THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Asset-Liability Management via Stochastic Programming for a Swedish Life Insurance Company

Fredrik Altenstedt

CHALMERS | GÖTEBORG UNIVERSITY

Department of Mathematics
CHALMERS UNIVERSITY OF TECHNOLOGY and GÖTEBORG UNIVERSITY
Göteborg, Sweden 2001

Asset-Liability Management via Stochastic Programming for a Swedish Life Insurance Company
Fredrik Altenstedt

ISSN 0347–2809/NO 2001:42
Department of Mathematics
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg
Sweden
Telephone +46 (0)31–772 1000

Chalmers University of Technology
Göteborg, Sweden 2001

Abstract

In this work we develop an asset liability model for a swedish life insurance company, incorporating Swedish laws and regulations. A method for generating representative scenario trees from a black box model of the worlds economy is developed as well as such a black box economy model. Stochastic programming is employed to find the optimal solution to the asset allocation problem given a scenario tree. Finally the model is tested numerically, and its performance is compared to a benchmark strategy, consisting of finding the best fixed mix for a given scenario tree. We further investigate the effect of arbitrage opportunities in the tree, as well as the sensitivity of the results to adding extra stages. The long term goal of developing and testing this model is to assemble a decision support system to be used by the managers of a Swedish life insurance company.

Keywords: Stochastic programming, Asset liability management.

MSC 2000 subject classifications: 90C15

Contents

Preface

Acknowledgement

First of all, I would like to thank my supervisor Associate professor Michael Patriksson for his help and support during this work. He has always taken the time to discuss my project as well as shown both interest and faith in my work. I would further like to thank LIVIA for supplying me with an interesting problem, as well as providing data, help and financial support.

On a more personal level I would like to thank my friends for filling my life with more than work, I am extra grateful for the times when doing so required mild force. Finally I would like to thank my sister and parents for giving me support and encouragement, during this work as well as always before.

Fredrik Altenstedt
Göteborg, July 2001

Notation

Notation for Chapter 1

Symbols

$g(\cdot)$	Utility function
ρ	Probability
$(\Omega, \mathscr{F}, P)$	Probability space
Ω	Outcome space
$\mathscr{F}$	σ-algebra
$\mathscr{F}_t$	Filtration
P	Probability measure
t	Time stage
$Y(t)$	Stochastic process
ξ	Random variables
ξ_t	Random variables known at time t
$f(\cdot)$	Objective function
x	Decision variables
X	Feasible set
E	Expectation
Q	Optimal objective value of recource problem
$\mathcal{Q}$	Recource function
S	Second stage feasibility set

$\overrightarrow{x}_t$	Collection of decsision variables up to and including stage t $(x_0, x_1, \dots, x_t)$
$A_{t,k}$	Constraint matrix of variables of stage k in constraints of stage t
W	Recource matrix
h	Right hand side
c	Objective coefficients
$p(i,t,k)$	Parent of stage k of outcome i of stage t
$p(i,t)$	Parent of outcome i of stage t
$R(i,t)$	Successors of outcome i at stage t
q	Number of outcomes

Subscripts

t, k	Time stage

Superscripts

i	Outcome

Notation for Chapter 3

Symbols

r	Interest rate
β	Constant in interest rate model
α	Constant in interest rate model
σ	Constant in interest rate model
γ	Constant in interest rate model
r_0	Mean reversion level
$\widehat{r}_b$	Long term mean value of 5 year bond rate
s_s	Long term difference between bond rate and treasury bill rate
ρ	Correlation factor
dz	Standard Wiener process increment
dt	Time-step

$l(\cdot)$	Log-likelyhood function
p	Asset price

Subscripts

b	Bond
s	Treasury bill
i	Asset type
m	Market portfoio
f	Risk free asset

Superscripts

t	Time stage

Notation for Chapter 4

Symbols

α	Constant in mortality model
β	Constant in mortality model
γ	Constant in mortality model
f	Constant in mortality model
$l(\cdot)$	Life function
$D(\cdot)$	Commutation function
$N(\cdot)$	Commutation function
s	Time to start of payments
r	Time to end of payments
o	Payment intensity to customer
p	Payment from customer
V	Prospective reserve
V'	Retrospective reserve
δ	Interest rate intensity
x	Age
c	Interest rate burden

Notation for Chapter 5

Symbols

T	Time horizon
ξ^t	Random variables revealed up to decision stage t
I	Set of asset classes
K	Set of capital cover rules
$I_k \subset I, k \in K$	Subset of asset classes affected by a capital cover type rule
Q	Set of penalties, prospective reserve
L	Set of penalties, consolidation
A	Set of penalties, bonus rate of return
x_i	Amount of asset i held
y_{+i}	Amount of asset i bought
y_{-i}	Amount of asset i sold
$\widehat{x}_i$	Amount of asset i used to cover the prospective reserve
r	Bonus rate of return
z	Penalty violation
V	Retrospective reserve
P	Payments to or from customers
S	Prospective reserve
η	Price development
ρ	Direct return
γ	Transaction cost
c	Maximum prospective cover
s	Penalty
f	Security factor for reserve violation
Δr	Interest rate offset
κ	Consolidation limit
θ	Tax level times period length
d	Discount factor
w	Objective function

Subscripts

i	Asset

k	Capital cover rule
p	Prospectve cover rule, type of assets
q	Prospective cover rule
a	Bonus rate of return rule
k	Maximum consolidation rule
l	Minimum consolidation rule
in	Inflow of funds
out	Outflow of funds
max	Maximum limit
min	Minimum limit
ref	Reference level
tot	Total assets

