Open access

Kjetil Kåre Haugen

STOCHASTIC DYNAMIC PROGRAMMING

UNIVERSITETSFORLAGET

Stochastic Dynamic Programming

Stochastic Dynamic Programming

Kjetil K. Haugen

The book was first published in 2016 by Universitetsforlaget.

The book is published with support by Molde University College, Specialized University in Logistics.

ISBN printed edition (print on demand): 978-82-15-02670-1
ISBN electronic pdf-edition: 978-82-15-02671-8

Enquiries about this publication may be directed to:
post@universitetsforlaget.no
www.universitetsforlaget.no

Cover Layout: Universitetsforlaget
Layout: Kjetil Kåre Haugen

Preface

This book was my first serious academic project after finishing my PhD-thesis (Haugen, 1991) back in 1991. The primal subject for this thesis was an application of stochastic dynamic programming in petroleum field scheduling for Norwegian oil fields. Soon after defending my thesis, I was contacted by a US publisher, asking whether I would like to write a chapter in a new OR[1]-series of books. This chapter should be related to stochastic (or probabilistic) dynamic programming and cover around half of the planned volume, the other half covering deterministic dynamic programming. Being somewhat young, inexperienced and ambitious, I said yes to the job.

Later on, after finishing this work, it turned out that the book series was cancelled. Naturally, I was not happy about such a decision. However, the job I did back in 1991–1994, turned out to be of decent quality – even today. This new version of the book covers most classical concepts of stochastic dynamic programming, but is also updated on recent research. A certain emphasis on computational aspects is evident.

The book discusses both classical probabilistic dynamic programming techniques as well as more modern subjects, including some of my own results from my PhD. As such, the book can perhaps be categorized as a classic monograph. As a consequence, some knowledge of probability calculus as well as optimization and economic theory is needed for the general reader. However, the book can (with some added material) be used as a text-book on the subject, but as mentioned previously, it is not written as one.

Kjetil K. Haugen

Trondheim, Molde
1991–1994, September 2015

[1] operations research

Acknowledgements

The most important person to thank is my PhD-supervisor Prof. Bjørn Nygreen at NTNU, Trondheim Norway. He was the main motivator behind my interest in operations research in the first place. Acting as my supervisor, he must also take much of the responsibility for my interest in stochastic dynamic programming as a separate subject. Furthermore, Prof. Stein Wallace, a previous colleague at both NTNU and HiMolde needs to be thanked for his continuous enthusiasm for everything uncertain, or stochastic as Stein likes to name it.

Previous colleagues at NTNU and SINTEF, perhaps especially Marielle Christiansen, Morten Lund, Nils J. Berland and Thor Bjørkvoll also need to be thanked for stimulating and encouraging discussions related to the project.

Colleagues at my present institution, Molde University College, Specialized University in Logistics also deserve thanks, especially for parts of my later work involving several of the topics discussed in this book. Some examples – strongly related to techniques, thoughts and methods in this book – may be found in (Haugen et al., 2007*a*), (Haugen et al., 2007*b*), (Haugen and Berland, 1996), (Haugen, 1996), (Haugen et al., 1998), (Haugen et al., 2001), (Haugen et al., 2010), (Haugen et al., 2012), (Lanquepin-Chesnais et al., 2012).

I am very grateful to all of you!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Dynamic programming may be viewed as a general method aimed at solving multistage optimization problems. Probabilistic or stochastic dynamic programming (SDP) may be viewed similarly, but aiming to solve *stochastic* multistage optimization problems. A stochastic multistage optimization problem is a problem where one or several of the parameters in the problem are modelled as stochastic variables or processes. As many of the problems in the field of operations research deals with future planning and many future events are hard to predict with certainty, it is not hard to imagine the importance of SDP and related techniques. According to Bellman and Dreyfus (Bellman and Dreyfus, 1962) this – that is; the stochastic case – is always the actual situation.

The history of SDP is closely related to the history of dynamic programming. In addition to Bellman and Dreyfus (Bellman and Dreyfus, 1962), significant contributions were made by Howard (Howard, 1960*b*), (Howard, 1960*a*) and d'Epenoux (d'Epenoux, 1960) in the late fifties and early sixties.

Today, most standard textbooks on operations research include SDP – at least to some extent: See for example Ravindran, Phillips and Solberg's latest edition (Ravindran et al., 1987) or Hillier and Lieberman (Hillier and Lieberman, 1989). However, these type of books tend to be sparse in their coverage of the topic. An excellent introductory text is written by Hastings (Hastings, 1973). A more modern approach, by Ross, can be found in (Ross, 1983).

We will return to more recent contributions later in the chapter.

In the next section, we will present an illustrative example. This example will be solved first by a decision tree approach and later by a SDP approach. We choose to do this as the decision tree approach is simple to grasp and widely known. We also get a nice way of comparing the two methods.

1.1 An illustrative example

Assume that a person owns an object which he wants to sell. The sale is taking place over a fixed set of time periods. In each time period, the price is assumed to be stochastic. We also assume that the price is identically and independently distributed over all possible sales periods and that a fixed cost is associated with selling the object. The problem facing our friend, is then to decide when to sell the object.

An important fact to consider, dealing with these type of problems, is what we might call the “information - decision structure”. That is; when is new information gathered and when must decisions be made. In the problem outlined above, at least two possibilities exist. The price is revealed before a selling decision is made. That is; in the given time period, the seller can observe the outcome of the stochastic price before the decision on whether to sell or not is made. Alternatively, we could face a situation where the seller must decide on selling or not before the price he gets is revealed.

The first situation might be named the “operating” situation. Typically when making operational decisions we observe some outcomes and make corrective actions. The other situation may be named the “investment” situation. That is, we have to make a decision before the outcome is known. Obviously most practical problems have structures involving both types, but to make things simple we stick to one of the situations for our example.

Surely, which structure we choose is determined by the practical situation we want to model. A natural choice in our example is to assume that the price is revealed before the selling decision is made. We could for instance assume that our example is a house sale model. The salesman has got a new job and needs to sell his house before he moves. In each time period one bidder arrives with an uncertain bid. Given an observed bid, he then has to decide whether to sell his house or not. Given that he decides to wait the bidder leaves and does not return.

Table 1.1 gives necessary data to the example. We observe that the price can take 3 values; 200, 150 or 50. The cost is assumed fixed and therefore independent of price.

Table 1.1: Data for the house selling example. (All numbers in $1000.)

probabilities	*prices*	*cost*
0.25	200	100
0.55	150	100
0.20	50	100

1.2 Solving the example by decision trees

This section will solve the example presented in 1.1 and introduce the concept of conditional solutions.

A decision tree is a graphical way of expressing decision problems which are not too complex. At the same time, the graphical approach gives a natural solving procedure. Decision trees are treated in almost any textbook of OR or decision theory. Refer for instance to Watson and Buede (Watson and Buede, 1987).

A decision tree consists of chance and decision nodes. A chance node is a tree structure picturing the stochasticity, while a decision node describes possible decisions. A circle is normally used to define a chance node while a square is used for a decision node. According to our discussion above, the decision tree for our example should be composed of structures as those showed in Figure 1.1.

To complete our model we need to decide on the number of time periods the sales offer is valid. Assuming that we use 2 periods, the full decision tree is showed in figure 1.2.

Note that it is impossible to sell the object several times. Therefore, if the object is sold in period 1, the decision tree stops. Now we are in a position to solve our example applying the decision tree approach. In order to do that, we need to put relevant numbers into the tree. We start at the end of the time horizon (period 2). Then the possible decision – in each decision node – is to sell or wait. Waiting implies not selling now (or ever) as this is the last period. Figure 1.3 shows the situation for the upper branch of the tree.

A sensible thing to do is to choose the decision in each decision node that maximizes profit. Doing this we obtain a profit of 100, 50 and 0 in each of the decision nodes. Now we continue to the time period one step earlier – period 1. However, the decision problem facing us now is a bit more tricky. We have

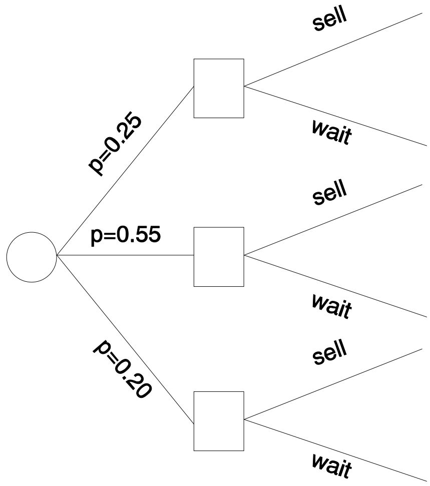


Figure 1.1: Basic decision/chance node structure for the house selling example

to choose between a certain outcome of 100 – obtained by selling in period 1 or an uncertain outcome of $(100, 50, 0)$ with probabilities $(0.25, 0.55, 0.20)$ respectively.

These problems are very popular in decision theory literature. They are often used to motivate utility theory – see for instance (Watson and Buede, 1987). We shall not use time to discuss these matters, just note that such a decision problem is not necessarily straightforward. (We will return to utility theory in section 1.2.) One possible way of thinking is that the decision maker should make a decision that yields the best average result. In such a situation, maximizing the expected value is a natural choice. Figure 1.4 sums up this discussion.

Continuing in this manner we obtain the solution. Table 1.2 sums up the solution. (Note that the period 2 solutions are conditioned on not selling the house in period 1. If the house is sold in period 1, we do nothing in period 2.)

We note a very important fact from table 1.2. The solution is conditioned on the stochastic variable. That is, depending on what instances we observe in future realisations of the stochastic variable, we plan to make different decisions. This fact is important to grasp when it comes to understanding stochastic optimization. If we compare our solution to the solution structure of a deterministic optimization problem the big difference is that we get sev-

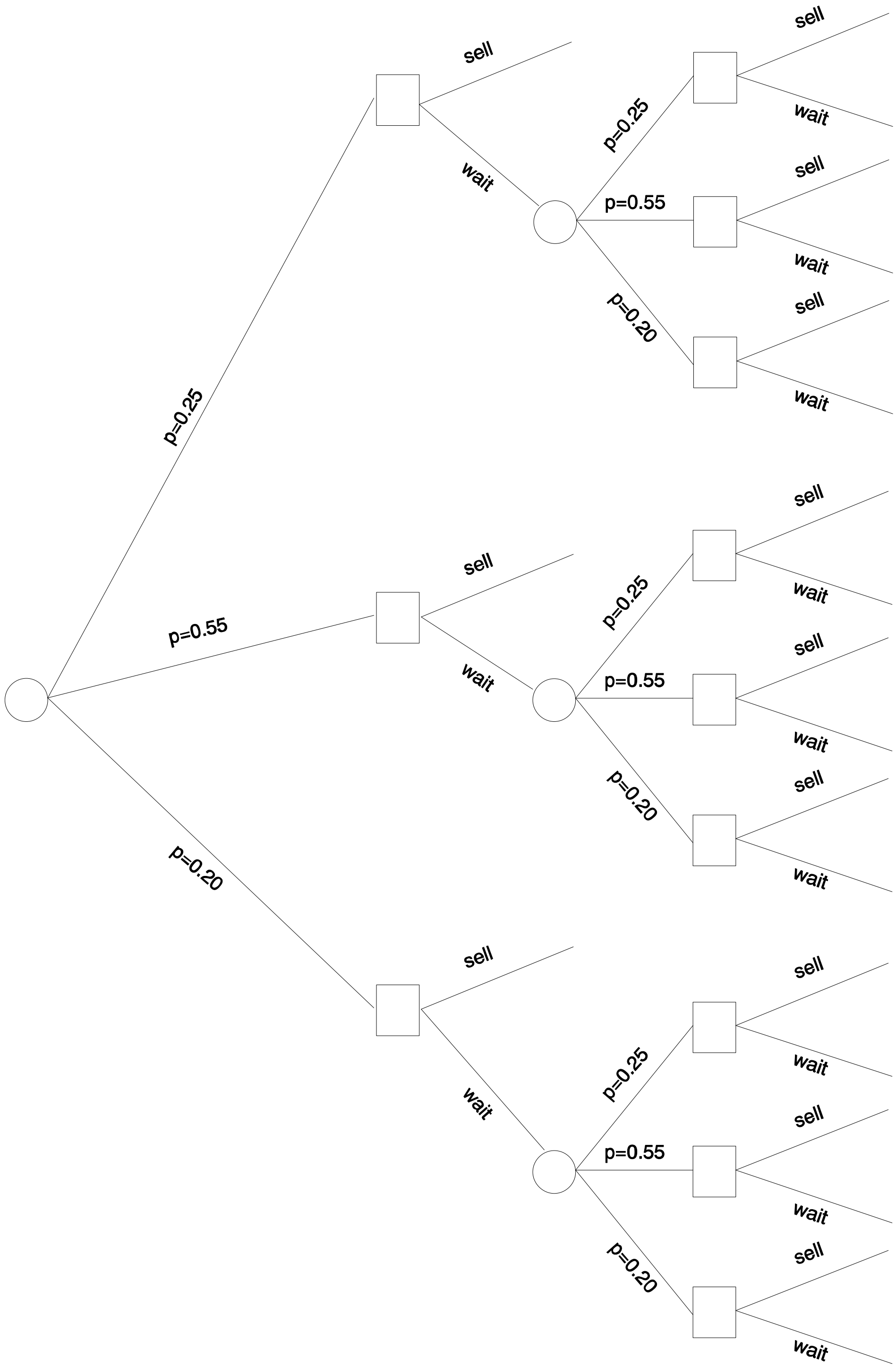


Figure 1.2: Full decision tree for the house selling example

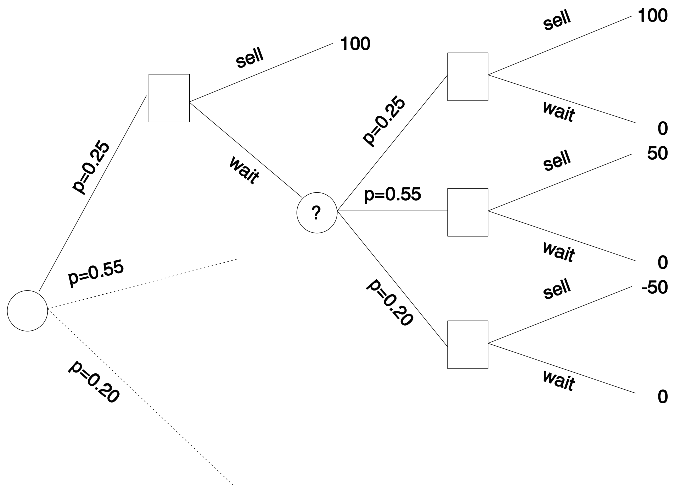


Figure 1.3: Upper branch of decision tree for the house selling example

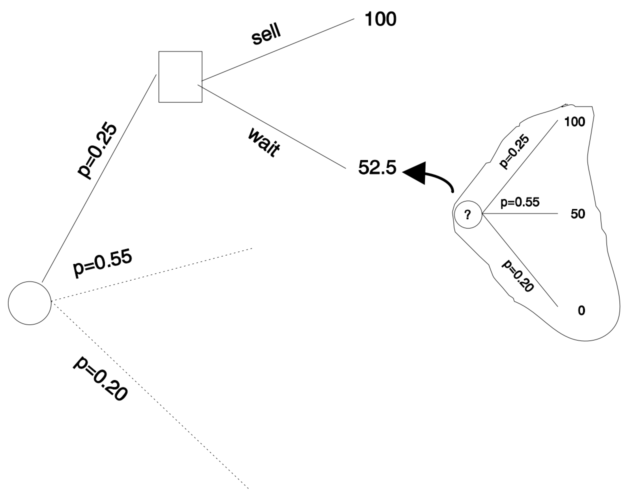


Figure 1.4: Evaluating uncertain outcomes by expected values in a decision tree

Table 1.2: Solution for the house selling example.

	high price	*medium price*	*low price*
period 1	sell	wait	wait
period 2	sell	sell	wait

eral conditional solutions. That is, we make alternative plans for all possible futures. As opposed to the deterministic case where we only get one plan.

We also observe another important fact from table 1.2. The optimal strategy is different between the two time periods. We see that it differs in the optimal decision given a medium price observation. The salesman waits in period 1 while he sells in period 2. This is an important distinction which is treated well in the literature of SDP. Especially if we look at infinite horizon problems, the possibility of obtaining *stationary policies* will prove to be interesting. A *stationary policy* is a solution which is unconditioned on time but conditioned on state. We will return to these topics later.

1.3 Solving the example by SDP

In this section we will introduce the fundamental equation of SDP and solve the example introduced in section 1.1 by SDP. We will also compare SDP with the decision tree method from section 1.2.

We will explain SDP in close connection to the decision tree calculations made in section 1.2. Table 1.3 defines a function which we call $R(i, a)$.

Table 1.3: Definition of the immediate return function $R(i, a)$ for the house selling example.

$R(i, a)$	$i =$*high price*	$i =$*medium price*	$i =$*low price*	$i =$ "*sold*"
$a =$*sell*	100	50	-50	-
$a =$*wait*	0	0	0	0

If we return to figure 1.3 we observe that the function corresponds with the decision nodes in period 2. That is, these numbers state possible returns for all *states* i and *actions* a. Note that we need to include a state telling us whether we have sold the object earlier. Note also that the state value "*sold earlier*" implies a certain immediate return of 0. This state value is implicitly incorporated in the decision tree as the tree stops after each selling decision. The next thing we did in our decision tree approach was to find, for all states i, the decisions that maximized immediate return. Mathematically, we can describe this operation as follows:

$$V_2(i) = \max_a R(i,a) \tag{1.1}$$

By performing this maximization over a we obtain a function in i which we have called $V_2(i)$. The actual values of the $V_2(i)$ function is displayed in table 1.4.

Table 1.4: $V_2(i)$ for the house selling example.

	$i =$*High Price*	$i =$*Medium price*	$i =$*Low Price*	$i =$ "*sold earlier*"
$V_2(i)$	100	50	0	0

The next step we performed in the solution process, was to move to period 1. Again we maximized over all states but also including expected values of waiting with the sales decision to period 2. If we look at the high price state, the actual computation we performed was;

$$\max\{100, 52.5\} \tag{1.2}$$

or in more formal terms:

$$\max\left\{R(i, a = \text{“}\textit{sell}\text{”}), \sum_i p_i V_2(i)\right\} \tag{1.3}$$

Alternatively, the same could be achieved by the following;

$$\max_a \left\{R(i,a) + \sum_i p_i V_2(i)\right\} \tag{1.4}$$

Note that the i subscript only takes on the three stochastic values in period 1 as the fourth alternative from period 2 – "*sold earlier*" is impossible. If we call the value function in period 1 $V_1(i)$, equation (1.4) becomes:

$$V_1(i) = \max_a \left\{R(i,a) + \sum_i p_i V_2(i)\right\} \tag{1.5}$$

Table 1.5 gives the results of performing the actual calculations.

