;

int travel_time_total[year]  =  { 0 , 78032435 , 74309932 , 70786686 , 67453443 ,
64301444 , 61322399 , 58508457 , 55852176 , 53346502 , 50984742 , 48760546 ,
46667884 , 44544154 , 42489500 , 40501635 , 38578370 , 36717612 , 34917359 ,
33175691 , 31490771 , 29860834 , 28884192 , 26759222 , 25284368 , 23858137 };
int policy[25] = { 1 , 1 , 1 , 1 , 2 , 1 , 1 , 0 , 1 , 2 , 1 , 1, 1 , 1 , 1 , 0 , 0 , 0 , 1 , 1 , 1 , 2
, 1 , 1 , 1};
int tot_revenue[year][state][ds];

class Metro{
public:
	double energy_cost ;
	int maint_cost ;
	double repair_cost ;
	double personnel_cost ;
	int personel_number;
```

```
int wage ;
int other_cost ;
int vehicle_cost;
double fee;
double project_length ;
double total_length  ;
double pop_rate_without ;
double pop_rate_project ;
double peak_hr_pop_rate ;
double peak_hr ;
double full_rate ;
int capacity ;
double pass_km_with ;
double pass_km_without ;
int train_number[year][state];
int cost[year][state][ds];
int peak_hr_pop_project[year][state];
int peak_hr_pop_without[year][state];
int tour_number_without[year][state];
int tour_number_with[year][state];
int train_km_without[year][state];
int train_km_with[year][state];



Metro (double a = 0.52, int b = 10275, double c = 0.04, double d = 0.096,

int e = 1000, int f = 15640, int g = 513135, double h = 4.6, double i = 16.64,

double j = 0.8, double k = 0.2, double l = 0.53, double m = 6.0,

double n = 0.9, int o=1872, double p = 5.86, double r = 5.1, int z = 1600000,

double v=0.70);
```

```
};
Metro :: Metro ( double a, int b, double c, double d, int e, int f, int g, double h,

double i, double j, double k, double l, double m, double n,int o, double p, double r,

int z, double v)
{
        energy_cost = a;
        maint_cost = b;
        repair_cost = c;
        personnel_cost = d;
        personel_number = e;
        wage = f;
        other_cost = g;
        project_length = h;
        total_length = i ;
        pop_rate_without = j;                      //0,8
        pop_rate_project = k;                      //0,2
        peak_hr_pop_rate = l;
        peak_hr  = m;
        full_rate = n;
        capacity = o;
        pass_km_with = p;
        pass_km_without = r;
        vehicle_cost = z;
        fee = v;
}

class Bus{
public:
        float op_costS ;
        float road_maint_costS;
        float road_accident_costS ;
        int capital_costS ;
```

```
	float travel_time_costS ;
	float environmental_costS ;
	float pop_rate ;
	float passenger_km  ;
	int capacity ;
	float tour_length ;
	int total_tour_length ;
	float bus_tour_number;
	float availability_rate ;
	int initial_number ;
	float conversion_factor ;
	int required_number_without[year][state];
	int required_number_project[year][state];
	int required_number_with[year][state][ds];
	int revenue[year][state][ds];
	int bus_pop_with[year][state][ds];
	int bus_pop_without[year][state];
	int bus_km_with[year][state][ds];
	int bus_km_without[year][state];
	int cost[year][state][ds];
	int purchased_train_pop[year][state][ds];

	Bus ( float a = 1.16, float b = 0.065, float c = 0.036208, int d = 200000,

	float e = 5.93, float f = 0.4, float h = 0.14, float i = 4.68, int j = 70,

	float k = 9.6, int l = 180, float m = 0.85, int n = 16, float y = 0.7 );
};

Bus :: Bus ( float a, float b, float c, int d, float e, float f,  float h, float i, int j, float k,

int l, float m, int n, float y)
{
	op_costS = a;
```

```
		road_maint_costS = b;
		road_accident_costS = c;
		capital_costS = d;
		travel_time_costS = e;
		environmental_costS = f;
		pop_rate = h;
		passenger_km = i;
		capacity = j;
		tour_length = k;
		total_tour_length = l;
		availability_rate = m;
		initial_number = n;
		conversion_factor = y;
}

class Minibus : public Bus{
public:
		int minibus_tour_number;
		int minibus_pop_with[year][state][ds];
		int minibus_pop_without[year][state];
		int minibus_km_with[year][state][ds];
		int minibus_km_without[year][state];