Superscripts

t	Time

Notation for Chapter 7

Symbols

x	Desicion variables
X	Feasible set
ξ	Random variables
E	Expectation
EV	Optimal value of mean value problem
EEV	Expected value of the mean value solution
EVPI	Expected value of perfect information
REVPI	Relative expected value of perfect information
RP	Optimal value of recource problem
WS	Optimal value of wait-and-see solution
VSS	Value of the stochastic solution

Subscripts

t	Time

Notation for Chapter 8

Symbols

x	Decision variables
ξ	Random variables
c	Objective function coefficient
A	Constraint matrix
B	Basis matrix
b	Right hand side
ϵ	Small disturbance
z	Optimal solution value
π	Dual variable
P	Probability
$N(\cdot,\cdot)$	Normal distribution

Subscripts

t	Time
i	Asset class

Notation for Chapter 9

Symbols

x	Desicion variables
y	Second stage decision variables
A	Constraint matrix
W	Recource matrix
h	Right hand side
ξ	Random variables
Q	Optimal objective value of second stage function
$\mathcal{Q}$	Recource function
S	Second stage feasibility set
D, d	Feasibility cut
E, e	Optimality cut
θ	Vaule of model of recource function

r	Number of feasibility cuts
s	Number of optimality cuts
μ	Dual variables feasibility problem
w	Optimal value of feasibility problem
ρ	Probability
ν	Dual variables
π	Lagrangean multipliers
u, v	Slack variables

Subscripts

l	Cut number

Superscripts

k	Iteration number
i	Outcome

Introduction

In this thesis we develop a stochastic programming asset liability management system to aid in the decision process of a Swedish life insurance company. As far as we know, this has never before been done for Swedish conditions. A short description of this problem is given in Chapter 2 and the full model is developed in Chapter 3 – Chapter 5. In this work, we have cooperated with the Swedish life insurance company LIVIA.

Stochastic programming is a branch of optimization where one tries to explicitly take into account random events which may influence the value of our decisions. We hence try to hedge against unfavourable outcomes of these random events in order to obtain a solution which will have the best performance on average.

1 USES OF STOCHASTIC PROGRAMMING

Problems which has the temporal structure of

$$\text{decision} \rightarrow \text{realisation of unknown entity} \rightarrow \text{recourse decision},$$

and where the outcome of the random event may have a large impact on the solution is where we may benefit from the use of stochastic programming. We thus have an initial decision which must be made with imperfect information of the outcome of some random event. Later this random event becomes known and we may take a corrective action. Stochastic programming problems are not limited to two stages; we may have a ladder structure with several stages such as

$$\begin{aligned} &\text{initial decision} \rightarrow \text{realisation} \\ &\quad \rightarrow \text{recourse decision} \rightarrow \ldots \\ &\rightarrow \text{realisation} \rightarrow \text{recourse decision}, \end{aligned}$$

where each decision is made with increasing knowledge of the outcomes of the random entities.

Capacity expansion problems constitute one area where the structure of alternating decisions and random outcomes naturally arise. In this kind of problem we must decide on how to invest in production capacity before knowing the actual demand for the produced goods. The realisation of the unknown is the demand for what we may produce, and the recourse decision is the actual production plan. Examples of this type include Eppen et. al. [21] who use this method to address the problem of determining a manufacturing strategy for GM. The first stage decision is which plants to close/keep, the unknown parameters are the actual demand for different models of cars, and the recourse action is a production plan stating where to produce the cars in demand in the market. A similar structure is found in [38] where Sen, Doverspike and Cosares use stochastic programming to plan the expansion of a telecom network. Where to install the communication lines is the first stage, the demand for private lines is the unknown factors and these are not revealed until after we build the lines. The recourse decision is how to route the calls. Other examples where stochastic programming has been applied to capacity expansion problems include [9] where the expansion of the chemical processing capabilities of Korea is investigated. A problem similar to capacity expansion is addressed in [12] where Carøe and Schultz try to determine which power plants to commit for future production. In this case the future demand is unknown and the recourse action is which of the committed power plants to actually use.

Asset liability management is a field in which it has become popular to apply stochastic programming. Within this problem framework, we have a stream of future liabilities and payments. We wish to invest our capital in a fashion that will meet these future liabilities with a high reliability and without requiring excessive amounts of capital. An early commercial application of this type is a system implemented by Cariño, Ziemba et.al. [13, 14, 15] for Yasuda-Kasai, a Japanese insurance company. This company may invest in a number of assets, via different legal entities (direct versus investment via subsidiaries). The goal is not only to maximize the total capital; some forms of income is preferred over capital gains, as higher income will allow higher bonuses to the customers.

Stochastic programming has been used for the management of pension funds by a number of authors. Dert [20] has developed an asset liability model for a Dutch pension fund, with chance constraints (see Subsection 1.2.e) regulating the probability of under-funding, and taking into account the participating member's status via Markovchains. The problem addressed is that of a fund manager managing assets for a group of companies, called the sponsors of the fund. The companies makes payments to the fund over the years, and the fund is expected to provide benefits to retiring employees. The benefits are dependent on the employees final and average salary. The job of the manager of the fund is to keep the fund sufficiently solvent while keeping the contributions low and predictable. In [23, 31] Kouwenberg and Gonzio address the same problem using large scale stochastic programming.