Table 1.5: $V_1(i)$ for the house selling example.

	$R(i,a)+\sum_i p_i V_2(i)$		
	$i=$*High Price*	$i=$*Medium price*	$i=$*Low Price*
sell	100+0	50+0	-50+0
wait	0+52.5	0+52.5	0+52.5
$V_1(i)$	100	52.5	52.5

Comparing table 1.5 and 1.4 with table 1.2 we observe that our latter approach produced the same answer as the decision tree approach.

If we look at equation (1.5) we see that we have identified a recursive method of computing the value function at different periods of time in our problem.

Ross (Ross, 1983) defines the *optimality equation* as follows:

$$V_n(i) = \max_a \left[R(i,a) + \sum_j P_{ij}(a) V_{n+1}(j) \right] \tag{1.6}$$

Here, a is an action chosen from a set A, $R(i,a)$ is the immediate return obtained by taking action a in state i, $P_{ij}(a)$ is the probability of reaching state j given that state i is observed at stage n and action a is taken while $V_n(i)$ is the value function in state i at stage n. Comparing equations (1.6) and (1.5) we observe that they are quite similar. The only difference is that equation (1.6) allows more general probability definitions. If we compare the term $P_{ij}(a)$ in equation (1.6) with P_i in equation (1.5) we observe that equation (1.6) allows the addition of two effects.

- The probability of reaching a state may depend on the observed state.
- The probability of reaching a state may depend on the action taken.

To be formal, the term $P_{ij}(a)$ in equation (1.6) states that the stochastic mechanism affecting our optimization problem is a family of discrete Markov processes with transition matrices $P_{ij}(a)$. Returning to our initial example we note that using this terminology, the price of our object can be described as follows:

$$P_{ij}(a) = \begin{array}{c|ccc} & H & M & L \\ \hline H & 0.25 & 0.55 & 0.20 \\ M & 0.25 & 0.55 & 0.20 \\ L & 0.25 & 0.55 & 0.20 \end{array} \quad \forall a \in \{\text{“}sell\text{”}, \text{“}wait\text{”}\} \tag{1.7}$$

This means that if we observe a high, medium or low price (H,M, L) in period 1, then the probability of observing the same set of prices in period 2 is independent of the observation in period 1.

Assume alternatively that $P_{ij}(a)$ had the following structure:

$$P_{ij}(a) = \begin{array}{c|ccc} & H & M & L \\ \hline H & 1-\alpha & \frac{11}{15}\alpha & \frac{4}{15}\alpha \\ M & 0.25 & 0.55 & 0.20 \\ L & \frac{5}{16}\beta & \frac{11}{16}\beta & 1-\beta \end{array} \quad \forall a \in \{\text{“}sell\text{”}, \text{“}wait\text{”}\},\ \alpha \in [0, 0.75],\ \beta \in [0, 0.8] \tag{1.8}$$

How may equation (1.8) be interpreted? Suppose we suspect that the price we get selling a house one year depends on the price we get the year before. A simple – but somewhat sensible assumption – may be to say that a high price one year would “lead” to a high price next year and vice versa. Equation (1.8) reflects such a reasoning. Note that if we choose $\alpha = 0.75$ and $\beta = 0.8$ we obtain equation (1.7). Choosing the parameters α and β equal to zero produces the other extreme – high and low price as absorbing states. Surely we could try to give precise estimates on α and β if we had additional information but that may be difficult. One way to attack such a problem may be to try to solve the problem parametrically. That is, solve it for all possible values of the parameters. In this example, this turns out to be very simple so we might as well do it.

The calculations which lead to table 1.4 does not change. However, a new version of table 1.5 is shown in table 1.6

The $V_1(i)$ -values in table 1.6 are obtained as follows:

$$Max \left[100, 100 - \frac{190}{3}\alpha\right]_{\alpha \in [0,0.75]} = 100 \tag{1.9}$$

$$Max \left[-50, \frac{525}{8}\beta\right]_{\beta \in [0,0.8]} = \frac{525}{8}\beta \tag{1.10}$$

This might be a somewhat surprising result. The solution structure is unchanged. That is, the optimal conditional decisions in table 1.2 are the

Table 1.6: $V_1(i)$ for the house selling example with alternative definition of $P_{ij}(a)$

	$R(i,a)+\sum_i p_i V_2(i)$		
	i =*High Price*	i =*Medium price*	i =*Low Price*
sell	100	50	-50
wait	100-$\frac{190}{3}\alpha$	52.5	$\frac{525}{8}\beta$
$V_1(i)$	100	52.5	$\frac{525}{8}\beta$

same. The only difference is the expected value of waiting in period 1 given a low price, which is moving linearly from 0 to 52.5 for permitted values of β.

Surely this is not a general result. However, it stresses an important point. *It is normally the leap from a deterministic model to a stochastic model that give a dramatic different solution, not the actual stochastic mechanism applied.*

Action dependence may be harder to imagine in our house selling example. Obviously we might imagine situations where a decision on selling a house or not may affect our predictions on the future price. However, as our alternative decision is to sell today, this example becomes somewhat artificial. If we alternatively look at the possible decisions we have modelled, it should be obvious that if we do not sell the house today, there is a whole pile of actions we can take to try to change the price we may get tomorrow. We can paint it, advertise more or differently, hire someone to set the house on fire and so on. Such situations may be compared to insurance. That is, we can – by actions – change our perspective of the future but the future is still hard to model without using probabilistic techniques. Weather phenomena fit nicely into this pattern. We can affect the probability of rain by locating our business in Sahara or New Foundland but still some probability of rain exists in Sahara. We can guard ourselves against theft by engaging a guard bureau, buying a gun or moving away from New York, but still some probability of theft exists. Surely we can also use insurance – not to remove the probability of the event – but to vary the consequences of the event.

Let us make a small change in our example to exemplify this. Suppose we now include the possibility of painting our house if we do not sell it.

Assume that $P_{ij}(a)$ is given as in equation (1.7) for $a \in \{\text{“}sell\text{”}, \text{“}wait\text{”}\}$ while equation (1.11) gives the transition matrix for the third alternative; $a =$ *“wait and paint”*.

$$P_{ij}(a) = \begin{array}{c|ccc} & H & M & L \\ \hline H & 0.90 & 0.10 & 0.00 \\ M & 0.50 & 0.40 & 0.10 \\ L & 0.20 & 0.50 & 0.30 \end{array} \tag{1.11}$$

We may explain equation (1.11) as follows: If we observe a high price today, the market (potential buyers) believes that our house is a good bargain and painting makes it even better. If we observe a medium price today, painting extends the value of the house but not so much. If – on the other hand – a low price is observed today, potential buyers have identified our house as a shack and trying to paint it makes the markets perception even worse. In this situation, potential buyers perceive our painting strategy as an attempt to hide the fact that the house is in a real bad state.

Let us then resolve the example under these assumptions. Suppose the cost of painting is unknown – we call it c. (It seems sensible to assume that $c > 0$. Nobody would pay us for the honour of painting our house.) If we perform the same type of calculations as those leading to table 1.6 we obtain table 1.7.

Table 1.7: $V_1(i)$ for the house selling example with alternative definition of $P_{ij}(a)$

	$R(i,a) + \sum_i p_i V_2(i)$		
	$i =$ *High Price*	$i =$ *Medium price*	$i =$ *Low Price*
sell	100	50	-50
wait	52.5	52.5	52.5
wait and paint	$95 - c$	$70 - c$	$45 - c$
$V_1(i)$	100	$\max[52.5, 70 - c]$	52.5

The results from table 1.7 show that the maximum price the owner of the house would be interested in paying for the painting operation is 17.5. This number is obtained as follows: Given that $c > 17.5$, the solution is unchanged hence no point in painting the house. Alternatively if $c < 17.5$, $70 - c > 52,5$ and it would pay off to use the painting strategy.

The classical problem which is treated in the literature to exemplify action dependence, is that of machine maintenance. See for instance Hillier and Lieberman (Hillier and Lieberman, 1989).

Chapter 2

SDP – basic concepts

In section 1.1 we have introduced SDP by an illustrative example. This chapter will try to sum up and define necessary terms.

2.1 Comparing Stochastic and Deterministic DP

If we compare the examples we have looked at with the chapter in Sandblom et al. (Sandblom et al., To appear – Never did) on deterministic dynamic programming, the fundamental concepts are unchanged. That is; concepts as **stages**, **states**, **stage transformation function** and **the principle of optimality** remains with unchanged meaning. The differences may be summed up as follows. If we use the notation from Sandblom et al. (Sandblom et al., To appear – Never did), the **stage transformation function** may be expressed as

$$x_{k-1} = t_k(x_k, d_k) \tag{2.1}$$

for the deterministic case. In the stochastic case, we may generalize this functional relationship as

$$x_{k-1} = t_k(x_k, d_k, \xi_k) \tag{2.2}$$

where ξ_k is a stochastic variable. (Note that we adopt the backward recursion scheme.) We do not know with certainty which state the system transforms into given state and decision at the former stage. The transformation is governed by a stochastic variable. Recall our example; if we made the decision to wait at a certain stage, observing the price (state value) simultaneously, the state value (price) at the next stage was not determined with certainty.

The other implication of the stochastic assumption relates to the calculation of the recursive relationship. As pointed out earlier, given an uncertain transition from one stage to the other, we need to decide how to deal with uncertain outcomes. In our examples, we have used expected value as a mean of dealing with uncertainty. As discussed in subsection 1.2 one answer to this problem is utility theory.

2.2 Illustrating expected utility

Expected utility can be defined as follows:

$$E[U(\xi)] = \int_{\Xi} u(\xi) f(\xi) d\xi \tag{2.3}$$

$u(\xi)$ is the utility function, ξ is a (multidimensional) stochastic variable, Ξ is the support of the stochastic variable while $f(\xi)$ is the density function. Note immediately that by choosing $u(\xi) = \xi$, equation (2.3) computes the expected value of the stochastic variable ξ.

It can be shown (refer for instance to Baumol (Baumol, 1972)) that an individual who accepts five simple axioms will choose to maximize expected utility. Let us look at a few simple consequences of the utility maximization hypotheses. Maximizing expected utility implies the following:

$$X \succ Y \Leftrightarrow u(X) > u(Y) \tag{2.4}$$

That is, if the certain event X is strictly preferred to ($\succ$) the certain event Y then the utility of X should be larger than the utility of Y and vice versa. Let us multiply the inequality part of equation (2.4) with a positive constant a and add another constant b.

$$au(X) + b > au(Y) + b \tag{2.5}$$

or

$$X \succ Y \Leftrightarrow au(X) + b > au(Y) + b \tag{2.6}$$

Equation (2.6) says that we cannot measure utility along an absolute scale. Or alternatively; any two points on a utility scale can be chosen arbitrarily.

Another concept that we need is that of greed. If our events X and Y are measured along the same scale – say money. Then we assume that if $X > Y \Leftrightarrow u(X) > u(Y)$. Alternatively we may state this assumption as $\frac{du(\xi)}{d\xi} > 0$.

Let us now return to our house selling example and show how expected utility may change our solution. Given a general utility function $u(\xi)$ we can adjust table 1.4 and obtain table 2.1

Table 2.1: $V_2(i)$ for the house selling example with utility function, $u(\xi)$

	$i =$*High Price*	$i =$*Medium price*	$i =$*Low Price*	$i =$*"sold earlier"*
$V_2(i)$	u(100)	u(50)	u(0)	u(0)

Continuing the calculations to stage 1, we get table 2.2

Table 2.2: $V_1(i)$ for the house selling example with utility function, $u(\xi)$

	$R(i,a) + \sum_i p_i V_2(i)$		
	$i =$*High Price*	$i =$*Medium price*	$i =$*Low Price*
sell	$u(100)$	$u(50)$	$u(-50)$
wait	$.25 + .55u(50)$	$.25 + .55u(50)$	$.25 + .55u(50)$
$V_1(i)$	$u(100)$	$Max[u(50), .25 + .55u(50)]$	$.25 + .55u(50)$

The numbers in table 2.2 may need some further explanation. The first line gives the utility values of selling now (at stage 1) yielding $u(100)$, $u(50)$ and $u(-50)$. If we wait at stage 1 we have to calculate the expected utility of that decision yielding;

$$EU(\text{"}wating\text{"}) = .25u(100) + .55u(50) + .20u(0). \tag{2.7}$$

However, we can utilize the implications of equation (2.6) and assign numbers to $u(100)$ and $u(0)$. As table 2.2 indicates, we have chosen $u(100) = 1$ and $u(0) = 0$.

To obtain the $V_1(i)$ values, we utilize the greedy assumption. That is;

$$u(100) > u(50) > u(0) \Rightarrow u(100) > .25u(100) + .55u(50) + .20u(0) \quad (2.8)$$

and

$$.25u(100) + .55u(50) + .20u(0) > u(-50) \quad (2.9)$$

The point of introducing utility theory is to show that we can use the theory to design different attitudes towards risk for the decision maker. If we look at our example, we see that the only decision that may change, is that of waiting in period 1 given a medium price observation. Surely, this should not be surprising. In period 2 the decisions are unchanged which should be more or less obvious. Given a high price observation in period 1, we will not obtain higher utility by waiting. Therefore we sell, independent of attitude towards risk. The same happens if we observe a low price in period 1. If we sell, we obtain $u(-50)$ with certainty. By waiting, we are better off as the worst thing that can happen is $u(-50)$ in period 2. Therefore, the interesting state is – as mentioned above – a medium price in period 1. In this situation, the decision maker is facing either getting $u(50)$ now or the uncertain outcome $[u(100), u(50), u(0)]$ with probabilities $[.25, .55, .20]$. Let us calculate the value of $u(50)$ needed for the decision maker to be indifferent between the two decision alternatives. Surely this value can be found by

$$u(50) = .25 + .55u(50) \Rightarrow u(50) \approx .56 \quad (2.10)$$

Let us now plot the three points we have found for our utility function. Figure 2.1 shows this graph.

As figure 2.1 indicates, the three points $u(0) = 0$, $u(50) = 0.56$ and $u(100) = 1$ shows a concave pattern. (This is indicated by the dashed line in figure 2.1.) The literature denotes this phenomena *risk aversion*. To sum up: The decision maker chooses the uncertain outcome if $u(50) < 0.56$. Then the degree of risk aversion is – in a sense – not big enough. On the other hand, if $u(50) > 0.56$ the certain outcome is chosen.

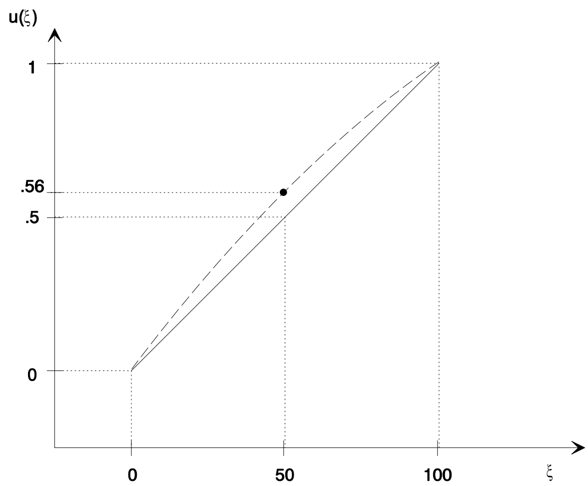


Figure 2.1: Graph of utility function given indifference between risky and certain decision

Chapter 3

SDP - Benefits

This chapter discusses some beneficial topics of SDP as opposed to the method of decision trees or alternative stochastic programming methods.

3.1 SDP versus Decision trees

Section 1.2 and 1.3 illustrates that certain problems allow application of either decision trees or SDP. In this perspective, it may be interesting to determine whether any of the methods is advantageous. If we look back on the decision tree in figure 1.2, we observe that this method involves a full enumeration of all possible decisions and states. Suppose we extend our time horizon in the house selling example to 15 periods. Then, the number of end leaves in the decision three (sell and wait nodes as in figure 1.2) would be 28,697,814 – an enormous number. This number is obtained by observing that each wait node produces 6 new sell and wait nodes. As half of the nodes at t are wait nodes, the following recursive equation holds:

$$n_{t+1} = 6\left(\frac{n_t}{2}\right) = 3n_t \text{ or } n_t = 6(3)^{t-1} \forall t \in \{0, 1, \dots, T-1\} \tag{3.1}$$

Substituting $T = 14$ into equation (3.1) yields 28,697,814.

If we compare this to the SDP approach, we observe that the calculations at each stage would not grow exponentially as in the decision tree. We would still be doing computations leading to tables close to table 1.5 – adding the fourth $i =$"*sold earlier*" state. That is, from a computational point of view, we may save a lot of work by applying SDP as opposed to decision trees.

Surely, this effect is not due to the stochasticity of the problem. An excellent example comparing the complexity of a deterministic decision tree and the corresponding deterministic DP may be found in Wallace and Kall (Kall and Wallace, 1994).

Hence, one important difference between SDP and decision trees is a computational superiority in favour of SDP.

3.2 Nondiscrete state space

So far we have looked at problems with discrete state and decision space. In order for a decision tree approach to work, this is more or less a necessity. However, SDP may work well also in the case of a continuous state space. The methods of Stochastic Programming, may also have difficulties dealing with non-discrete state space descriptions.

Suppose we make a slight change in the description of our house selling example. Assume that the price is described by a continuous density function. Let us make things simple and use a uniform density. That is; $p \sim U[50, 200]$. To simplify, we introduce another stochastic variable called x defined as

$$x = p - C \tag{3.2}$$

Where C is the deterministic cost associated with the selling decision from table 1.1. Consequently, x is uniform; $x \sim U[-50, 100]$, and it gives the net profit obtained by performing a sale decision. The density of x becomes

$$f(x) = \begin{cases} \frac{1}{150} & x \in [-50, 100] \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

Let us resolve the example under the new assumption. In period 2, we will sell if the observed x is positive. Alternatively, we wait or do noting as this is the last period. Mathematically;

$$V_2(x) = \begin{cases} x & x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

Note that the value function V_2 is a continuous function of the continuous state variable x. Continuing to period 1 we find $V_1(x)$ by the recursion

$$V_1(x) = \max\left[x, E\left\{V_2(x)\right\}\right] \tag{3.5}$$

(Formally, equation 3.5 is incorrect. This is due to the fact that the x is not the same. The first x is the outcome of the stochastic variable in period 1 while the x in $E\{V_2(x)\}$ is the outcome in period 2. However, as we compute the expectation, the x (from period 2) vanishes yielding no further notational problems.)