		Minibus ( float a = 0.29 , float b = 0.000258 , float c = 0.036208 ,

		float d = 63000.0 , float e = 5.93 , float f = 0.4 , float h = 0.1 , double i = 6.46,

		float j = 14.0 , float k = 9.6 , float l = 180 , float m = 0.90 , float n = 86.0 ,

		float y = 0.7 ) : Bus ( a , b , c , d , e , f , h , i , j , k , l , m , n , y ) { };
};

class Car{
public:
```

```
        double op_costS;
        double road_maint_costS ;
        double road_accident_costS ;
        double travel_time_costS ;
        double environmental_costS ;
        double conversion_factor ;
        int revenue[year][state][ds];

        Car ( double a = 0.39 , double b = 0.00001616 , double c = 0.036208 ,

        double d = 5.93 , double e = 0.15 , double f = 0.7 );
};

Car :: Car ( double a , double b , double c , double d , double e , double f )
{
        op_costS = a ;
        road_maint_costS = b ;
        road_accident_costS = c ;
        travel_time_costS = d ;
        environmental_costS = e ;
        conversion_factor = f ;
}



void main()
{
        int y, s , i , t ;
        Car c ;
        Minibus min ;
        Bus b ;
        Metro m ;
        int initial_train_number[25] ;
        initial_train_number[0] = 4 ;
```

```
int max_state ;                              // maximum value of the state
int V_max[year][state] ;
V_max[0][0] = 0 ;
V_max[0][1] = 0 ;
V_max[0][2] = 0 ;
double prob[3][3] = { { 0.6 , 0.3 , 0.1} , { 0.0 , 0.7 , 0.3} , { 0.0 , 0.5 , 0.5} };

int tot_prob[state] ;
int dec[year][state] ;
min.minibus_tour_number = min.total_tour_length / min.tour_length ;
b.bus_tour_number = b.total_tour_length / b.tour_length ;

for ( y = 1 ; y < year ; y++ )
{
for ( s = 0 ; s < state ; s++ )
{
        //population[y][s] = population[y][s] * (1 + rate) ;

        project_pop[y][s] = ( population[y][s]  * m.pop_rate_project *
        m.peak_hr_pop_rate ) / 330 ;

        // information for subway

        m.peak_hr_pop_project[y][s] = population[y][s]  *
        m.peak_hr_pop_rate * m.pop_rate_project ;

        m.peak_hr_pop_without[y][s] =  population[y][s]  *
        m.peak_hr_pop_rate * m.pop_rate_without ;

        m.tour_number_without[y][s] = population[y][s]  * m.full_rate *
        m.pop_rate_without / ( 330 * m.capacity * 2 ) ;

        m.train_km_without[y][s] = m.tour_number_without[y][s] * 330 * 8 *
        (m.total_length - m.project_length) * 2 ;
```

```
        m.tour_number_with[y][s] =  population[y][s] * m.full_rate  /   ( 330 *
        m.capacity * 2 ) ;

        m.train_number[y][s] = m.peak_hr_pop_without[y][s] * m.full_rate / (
        330 * m.capacity * 2 * m.peak_hr ) ;

        //information for minibus

        min.minibus_pop_without[y][s] = population[y][s] * min.pop_rate *
        m.pop_rate_project ;

        min.minibus_km_without[y][s] = min.minibus_pop_without[y][s] *
        min.tour_length / min.capacity ;

        min.required_number_project[y][s] = ( project_pop[y][s] * 0.5 *
        min.availability_rate ) / (min.minibus_tour_number * min.capacity ) ;