A similar problem in a British setting is treated by Consigli and Dempster in [17].

There is a major difference between these problem and the problem we study in this work. When we deal with fund management, the remedy to under funding is to ask the sponsor for more money, while the value of our liabilities is unaffected by our actions. What we try to optimize is the amount spent to honour the liabilities. In the Swedish setting, where we deal with an insurance company, we have no control of the inflow of funds. The contributions may hence be treated as exogenous variables and we try to maximize the yield of the fund in order to maximize the amount paid back to the customers. A work treating a case similar to ours is [26] in which Høyland discusses an asset liability management system for a Norwegian life insurance company. The regulations and hence the constraints are however different in the Swedish and Norwegian cases. The primary difference is that Norwegian laws require the company to have at least a specified yield on the invested assets over a defined period of time.

2 DISPOSITION OF THIS THESIS

2.a CHAPTER 1

A short description of different types of stochastic programming problems, and how they are related is given here. We also introduce scenario-trees, which is needed to describe the structure of the random components of our problem.

2.b CHAPTER 2

In this chapter we give a short description of our specific problem, and how we divide it into separate models, a model of the surrounding economy, a model of the reserves, and a model of the company. These models are further described in chapters 3 – 5.

2.c CHAPTER 3

Here we specify a model describing the surrounding economy. We model the price development of different asset classes as well as the development of interest rates. We further make sure that the interest rates and prices of bonds are consistent.

2.d CHAPTER 4

In order to determine the actions of the company, we must know how the customers behave. In this chapter we describe a simple model of the customers actions, as well as how to compute the reserve requirements forced upon the company by Swedish laws.

2.e CHAPTER 5

In this chapter we describe the model of the company. This model is formulated as a stochastic programming problem. The model includes how our actions determine the value of the assets held as well as enforces constraints given by laws and policy of the company.

2.f CHAPTER 6

In order to test the model numerically, it must actually be implemented. This chapter gives a description of the different parts of this implementation. We describe how the data for simulations is generated, as well as how we use a modified algebraic modelling language to express the model of our company in a short and easily modified way.

2.g CHAPTER 7

An overview of some aspects of the behaviour of the model is given. We see how far from linear the problem is as well as take a look at the matrices describing the problem. We try to measure how dependent the problem is on the random entities given, which give an indication of what may be gained by using stochastic programming.

2.h CHAPTER 8

In order to see if our model may be useful, and to explore some properties of the model, we devise and perform a series of numerical experiments, which are described in this chapter. Among other things we investigate how the size of the tree affects the quality of the solution, as well as try to determine whether arbitrage possibilities may disturb the solution in a negative way.

2.i CHAPTER 9

We give a short description of some different specialized solution techniques designed to improve the solution efficiency of stochastic programs. We further shortly discuss which effect these techniques may have on our problem and which ones are suitable candidate for implementation.

2.j CHAPTER 10

Here we sum up the results of this thesis as well as outline the work to be done in the future.

Chapter 1

Stochastic programming

In this chapter we present a number of different types of stochastic programming problems, and how these are generalisations of deterministic optimization problems. We describe how the time-structure of stochastic problems may be described with or without split variables, and how we may represent the random variables and their interdependence using scenario-trees.

1.1 UTILITY

In simpler cases of deterministic optimization it is clear what we try to optimize. We may wish to find the shortest path from A to B, find the cheapest lunch, and so forth. In order to make an optimal decision we must have a way of comparing the relative value of different decisions. We do this by the use of a real valued objective function. In the lunch case, this may simply be the price of the meal, and we prefer a lower price to a higher. Note that although we get a numerical value of how good a solution is, the only thing we use these values for is to rank the different solutions. It is by no means clear that the difference between objective function values tell us how much we prefer one alternative over another.

When we deal with randomness in our problems, an ordering is not enough. To illustrate this we use a simple coin-tossing game. A player has the safe alternative of receiving 10 SEK regardless of the outcome of the toss, or gamble and receive 30 SEK if heads comes up and nothing if tails comes up.

If we offer a person to play this game, most people would take the toss, as it has a higher expected gain. However, if we raise the bets by a factor 1,000,000 most people would choose the certain money. Still nothing in the problem has really changed the relative values of the outcomes:

$\frac{10}{30} = \frac{10,000,000}{30,000,000}$. Thus when dealing with optimization problems with random outcomes we need something that do accurately measure our preferences. The function we use to do this is a *utility function*, as described by Von Neumann and Morgenstern in [43]. A utility function is not only an ordering of the possible outcomes, it specifies the distance between the outcomes, how much we prefer one outcome over another. By definition a rational person will, when given a choice, choose the alternative with the highest expected utility.

To exemplify this, assume we have a person who may choose to participate in a game of tossing a non-symmetric coin. If the person choose not to play, the outcome is A (in the example above getting 10SEK). If the person chooses to play the outcome will be B (getting 30 SEK) with probability ρ and C (the player gets nothing) with probability $1 - \rho$. We denote the utility of these outcomes with $g(A), g(B)$ and $g(C)$. Clearly we have $g(B) > g(A) > g(C)$

As a rational person maximizes expected utility, she will choose the toss if $g(A) \leq \rho g(B) + (1 - \rho)g(C)$ and not otherwise. By constructing a large number of games like this we may determine a persons utility function. We specify the three events A B and C and let our subject rank them. Assume that the person prefers B to A and A to C and assign arbitrary values to $g(B)$ and $g(C)$ so that $g(B) > g(C)$. We now let the person choose a value $\bar{\rho}$ of the probability p in the game above so that she does not prefer playing to not plying or vice versa and conclude that $g(A) = \bar{\rho}g(B) + (1 - \bar{\rho})g(C)$.