The expectation in equation (3.5) is calculated as

$$E\{V_2(x)\} = \int_{-50}^{0} 0\frac{1}{150}dx + \int_{0}^{100} x\frac{1}{150}dx = 33\frac{1}{3} \tag{3.6}$$

giving

$$V_1(x) = \begin{cases} x & x \geq 33\frac{1}{3} \\ 33\frac{1}{3} & \text{otherwise} \end{cases} \tag{3.7}$$

Hence, in period 2 the house is sold if the observed x is larger than $33\frac{1}{3}$ otherwise we wait. To show the structure we may construct a table as 1.2.

Table 3.1: Solution for the house selling example with price uniformly distributed.

	$x \in< 33\frac{1}{3}, 100]$	$x \in< 0, 33\frac{1}{3}]$	$x \in [0, -50]$
period 1	sell	wait	wait
period 2	sell	sell	wait

If we compare table 3.1 with table 1.2 we observe an unchanged solution structure. This should not be surprising as we more or less solve the same problem. Note however that we get intervals instead of the discrete events *High*, *Medium* and *Low Price*.

3.3 Nondiscrete action space

In the former section we introduced the possibility of using a continuous state space. SDP allows us to use continuous action space as well. Suppose again that we change our assumptions in the house selling example. Assume now that our asset is an area of land and that we are able to sell parts of this land in each period. To make things as simple as possible we assume that the total area of the land is 1 unit of something and that we want to find an optimal sales strategy over a two period horizon. Hence, we need decision variables in each of the two periods. Let us define α_t as the proportion of the remaining land we sell in period t – $\alpha_t \in [0, 1]$. We keep the assumption on p and x; $x \sim U[-50, 100]$.

Let us solve this example using SDP. In period 2 we need to have information on how much land we have available for sale in this period. Surely this is determined by the decision we make in period 1. Additionally (as in our earlier examples), we need information on the outcome of the stochastic mechanism. Hence, a SDP formulation will contain a two dimensional state space in period 2. The value function in period 2 is;

$$V_2(x, \alpha_1) = \max_{0 \leq \alpha_2 \leq 1} [x(1 - \alpha_1)\alpha_2] \tag{3.8}$$

The term $(1 - \alpha_1)$ in equation (3.8) computes the available area for sale in period 2. As α_1 is $\in [0, 1]$, $(1 - \alpha_1)$ is positive. Therefore, we will sell all of our remaining land if $x > 0$ – $(\alpha_2 = 1)$. If on the other hand, $x < 0$, we sell nothing – $(\alpha_2 = 0)$. Then, $V_2(x, \alpha_1)$ may be written:

$$V_2(x, \alpha_1) = \begin{cases} x(1 - \alpha_1) & x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

The optimization problem for period 1 is formulated as;

$$V_1(x) = \max_{0 \leq \alpha_1 \leq 1} [x\alpha_1 + E\{V_2(x, \alpha_1)\}] \tag{3.10}$$

The expectation in equation (3.10) is computed as in equation (3.6) giving

$$E\{V_2(x, \alpha_1)\} = 33\frac{1}{3}(1 - \alpha_1) \tag{3.11}$$

and $V_1(x)$ as

$$V_1(x) = \max_{0 \leq \alpha_1 \leq 1} \left[\alpha_1(x - 33\frac{1}{3}) + 33\frac{1}{3}\right] \tag{3.12}$$

Solving the optimization problem (3.12) is straightforward. If $x < 33\frac{1}{3}$, α_1 is multiplied with a negative number. Therefore, the maximal $V_1(x|x < 33\frac{1}{3})$ is obtained by minimizing α_1. Alternatively, if $x \geq 33\frac{1}{3}$, the optimal α_1 is 1. Consequently, $V_1(x)$ becomes

$$V_1(x) = \begin{cases} x & x \geq 33\frac{1}{3} \\ 33\frac{1}{3} & \text{otherwise} \end{cases} \tag{3.13}$$

As in deterministic DP we need to "roll back" over the deterministic state variable (α_1 in this case) in order to find a complete solution. The case $x \geq 33\frac{1}{3}$ yields selling the whole area in period 1 and nothing happens in period 2. On the other hand, if $x < 33\frac{1}{3}$ we sell nothing in period 1, $\alpha_1 = 0$ and

$$V_2(x) = \begin{cases} x & x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.14}$$

If we compare the solution of this example – equations (3.13) and (3.14) – to the example in section 3.2 – equations (3.7) and (3.4), we observe that the solutions are identical. This is of course due to the fact that the optimization problems (3.8) and (3.12) are parametrical linear programming problems. We know from LP theory that we always obtain corner solutions. In our case we have two corner solutions, $\alpha = 1$ or $\alpha = 0$.

Actually, this is a quite general result. As an outline of a proof is instructive for later purposes, we will carry it through. Let us make some simple generalizations. Assume that we look at our problem in a time frame of N periods. Assume also that we generalize our uniform distribution for x to a general distribution $f[a, b]$ where $a < 0$. Let us start the backward recursion in period N. Our optimization problem becomes:

$$\max_{0 \leq \alpha_N \leq 1} \left[x \left\{ \prod_{i=1}^{N-1} (1 - \alpha_i) \right\} \alpha_N \right] \tag{3.15}$$

The term $\prod_{i=1}^{N-1}(1 - \alpha_i)$ is merely the remaining area for sale in period N. The solution to (3.15) is straightforward giving

$$V_N(x, \alpha_1, \ldots, \alpha_{N-1}) = \begin{cases} x \prod_{i=1}^{N-1}(1 - \alpha_i) & x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.16}$$

If we move to stage $N - 1$ we obtain the following optimization problem;

$$\max_{0 \leq \alpha_{N-1} \leq 1} \left[x \left\{ \prod_{i=1}^{N-2} (1 - \alpha_i) \right\} \alpha_{N-1} + \left\{ \prod_{i=1}^{N-2} (1 - \alpha_i) \right\} (1 - \alpha_{N-1}) \int_0^b x f(x) dx \right] \tag{3.17}$$

The product term $\left\{\prod_{i=1}^{N-2}(1-\alpha_i)\right\}$ is a common factor and a constant in the optimization, and may hence be removed. Let us also define:

$$\theta_{N-1} = \int_0^b xf(x)dx \tag{3.18}$$

θ_{N-1} is a positive number. Then we can rewrite the optimization problem (3.17) as

$$\max_{0\leq\alpha_{N-1}\leq 1} [\alpha_{N-1}(x-\theta_{N-1})+\theta_{N-1}] \tag{3.19}$$

with the solution

$$V_{N-1}(x,\alpha_1,\ldots,\alpha_{N-2}) = \begin{cases} x\left\{\prod_{i=1}^{N-2}(1-\alpha_i)\right\} & x\geq\theta_{N-1} \\ 0 & \text{otherwise} \end{cases} \tag{3.20}$$

or at a general stage $N-j$,

$$V_{N-j}(x,\alpha_1,\ldots,\alpha_{N-j-1}) = \max_{0\leq\alpha_{N-j}\leq 1} [\alpha_{N-j}(x-\theta_{N-j})+\theta_{N-j}] \;\; \forall j\in\{1,\ldots,N-1\} \tag{3.21}$$

θ_{N-j} is calculated by the recursion

$$\theta_{N-j} = \int_{\theta_{N-j+1}}^b xf(x)dx, \tag{3.22}$$

and is positive for any j as θ_{N-1} is positive. After this somewhat lengthy mathematical development, the clue is to observe the structure of equation (3.21). We observe that this family of optimization problems are all linear. Hence, at any stage j we get solutions where we either sell all or nothing. That is, it will never be optimal to split the area between periods given our assumptions.

This somewhat cumbersome exercise, shows that SDP may be used to obtain quite general problem characteristics. Surely this is not the case generally, but for some classes of problems, SDP may be a valuable tool.

3.4 Handling non linearities

As described in Sandblom et al. (Sandblom et al., To appear – Never did), DP may be freely applied in non linear problems. This is also the case in SDP applications. This section will elaborate further on the example introduced in section 3.3.

As section 3.3 showed, the optimal solution would never contain splits between periods. The reason for this was showed to be due to linear subproblems at any stage. Suppose we introduce a risk averse utility function. What would be the consequences? Surely, this would introduce non linear subproblems due to the concave structure of a risk averse utility function. Would it also change the solution such that under certain assumptions, we would obtain a split between periods? The answer to this question is yes, as we soon shall see. The reason is due to the fact that introducing risk aversion leads to a trade off between taking a risky decision of postponing the sale to period 2 or selling now. By selling now, we insure ourselves against potential loss if the price is low in period 2. However, we will not sell all our property, as leaving some for sale to the next period may prove advantageous.

Let us, for the time being, introduce a general utility function $u(w)$. We assume greed and risk aversion, $u'(w) > 0$ and $u''(w) < 0$. Performing the SDP calculations at stage 2 then implies the following optimization problem; (Note that we return to our original density for x, $x \sim U[-50, 100]$.)

$$V_2(x, \alpha_1) = \max_{0 \leq \alpha_2 \leq 1} \left[u\left(x(1-\alpha_1)\alpha_2 \right) \right] \tag{3.23}$$

The solution to the optimization problem (3.23) is identical to problem (3.8) as we maximize a monotone function ($u()$) of the same goal. Hence,

$$V_2(x, \alpha_1) = \begin{cases} u(x(1-\alpha_1)) & x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.24}$$

Moving on to period 1, the SDP recursion becomes:

$$V_1(x) = \max_{0 \leq \alpha_1 \leq 1} \left[u(x\alpha_1) + E\left\{ u(V_2(x, \alpha_1)) \right\} \right] \tag{3.25}$$

If we calculate $E\left\{ u(V_2(x, \alpha_1)) \right\}$, equation (3.25) can be written;

$$V_1(x) = \max_{0 \leq \alpha_1 \leq 1} \left[u(x\alpha_1) + \frac{1}{150} \int_0^{100} u\left(y(1-\alpha_1) \right) dy \right] \tag{3.26}$$

Note that we have introduced the variable y for the stochastic variable x in period 2 in order to avoid confusion with x in period 1.

In order to keep the mathematics at a reasonable level we introduce a family of utility functions at this point. Assume that $u(w)$ may be expressed as follows:

$$u(w) = Aw^2 + Bw \tag{3.27}$$

$u(w)$ is called a quadratic utility function, it should not be hard to understand why. Let us set the scale as we did in the example in section 2.2. Fist we observe that $u(0) = 0$ directly from equation (3.27). Second, we need to obtain $u(100) = 1$. This is done by defining $u(w)$ as follows:

$$u(w) = (.0001 - .01B)w^2 + Bw \tag{3.28}$$

Finally we want $u'(w)$ positive and $u''(w)$ negative. $u''(w) < 0$ implies

$$2(.0001 - .01B) < 0 \Rightarrow B > .01 \tag{3.29}$$

while $u'(w) > 0$ leads to

$$2(.0001 - .01B)w + B > 0 \tag{3.30}$$

Utilizing the fact that the maximal value of w is 100 we obtain an upper limit on B from equation (3.30)

$$2(.0001 - .01B)100 + B > 0 \Rightarrow B < .02 \tag{3.31}$$

Hence we may choose B in the interval $< .01, .02 >$. Figure 3.1 shows $u(w)$ for some values of B. Note that the degree of risk aversion is increasing with increasing B in figure 3.1.

Now we are in a position to evaluate the integral in equation (3.26). Using (3.28), equation (3.26) may be expressed as

$$V_1(x, B) = \max_{0 \le \alpha_1 \le 1} \left[C_1(B, x)\alpha_1^2 + C_2(B, x)\alpha_1 + C_3(B)(1 - \alpha_1)^2 + C_4(B)(1 - \alpha_1) \right] \tag{3.32}$$

where

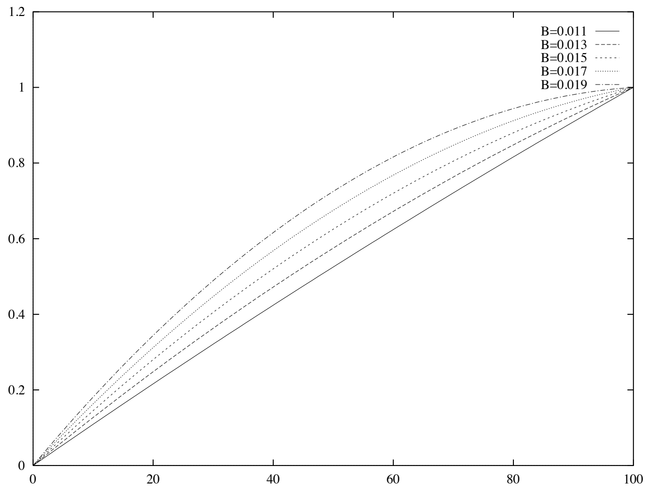


Figure 3.1: Graph of utility function $u(w) = (.0001 - .01B)w^2 + Bw, B \in [0.01, 0.02]$

$$\begin{array}{lcl} C_1(B,x) & = & x^2(0.0001 - 0.01B) \\ C_2(B,x) & = & Bx \\ C_3(B) & = & 2222.22(0.0001 - 0.01B) \\ C_4(B) & = & 33.33B \end{array} \tag{3.33}$$

Let us start out simple and just choose a set of values for x and B and plot the objective from equation (3.32). Let us choose $x = 70$ and $B = 0.019$. Inserting these values into (3.32) gives;

$$V_1 = \max_{0 \leq \alpha_1 \leq 1} \left[-0.441\alpha_1^2 + 1.33\alpha_1 - 0.2(1-\alpha_1)^2 + 0.6333(1-\alpha_1) \right] \tag{3.34}$$

The objective in equation (3.34) is plotted in figure 3.2.

We observe from figure 3.2 that our hypothesis of a split in the solution is correct for this case. The maximal α_1 is easily found by differentiating giving $\alpha_1 \approx 0.86$. Hence, our initial hypothesis on a possible split has been confirmed. If we observe $x = 70$ and use an utility function with $B = 0.019$ the optimal solution states that we shall sell 86% of our land in period 1 and 14% i period 2.

The general solution to this example is a bit harder to obtain. Let us first give the structure and then show how we may find it. The solution may be stated as follows:

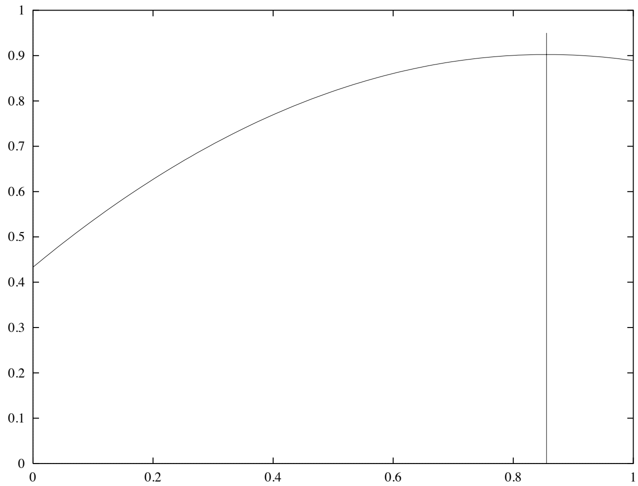


Figure 3.2: Graph of $-0.441\alpha_1^2 + 1.33\alpha_1 - 0.2(1-\alpha_1)^2 + 0.6333(1-\alpha_1)$

$$\begin{array}{ll} x < \underline{x}(B) & \alpha_1^* = 0 \\ \underline{x}(B) \leq x \leq \bar{x}(B) & \alpha_1^* = \frac{C_4(B)+2C_3(B)-C_2(B,x)}{2C_1(B,x)+2C_3(B)} \\ x > \bar{x}(B) & \alpha_1^* = 1 \end{array} \tag{3.35}$$

$C_1, \ldots, C_4$ are defined in equation (3.33), while $\bar{x}(B)$ is found by solving the inequality

$$H(B)(33\frac{1}{3} - x) - 2x^2 < 0 \tag{3.36}$$

$H(B)$ is defined

$$H(B) = \frac{B}{0.0001 - 0.01B} \tag{3.37}$$

$\underline{x}(B)$ is defined as

$$\underline{x}(B) = 33\frac{1}{3} + 4444.44\left(\frac{0.0001}{B} - 0.01\right) \tag{3.38}$$

Let us start investigating the solution (3.35), (3.36), (3.37) and (3.38) by differentiating (3.32) with respect to α_1 and solve for first order conditions. (Note that we have simplified the notation of $C_1, \ldots, C_4$ by omitting the parametric dependence of B and x.)

$$2C_1\alpha_1 + C_2 + 2C_3(1-\alpha_1)(-1) - C_4 = 0 \tag{3.39}$$

Solving the linear equation (3.39) yields;

$$\alpha_1^* = \frac{C_4 + 2C_3 - C_2}{2C_1 + 2C_3} \tag{3.40}$$

To secure that α_1 maximizes the problem (3.32), the second order conditions must be satisfied. This is a simple one variable optimization problem, and the second order conditions are checked by differentiating with respect to α_1 again, giving

$$2C_1 + 2C_3 < 0 \tag{3.41}$$

Substituting values for C_1 and C_3 from equation (3.33) into (3.41) gives

$$(.0001 - .01B)\left[x^2 + 2222.22\right] < 0 \tag{3.42}$$

As the expression $x^2 + 2222.22$ is always positive, equation (3.42) yields $B > .01$. This constraint is already imposed on our problem – see equation (3.29). That is, our optimization problem behaves well. (We maximize a concave function subject to a linear constraint set.)

Now we need to secure that α_1^* is non negative and less than or equal to 1. Let us investigate the inequality; $\alpha_1^* \geq 0$.