        // information for bus

        b.bus_pop_without[y][s] =  population[y][s] * b.pop_rate *
        m.pop_rate_project ;

        b.bus_km_without[y][s] = b.bus_pop_without[y][s] * b.tour_length /
        b.capacity ;

        b.required_number_project[y][s] = ( ( project_pop[y][s] * 0.5 ) / (
        b.bus_tour_number * b.capacity ) ) * b.availability_rate ;

        }
}
for ( y = 1; y < year ; y++ )
{
```

```
for (  t  = 0 ; t < 3 ; t++ )
{
        tot_prob[t] = 0 ;
}
for ( t = 0 ; t < 3 ; t++ )
{
for ( int k = 0 ; k < 3 ; k++ )
{
        tot_prob[t] = tot_prob[t] + prob[t][k] * V_max[y-1][k] ;
}
}
for ( s = 0 ; s < state ; s++ )
{
max_state = 0 ;

for ( i = 0 ; i < 5 ;  i++ )
{
        tot_revenue[y][s][i] =  population[y][s] * m.pop_rate_project * 1 ;
        //revenue from fee of bus and minibus

        if ( i != 0 )                    // for the project case
        {



        //subway cost

        m.train_km_with[y][s] = m.tour_number_with[y][s] * 330 * 8 *
        m.project_length * 2 + ( m.tour_number_with[y][s] /
        m.train_number[y][s] ) * decision[i] * 2 * 8 * 330 * m.project_length ;

        m.cost[y][s][i] = ( ( m.energy_cost * m.train_km_with[y][s] +
        m.maint_cost * m.project_length * pow( 1.02 , year – y ) +
        m.repair_cost * m.train_km_with[y][s] + m.personnel_cost *
        m.train_km_with[y][s] + m.personel_number * m.wage +
```

```
m.other_cost * m.project_length * pow( 1.02 , year – y ) )  /  5 ) + (
decision[i] *  m.vehicle_cost * 8 ) ;


//minibus revenue

min.purchased_train_pop[y][s][i] = decision[i] * m.capacity * 2 *
m.pop_rate_project * m.peak_hr_pop_rate * m.full_rate *
 ( m.tour_number_with[y][s] / ( m.train_number[y][s]+ decision[i] ) );

// population not met with subway

min.minibus_pop_with[y][s][i] = ( project_pop[y][s] –
min.purchased_train_pop[y][s][i] ) * 0.5 ;

min.required_number_with[y][s][i] = ( min.minibus_pop_with[y][s][i]
* min.availability_rate ) / ( min.minibus_tour_number * min.capacity
) ;

min.minibus_km_with[y][s][i] = ( population[y][s] * min.pop_rate *
m.pop_rate_project* min.tour_length / min.capacity) - (  (
min.required_number_project[y][s] -
min.required_number_with[y][s][i] ) * min.total_tour_length * 330 ) ;

min.revenue[y][s][i] = ( min.op_costS *
min.minibus_km_with[y][s][i] * min.conversion_factor  )  + (
min.road_maint_costS * min.minibus_km_with[y][s][i]  *
min.conversion_facto r) + ( min.road_accident_costS *
min.minibus_km_with[y][s][i]  ) + ( min.environmental_costS *
min.minibus_km_with[y][s][i] ) + ( (
min.required_number_project[y][s] -
min.required_number_with[y][s][i] ) * min.capital_costS *
min.conversion_factor  ) ;
```

```
//bus revenue

b.purchased_train_pop[y][s][i] = decision[i] * m.capacity * 2 *
m.pop_rate_project * m.peak_hr_pop_rate * m.full_rate *
(m.tour_number_with[y][s] / ( m.train_number[y][s] + decision[i] ) ) ;

b.bus_pop_with[y][s][i] = ( project_pop[y][s] -
b.purchased_train_pop[y][s][i] ) * 0.5 ;

b.required_number_with[y][s][i] = ( b.bus_pop_with[y][s][i] *
b.availability_rate ) / ( b.bus_tour_number * b.capacity ) ;

b.bus_km_with[y][s][i] = ( population[y][s] * b.pop_rate *
b.tour_length * m.pop_rate_project / b.capacity) - ( (
b.required_number_project[y][s] - b.required_number_with[y][s][i] )
* b.total_tour_length * 330 ) ;