In order for a utility function to exist the preferences must fulfil a number of conditions given in [43], such as guaranteeing that no circular preferences exists. By circular preferences we mean that a person prefers A to B and B to C but still prefers C to A.

1.2 DIFFERENT TYPES OF STOCHASTIC PROGRAMMING PROBLEMS

1.2.a PROBABILITY THEORY

Before formulating any stochastic programming problems we provide a short overview of some necessary notions from probability theory.

A *probability space* consists of three components, a measurable space of outcomes Ω, a σ algebra $\mathscr{F}$ defined on this space and a probability measure $P : \mathscr{F} \rightarrow [0, 1]$. We define multivariate random variables as $\mathscr{F}$-measurable functions from Ω to $\mathbb{R}^n$. For the random variable $\xi(\omega)$ the probability of $\xi(\omega)$ belonging to the Borel set $B \subset \mathbb{R}^n$ will then be given by $P(\xi^{-1}(B))$.

A *discrete stochastic process* is an ordered collection of random variables, indexed over a countable ordered set of discrete times $t \in 0 \dots T$.

A *filtration* $\mathscr{F}_t$ on Ω is a sequence of increasing σ algebras ($\mathscr{F}_{t-1} \subset \mathscr{F}_t$) on Ω. A process $Y(t)$ is said to be *adapted* to a filtration $\mathscr{F}_t$ if $Y(\tau)$ is measurable with respect to $\mathscr{F}_t$ for all $\tau \leq t$. A filtration is said to be generated by a process if it is the smallest filtration to which the process is adapted.

An event is said to hold *almost surely* if it occurs with probability 1. As an example we may take a continuous random variable uniformly distributed on [0,1]. An outcome of such a variable will almost surely not take the value 0.5, although this outcome is still possible.

1.2.b STAGES

Throughout this thesis we assume that we have a finite number of time-stages in our problem. By a stage we mean a point in time where we may make a decision. Different time-stages are separated by the amount of information we have, two different decisions belong to the same stage if they are made with the same information available. We assume that all our problems start at $t = 0$ and that no random information is revealed before the first decision is taken.

From now on we let ξ denote a random vector including all the random variables affecting the problem. If we have random processes affecting the problem, we let ξ_t denote all the outcomes known at time-stage t. Specifically, if we know the outcome of ξ_t we know the outcome of ξ_{t-1}. In order to simplify notation we let the deterministic information available at time $t = 0$ be denoted by ξ_0.

1.2.c GENERAL PROBLEM

A general formulation of a *deterministic* optimization problem is

$$\begin{aligned} &\text{minimize} \quad f(x), && (1.1a)\\ &\text{subject to} \quad x \in X. && (1.1b)\end{aligned}$$

Usually we assume that $X \subset \mathbb{R}^n$ and that f is a continuous function $f : \mathbb{R}^n \to \mathbb{R} \cup +\infty$ on X. If we are to introduce randomness into this problem we must know whether the value of our decision variables are to be allowed to vary with the random outcome or not, i.e, if we know the outcome of the random variable before or after our decision. If we know the outcome of our random variables before the decision is made, and this decision is a direct function of ξ, the decision itself will be measurable w.r.t. the same algebra as ξ. This leaves us with the problem to

$$\begin{aligned} \text{minimize} \quad & f(x(\xi), \xi), && (1.2a)\\ \text{subject to} \quad & x(\xi) \in X(\xi). && (1.2b) \end{aligned}$$

Obviously this problem reduces to an instance of the general problem (1.1) for each outcome of ξ.

1.2.d MEAN PROBLEM

A more interesting problem is the *mean problem*. We now assume that we do not know the values of ξ before we decide on our actions. As such, we must weigh the different outcomes in some fashion, and we do this by minimizing the expected value of the utility function:

$$\begin{aligned} \text{minimize} \quad & \mathsf{E}_\xi[f(x, \xi)],\\ \text{subject to} \quad & x \in X(\xi) \quad \text{almost surely}. \end{aligned}$$

As previously we have $X \subset \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R} \cup +\infty$ on X. Here E_ξ denotes the expectation operator with respect to ξ, $\mathsf{E}_\xi[f(\cdot, \xi)] = \int_\Omega f(\cdot, \xi(\omega))dP(\omega)$. This is obviously an instance of the deterministic problem (1.1).

1.2.e CHANCE CONSTRAINED PROBLEMS

Should we not require the constraints to hold almost surely, but with probability $1 - \alpha$ we need to formulate the *chance constrained problem*,

$$\begin{aligned} \text{minimize} \quad & \mathsf{E}_\xi[f(x, \xi)],\\ \text{subject to} \quad & P(x \in X(\xi)) \geq 1 - \alpha. \end{aligned}$$

These problems have the undesirable property of possibly being non-convex, even if $f(\cdot, \xi)$ and $X(\xi)$ are convex for all ξ. For an example of this see section 3.2 in [6]. Chance constraints may occur wherever we try to limit the probability of something undesirable happening, but where guaranteeing that such an event does not happen almost surely would be prohibitively expensive. For our problem this kind of formulation may be interesting for the reserve requirements, see Chapter 4, as these are impossible to fulfil with probability 1 (there is always a positive probability of a full blown financial disaster resembling the crash during the great depression, combined with hyper-inflation, rendering all our assets worthless).