Substituting values for $C_1, \ldots, C_4$ from equation (3.33) into (3.40) gives

$$\frac{1}{2\left[x^2 + 2222.22\right]}\left[\frac{B(33\frac{1}{3} - x)}{.0001 - .01B} + 4444.44\right] \geq 0 \tag{3.43}$$

Further manipulations on equation (3.43) yields

$$x > 33\frac{1}{3} + 4444.44\left(\frac{0.0001}{B} - 0.01\right) \tag{3.44}$$

We see that the right hand side of inequality (3.44) is what we have defined as $\underline{x}(B)$ in equation (3.38).

It is probably simplest to explain the meaning of the inequality by looking at a simple graph. Suppose that we fix B to 0.011. Then, the inequality (3.44) is simplified to

$$x > 29.29 \tag{3.45}$$

In figure 3.3, the objective from equation (3.32) is plotted for various values of x.

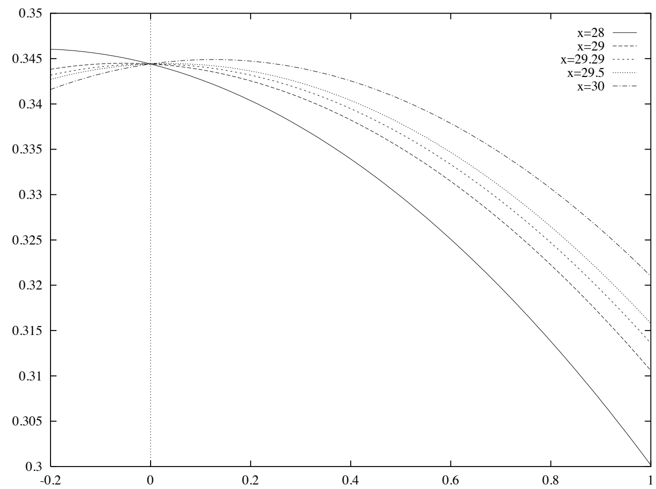


Figure 3.3: Graph of objective with $B = 0.011$

We observe from figure 3.3, that if x is smaller than 29.29, the interior optimal α_1 is outside the region $[0, 1]$. Hence, the optimal solution is to set α_1^* equal to 0. Alternatively, if x is larger than 29.29, the optimal α_1 falls inside the interval $[0, 1]$ and we sell parts of the land in period 1 according to equation (3.35). Note that the lover limit for split, $\underline{x}(B)$ is a decreasing function in B. ($\frac{d\underline{x}(B)}{dB} = \frac{-0.44}{B^2} < 0$.) Hence, increasing risk aversion (increasing B) implies a decreased lower acceptable value $\underline{x}(B)$. If for instance an individual with $B = 0.011$ observes $x = 28$ in period 1, nothing is sold in this period. If a more cautious individual – with $B = 0.015$ – observed $x = 28$ in period 1, he would sell $\approx 47.33\%$ in period 1.

Note also that we are able to find an absolute lower limit $\underline{x}(B)$ in this example. Look again at inequality (3.43). Our choice of utility function made it necessary to limit the parameter measuring risk aversion B to the interval $[0.01, 0.02]$. The inequality (3.43) states that in order to obtain a split solution, x must be larger than the right hand side expression for **any possible** B. Therefore, if we can solve the following problem,

$$x > \max_{0.01 < B \leq 0.02} \left[33\frac{1}{3} + 4444.44 \left(\frac{0.0001}{B} - 0.01 \right) \right] \tag{3.46}$$

we would find a lower limit for x which would guarantee a non-split solution. Solving the reformulated inequality (3.46) yields

$$x > 11.11 \tag{3.47}$$

That is, if the land salesman observes an x smaller than 11.11 he will never sell anything in period 1, independently of his attitudes towards risk. The reason for this somewhat unexpected result is due to the choice of utility function. The upper limit of 0.02 imposed on the B parameter really means nothing else than a constraint on the degree of risk aversion we can use. So if we had chosen another family of utility functions, for instance the exponential $(a - be^{-cx})$, we would not have got this type of result.

The upper bound for x, $\bar{x}$ is somewhat harder to investigate. Principally, it is established analogously with $\underline{x}(B)$ by solving the inequality $\alpha_1^* \leq 1$. However, this inequality is non linear in x and makes it harder to establish a clean cut solution structure.

$$\frac{1}{2\left[x^2 + 2222.22\right]}\left[\frac{B(33\frac{1}{3} - x)}{.0001 - .01B} + 4444.44\right] \leq 1 \tag{3.48}$$

This inequality is obtained from inequality (3.43). Multiplying by $2[x^2 + 2222.22]$ and rearranging terms give

$$H(B)(33\frac{1}{3} - x) - 2x^2 < 0 \tag{3.49}$$

which is inequality (3.36). Let us plot the left hand side of this inequality as a function of x for a range of B values. Figure 3.4 shows the results. (Note that B is increasing from top to bottom in figure 3.4.)

Some parts of figure 3.4 are simple to explain. Let us start by investigating small B's. We see that if B is sufficiently small, approximately; $B < 0.015$, we obtain a structure with an upper limit on x. Look for instance at the case $B = 0.013$. Then if x is smaller than 41, the value of inequality (3.49) is true and we get a split solution. Alternatively if $x > 41$, α_1 is larger than 1 and the optimal solution is to sell all the land in period 1. If the degree of risk aversion is increased, for instance by choosing $B = 0.014$, we get the same type of solution but with a larger upper bound on x. Surely this seems sensible, the decision maker is more cautious and needs a higher observed x in period 1 in order to sell all his land. The bottom part of the figure is also easy to explain. If $B \geq 0.16$, we observe that the function is always negative.

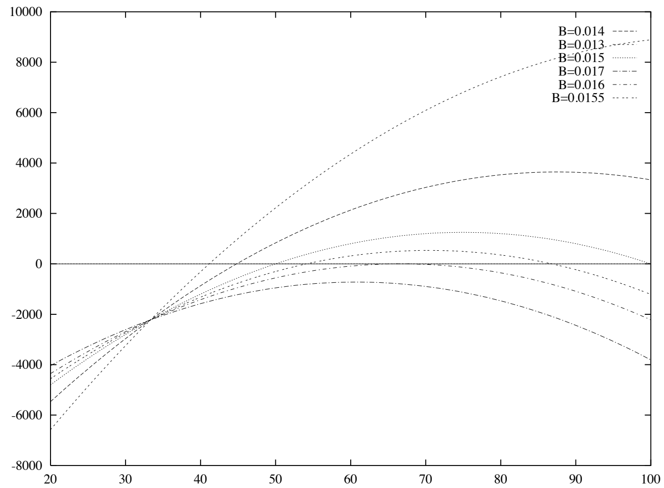


Figure 3.4: Graph of $H(B)(33\frac{1}{3}-x)-2x^2$ as a function of x with B ranging from 0.013 to 0.017

This implies that the degree of risk aversion is so large that we always obtain a split solution. In this area, the decision maker is so cautious that he always insures himself against a low future x by selling some of the land in period 1.

However, the mid part of figure 3.4 ($B = 0.0155$) is harder to explain. Note that for this value of B, the inequality (3.49) is true for 2 intervals – $x < 55$ and $x > 87$ approximately. To stress the implications of this pattern we have constructed another graph. This graph is shown in figure 3.5.

In figure 3.5, we have plotted the objective function in the optimization problem (3.32) as a function of α_1. We have done this for various values of x and a fixed value of $B = 0.0155$. Additionally, we have plotted the optimal values of α_1 and connected these by a line.

This graph ($\alpha_1^*(x)$), shows how the optimal values of α_1 changes as x changes. In order to explain this structure we have taken it out an plotted it in figure 3.6.

If we start examining figure 3.6 in point A, we observe that when we move from point A to B, the optimal proportion sold in period 1 is increasing towards 1. The horizontal line between points B and C is obtained because α_1^* must be ≤ 1. This makes sense. However, the behaviour between points C and D seems weird. Here, we obtain a solution where the optimal α_1 is decreasing when x increases. The interpretation is that when x becomes

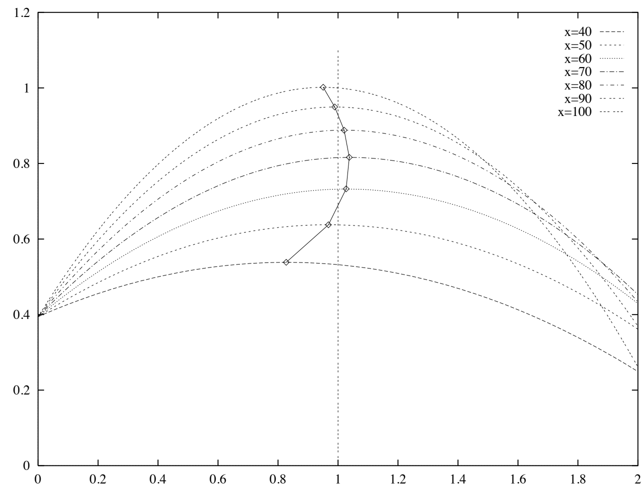


Figure 3.5: Graph of objective as a function of α_1 for various values of x; $B = 0.0155$

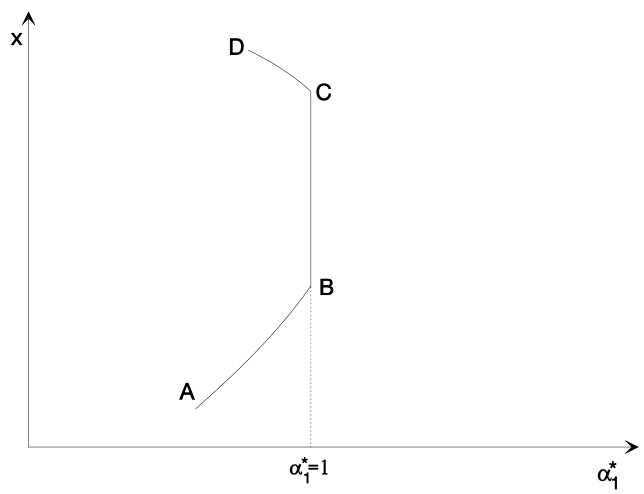


Figure 3.6: Caption of $x(\alpha_1^*)$

sufficiently large, the solution moves from selling everything in period 1 to a split solution.

In order to understand why this happens, we have to look closer at the utility function we have used. One way of measuring the degree of risk aversion associated with a utility function is by computing the so called *absolute risk aversion.* This measure is defined; see for instance Copeland and Weston (Copeland and Weston, 1983).

$$ARA = -\frac{u''(w)}{u'(w)} \tag{3.50}$$

Loosely speaking, the idea is to use ARA as a measure of how risk aversion changes when wealth (w) changes. Hence, you would expect ARA to be decreasing with w. It is easier to engage in a high stake bet for a rich person who can bear the loss than for a poor person who can not. The larger ARA is, the more risk averse the utility function is.

Let us calculate ARA and $\frac{dARA}{dw}$ for the quadratic family of utility functions[2].

$$ARA = \frac{2A}{B - 2Aw} \text{ and } \frac{dARA}{dw} = \frac{4A^2}{(B - 2Aw)^2} > 0 \tag{3.51}$$

We observe from equations (3.51) that ARA is dependent on the argument w and more important, ARA increases with the argument of the utility function. This last property shows why we get our weird result. When x in figure 3.6 becomes very big, this has the effect of increasing risk aversion, even if B is fixed at 0.0155. Hence when x becomes close to 100, the decision maker gets more risk averse and shifts from the less risk averse decision of selling all in period 1 towards the more risk averse decision of split sale between the two periods.

By aid of figure 3.4, we can construct a more precise solution structure than equation (3.36) indicates. It should be easy to realize that we get three different solution types in this area depending on the value of B. If B is smaller than a value, call this value B_1, we obtain a solution corresponding to the upper part of figure 3.4. That is, if B is smaller than B_1, we split the solution. On the other hand, if B is larger than B_1 we sell everything in period 1 - ($\alpha_1^* = 1$). It should also be evident that this solution structure

[2]We use the normal way of defining a quadratic utility function; see (Copeland and Weston, 1983); $u(w) = Aw^2 - Bw$ for these calculations

is valid for values of B smaller than the B which is such that the larger of the roots in equation (3.52) equals 100. (Note from figure 3.4 that the value we are looking for should be close to 0.015, and that $H(B)$ is defined in equation (3.36).)

$$H(B)(33\frac{1}{3} - x) - 2x^2 = 0 \tag{3.52}$$

The problem we have formulated in words above, may be formulated mathematically as follows: Find the roots r_1, r_2 from equation (3.52) and solve equation (3.53) for B;

$$\max(r_1, r_2) = 100 \tag{3.53}$$

This is a simple problem. The first step yields:

$$r_1 = -\frac{H(B)}{4}\left[1 + \sqrt{1 + \frac{266\frac{2}{3}}{H(B)}}\right] \text{ and } r_2 = -\frac{H(B)}{4}\left[1 - \sqrt{1 + \frac{266\frac{2}{3}}{H(B)}}\right] \tag{3.54}$$

It is easy to show that $max(r_1, r_2) = r_1$. Therefore, the second step in our procedure involves solving the following equation.

$$-\frac{H(B)}{4}\left[1 + \sqrt{1 + \frac{266\frac{2}{3}}{H(B)}}\right] = 100 \tag{3.55}$$

The solution of equation (3.55) is straightforward. Multiplying over the factor $-\frac{H(B)}{4}$, rearranging and squaring yields a quadratic equation in $H(B)$. The solution of this equation gives $B = B_1 = 0.015$.

If B is larger than 0.015 we get the solution structure we described as "weird" above. This structure is characterized by the fact that we obtain two roots within the interval $[0, 100]$ for x – refer to figure 3.4. We will get this structure for gradually increasing B's until the maximal value of the function on the left of inequality (3.49) is zero. Hence, we may find another critical value for B say B_2 by the following procedure. First we maximize;

$$\frac{d}{dx}\left\{H(B)(33\frac{1}{3} - x) - 2x^2\right\} = 0 \tag{3.56}$$

which yields x^*

$$x^* = -\frac{H(B)}{4} \tag{3.57}$$

The maximal value should equal 0;

$$H(B)\left(33\frac{1}{3} + \frac{H(B)}{4}\right) - 2\left(\frac{H(B)^2}{16}\right) = 0 \tag{3.58}$$

Equation (3.58) is a simple quadratic equation in $H(B)$ with solution;

$$H(B) = -266\frac{2}{3} \tag{3.59}$$

Hence, B_2 is found to be 0.016. (Refer to figure 3.4.) Let us try to sum up the solution in a table.

Table 3.2: Solution to the house selling example with quadratic utility function and uniform density.

$x < \underline{x}(B)$	$x \geq \underline{x}(B)$					
	$B < 0.015$		$B \in [0.015, 0.016]$			$B > 0.016$
	$x < r_2$	$x \geq r_2$	$x < r_2$	$x \in [r_2, r_1]$	$x > r_1$	
$\alpha_1^* = 0$	Split	$\alpha_1^* = 1$	Split	$\alpha_1^* = 1$	Split	Split

$\underline{x}(B)$ in Table 3.2, is defined in equation (3.38). The term "Split" refers to a split solution computed by equation (3.40). r_1 and r_2 are defined in equation (3.54).

This section has demonstrated that SDP may be applied in solving SDP's with non linearities. However, the solution structure to this seemingly simple two periodic problem, turned out to be quite complex. Partially, this was due to a somewhat special choice of utility function.

3.5 Analytic solutions

In the former sections, we have demonstrated SDP's ability to solve simple two periodic stochastic optimization problems. In order to obtain solutions, we had to specify probability densities and/or utility functions. In this section we will demonstrate that SDP may give more general solution structures if the problem structure is somewhat different.

Very often, an assumption of infinite horizon proves simplifying to this type of problems. (Note that we will return to infinite horizon problems as such in a later section.) In this section we will use an infinite horizon assumption to show that a general solution may be obtained. We will also use this analytic solution to discuss some general differences between stochastic and deterministic optimization problems.

Assume now that we return to the house selling example with expected value as our objective, binary decision structure and a general density function for x. That is, x is independently and identically distributed in any period with density $f[a, b]$, $a, b > 0$. (Note that we change our earlier assumption of a negative lower value a.) Assume also that we introduce impatience in the problem. If we postpone the selling decision from one period to another, a cost c occurs. Similar models have been treated by Haugen (Haugen, 1991) and Kaufman (Kaufman, 1963).

Under these assumptions, the optimality equation may be expressed as

$$V_t(x) = \max\left[x, \int_a^b V_{t+1}(x)f(x)dx - c\right] \tag{3.60}$$

recursively expanding equation (3.60) assuming N is the last period gives;

$$V_N(x) = x \tag{3.61}$$

That is, in the last period the best we can do is to sell as any outcome in $[a, b]$ yields positive contribution to the objective. Let us move to period $N - 1$. The optimality equation (3.60) then becomes:

$$V_{N-1}(x) = \max\left[x, \int_a^b xf(x)dx - c\right] \tag{3.62}$$

If we had known the density $f(x)$, the expression $\int_a^b xf(x)dx - c$ could have been computed as a number. Let us call this number p_1 and rewrite equation (3.62);

$$V_{N-1}(x) = \max[x, p_1] \tag{3.63}$$

Figure 3.7 shows $V_{N-1}(x)$.

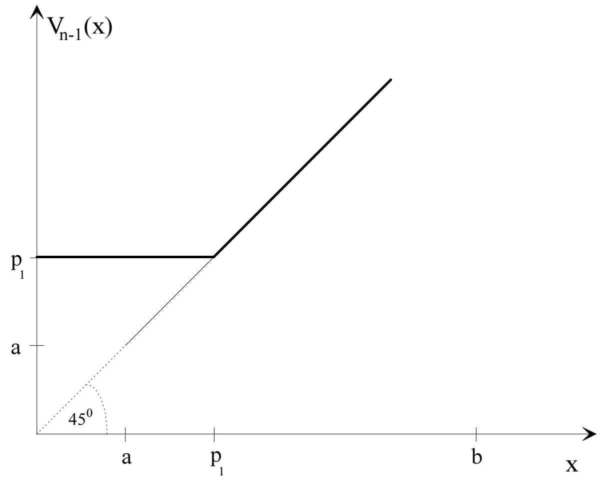


Figure 3.7: $V_{N-1}(x)$ in the house selling example with infinite horizon.