b.revenue[y][s][i] = ( b.op_costS * b.bus_km_with[y][s][i] *
b.conversion_factor ) + ( b.road_maint_costS *
b.bus_km_with[y][s][i]  * b.conversion_factor ) + (
b.road_accident_costS  *  b.bus_km_with[y][s][i]  ) +
(b.environmental_costS  *  b.bus_km_with[y][s][i] ) + ( (
b.required_number_project[y][s] - b.required_number_with[y][s][i] )
* b.capital_costS * b.conversion_factor) + (
b.purchased_train_pop[y][s][i] * m.fee * 330 ) ;


//car revenue

c.revenue[y][s][i] = car_km[y][s] * c.op_costS * c.conversion_factor
+ car_km[y][s] * c.road_maint_costS * c.conversion_factor +
car_km[y][s] * c.road_accident_costS + car_km[y][s] *
c.environmental_costS ;
```

```
        // total revenue

        tot_revenue[y][s][i] =  b.revenue[y][s][i] + min.revenue[y][s][i] +
        c.revenue[y][s][i] + travel_time_total[y] - m.cost[y][s][i] ;
                    }
        }                   // end of i (decisions)

        for ( t = 0 ; t < 5 ;  t++ )
        {
        if ( tot_revenue[y][s][t] > max_state )
        {
        max_state = tot_revenue[y][s][t];       // maximum value of the state
        dec[y][s] = t;                          // decision which provides max profit

        }
        }

        V_max[y][s] = max_state + ( 1 / ( 1 + interest_rate ) ) * tot_prob[s] ;

        }               // end of s (states)
}                       // end of y (years)


cout <<  "****************************************** "  <<  endl ;
cout <<  "********  initial number of trains = 4  ******* "  << endl ;
cout <<  "************************************** "  <<  endl << endl ;

for ( y = 1; y < year ; y++ )
{
        cout  <<   "Year: "  <<   y   <<   endl;

        initial_train_number[y] = initial_train_number[y-1] +
        dec[year-y][policy[y]] ;
```

```
        cout <<"  \n          Decision:"  <<  dec[year-y][policy[y]] <<  "    " <<
        "Number of Trains:"  <<  initial_train_number[y]  <<  endl  <<  endl ;

    }

    cout << "  \n\n  ---------> with profit: " << V_max[25][policy[0]] << endl <<
    endl << endl ;

}
```

RESUME

Miss İnce was born in Kırıkkale in 1980. She has graduated from Kırıkkale Fen Lisesi in 1998. She has completed her undergraduate work at the Department of Industrial Engineering of İstanbul Kültür University. In year 2001, she has started to the double major program under the Department of Computer Engineering of the same university. She has graduated from the Department of Industrial Engineering in 2003 with the first rank and she has started to work as Research Assistant in the same university. She has completed her study in the Department of Computer Engineering in 2004 with the second rank. In year 2004, she has enrolled to the Master Program of Industrial Engineering Department of İstanbul Technical University. She has been still working in İstanbul Kültür University. She has participated in many conferences, i.e.:

- İnce, Ö., Ayağ, Z., Özdemir, R.G., *Montaj Hatlarında Ara Stok Alanı Kapasitesi Belirleme,* YA/EM 2005 25.Ulusal Yöneylem ve Endüstri Mühendisliği Kongresi, Koç University, 04-06 July 2005, İstanbul.
- İnce, Ö., Ayağ, Z., Özdemir, R.G., *A Simulation Model for Performance Evaluation in the Supply Chain,* International Logistics and Supply Chain Congress, 2005, Galatasaray University, Faculty of Economics and Administrative Sciences, Loder Logistics Association, University Paris 1-Sarbonne, 23-24 November 2005, İstanbul.
- Özdemir, R.G., Ayağ, Z., Kula, U., İnce, Ö., *0-1 Tamsayılı Hedef Programlama ile Tedarik Zinciri Yönetiminde bir Vaka Çalışması*, YA/EM 2006 26. Ulusal Kongresi, Kocaeli University, 3-5 Temmuz 2006, Kocaeli.