1.2.f TWO STAGE PROBLEMS WITH RECOURSE

Still more interesting problems arise if we assume that we may take corrective actions once the outcome of the stochastic variables are known, but we still have some decisions which must be made before the random event has happened. These uninformed decisions will hence link across the different outcomes of the stochastic variables. Assuming we divide our decision variables into two vectors, $x_1(\xi)$ dependent on ξ, and x_0 independent of ξ, we can express our problem as

$$\begin{aligned} \text{minimize} \quad & f_0(x_0) + \mathsf{E}_\xi[f_1(x_0, x_1(\xi), \xi)], && (1.3a) \\ \text{subject to} \quad & x_0 \in X_0, && (1.3b) \\ & x_1(\xi) \in X_1(x_0, \xi) \qquad \text{almost surely.} && (1.3c) \end{aligned}$$

This is the *two-stage stochastic problem with recourse*. The *recourse problem* is now obtained as a function of x_0:

$$\begin{aligned} & Q(x_0, \xi) = && (1.4a) \\ & \underset{x_1}{\text{minimize}} \quad f_1(x_0, x_1, \xi)), && (1.4b) \\ & \text{subject to} \quad x_1 \in X_1(x_0, \xi). && (1.4c) \end{aligned}$$

We apply the convention that $Q(x_0, \xi) = \infty$ whenever the problem is infeasible.

In addition we apply the convention that $\infty - \infty = \infty$ in order to resolve the ambiguity when x_0 makes (1.4) unbounded for some ξ and infeasible for others.

If we further define $\mathcal{Q}(x_0) = \mathsf{E}_\xi[Q(x_0, \xi)]$ we may write the two-stage stochastic programming problem as the *deterministic equivalent problem*

$$\begin{aligned} \text{minimize} \quad & f_0(x_0) + \mathcal{Q}(x_0), && (1.5a) \\ \text{subject to} \quad & x_0 \in X_0, && (1.5b) \end{aligned}$$

(note that $\mathcal{Q}(x_0)$ is an implicit function).

We further note that requiring that $\mathcal{Q}(x_0) < \infty$ is equivalent to requiring that

$$x_1 \in X_1(x_0, \xi), \tag{1.6a}$$

should hold almost surely, exactly as previously.

In order to specify when the second stage is feasible, we define the *first stage feasibility set* as $S = \{x_0 | \mathcal{Q}(x_0) < \infty\}$.

Naturally, if we fix the value of x_0 in (1.3) this problem will separate into different subproblems for each outcome of ξ just as in (1.2).

1.2.g MULTISTAGE PROBLEMS WITH RECOURSE

We now let our random variables ξ_t be defined by a number of discrete stochastic processes (of finite horizon T). As mentioned earlier we assume that ξ_0 describes the initial state of the problem, and hence that these values are deterministic. We remember that ξ_t is assumed to contain all information known at time t and hence that $x_t(\xi_0, \ldots, \xi_t) = x_t(\xi_t)$.

We extend the two-stage stochastic problem to the *multi-stage stochastic problem.* To simplify notation, we add the definition $\overrightarrow{x}_t(\xi_t) = (x_0(\xi_0), x_1(\xi_1), \ldots, x_t(\xi_t))$ and we may now write the multistage stochastic problem as

$$\begin{aligned}
\text{minimize} \quad & f_0(x_0(\xi_0), \xi_0) + \mathsf{E}_{\xi_1}[f_1(\overrightarrow{x}_1(\xi_1), \xi_1) + \mathsf{E}_{\xi_2|\xi_1}[f_2(\overrightarrow{x}_2(\xi_2), \xi_2) + \\
& \ldots + \mathsf{E}_{\xi_T|\xi_{T-1}}[f_T(\overrightarrow{x}_T(\xi_T), \xi_T)]]], && \text{(1.7a)} \\
\text{subject to} \quad & x_0(\xi_0) \in X_0(\xi_0), && \text{(1.7b)} \\
& x_t(\xi_t) \in X_t(\overrightarrow{x}_{t-1}(\xi_t), \xi_t), \quad \forall t = 1, \ldots, T. && \text{(1.7c)}
\end{aligned}$$

Here, as always, $x_t(\xi_t)$ will be presumed measurable with respect to $\mathscr{F}_t$, the filtration generated by ξ_t (see Subsection 1.2.a). Measurability is important as it guarantees the *non-anticipativity* of the variables, in other words that $x_t(\xi_t)$ will be independent of any events occurring after time t.

The deterministic equivalent of a multistage stochastic programming problem is recursively defined by first defining $Q_T(\overrightarrow{x}_{T-1}(\xi_{T-1}), \xi_T)$ as

$$\begin{aligned}
& Q_T(\overrightarrow{x}_{T-1}(\xi_{T-1}), \xi_T) = \\
& \qquad \underset{x_T}{\text{minimize}} \quad f_T(\overrightarrow{x}_{T-1}(\xi_{T-1}), x_T(\xi_T), \xi_T)), \\
& \qquad \text{subject to} \quad x_T(\xi_T) \in X_T(\overrightarrow{x}_{T-1}(\xi_{T-1}), \xi_T),
\end{aligned}$$

and $\mathcal{Q}_T(\overrightarrow{x}_{T-1}(\xi_{T-1}), \xi_{T-1}) = \mathsf{E}_{\xi_T|\xi_{T-1}}[Q_T(\overrightarrow{x}_{T-1}(\xi_{T-1}), \xi_T)]$.