Note that figure 3.7 implicitly contains an assumption, $p_1 > a$. This is perfectly reasonable, as the opposite ($p_1 < a$) would lead to an uninteresting problem in which we would never postpone the sales decision. The optimality equation at $N-2$ becomes;

$$V_{N-2}(x) = \max\left[x, E\left\{V_{N-1}(x)\right\}\right] = \max\left[x, E\left\{\max\left[x, p_1\right]\right\}\right] \tag{3.64}$$

Utilizing the implicit assumptions in figure 3.7, equation (3.63) may be expressed:

$$V_{N-1}(x) = \max\left[x, p_1\right] = \begin{cases} x & x \geq p_1 \\ p_1 & x < p_1 \end{cases} \tag{3.65}$$

Thus, the expectation in equation (3.64) may be computed as;

$$E\left\{\max\left[x, p_1\right]\right\} = \int_a^{p_1} p_1 f(x)dx + \int_{p_1}^b x f(x)dx - c \tag{3.66}$$

giving

$$V_{N-2}(x) = \max\left[x, \left\{\int_a^{p_1} p_1 f(x)dx + \int_{p_1}^b x f(x)dx - c\right\}\right] \tag{3.67}$$

Now we can repeat the argument that led to equation (3.63) in this period. Hence,

$$V_{N-2}(x) = \max\,[x, p_2] \tag{3.68}$$

or in the general case

$$V_{N-j}(x) = \max\,[x, p_j] \tag{3.69}$$

where

$$p_j = \int_a^{p_{j-1}} p_{j-1} f(x)dx + \int_{p_{j-1}}^b x f(x)dx - c \tag{3.70}$$

and

$$p_1 = \int_a^b x f(x)dx - c \tag{3.71}$$

Let us at this point look at a numerical example. Suppose that $f[a, b] = U[0, 100]$ and $c = 30$. Given this information, the recursion in equations (3.70), (3.71) may be calculated giving $p_1, p_2, \ldots, p_N$. Table 3.3 shows this values for $N = 10$.

Table 3.3: $p_1, p_2, \ldots, p_{10}$

i	p_i
1	20.000
2	22.000
3	22.420
4	22.513
5	22.534
6	22.540
7	22.540
8	22.540
9	22.540
10	22.540

As Table 3.3 shows, p_i shows limiting behaviour. It is possible to show that this behaviour is a general characteristic for several classes of stochastic optimization problems, among them this example. (Refer for instance to Ross (Ross, 1983).) We will not fulfil this matter further at this point, merely accept the existence of such a limiting behaviour. Given this behaviour, the following must hold:

$$\lim_{j \to \infty} p_j \to p^* \tag{3.72}$$

Consequently,

$$V_{N-j}(x) \to V(x) = \max\left[x, p^*\right] = \begin{cases} x & x \geq p^* \\ p^* & x < p^* \end{cases} \tag{3.73}$$

Thus, if we view our problem in an infinite horizon perspective, we have identified a strategy independent of time (stages). The strategy may be interpreted as follows: If we observe an x at any time, smaller than p^*, we do nothing. Alternatively, if we observe an x larger than p^*, we sell our land immediately.

We have earlier referred to such a strategy as a stationary policy. Not only have we identified such a strategy, we have also an equation to find it. Namely,

$$p^* = \int_a^{p^*} p^* f(x)dx + \int_{p^*}^b x f(x)dx - c \tag{3.74}$$

Adding and subtracting the term $\int_a^{p^*} x f(x)dx$ yields

$$p^* = E(x) - c + \int_a^{p^*} \left(p^* - x\right) f(x)dx \tag{3.75}$$

The term $E(X)$ is merely the expected value of x.

$$E(X) = \int_a^b x f(x)dx \tag{3.76}$$

In problems similar to ours, a stationary policy of this type is often referred to as a *reserve price strategy*.

Let us return to our numerical example leading to Table 3.3 and check that equation (3.75) gives the same answer. Inserting $c = 30$ and $f[a, b] = U[0, 100]$ in equation (3.75) gives;

$$p^* = 50 - 30 + \frac{1}{100}\int_0^{p^*} (p^* - x) \tag{3.77}$$

or

$$p^{*2} - 200p^* + 4000 = 0 \tag{3.78}$$

The quadratic equation (3.78) has solution:

$$p^* = 22.540 \text{ or } p^* = 177.460 \tag{3.79}$$

If we compare the smallest root ($p^* = 22.540$) with the limiting value from Table 3.3, we observe that these values are equal. Hence, equation (3.75) seems to give a correct expression for the reserve price p^*.

It is not much point in establishing an analytic solution to a problem unless we want to use it to something. Therefore, let us use the solution to compare our example to a deterministic version of the problem and discuss the differences. A deterministic version of our problem may for instance be formulated as follows: Assume that δ_i is a binary decision variable;

$$\delta_t = \begin{cases} 1 & \text{if we sell the land in period } t \\ 0 & \text{otherwise} \end{cases} \tag{3.80}$$

The normal way of going from a stochastic optimization problem to a deterministic equivalent, is to substitute stochastic variables with their expectations. Doing this, our deterministic problem can be formulated as an infinite horizon integer programming problem as follows:

$$\begin{array}{rl} Max\ Z = & \sum_{t=1}^{\infty} \left[E(x) - (t-1)c\right]\delta_t \\ s.t. & \sum_{t=1}^{\infty} \delta_t \leq 1 \\ & \delta_t \in \{0,1\} \end{array} \tag{3.81}$$

Given our earlier assumptions of $E(x) - c > a$ and $a > 0$, it is easy to realize that $E(x)$ must be positive. Hence, the optimal solution to problem (3.81) is easily found as

$$\delta_1^* = 1 \text{ and } \delta_t^* = 0 \ \forall t \geq 2 \tag{3.82}$$

That is, the land is sold in period 1. One way to compare the deterministic solution to the problem and the stochastic solution to the problem can be to compute expected waiting time before a sales decision is made in the stochastic case. Let us define the following probability:

$$P(x \geq p^*) = q \tag{3.83}$$

Hence, q is the probability that x is larger than or equal to the reserve price - p^*. Then, the expected number of periods before the sale is made can be computed as follows:

$$E(\text{“Waiting time”}) = q + 2(1-q)q + 3(1-q)^2 q + \dots \tag{3.84}$$

$$= q \sum_{t=1}^{\infty} t(1-q)^{t-1} \tag{3.85}$$

Integrating to obtain a geometric series and later differentiating equation (3.85) gives

$$E(\text{“Waiting time”}) = \frac{1}{q} \tag{3.86}$$

Utilizing equation (3.86) we can compare the stochastic and the deterministic model. If $q = 1$, the probability of observing an x larger than p^* equals 1. Then, we obtain an expected waiting time of one period or the same solution as in the deterministic case. (Note that our definition of periods means that immediate sale gives a waiting time of 1 period.) On the other hand, if this probability decreases, the expected time before the sale is made increases geometrically.

This example has showed that in some situations, SDP may be applied to obtain analytic solutions to stochastic optimization problems. The reader should not make the mistake of believing that this is a common situation. In some situations, it may be helpful – at least as a way of obtaining principal information on problem behaviour.

However, this example may be used to stress another important point. In practice, many people tend to apply scenario analysis as a method of taking care of uncertainty. By the term scenario analysis, we here refer to a process where various scenarios or possible future developments of random structures are substituted for stochastic variables. Then for each scenario, a deterministic optimization problem is solved. Finally, one tries to weigh these deterministic solutions together in order to find some solution that takes uncertainty into consideration. The point we want to stress here, is the fact that such a strategy may be dangerous. It may not exist any weighing strategy that captures the stochasticity of the problem.

Let us look at a “scenario analysis” way of solving our example. Suppose that two scenarios are formulated; *Good* and *Bad*. Suppose that the *Good* scenario is characterized by a fixed $x = 70$ in all future periods while the *Bad* scenario has $x = 10$ in all future periods. If we then solve the deterministic optimization problem (3.81) for these two scenarios, we obtain the same solution in both cases; namely

$$\delta_1^* = 1 \text{ and } \delta_t^* = 0 \ \forall t \geq 2 \tag{3.87}$$

Actually, we could formulate as many scenarios we want for x, but any one of them would produce the same solution. Surely, it is impossible to weigh a set of solutions (3.87) together to capture the stochasticity of the solution to the stochastic problem as it is described in equations (3.73) and (3.75). This type of arguing is often used to justify why stochastic optimization problems must be solved by stochastic optimization methods. Refer for instance to the chapter on *scenario aggregation* in Wallace and Kall (Kall and Wallace, 1994).

3.6 Concluding remarks

In the former sections, we have looked at different versions of the same example. This example is often referred to as a *secretary problem* in the literature. Smith (Smith, 1991) treats such problems and stresses the fact that in spite of their obvious simplicity, they may represent a wide range of interesting practical problems.

The classical secretary problem is treated by Gilbert and Mosteller (Gilbert and Mosteller, 1966). This problem may be described as follows: Assume that a manager wants to employ a secretary. Each candidate is assumed to enter independently of each other and when a secretary is employed, it is illegal to change the decision. The qualities of a secretary is then assumed to be monitorable when the secretary arrives asking for employment, but uncertain before. Hence, it should be simple to see the similarities with our examples. If we allow the manager to employ a secretary part time, we have a situation similar to our example in sections 3.2 and 3.3.

As Smith (Smith, 1991) and others stress, such a situation is common in many decision problems. Typically all kind of problems involving stop and start decisions may be viewed in this perspective. Hence, project scheduling problems as those discussed by Haugen (Haugen, 1991), (Haugen, 1996), various option pricing problems, search problems etc. may be categorized among secretary problems.

Recent work by Tamaki (Tamaki, 1991), Rose (Rose, 1984), Smith (Smith, 1975) and others provide further insight into such problems. Typically, various authors discuss the possibilities of relaxing one or several of the assumptions underlying the classical secretary problem. Tamaki and Rose look at the problem when more than one candidate is to be selected, while Smith looks at the situation if a secretary is allowed not to accept a job offer.

Chapter 4

SDP - difficulties

So far we have discussed simple problems. Not necessarily in a conceptual framework, but surely in a computational. It makes little sense to discuss SDP, or DP for that matter, without taking computers and computational problems into consideration. As opposed to mathematical programming methodology, little commercial software aimed at solving SDP problems exists. This is obviously due to the generality of the method. Remember that SDP is merely a search/decomposition technique which works on stochastic optimization problems under quite general assumptions. Another important reason for the lack of commercial SDP or DP software is the so called *curse of dimensionality*. It is essential to understand this concept in order to be able to approach tractable solution techniques for large scale problems. Therefore, we will use the next paragraph to discuss this somewhat unpleasant property.

4.1 Curse of dimensionality

This famous problem was discovered early in the history of DP. Bellman and Dreyfus (Bellman and Dreyfus, 1962) discuss the problem under the less dramatic description "dimensionality difficulties". However, they used the term "curse" in a more ironic fashion than today's language habits should indicate. Actually, the term "curse" is a very good description of the problem, and this fact is probably the reason why the term has been adopted as the standard way of describing this phenomena. Webster's dictionary defines the term curse as follows:

> **curse** *n, v* **cursed** or **curst**, **cursing.** **-*n*.** **1.** the expression of a wish or misfortune, evil, doom, etc., befall another. **2.** a formula or charm intended to cause such misfortune to another. **3.** the act of reciting such a formula. **4.** an ecclesiastical censure or anathema. **5.** a profane oath. **6.** an evil that has been invoked upon one. **7.** something accused. **8.** the cause of evil, misfortune or trouble. ...

This somewhat drastic definition should indicate the seriousness of the problem. Let us explain what we mean by the curse of dimensionality. So far, our examples have been simple in the sense that we have used one or very few state variables. Let us return to the optimality equation (1.6).

$$V_n(i) = \max_a \left[R(i,a) + \sum_j P_{ij}(a) V_{n+1}(j) \right] \tag{4.1}$$

This equation (4.1), is written in a form where the state space (possible values for the state variable i) is one dimensional. Suppose our problem needs a multidimensional state space definition; say $i_1, \ldots, i_m$ where we assume that each state variable can take a set of discrete values. Then equation (4.1) may be expressed:

$$V_n(i_1, \ldots, i_m) = \max_a \left[R(i_1, \ldots, i_m, a) + E\left\{ V_{n+1}(i_1, \ldots, i_m, a) \right\} \right] \tag{4.2}$$

(Note that equation 4.2 is written merely with an E-symbol as opposed to equation 4.1. The reason for this is the fact that we assume that all states are stochastic states in equation 4.1. This is seldom the case in practical situations, and we do not need to partition the state space further than equation 4.2 indicates, to explain the curse of dimensionality.)

Let us assume that state variable i_k can take I_k values - ($k \in \{1, 2, \ldots, m\}$). Then, the number of optimization problems we have to solve at **each stage** n, is

$$N = \prod_{k=1}^{m} I_k \tag{4.3}$$

It is the size of N and the consequences of this size which is referred to as the curse of dimensionality. Let us look at a simple numeric example. Assume for simplicity that $I_k = I \ \forall\ k$ and that $m = 10$. Table 4.1 shows N as a function of I.

Table 4.1: State space size N as a function of I

I	N
2	1024
3	59049
4	1048576
5	9765625
6	60466176
7	282475249
8	1073741824

If we look at Table 4.1, we observe the enormous growth of the state space. It is the handling – or should we perhaps say – the lack of handling of this growth, which is called the curse of dimensionality. Surely, almost any algorithm that solves optimization problems have difficulties with increasing problem size, but such a growth is seldom occurring. There are really two types of problems we encounter as a consequence of the curse of dimensionality.

First, the number of computations reach enormous amounts. The example above, shows that with $m = 10$ and $I = 8$ we must solve more than a billion optimization problems at each stage in the problem. Clearly, this is intractable if the subproblems take any time at all.

Second, we need to store values from one stage to another to be able to compute the value function. Hence, if we use 4 byte reals[3] to store $V_t(i_1, \ldots, i_m)$, we need storage capacity of $\frac{1073741824 \cdot 4}{1024^3} = 4$ Gb[4] for the case with $m = 10$ and $I = 8$. This is a really huge number. Today's PC is seldom equipped with disks of such magnitude[5], even a standard workstation configuration, would have problems storing such enormous amounts of data.

Therefore, the second problem is often the problem we meet first if we try to implement SDP problems on a computer.

[3] a real is a computer language term describing what type of number we can store in this type. As the name should indicate this type stores real numbers. Byte is a unit measuring the space occupied by data elements in a computer. Hence, a 4 byte real stores real numbers in an "area" of 4 bytes.

[4] 1 Gb= 1024^3 Bytes

[5] Today was 1994. Now, in 2016, things have changed, but the principle remains.

Surely, if we overcome the storage problem, the problem of a huge number of computations still remain.

In this perspective, it is interesting to try to judge whether a huge stage space is a “normal” situation. In the examples we have looked at so far, this was not the case. However, if we want to deal with real world problems, a huge state space is unluckily very often a problem. For instance, a real world situation is seldom characterized by single decisions. Very often, one decision is followed by other decisions dependent on the first decision. If we want to decide on where and when to build a factory, we normally also want to run the factory afterwards. If we want to locate a warehouse, we need to make transport decisions in subsequent stages. If we want to decide what dimensions a natural gas pipeline should have, we later on need to take care of pricing schemes for various users of the pipeline – possibly dependent on which dimension we chose earlier.

Let us extend our problem of selling a house to illustrate these points. Assume that we change our focus from private house selling to a real estate perspective. Hence, we face a set of houses to sell, and a possible decision problem could be to decide when to sell a house. A natural set of decision variables may be:

$$a_{kn} = \begin{cases} 1 & \text{if house } k \text{ is sold at stage } n \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

If the decision of selling a house imply no other consequences for our real estate business, we may look at each house apart and solve problems equivalent to those in sections 1.3 – 3.6. However, assume that a house selling decision implies resource consequences for our firm at subsequent stages. For instance, it may be necessary (for our firm) to maintain the house after we have sold it, we may need to develop infrastructure – roads, sewerage systems, water pipes and so on. Surely, such commitments will depend on the type of property we sell, but at least for apartments, various type of after-sale commitments exist. If the after-sale commitments are of geographical type, it seems sensible to assume that they may vary from house to house. Surely, it may be hard to predict such future commitments. Nevertheless, let us assume that we are able to predict them with certainty. Hence, let q_{ku} be the resources needed after selling house k in time period u independent of selling time. Figure 4.1 show the structure of q_{ku}.

Observe from figure 4.1 that q_{ku} is independent of when the sale is performed. That is, if house k is sold at stage n, we need to cope with resources

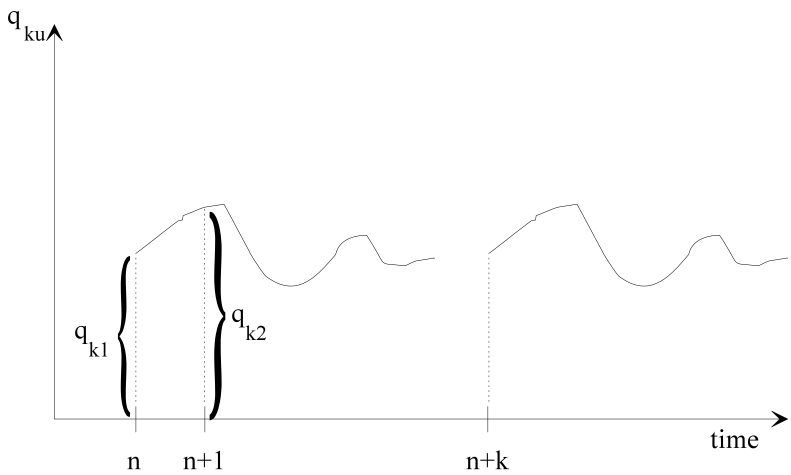


Figure 4.1: Future resource needs after selling a house

q_{k1} at stage n, q_{k2} at stage $n+1$ and so on. Alternatively, if the house is sold at stage $n+k$, resources needed at $n+k$ is q_{k1}, q_{k2} at $n+k+1$ etc.

To make these resource commitments interesting, we need some limitations on total use of resources at each stage. Let Q_n be the available amount of resource at stage n. For instance, the actual resources may involve building roads connecting the houses to an available road system, and we have a limited crew of road builders.

So far, we have not said anything about uncertainty. In previous examples, the sales price of houses were modelled as a stochastic variable. This seems still to be a sensible assumption. Actually, the point we want to make is not necessarily dependent on which stochastic mechanism we implement. Therefore, we need not specify how the stochastic sales price may be modelled. Let us instead look at a certain stage n and ask the following question: What is the necessary information we need to make a decision at this stage. Surely, we need the value function at stage $n+1$ and a stochastic mechanism to compute the expected value of this function. But we also need some additional information. Not only do we need information on whether a house has been sold or not, but if it has been sold, we need to know when. This information is necessary in order to check the resource constraints. That is, whether a decision/state combination is legal. Hence, we need for each house a state variable describing whether the house has been sold earlier and if so when. The following state variable structure yield the necessary information:

Let i_{kn} be a state variable associated with house k at stage n and let i_{kn} take on the following values:

$$i_{kn} \in \{0, 1, \ldots, n-1\} \tag{4.5}$$

That is, if $i_{kn} = 0$, the house has not been sold before stage n. Alternatively, if $i_{kn} = 1, 2 \ldots, n-1$, the house was sold at stage 1, 2 etc.

We will not pursue the solution process of this example any further. The point is not to solve it, but to illustrate how easy a multidimensional and huge state space may be generated. Suppose that the real estate firm want to look at this problem over a 10 period time horizon, and that 10 houses are available for sale. At the last stage, each of the i_{kn} variables can take 10 values; $0, 1, \ldots, 9$. As there are 10 houses available, the possible number of state combinations will be 10^{10} – a very nasty number. Note that this is even before a possible set of stochastic state variables are included.

Obviously, dependent of the actual values of Q_n and q_{ku}, we may reduce the effective state space even before we start solving our SDP. If the Q_n's are "small" compared to the q_{ku}'s, we have a somewhat heavily constrained problem and a lot of possible state combinations will become illegal. This fact may give an alternative explanation of why standard software for SDP and DP are more or less absent. As this example shows, the data of the problem may be crucial when it comes to the determination of whether a problem is practically solvable or not.

As the discussion above has shown, the curse of dimensionality is not limited to the stochastic case. As a matter of fact, the dimensionality problems of our example came as a result of a huge "deterministic state space". Obviously, things do not become easier if we are to move from a deterministic to a stochastic problem as we – at least in a normal situation – would need a larger state space to take care of the stochasticity. So even if the curse of dimensionality is characteristic of any type of DP problem, deterministic or stochastic, a stochastic problem is normally even harder to solve than a deterministic one.

Surely, our example and discussion of the curse of dimensionality should have stressed the importance of the problem. As a consequence, a lot of effort has been put into finding methods to cure the "curse". This research has not given conclusive results. Hence, the topic is still undergoing a lot of research. We will return to discuss some of the possible "cures" in a later section.

4.2 Problem structure

Very often, the curse of dimensionality is explained by an example along the following lines. (See for instance Bellman and Dreyfus (Bellman and Dreyfus, 1962) or Ravindran et al. (Ravindran et al., 1987).) Look at the following deterministic optimization problem:

$$\begin{array}{ll} Max & \sum_{i=1}^{N} g_i(x_i) \\ s.t. & \sum_{i=1}^{N} b_{ij} x_j \leq c_j \ \forall j \in \{1, 2, \ldots, M\} \\ & x_i \geq 0, b_{ij} \geq 0 \end{array} \tag{4.6}$$

Hence, we are facing an optimization problem with separability in the objective and M linear constraints. The normal way of solving such a problem by dynamic programming, is to assign a stage to each variable. The state space is constructed by noting that if the remaining resources available is known at each stage, the constraints may be checked. As an example, suppose one of the constraints looks as follows:

$$2x_1 + 3x_3 + 4x_4 + 5x_5 \leq 10 \tag{4.7}$$

Surely, if we start at stage 5 the remaining resources to use at stage 4 is;

$$S_4 = 10 - 5x_5 \tag{4.8}$$

Moving on to stage 3, the remaining resources is;

$$S_3 = 10 - 5x_5 - 4x_4 = S_4 - 4x_4 \tag{4.9}$$

The general state space structure then implies M state variables – one for each constraint – which can be recursively updated as follows:

$$S_{n-1}^j = S_n^j - b_{nj} x_n \ \forall j \in \{1, 2, \ldots, M\} \tag{4.10}$$

The point to note here, is the M state variables. Surely, if M is a large number, our problem very soon becomes intractable. It is very easy to interpret this example as a general weakness of DP (and SDP) in handling multiple constraints. Surely, the state space definition of example (4.6) gives such a result, but additional constraints do not need to increase the state space. Let us discuss this fact by returning to the example in section 4.1. Assume that the real estate firm cannot sell any house in any time period, and that the firm is able to decide which periods are legal sale periods for each house. (A simple practical situation explaining the existence of such constraints, may be that the firm does not own the houses yet. However, a certain buying plan exists. For instance, house 1 may be bought in period 1 while house 5 is bought in period 6.) Such constraints may easily be handled by the existing state space description. Assume for instance that house k is bought in period t_k. Then, the general state space structure at stage n may be reduced directly as follows:

$$i_{kn} \in \{0, t_k, t_{k+1}, \ldots, n-1\} \ \forall k \tag{4.11}$$

As an example of such a reduction; assume that house k is bought in time period k. That is, house 1 is bought in period 1, house 2 is bought in period 2 etc. Then the possible number of state combinations is 10! which is a much smaller number than 10^{10}.

The point we wanted to make is indeed very simple. It is not necessarily the number of constraints that blows up the state space.

We might say that the important thing to achieve if DP or SDP is to be applied on a problem, is to find an effective state space definition. Surely, the whole problem structure must be taken into consideration when this modelling decision is to be made.

Chapter 5

Infinite horizon problems

If we look back on section 3.5 we solved an infinite horizon problem. The key point in such problems is to find a so called stationary policy. Such a policy is characterized by being independent of the stages in the model. Hence, in a simple SDP model, we want to find decisions that only depend on the states of the system. The literature tends to use an alternative name for such problems – Markov decision processes or MDP's. To understand the reason for this somewhat confusing term, look again at the fundamental optimality equation;

$$V_n(i) = \max_a \left[R(i,a) + \sum_j P_{ij}(a) V_{n+1}(j) \right] \tag{5.1}$$

As mentioned earlier, $P_{ij}(a)$ is a family of discrete Markov transition matrices – hence the name MDP. So the focus of this section is to discuss various solution possibilities to problem (5.1) under an infinite horizon assumption. As our example in section 3.5 showed, such an assumption is normally simplifying. Surely, it should be simpler to search for a stationary than a non stationary policy.

There are several methods available for solving this type of problem. Let us describe some of them by an example. Basic data for this example are presented in the next section.

5.1 Data for the MDP-example

Assume that you own a flat. This flat is rented to some people with a somewhat unstable paying pattern. The full rent per time period is \$200. Unluckily, your tenants do not always pay the full rent. Hence, the tenants pay you either \$200 (H), \$150 (M) or \$100 (L). The reason for this pattern of payments are due to the fact that you as the owner of the flat, have some obligations when it comes to maintenance. From your point of view, two possible maintenance strategies exist, high effort (HE) which costs \$100 or low effort (LE) which costs \$20. The net profit in each period may be summed up as in Table 5.1.

Table 5.1: Net profit in each time period for various payment and maintenance possibilities

	HE	LE
H	100	180
M	50	130
L	0	80

The link between your choice of strategy {HE, LE} and your tenants choice of strategy {H, M, L} is not certain. Table 5.2 gives information on the probabilistic nature of the causality between strategic choices from your point of view.

Note from Table 5.2 the dual nature of the causality. Your maintenance effort is only partly determining the probabilities for the state values in the next period, also the observed state value has impact. The probability of obtaining a high payment in the next period, if today's payment is high, is larger than if today's payment is low – independently of your action. Hence, you assume some kind of underlying stochastic process guarding the payment scheme. A sensible practical explanation on such, may be that today's payment is an indication on the present economic situation for your tenants. Hence, if you observe a high payment today, you assume that your tenants also have money in the next period.

The decision problem facing you, may then be described as follows: In each time period, you observe which payment you receive. Based on this

Table 5.2: Probabilities for High (H), Medium (M) or Low (L) payments in the next period, given observed state values and your decisions

State	Decision	H	M	L
H	HE	0.8	0.1	0.1
	LE	0.4	0.3	0.3
M	HE	0.4	0.4	0.2
	LE	0.3	0.3	0.4
L	HE	0.4	0.3	0.3
	LE	0.1	0.1	0.8

information, you must decide on which maintenance effort to use. This decision picks a future described by Table 5.2. This information decision pattern is then repeated infinitely.

A sensible objective may seem to be a maximization of the total expected net profit. However, it is easily observed from equation 5.1 that such a strategy may be hard to implement. As our problem lacks any impatient cost, refer to the example in section 3.5, the objective will be unbounded; i.e. grow to infinity. As we are interested in finding policies – that is strategies independent of time – we might as well maximize the expected net profit **per period**. The alternative to this interpretation is to introduce explicit impatience costs, normally in the form of discounting. We will briefly return to this type of models later.

5.2 Full enumeration

The conceptually simplest way of solving the problem outlined in section 5.1 is to go through a full enumeration of all possible policies and choose the one with the largest expected net profit per period. Let us start by finding all possible policies. Remember that a policy is a state but not time dependent choice of strategies. In our case, we have 3 states; H, M, L and 2 decisions; HE, LE. This give 2^3 possible policies. Each of these policies have an associated state dependent net profit. Table 5.3 shows all possible policies and associated net profits.

Each of the policies outlined in Table 5.3 has an associated Markov tran-

Table 5.3: Possible policies and associated net profits for the MDP-example

	State	1	2	3	4	5	6	7	8
Policy	H	HE	HE	HE	HE	LE	LE	LE	LE
	M	HE	HE	LE	LE	HE	HE	LE	LE
	L	HE	LE	HE	LE	HE	LE	HE	LE
Net profit	H	100	100	100	100	180	180	180	180
	M	50	50	130	130	50	50	130	130
	L	0	80	0	80	0	80	0	80

sition matrix. For instance, using Table 5.2 it is easily seen that policy 1 has the following matrix of transition probabilities;

$$\mathbf{P}_1 = \begin{array}{c|ccc} & H & M & L \\ \hline H & 0.8 & 0.1 & 0.1 \\ M & 0.4 & 0.4 & 0.2 \\ L & 0.4 & 0.3 & 0.3 \end{array} \tag{5.2}$$

while policy 4 has the following matrix of transition probabilities:

$$\mathbf{P}_4 = \begin{array}{c|ccc} & H & M & L \\ \hline H & 0.8 & 0.1 & 0.1 \\ M & 0.3 & 0.3 & 0.4 \\ L & 0.1 & 0.1 & 0.8 \end{array} \tag{5.3}$$

Note that we use the notation $\mathbf{P}_p$ for the matrix of transition probabilities associated with policy p, $p \in \{1, 2, \ldots, 8\}$.

If we knew the probabilities of observing states H, M and L for each of the possible policies in the long run, we could merely compute the expected profit for each policy and choose the policy with the largest expected profit. Luckily, such long run probabilities are easily obtained. The theory of stochastic processes give a direct answer to this problem – refer for instance to (Ross, 1996). Hence, long run or steady state probabilities may be calculated for each policy p by the following set of linear equations:

$$\pi_j^p = \sum_{i=1}^{I} \pi_i^p p_{ij}^p \ , \ \sum_{j=1}^{I} \pi_j^p = 1 \ ; \forall j \in \{1, 2, \ldots, I\} \text{ and } \forall p \in \{1, 2, \ldots, P\} \tag{5.4}$$

The notation in equation (5.4) has the following meaning: π_j^p is the steady state probability of observing state j given policy p. p_{ij}^p is element ij in matrix P_p. I is the number of states (3 in our example), while P is the number of policies (8 in our example). To show how one of these π's can be calculated let us look at π^1. Using equation (5.4) we get the following equational system:

$$\begin{array}{rcrcrcr} \pi_1^1 & = & 0.8\pi_1^1 & + & 0.4\pi_2^1 & + & 0.4\pi_3^1 \\ \pi_2^1 & = & 0.1\pi_1^1 & + & 0.4\pi_2^1 & + & 0.3\pi_3^1 \\ \pi_3^1 & = & 0.1\pi_1^1 & + & 0.2\pi_2^1 & + & 0.3\pi_3^1 \\ 1 & = & \pi_1^1 & + & \pi_2^1 & + & \pi_3^1 \end{array} \tag{5.5}$$

Any one of the three first equations in system (5.5) may be removed as one is redundant. (Refer to (Ross, 1996) for an explanation.) Solving system (5.5) yields the following solution:

$$\pi_1^1 = 0.6667 \ , \ \pi_2^1 = 0.1852 \ , \ \pi_3^1 = 0.1481 \tag{5.6}$$

Table 5.4 shows the results of performing the same type of calculations for the other 7 policies: (Note that we use the vector notation $\pi^k = [\pi_1^k, \pi_2^k, \pi_3^k]$.)

Table 5.4: Stationary distributions for all possible policies

State	π^1	π^2	π^3	π^4	π^5	π^6	π^7	π^8
H	.6667	.4762	.6379	.4167	.4000	.2326	.3700	.1923
M	.1852	.1429	.1724	.1250	.3333	.2093	.3000	.1731
L	.1481	.3809	.3809	.4583	.2667	.5581	.3300	.6346

Now we have the necessary information to solve our problem. Computing the scalar products of the vectors π^k and the corresponding vectors of state dependent net profits from Table 5.3 we obtain Table 5.5.

If we look at Table 5.5, we observe immediately that the optimal policy is policy 8. This policy, applying LE (Low Effort) in any state is the policy with

Table 5.5: Expected per period net profits for all possible policies

1	2	3	4	5	6	7	8
75.930	85.237	86.202	94.584	88.665	96.981	105.600	107.885

the maximal expected per period net profit. Hence, the landlord's strategy is the somewhat cynical one of maintaining the flat as little as possible. Even though many people may feel that such a strategy is quite common in practice, it is easy to change the data somewhat such that an alternative strategy yields higher expected per period net profit than the given one. For instance, assume that the cost associated with making a low effort is changed from \$20 to \$40, all other data unchanged. Then, the expected per period net profits for policy 7 and 8 becomes $\{160, 110, 0\}$ for policy 7 and $\{160, 110, 60\}$ for policy 8. Computing expected values for the two policies yields 92.200 for policy 7 and 87.885 for policy 8. Hence under these assumptions, the optimal strategy has changed.

5.3 Using LP to solve MDP's

In the former section, we showed how we could solve an MDP problem by full enumeration. Surely, for small problems like our example, a full enumeration is feasible. Suppose however that we looked at a problem with 10 possible actions and 10 states. Such a situation would yield 10^{10} possible policies and we would have to solve 10^{10} linear equational systems with 10 variables in each to find the stationary probabilities. Such a task is formidable. As much of the point of OR-techniques is to avoid full enumeration, we might suspect that somebody have designed algorithms to solve such problems without performing a full enumeration. Indeed this is the case. Manne (Manne, 1960) designed a linear programming formulation of this problem. According to White (White and White, 1989), linear programming is the only feasible solution technique for practical MPD's.

Let us describe Manne's formulation. We start by introducing some notation. Let

$$\delta_{ia} = \begin{cases} 1 & \text{if decision } a \text{ is chosen for observed state } i \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

where a is a decision, chosen from the set $\{1, 2, \ldots, A\}$ and i is a state, chosen from the set $\{1, 2, \ldots, I\}$.

We add the following set of constraints on δ_{ia}:

$$\sum_{a=1}^{A} \delta_{ia} = 1 \; \forall i \in \{1, 2, \ldots, I\} \tag{5.8}$$

Given the definition (5.7) and the constraints (5.8), δ_{ia} may be interpreted as a binary variable picking all possible policies. For instance, policy 5 from Table 5.3 can be picked by assigning the following values to δ_{ia};

$$\begin{array}{ll} \delta_{11} = 0 & \delta_{12} = 1 \\ \delta_{21} = 1 & \delta_{22} = 0 \\ \delta_{31} = 1 & \delta_{32} = 0 \end{array} \tag{5.9}$$

where state values 1, 2, 3 corresponds with H, M, L and decisions 1, 2 corresponds with HE, LE. Hence, the problem we would like to solve, is to assign a set of values to δ_{ia} (picking a policy) which maximizes expected per period net profit. If such an optimization problem can be formulated, we can use mathematical programming methods to find the solution. As the title to this section indicates, we are looking for a linear programming formulation. The set of decision variables δ_{ik} which we have formulated, indicate however an integer program. To avoid an integer program formulation, we perform a little trick. We introduce the term *randomized policy*. A randomized policy is an extension to a *deterministic policy*, characterized by the fact that we allow ourselves to choose the probability of performing an action. Suppose we change the values of policy 5 as follows:

$$\begin{array}{ll} \delta_{11} = 0 & \delta_{12} = 1 \\ \delta_{21} = \frac{1}{2} & \delta_{22} = \frac{1}{2} \\ \delta_{31} = 1 & \delta_{32} = 0 \end{array} \tag{5.10}$$

Then, policy 5 may be interpreted as follows: If state 1 (H) is observed, we make decision 2 (LE) with certainty. If state 3 (L) is observed, we make decision 1 (HE) with certainty. However, if state 2 (M) is observed, we toss a coin to decide on which action (HE or LE) to do. That is, we may interpret δ_{ia} as follows:

$$\delta_{ia} = P(decision = a|state = i) \tag{5.11}$$

Still, δ_{ia} must sum to 1 for each a, but the binary definition (5.7) is changed to

$$0 \leq \delta_{ia} \leq 1 \ \forall i \text{ and } a \tag{5.12}$$

It is convenient to formulate our linear program with unconditional probabilities as decision variables. That is, let

$$y_{ia} = P(state = i \cap decision = a) \tag{5.13}$$

Applying the definition of conditional probability:

$$P(state = i \cap decision = a) = P(decision = a|state = i)P(state = i) \tag{5.14}$$

or

$$y_{ia} = \delta_{ia}\pi_i \tag{5.15}$$

Performing a summation over all a in equation (5.15) yields

$$\sum_{a=1}^{A} y_{ia} = \sum_{a=1}^{A} \delta_{ia}\pi_i = \pi_i \sum_{a=1}^{A} \delta_{ia} \tag{5.16}$$

Using equation (5.8), equation (5.16) becomes

$$\pi_i = \sum_{a=1}^{A} y_{ia} \tag{5.17}$$

Combining equation (5.17) and equation (5.15) we get

$$\delta_{ia} = \frac{y_{ia}}{\sum_{a=1}^{A} y_{ia}} \tag{5.18}$$

Therefore, our original decision variables δ_{ia} are easy to calculate by equation (5.18) when the y_{ia}'s are known.