Now we define $Q_{T-1}(\overrightarrow{x}_{T-2}(\xi_{T-2}), \xi_{T-1})$ as

$$
\begin{aligned}
&Q_{T-1}(\overrightarrow{x}_{T-2}(\xi_{T-2}), \xi_{T-1}) = \\
&\qquad \underset{x_{T-1}}{\text{minimize}} \quad f_{T-1}(\overrightarrow{x}_{T-2}(\xi_{T-2}), \xi_{T-1}) + \mathcal{Q}_T(\overrightarrow{x}_{T-1}(\xi_{T-1})), \\
&\qquad \text{subject to} \quad x_{T-1} \in X_{T-1}(\overrightarrow{x}_{T-2}(\xi_{T-2}), \xi_{T-1}),
\end{aligned}
$$

and $\mathcal{Q}_{T-1}(\overrightarrow{x}_{T-2}(\xi_{T-2}), \xi_{T-2}) = \mathsf{E}_{\xi_{T-1}|\xi_{T-2}}[Q_{T-1}(\overrightarrow{x}_{T-2}(\xi_{T-2}), \xi_{T-1})]$.

Recursively in the same way we define $\mathcal{Q}_t(\overrightarrow{x}_{t-1}(\xi_{t-1}), \xi_{t-1})$ until we may write the full deterministic equivalent as

$$
\begin{aligned}
&\text{minimize} \quad f_0(x_0(\xi_0), \xi_0) + \mathcal{Q}_1(x_0(\xi_0)) \\
&\text{subject to} \quad x_0 \in X_0(\xi_0).
\end{aligned}
$$

1.3 DISCRETE DISTRIBUTIONS AND SCENARIO TREES

1.3.a STOCHASTIC LINEAR PROGRAMS

In this work we will be concerned with multi-stage stochastic linear programs. If we restrict ourselves to linear programs, the problem (1.7) will reduce to the following form

$$
\begin{aligned}
\text{minimize} \quad & c_0^{\mathrm{T}} x_0 + \mathsf{E}_{\xi_1}[c_1(\xi_1)^{\mathrm{T}} x_1(\xi_1) + \mathsf{E}_{\xi_2|\xi_1}[c_2(\xi_2)^{\mathrm{T}} x_2(\xi_2) + \\
& \ldots + \mathsf{E}_{\xi_T|\xi_{T-1}}[c_T(\xi_T)^{\mathrm{T}} x_T(\xi_T)]]], && (1.11a) \\
\text{subject to} \quad & W_0(\xi_0) x_0(\xi_0) = h_0(\xi_0), && (1.11b) \\
& \sum_{k=0}^{t-1} A_{t,k}(\xi_t) x_k(\xi_k) + W_t(\xi_t) x_t(\xi_t) = h_t(\xi_t), \quad t = 1, \ldots, T, \\
& && (1.11c) \\
& x_t(\xi_t) \geq 0, \quad t = 0, \ldots, T, && (1.11d)
\end{aligned}
$$

where all constraints must hold almost surely. We have $x_t(\xi_t) \in \mathbb{R}^{n_t}$, $W_t(\xi_t) \in \mathbb{R}^{m_t \times n_t}$, $A_{t,k}(\xi_t) \in \mathbb{R}^{m_t \times n_k}$ and $h_t(\xi_t) \in \mathbb{R}^{m_t}$.

1.3.b RELATIVELY COMPLETE, FIXED AND SIMPLE RECOURSE

Many solution methods for solving stochastic programming problems work by fixing the variables of earlier time stages, and then solving the recourse problem, which will decompose as described above. It would be comforting to know that we may not render the later stages infeasible however bad we choose these fixed variables. A problem having this property is said to have *relatively complete recourse*.

If we view (1.7) this property may be expressed as

$$\begin{gathered} X_1(x_0(\xi_0), \xi_1) \neq \emptyset, \quad \forall x_0(\xi_0) \in X_0(\xi_0), \\ x_{t-1}(\xi_{t-1}) \in X_{t-1}(\overrightarrow{x}_{t-2}(\xi_{t-2}), \xi_{t-1}) \Rightarrow X_t(\overrightarrow{x}_{t-1}(\xi_{t-1}), \xi_t) \neq \emptyset, \\ \forall t \in 1, \ldots, T, \end{gathered}$$

which should hold almost surely.

The notion of relatively complete recourse is computationally difficult to use as we must check the feasibility of later stages for all feasible variables of earlier stages. Noting that we may rewrite the constraints for linear problems above as

$$W_0(\xi_0)x_0(\xi_0) = h_0(\xi_0), \tag{1.12}$$

$$W_t(\xi_t)x_t(\xi_t) = h_t(\xi_t) - \sum_{k=0}^{t-1} A_{t,k}(\xi_t)x_k(\xi_i), \quad t = 1, \ldots, T, \tag{1.13}$$

$$x_t(\xi_t) \geq 0, \quad t = 0, \ldots, T. \tag{1.14}$$

We realize that it is sufficient that $\text{pos}(W_t(\xi_t)) = R^{m_t}$ holds in order to guarantee the existence of a feasible $x_t(\xi_t)$. If all our Ws have this property we say that the problem has *complete recourse*.

If we have $W_t(\xi_t) = W_t, \forall t = 0, \ldots, T$ independent of ξ we say that the problem has *fixed recourse*. This may be an advantage in decomposition schemes, since all last-stage problems will have the same constraint matrix.

1.3.c DISCRETE DISTRIBUTIONS

If we assume that the number of outcomes of our processes is finite, we may enumerate all the outcomes and simplify our problem further.

As the number of outcomes is finite, the number of outcomes of a given time-stage is also finite. If we assume that we have q_t possible outcomes at stage t we may enumerate these as $\xi_t^i, i = 1, \ldots, q_t$. We get the probabilities as

$$\rho_t^i := P(\xi_t = \xi_t^i) > 0.$$

As a specific outcome at time t includes everything known at that time, this outcome will be predescented by a specific sequence of outcomes of $\xi_0, \ldots, \xi_{t-1}$. This sequence is called the *ancestors* of our outcome, and we call the closest one the *parent outcome*. We denote the index of the parent outcome as $p(i,t)$. We may iterate the parent operator, letting $p^2(i,t)$ denote $p(p(i,t), t-1)$. For simplicity we also define $p(i,t,k) = p^{t-k}(i,t)$, the ancestor of the outcome $\{t,i\}$ at time-stage k.

In the same fashion as the predecessors we may define the set of successors of an outcome as $R(i,t), t < T$, given by

$$R(i,t) = \left\{ j : P(\xi_{t+1} = \xi_{t+1}^j | \xi_t = \xi_t^i) > 0 \right\},$$

or, equivalently,

$$R(i,t) = \{ j : p(j,t+1) = i \}.$$

We collect all our outcomes into a *scenario-tree*, which may be seen in Figure 1.1

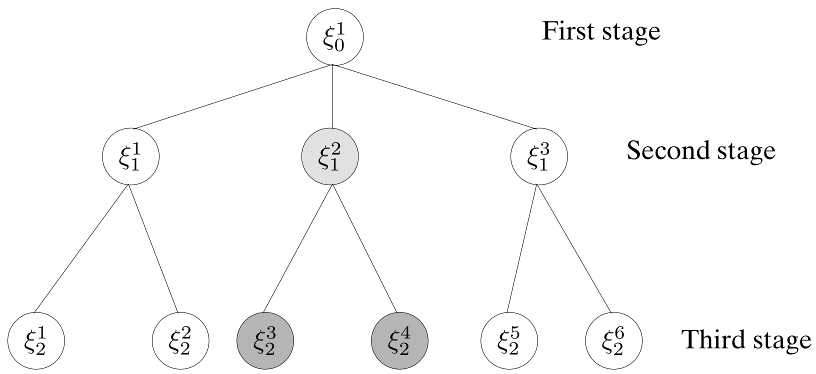

Figure 1.1: An example of a scenario tree.

In this tree the outcomes $\xi_T^i, i = 1, \ldots, q_T$, are leaves as they have no descendants, and the outcome ξ_0 is the root, as it has no ancestors.

If we consider the outcome ξ_1^2 in Figure 1.1 we have $p(2,1) = 1$ and $R(2,1) = \{3,4\}$.

1.3.d EXTENSIVE FORM WITH DISCRETE DISTRIBUTION

As we have a discrete number of outcomes, we may enumerate the components of the multistage stochastic linear problem. We may write c_t^i for $c_t(\xi_t^i)$, $A_{t,k}^i$ for $A_{t,i}(\xi_t^i)$, etc. The multistage stochastic linear program may then be expressed as

$$\text{minimize} \quad \sum_{t=0}^{T} \sum_{i=1}^{q_t} \rho_t^i (c_t^i)^{\mathrm{T}} x_t^1, \tag{1.15a}$$

$$\text{subject to} \quad W_0^i x_0^1 = h_0^1 \tag{1.15b}$$

$$\sum_{k=0}^{t-1} A_{t,k}^i x_k^{p(i,t,k)} + W_t^i x_t^i = h_t^i, \quad t = 1, \ldots, T, \quad i = 1, \ldots, q_t, \tag{1.15c}$$

$$x_t^i \geq 0, \quad t = 0, \ldots, T, \quad i = 1, \ldots, q_t. \tag{1.15d}$$

This is the *extensive form* of the stochastic linear program. With the definitions from Figure 1.2 the extensive form will simply be

$$\begin{array}{ll} \text{minimize} & \widehat{c}^{\mathrm{T}} \widehat{x}, \\ \text{subject to} & \widehat{A}\widehat{x} = \widehat{h}, \\ & \widehat{x} \geq 0. \end{array}$$

This is an ordinary linear program, and may be solved as such. More efficient methods do however exist, of which some are described in Chapter 9.

If we order our sub-matrices according to ascending time, the constraint matrix will be lower block triangular, and will have the shape described in Figure 1.2.

1.4 SPLIT VARIABLE FORMULATION

If we take the problem (1.3) but assume that we have a discrete distribution with a finite number of outcomes, we may generate the extensive form in the same manner as in the previous section. We may enumerate the outcomes of ξ as in Subsection 1.3.c. We do hence assume that we have n outcomes with the probabilities $\rho^i := P(\xi = \xi^i), i = 1, \ldots, n$.