Let us now turn to the expression for the objective function. Let R_{ia} be the per period net profit of observing state i and making decision a. If we return to the example in section 5.1 this values are readily available from Table 5.1. That is,

$$\begin{array}{ll} R_{11} = 100 & R_{12} = 180 \\ R_{21} = 50 & R_{22} = 130 \\ R_{31} = 0 & R_{32} = 80 \end{array} \tag{5.19}$$

As these values are the outcomes of the stochastic variable y_{ia}, the expected per period net profit, which is our objective, is easily calculated as follows:

$$E(R) = \sum_{i=1}^{I} \sum_{a=1}^{A} R_{ia} y_{ia} \tag{5.20}$$

We have three sets of constraints we need to take into account. First, our unconditional probability density y_{ia} must sum to 1.

$$\sum_{i=1}^{I} \sum_{a=1}^{A} y_{ia} = 1 \tag{5.21}$$

Second, the stationary distribution must be computed correctly,

$$\pi_j = \sum_{i=1}^{I} \pi_i p_{ij}(a) \ , \forall j \in \{1, 2, \ldots, I\} \tag{5.22}$$

where $p_{ij}(a)$ are the transition probabilities from Table 5.2. Equation (5.22) must be expressed in the y_{ia} variables. This is easily achieved by utilizing equation (5.17) giving

$$\sum_{a=1}^{A} y_{ja} = \sum_{i=1}^{I} \sum_{a=1}^{A} y_{ia} p_{ij}(a) \ , \forall j \in \{1, 2, \ldots, I\} \tag{5.23}$$

Last, we need to incorporate the bounds (5.12). It is easy to realize that

$$\delta_{ia} \geq 0 \Rightarrow y_{ia} \geq 0 \tag{5.24}$$

The lower bound, $\delta_{ia} \leq 1$, is equivalent to (by equation (5.18))

$$\frac{y_{ia}}{\sum_{a=1}^{A} y_{ia}} \leq 1 \tag{5.25}$$

or

$$y_{ia} \leq \sum_{a=1}^{A} y_{ia} \tag{5.26}$$

as equation (5.24) holds. It is easy to realize that equation (5.26) is always satisfied.

The linear programming formulation may be summed up as follows:

$$\begin{array}{ll} Max & \sum_{i=1}^{I}\sum_{a=1}^{A} R_{ia}y_{ia} \\ s.t. & \sum_{i=1}^{I}\sum_{a=1}^{A} y_{ia} = 1 \\ & \sum_{a=1}^{A} y_{ja} - \sum_{i=1}^{I}\sum_{a=1}^{A} y_{ia}p_{ij}(a) = 0 \ , \ \forall j \in \{1, 2, \ldots, I\} \\ & y_{ia} \geq 0 \ , \ \forall i \in \{1, 2, \ldots, I\} \ , \ \forall a \in \{1, 2, \ldots, A\} \end{array} \tag{5.27}$$

Let us now use this formulation to formulate and solve the example in section 5.1. Using the data from section 5.1 and equation (5.27) we obtain the following linear programming problem:

$$\begin{array}{ll} Max & 100y_{11} + 180y_{12} + 50y_{21} + 130y_{22} + 80y_{32} \\ s.t & y_{11} + y_{12} + y_{21} + y_{22} + y_{31} + y_{32} = 1 \\ & y_{11} + y_{12} - (0.8y_{11} + 0.4y_{12} + 0.4y_{21} + 0.3y_{22} + 0.4y_{31} + 0.1y_{32}) = 0 \\ & y_{21} + y_{22} - (0.1y_{11} + 0.3y_{12} + 0.4y_{21} + 0.3y_{22} + 0.3y_{31} + 0.1y_{32}) = 0 \\ & y_{31} + y_{32} - (0.1y_{11} + 0.3y_{12} + 0.2y_{21} + 0.4y_{22} + 0.3y_{31} + 0.8y_{32}) = 0 \\ & y_{11}, y_{12}, y_{21}, y_{22}, y_{31}, y_{32} \geq 0 \end{array} \tag{5.28}$$

Solving the linear program above, yields the following solution:

$$y_{12} = 0.1923 \ , \ y_{22} = 0.1731 \ , \ y_{32} = 0.6346 \ , \ y_{11} = y_{21} = y_{31} = 0 \tag{5.29}$$

Using equation (5.18), the corresponding optimal policy can be calculated as:

$$\delta_{12} = \delta_{22} = \delta_{32} = 1 \ , \ \delta_{11} = \delta_{21} = \delta_{31} = 0 \tag{5.30}$$

Note that this is the same policy as the one we found by full enumeration in section 5.2. This may come as a surprise as the linear programming formulation introduced a more general problem than the problem we solved in the full enumeration case. Remember that the linear programming formulation allowed for randomized policies as opposed to the case in section 5.2.

However, our example gave a deterministic policy as the optimal solution. Actually, it can be shown that this will always be the case. That is, a randomized policy will never be optimal. Refer to Derman (Derman, 1962) for a formal proof of these characteristics of the Linear Program (5.27).

So far, we have presented two methods for solving MDP's – full enumeration and linear programming. There exists also another well known method often referred to as *policy improvement* due to Howard (Howard, 1960*b*). It is however more suitable to present that method when we turn our attention to problems with discounting in the next section.

5.4 Discounted returns

As mentioned in section 5.1, an alternative way of looking at infinite SDP problems, as opposed to the per period approach in the former sections, is to introduce discounted returns. Let us make a simple reformulation of the optimality equation (1.6) and discuss the implications.

$$V_n(i) = \max_a \left[R(i,a) + \alpha \sum_j P_{ij}(a) V_{n+1}(j) \right] \tag{5.31}$$

The simple reformulation consists merely of the introduction of a parameter α which we normally refer to as the *discount factor*, $0 < \alpha < 1$. Surely, the role of the discount factor, is to bound the objective such that we do not get the problem of an infinite objective. Hence, we may look at the total expected discounted cost instead of the per period approach in the former sections.

The motivation for introducing discounting is picked up from economic theory. One of the basic assumptions in all economic theory is the existence of a so called time preference. This term refers to the fact that the value of a cash-flow depends on when an economic agent receives it. Normally, one assumes that the cash flow is more valuable the sooner it is received. One simple example may clarify. Assume that you have \$100 available today, and that you may invest your \$100 in a bank to 10% interest. Under the assumption of no taxes and inflation, you would receive \$110 next year if you put your money in the bank. Hence, the value of \$100 today equals the value of \$110 in one year. Or,

$$100 = \alpha 110 \Rightarrow \alpha = \frac{100}{110} \tag{5.32}$$

Suppose alternatively that you had several investment opportunities say a whole pile of banks giving different interests. As a rational investor, you would check the market and pick the best opportunity. Hence, a discount factor is often said to be the best alternative investment possibility facing an agent. All these seems simple and easy. There are however some problems involved. Suppose that instead of investing \$100 you have $\$10^9$ to invest. What would happen in your local bank if that amount of money was handled over the desk? Surely, several possibilities exist. You could get your 10%, or you could get a higher interest. But you could also be rejected. The bank might fear the possibility of paying you $\$1.1 \cdot 10^9$ next year. This somewhat exaggerated example tries to point to the fact that the discount factor, or the return an investor can get on his money, partly is determined by his own actions. Of course, other agents also influence possible returns. The problem in our setting is that the decision maker's actions may influence the value of α while the problem (5.31) is establishing the actions – a somewhat "tail-biting" type of problem.

Simultaneously, we face the problem of uncertainty. As we deal with stochastic problems, we have to ask whether stochasticity influences time preference. Surely, it does. Suppose you changed your focus from bank investing to stock, options or other financial instruments. Very soon, you would observe that the expected return depends on the stochastic structure of the investment. Loosely – the more risk involved in an investment, the higher the expected return. Hence, it is very sensible to question the problem formulation in equation (5.31) in this perspective. Many books have been written on these subjects and many more books will be written on them. From our perspective, the point is more or less to ask some questions, not to give answers. The interested reader is referred to Hirschleifer (Hirschleifer, 1970). Let us therefore assume that the task of obtaining a discount factor is feasible and that the model (5.31) may be used. In the next sections we will briefly discuss some methods available for solving MDP's with discounted returns.

5.5 Method of successive approximations

It can be shown under quite general assumptions (Ross, 1983) that the optimal policy in the infinite case must satisfy the following equation

$$V(i) = \max_a \left[R(i,a) + \alpha \sum_j P_{ij}(a) V(j) \right] \tag{5.33}$$

Note that the basic assumption which leads to equation (5.33) is the convergence of V_n. That is:

$$n \to \infty \Rightarrow V_n(\cdot) \to V_{n+1}(\cdot) \to V(\cdot) \tag{5.34}$$

Refer also to the example in section 3.5. In this example, we computed the reserve price for various stage values (Table 3.3) and observed the convergence.

Surely, we could try to solve the problem formulated in equation (5.33) by attacking it directly. For instance, if we have a continuous action space, we may try to solve the maximization problem, possibly by methods of calculus. Unluckily this is seldom possible. An alternative approach is to use the so called **Method of successive approximations**. This is a very simple method and involves solving the finite time problem until convergence of the value function. That is, we start out in iteration 0 by setting all $V(i)$'s to 0 and iterate as a normal finite problem.

Then the iteration goes:

0 **Initialize**	set $n := 1$ and $V^0(i) = 0, \forall i \in \{1, 2, \ldots, I\}$
1 **Solve**	$V^n(i) = \max_a \left[R(i,a) + \alpha \sum_j P_{ij}(a) V^{n-1}(j) \right]$
2 **Check convergence**	if V^n is "suitably close" to V^{n-1} **stop**. Otherwise, set $n := n + 1$ and **go to** step 1.

(5.35)

Using the data from section 5.1 we can test this approach. Table 5.6 shows the development of the value function for 5 iterations.

To obtain the numbers in Table 5.6 we must specify a discount factor. We have used $\alpha = 0.1$. This is not necessarily a very sensible choice, but it gives reasonably fast convergence as Table 5.6 should indicate. (It should be easy to realize that the convergence speed and discount factor are inversely related in this algorithm.) To explain one of the numbers: First, we perform

Table 5.6: Behaviour of the Method of successive approximations

State(i)	$V^0(i)$	$V^1(i)$	$V^2(i)$	$V^3(i)$	$V^4(i)$	$V^5(i)$
1 (H)	0.00	180.00	193.50	194.70	194.81	194.82
2 (M)	0.00	130.00	142.50	143.66	143.77	143.78
3 (L)	0.00	80.00	89.50	90.52	90.63	90.64

the initialization step, $V^0(i) = 0\ ,\ \forall i$. Then, we obtain the the values for $V^1(i)$ in Table 5.6 merely by maximizing each row in Table 5.1. $V^2(1)$ is found by the following expression:

$$\begin{aligned} V^2(1, a = 1) &= 100 + 0.1\,(0.8 \cdot 180 + 0.1 \cdot 130 + 0.1 \cdot 80) = 116.5 \\ V^2(1, a = 2) &= 180 + 0.1\,(0.4 \cdot 180 + 0.3 \cdot 130 + 0.3 \cdot 80) = 193.5 \end{aligned} \tag{5.36}$$

Hence, $V^2(1) = max[116.5, 193.5] = 193.5$.

Note that the policies associated with each iteration is missing in Table 5.6. This is due to the fact that the optimal policy (Low effort in any state) is obtained immediately. This is not a general characteristic. However, it is not uncommon to obtain the optimal policy faster than the optimal value. We do however need to carry on until convergence is established in the value functions to “prove” that the policy we have identified is the optimal one.

Note that this type of algorithm is not applicable in the case investigated in sections 5.2 – 5.3 due to lack of discounting.

5.6 Method of policy improvement

This method is somewhat different, but still conceptually simple. Assume that you pick one policy p – maybe at random. If we return to equation (5.33) it should be easy to realize that we can use it to evaluate this policy. The following linear equational system (I equations in I unknowns) may thus be solved;

$$V_p(i) = R(i, p(i)) + \alpha \sum_j P_{ij}(p(i)) V_p(j)\ ,\ \forall i \in \{1, 2, \ldots, I\} \tag{5.37}$$

where we use the notation V_p for the value function of the given policy p, and $p(i)$ for the policy to emphasize the fact that it is state dependent. If we are able to find another policy which is better then p then we can evaluate the new one in the same fashion. According to the title of this section, the fact that this is possible should not be surprising. The improvement step is done by performing the following set of calculations.

$$\max_a \left[R(i,a) + \alpha \sum_j P_{ij}(a) V_p(j) \right] \tag{5.38}$$

Hence, this algorithm has a two stage structure. First, we evaluate a given policy – often referred to as *value determination*. Second, we improve the policy and find a better one – often referred to as *Policy improvement*. This algorithm is due to Howard (Howard, 1960*b*). Proofs of convergence can be found in Ross (Ross, 1983).

The algorithm may be outlined as follows:

0 **Initialize**	set $n := 1$ and choose some policy. Name the policy p^n
1 **value determination**	Solve $V^n(i) = R(i, p^n(i)) + \alpha \sum_j P_{ij}(p^n(i)) V^n(j)$
2 **policy improvement**	Solve $max_a \left[R(i,a) + \alpha \sum_j P_{ij}(a) V^n(j) \right]$ $\Rightarrow a^* = p^{n+1}$
3 **check convergence**	if $p^n = p^{n+1}$**stop**. Otherwise, set $n := n+1$ and **go to** step 1.

(5.39)

Let us apply this algorithm to our example. The simplest thing to do would of course be to start with the optimal policy. Then, the algorithm should terminate after only 1 iteration. Let us check it out. Using the policy "Low Effort in any state" yields the following step 1 calculations:

$$\begin{aligned} V^1(1) &= \ 180 + 0.1\,(0.4V^1(1) + 0.3V^1(2) + 0.3V^1(3)) \\ V^1(2) &= \ 130 + 0.1\,(0.3V^1(1) + 0.3V^1(2) + 0.4V^1(3)) \\ V^1(3) &= \ 80 + 0.1\,(0.1V^1(1) + 0.1V^1(2) + 0.8V^1(3)) \end{aligned} \tag{5.40}$$

The system of linear equations (5.40) gives the following solution:

$$V^1(1) = 194.826,\ V^1(2) = 143.784,\ V^1(3) = 90.637 \tag{5.41}$$

If we compare this numbers to the numbers in Table 5.6 we observe a close correspondence. Hence, our calculations should be correct. As this indeed is the optimal solution, we should expect a policy improvement step without effect. This is also the case. The calculations in the policy improvement step is illustrated in Table 5.7.

Table 5.7: Policy improvement step

State(i)	$a = 1$	$a = 2$	$\max_a$	a^*
1 (H)	117.930	194.826	194.826	2
2 (M)	65.357	143.784	143.784	2
3 (L)	13.826	90.637	90.637	2

As an example on how one of the numbers in Table 5.7 is found, let us calculate the value in the upper left corner ($i = 1, a = 1$). This value is computed as:

$$117.930 = 100 + 0.1\,(0.8 \cdot 194.826 + 0.1 \cdot 143.784 + 0.1 \cdot 90.637) \qquad (5.42)$$

We observe from Table 5.7 that the policy is unchanged. Thus, it is optimal.

5.7 Concluding remarks

In these sections, we have discussed two versions of Markov decision processes – with and without discounting. We have looked at different methods for the two cases. Note however, that it exists a linear programming formulation due to d'Epenoux (d'Epenoux, 1960) for the case with discounting. This formulation is very close to the one we have presented in section 5.3 so it is not discussed any further here. Both Ross (Ross, 1983) and Hillier and Lieberman (Hillier and Lieberman, 1989) discuss this method. A version of the policy improvement method is also available for problems without discounting. An example may for instance be found in Hillier and Lieberman (Hillier and Lieberman, 1989).

Chapter 6

Recent research

This concluding chapter will briefly discuss some important research issues. We will not give detailed descriptions of methods, but point to some relevant literature. A good general introduction to problems and recent research on MPD's may be found in White (White and White, 1989).

6.1 "Cures" for the curse of dimensionality

As discussed in chapter 4, the curse of dimensionality is a fundamental problem in nearly all practical DP or SDP applications. Thus, it should not be surprising that this topic has seen a lot of research attention. In fact, this property is often referred to as "why SDP does not work". In the next sections, we will discuss some of the possible angles of attack to cope with the curse of dimensionality.

6.2 Compression methods

The traditional approach is that of compression. Bellman and Dreyfus (Bellman and Dreyfus, 1962) discussed this approach already in 1962. They propose to replace the value function in explicit form by a polynomial. That is,

$$V_n(i) \to f(i) \approx \sum_k a_k i^k \tag{6.1}$$

(Note that i is assumed continuous in the summation in equation (6.1)). Then, we could store a set of coefficients $a_1, a_2, \ldots$ instead of the actual $V_n(i)$'s. Hopefully, the amount of information needed is significantly smaller. Surely, this is an approximate method and it will depend critically on the "shape" of $V_n(i)$. If $V_n(i)$ is smooth, we may find good approximations by (6.1) using relatively few a-parameters. Alternatively, if $V_n(i)$ "behaves" non-smoothly, we may need more a's to get a reasonable approximation. An important problem in this area may be illustrated by the following example. Suppose $V_n(i)$ is given as in Table 6.1 at a certain stage in some problem.

Table 6.1: Example illustrating the compression problem

State(i)	$V_n(i)$
1	100
2	10
3	20
4	50
5	70
6	60
7	30
8	40
9	90
10	80

We have plotted $V_n(i)$ from Table 6.1 as a function of i in figure 6.1. Surely, this pattern may be hard to approximate by a polynomial approximation. However, suppose that we sorted the states differently. for instance as in Table 6.2. This is surely possible. The state numbering is free for us to choose.

Now, it is easy to see that $V_n(i)$ may be perfectly described by the one variable function:

$$f(i) = 10 \cdot i \tag{6.2}$$

The point of this example is to show that how we number our state space may have an important effect on how the approximation in equation (6.1)

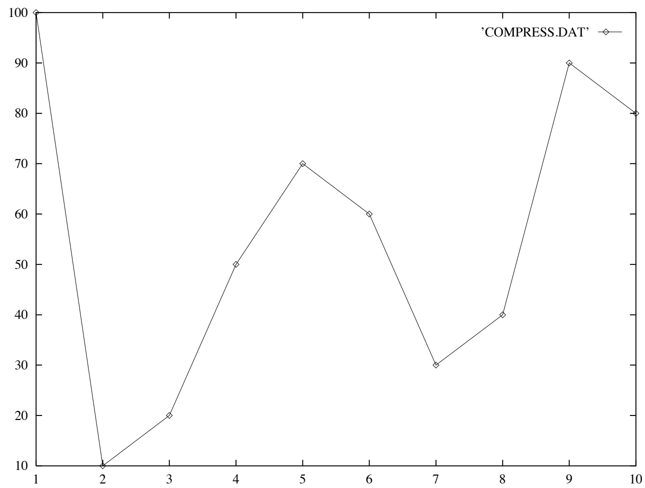


Figure 6.1: Graph of $V_n(i)$ as a function of i

Table 6.2: Example illustrating the compression problem with resorted state space

State(i)	$V_n(i)$
1	10
2	20
3	30
4	40
5	50
6	60
7	70
8	80
9	90
10	100

performs. Surely, the example was simple to investigate, but if we had a multidimensional state space such a reordering may be hard to find.

The compression techniques we have discussed so far, uses Euclidean geometry to represent data by a function. Recently, fractal geometry has gained increased popularity, especially in compression applications. Fractal geometry uses an algorithmic (or iterated) representation of “pictures” as opposed to the classical functional approach. A surprisingly complex set of graphical patterns may be represented by very simple algorithms often including only one or two parameters. To this authors knowledge, fractal compression has not yet been tested out in SDP applications. Surely, this may prove worthwhile.

A nice reference on fractal geometry and compression may be found in Barnsley (Barnsley, 1988). Haugen (Haugen, 1991) discusses various forms of compression possibilities in SDP.

6.3 State space relaxation

Another set of methods to “cure” the curse of dimensionality attacks the state space and tries to reduce it. This methods are described by Bellman and Dreyfus (Bellman and Dreyfus, 1962), Nemhauser (Nemhauser, 1966) and Ravindran (Ravindran et al., 1987). To understand this approach, return to the formulation in equation (4.6). Utilizing the general technique of Lagrange relaxation (Fisher, 1981), the problem may be converted to a problem with 1 state variable and $N+M-1$ decision variables, N original decision variables and $M-1$ Lagrange multipliers. That is, we associate Lagrange multipliers with all but one constraint, put these into the objective and solve a new problem with more decision variables but less constraints. As this formulation used one state variable for each constraint, we have reduced the state space. Surely, this does not come free and the number of decision variables are increased. Normally, such problems are solved by a search on the multipliers involving the solution of DP’s (or SDP’s) as sub problems. Other similar methods are discussed by Greenberg and Pierskalla (*surrogate multipliers*) (Greenberg and Pierskalla, 1970) and Morin and Esogbue (*Embedded state variables*) (Morin and Esogbue, 1974).

6.4 Aggregation methods

The basic idea in aggregation methods is to approximate the state and/or decision space with a new and smaller one in order to obtain a problem size that is computationally feasible. The field has two main directions:

- Aggregation based on SDP
- Aggregation based on LP

Aggregation methods based on LP is normally designed for infinite horizon problems. Remember that we could reformulate a MDP by linear programming. Then, aggregation theory of linear programming may be applied directly to the reformulated problem. Work by Zipkin (Zipkin, 1980), Mendelssohn (Mendelssohn, 1980) and Heyman and Sobel (Heyman and Sobel, 1984) cover this subject. The main idea is to obtain lower and/or upper bounds on the true objective value of the disaggregated problem by solving a smaller aggregated problem.

The main contribution of aggregation applied directly to a SDP problem is probably written by Hinderer (Hinderer, 1979). Here, the assumption of infinite horizon is not used, and Hinderer's methods may be viewed as a more general approach than the methods mentioned above.

6.5 Forecast horizon

The concept of forecast horizon relates to work by Bean and Smith (Bean and Smith, 1984), Bhaskaran and Seth (Bhaskaran and Sethi, 1985) and Hopp, Bean and Smith (Hopp, 1989). Bean and Smith (Bean and Smith, 1984) proves existence of forecast horizons for a large class of deterministic sequential optimization problems. Bhaskaran and Seth (Bhaskaran and Sethi, 1985) extends the framework to include stochastic problems with discounting. Hopp et. al. (Hopp, 1989) show that properties of the stochastic process itself is enough to ensure existence of a forecast horizon. All results are developed for infinite horizon problems (MDP's).

A forecast horizon is defined as the shortest time horizon needed in an optimization problem, in order to get a correct first period optimal solution. Thus, given the existence of a forecast horizon in a problem, we should be able to reduce the number of time periods used when solving the model.

This is if we only need the optimal solution in the first period. Normally, this is the case in a practical situation, as we would solve the model again in the next period anyway. However, if we want to inspect solution structures further into the future, the method is obviously limited.

6.6 SDP and supercomputing

According to Zenios (Zenios, 1989), the term Supercomputer was introduced around 1976 with the introduction of the CRAY 1S computer. The meaning of the term may seem vague. However, common for all supercomputers seem to be the fact that they are much faster than other computers, and that they have architectures involving some kind of parallel processing properties – refer for instance to Beasley (Beasley, 1987).

Supercomputers may be divided into two subgroups:

- Vector computers
- Parallel computers

A vector computer parallelizes at operational level, while a parallel computer duplicates the whole instruction set (processor). An excellent introduction to these topics may be found in Bertsekas and Tsitsiklis (Bertsekas and Tsitsiklis, 1989).

Today[6], it seems as if the parallel computing concept is the survivor of the two. Parallel computers come in many forms. Typically, memory and storage handling may be handled differently on different platforms. We will not pursue these matter further, but regard a parallel computer as a collection of computers able to perform computational tasks and to communicate with each other.

Such a computer framework raises interesting possibilities and problems in numerical optimization. A vast literature on related subjects should prove this – refer for instance to Zenios (Zenios, 1989).

As noted above, the introduction of parallel computers has introduced new possibilities and new problems. A "good" traditional (serial) algorithm[7] is often based on a principle of high information gathering at each iteration.

[6]Still back in 1994, still valid today (2015).

[7]We often refer to a traditional algorithmic concept as a serial type of algorithm as opposed to a parallel type of algorithm.

The simplex algorithm is a typical example of the traditional approach. At each step in the iteration, a new search direction is established by determination of which basic variable to leave and which non basic variable to enter the basis. In a parallel framework, such an algorithmic concept is not necessarily good. The reason is that successful utilization of a parallel computer involves parallel operations which again lead to a preference towards decomposition-type of algorithms. A decomposition type of algorithm is characterized by generation of sub problems often with minimal exchange of information between these sub problems. The two structures are visualized in figure 6.2.

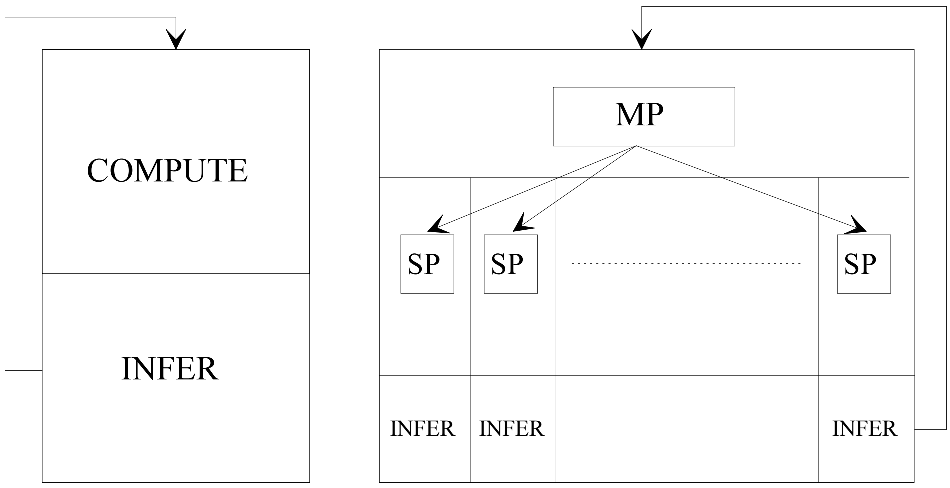


Figure 6.2: Serial type and decomposition type algorithms

As figure 6.2 shows, the traditional serial algorithmic approach on the left, has a repeated computation/inference structure. A decomposition approach splits a master problem (MP) into subproblems (SP) with individual inference. Surely, decomposition algorithms are not new. However, the introduction of usable parallel computers has to some extent rediscovered some older approaches. Refer for instance to the importance of decomposition methods in Stochastic programming. Parallel computing has also indisputably influenced modern algorithmic research, refer for instance to the method of **scenario aggregation** and the *progressive hedging algorithm* by Rockafellar and Wets (Rockafellar and Wets, 1991).

So, what has this to do with SDP? It is interesting to note that Bellman and Dreyfus (Bellman and Dreyfus, 1962) actually discuss parallel operations in relation to dynamic programming already in 1962. Surely, no usable parallel computers existed at that point. Let us return to the optimality equation and investigate it in this perspective.

$$V_n(i) = \max_a \left[R(i,a) + \alpha \sum_j P_{ij}(a) V_{n+1}(j) \right] \tag{6.3}$$

As discussed earlier, SDP is nothing more than a search technique involving decomposition. That is, for each state i, we can solve an optimization problem and each of these problems can be solved simultaneously. Of course, in some situations, we may want to use information on the solution on some state to simplify the solution to other states, but the nature of the SDP approach is well suited for parallelization. Combining these thoughts with reformulation techniques as those described in section 6.1 yield a versatile set of parallelization possibilities for SDP. A general introduction to parallelization in dynamic programming is given by Bertsekas and Tsitsiklis (Bertsekas and Tsitsiklis, 1989).

Bibliography

Barnsley, M. (1988), *Fractals everywhere*, Academic Press, New York.

Baumol, W. J. (1972), *Economic Theory and Operations Analysis*, Prentice-Hall, Englewood Cliffs, NJ.

Bean, J. and Smith, R. (1984), 'Conditions for the existence of planning horizons', *Mathematics of Operations Research* **9**, 391–401.

Beasley, J. E. (1987), 'Supercomputers and or', *J. Opl. Res. Soc.* **38**, 1085–1089.

Bellman, R. E. and Dreyfus, S. E. (1962), *Applied Dynamic Programming*, Princeton University Press, Princeton, New Jersey.

Bertsekas, D. P. and Tsitsiklis, J. N. (1989), *Parallel and distributed computation. Numerical Methods*, Prentice Hall, New York.

Bhaskaran, S. and Sethi, S. (1985), 'Conditions for the existence of decision horizons for discounted problems in a stochastic environment', *Operations Research Letters* **4**, 61–65.

Copeland, T. E. and Weston, J. F. (1983), *Finacial Theory and Corporate Policy - second edition*, Addison-Wesley, Reading Massachusetts, USA.

d'Epenoux, F. D. (1960), 'Sur un problème de production et de stockage dans l'aléatoire', *Revue Francaise Recherche Opérationelle* pp. 3–16.

Derman, C. (1962), 'On sequential decisions and markov chains', *Management Science* **9**, 16–24.

Fisher, M. L. (1981), 'The lagrangian relaxation method for solving integer programming problems', *Management Science* **27**, 1–18.

Gilbert, J. P. and Mosteller, F. (1966), 'Recognizing the maximum of a sequence', *J. Am. Statist. Assoc.* **61**, 35–73.

Greenberg, H. J. and Pierskalla, W. P. (1970), 'Surrogate mathematical programming', *Operations Research* **18**, 924–939.

Hastings, N. A. J. (1973), *Dynamic Programming with Management Applications*, Butterworth, London.

Haugen, K. K. (1991), Possible Computational Improvements in a Stochastic Dynamic Programming model for Scheduling of off-shore Petroleum fields, PhD thesis, Norwegian Institute of Technology, 7034 Trondheim NORWAY.

Haugen, K. K. (1996), 'A stochastic dynamic programming model for scheduling of offshore petroleum fields with resource uncertainty', *European Journal of Operational Research* **88**, 88–100.

Haugen, K. K. and Berland, N. J. (1996), 'Mixing stochastic dynamic programming and scenario aggregation', *Annals of Operations Research* **64**, 1–19.

Haugen, K. K., Lanquepin-Chesnais, G. and Olstad, A. (2012), 'A fast lagrangian heuristic for large-scale capacitated lot-size problems with restricted cost structures', *Kybernetika* **48**(2), 329–345.

Haugen, K. K., Løkketangen, A. and Woodruff, D. (2001), 'Progressive hedging as a meta-heuristic applied to stochastic lot-sizing', *European Journal of Operational Research* **132**, 116–122.

Haugen, K. K., Nygreen, B., Christiansen, M., Bjørkvoll, T. and Kristiansen, Ø. (1998), 'Modeling norwegian petroleum production and transportation', *Annals of Operations Reserach* **82**, 251–267.

Haugen, K. K., Olstad, A., Bakhrankova, K. and Eikenhorst, E. V. (2010), 'The single (and multi) item profit maximizing capacitated lot-size problem with fixed prices and no set-up', *Kybernetika* **46**(3), 415–422.

Haugen, K. K., Olstad, A. and Pettersen, B. I. (2007*a*), 'The profit maximising capacitated lot-size (pclsp) problem', *European Journal of Operational Research* (176), 165–176.

Haugen, K. K., Olstad, A. and Pettersen, B. I. (2007*b*), 'Solving large-scale profit maximization capacitated lot-size problems by heuristic methods', *Journal of Mathematical Modelling and Algorithms* **6**(1), 135–149.

Heyman, D. P. and Sobel, M. J. (1984), *Stochastic Models in Operations Research*, Vol. II, McGraw-Hill, New York.

Hillier, F. S. and Lieberman, G. J. (1989), *Introduction to Operations Research*, McGraw-Hill, Oakland.

Hinderer, K. (1979), On approximate solutions of finite-stage dynamic programs, Academic Press.

Hirschleifer, J. (1970), *Investment, Interest and Capital*, Prentice Hall, New York.

Hopp, W. J. (1989), 'Identifying forecast horizons in nonhomogenous markov decision processes', *Operations Research* **37**, 339–343.

Howard, R. (1960*a*), *Dynamic Probabilistic Systems*, Vol. I and II, Wiley, New York.

Howard, R. A. (1960*b*), *Dynamic Programming and Markov Processes*, John Wiley and Sons, New York.

Kall, P. and Wallace, S. W. (1994), *Stochastic Programming*, Wiley, New York.

Kaufman, G. M. (1963), 'Sequential investment analyses under uncertainty', *Journal of Business* pp. 39–64.

Lanquepin-Chesnais, G., Haugen, K. K. and Olstad, A. (2012), 'Large-scale joint price-inventory decision problems, under resource limitation and a discrete price set', *Journal of Mathematical Modeling and Algorithms* **11**(3), 269–280.

Manne, A. S. (1960), 'Linear programming and sequential decisions', *Management Science* **6**, 259–267.

Mendelssohn, R. (1980), 'Improved bounds for aggregated linear programs', *Operations Research* pp. 1451–1452.

Morin, T. L. and Esogbue, A. M. O. (1974), 'The embedded space approach to reducing dimensionality in dynamic programs of higher dimensions', *J. Math. Anal. appl.* **48**.

Nemhauser, G. L. (1966), *Introduction to Dynamic Programming*, Wiley, New York.

Ravindran, A., Phillips, D. T. and Solberg, J. J. (1987), *Operations research: Principles and Practice*, John Wiley and Sons, New York.

Rockafellar, R. T. and Wets, R. J.-B. (1991), 'Scenarios and policy aggregation in optimization under uncertainty', *Mathematics of Operations Research* pp. 119–147.

Rose, J. S. (1984), 'Optimal sequential selection based on relative ranks with renewable call options', *J. Am. Stat. Assoc.* **78**, 430–435.

Ross, S. M. (1983), *Introduction to Stochastic Dynamic Programming*, Academic Press, New York.

Ross, S. M. (1996), *Stochastic Processes, 2nd edition*, John Wiley & Sons, New York.

Sandblom, C.-L., Pederzoli, G. and Eiselt, H. A. (To appear – Never did), *Operations Research Theory - Techniques - Applications: Discrete Optimization models*, Vol. II, deGruyter, Berlin.

Smith, D. K. (1991), *Dynamic Programming, A Practical Introduction*, Ellis Horwood, New York.

Smith, M. H. (1975), 'A secretary problem with uncertain employment', *J. Appl. Prob.* **12**, 620–624.

Tamaki, M. (1991), 'A secretary problem with uncertain employment and best choice of available candidates', *Operations Research* **39**, 274–284.

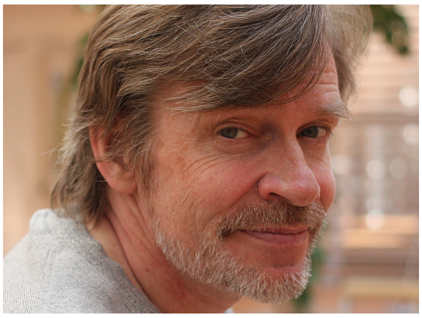
Watson, S. R. and Buede, D. M. (1987), *Decision Synthesis: The principles and practice of decision analyses*, Cambridge University Press, Cambridge.

White, C. C. and White, D. J. (1989), 'Markov decision processes', *European Journal of Operations Research* pp. 1–15.

Zenios, S. A. (1989), 'Parallel numerical optimization: Current status and an annotated bibliography', *Orsa Journal of Computing* **1**, 20–43.

Zipkin, P. (1980), 'Bounds on the effect of aggregating variables in linear programs', *Operations Research* pp. 903–916.

Index



Author Biography: Kjetil K. Haugen is 56 years old (born 1959), living in Molde, Norway. He is a professor of Logistics and Sport Management at Molde University College, Specialized University in Logistics. Haugen holds a PhD in Management Science from the Norwegian Institute of Technology from 1991. He achieved a (full) professorship in Logistics in 2005 and (full) professorship in Sport Management in 2011. Lately, his research interests have moved from operations management-/logistics into sports economics/strategy. His main interest has been game theory applied in football, both as a tool to understand football economics as well as the game itself. Professor Haugen has published in Journals such as *Public Choice*, *EJOR*, *Annals of Operations Research*, *Kybernetika*, *INTERFACES*, *Operations Research Perspectives*, *Annals of Regional Science*, *PLoS ONE*, *Journal of Sports Economics*, *European Sport Management Quarterly*, *Sport Management Review* and *Sport in Society* among others.