$$\widehat{x}^{\mathrm{T}} = [\overbrace{x_1^{1\mathrm{T}}}^{\text{Stg. 1.}}, \overbrace{x_2^{1\mathrm{T}}, x_2^{2\mathrm{T}}, x_2^{3\mathrm{T}}}^{\text{Stg. 2.}}, \overbrace{x_3^{1\mathrm{T}}, x_3^{2\mathrm{T}}, x_3^{3\mathrm{T}}, x_3^{4\mathrm{T}}, x_3^{5\mathrm{T}}, x_3^{6\mathrm{T}}}^{\text{Stg. 3.}}]$$

$$\widehat{c}^{\mathrm{T}} = [c_1^{1\mathrm{T}}, c_2^{1\mathrm{T}}, c_2^{2\mathrm{T}}, c_2^{3\mathrm{T}}, c_3^{1\mathrm{T}}, c_3^{2\mathrm{T}}, c_3^{3\mathrm{T}}, c_3^{4\mathrm{T}}, c_3^{5\mathrm{T}}, c_3^{6\mathrm{T}}]$$

$$\widehat{A} = \begin{bmatrix}
W_1^1 & & & & & & & & & \\
A_{2,1}^1 & W_2^1 & & & & & & & & \\
A_{2,1}^2 & & W_2^2 & & & & & & & \\
A_{2,1}^3 & & & W_2^3 & & & & & & \\
A_{3,1}^1 & A_{3,2}^1 & & & W_3^1 & & & & & \\
A_{3,1}^2 & A_{3,2}^2 & & & & W_3^2 & & & & \\
A_{3,1}^3 & & A_{3,2}^3 & & & & W_3^3 & & & \\
A_{3,1}^4 & & A_{3,2}^4 & & & & & W_3^4 & & \\
A_{3,1}^5 & & & A_{3,2}^5 & & & & & W_3^5 & \\
A_{3,1}^6 & & & A_{3,2}^6 & & & & & & W_3^6
\end{bmatrix}
\qquad
\widehat{h} = \begin{bmatrix} h_1^1 \\ h_2^1 \\ h_2^2 \\ h_2^3 \\ h_3^1 \\ h_3^2 \\ h_3^3 \\ h_3^4 \\ h_3^5 \\ h_3^6 \end{bmatrix}$$

Figure 1.2: Extensive form corresponding to Figure 1.1

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} \rho^i f(x_0, x_1^i, \xi^i), \\
\text{subject to} \quad & x_0 \in X_0, \\
& x_1^i \in X_1(x_0, \xi^i), i = 1, \dots, n.
\end{aligned}$$

We guarantee that x_0 is independent of ξ by simply having one single copy of x_0, hence x_0 is constant regardless of the outcome of ξ, and we do implicitly enforce the non-anticipativity. We may however choose to have several copies of x_0 and explicitly handle the non-anticipativity via constraints. If we split x_0 into $x_0^i, i = 1, \dots, n$, we may write the split variable formulation of the extensive form of the two-stage stochastic programming problem as

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} \rho^i f(x_0^i, x_1^i, \xi^i), && (1.16a) \\
\text{subject to} \quad & x_0^i \in X_0, \quad i = 1, \dots, n, && (1.16b) \\
& x_1^i \in X_1(x_0, \xi^i), \quad i = 1, \dots, n, && (1.16c) \\
& x_0^i - x_0^{i+1} = 0, \quad i = 1, \dots, n-1. && (1.16d)
\end{aligned}$$

The extra constraints (1.16d) will now guarantee the non-anticipativity of our problem, as all first stage variables are forced to take the same value by the constraint (1.16d).

This technique may be extended in a straight-forward manner to multi-stage problems, with a group of non-anticipativity constraints for each time-stage (except the last stage).

Split variable formulations of different kinds lead to a larger number of constraints and variables. However, some solution techniques works via relaxing the non-anticipativity constraints, which is why we have included this formulation here. For examples of this, see Chapter 9.

Chapter 2

Problem background and model partitioning

2.1 BACKGROUND

The term life insurance may be confusing as it tends to be associated with insurance against premature deaths. Most life insurances are however taken by people saving up for their retirement. As life insurance deals with large amounts of money to be managed for long periods of time, and as this money is most important for the future well-being of the customers, this line of business tend to be heavily regulated. Overseeing the insurance business (as well as other aspects of financial businesses in Sweden) is the authority *Finansinspektionen*, which we from now on will refer to as FI. Life insurance is regulated by law, most notably the law Försäkringsrörelselag 1982:713 [1] which includes requirements on how a company may invest the customer's money. These requirements are described in Section 4.2. This law has recently had a major overhaul. Before January 1 2000 a life insurance company was not allowed to pay dividends to the shareholders; all excess profits should be distributed among the customers. This is no longer true, but as our partner company does not currently plan to change their status from a mutual company, we assume that this restriction is still valid.

A natural question is who would like to own a company when you may not be paid dividends? In the case of our partner company it was founded by Nordbanken (now a part of Nordea). By owning a life insurance company Nordea is able to provide their customers with a wider range of services, and they may hence attract customers to their other branches of operations. In addition, LIVIA buys services regarding management of their assets from Nordea, and Nordea receives commissions for the policies they sell.

We mentioned earlier that as no dividends are paid, all the excess gains should be distributed among the customers. This is done via the *bonus rate of return* which is one of the major weapons in the competition between the different life insurance companies. The bonus rate of return and the requirements associated with it are discussed in Section 4.2.

There exist a number of different contracts which a customer may sign with a company. The basic savings related life insurance consists of a contract where the customer pays a monthly fee to the company until he/she turns 65. After this date the company will pay a monthly sum to the customer until he/she reaches the age of 70. Should the customer die before receiving all or some of his payments, these funds stay with the company and are distributed to the other customers. The customer is not bound to a specific pay-out scheme by the contract. He/she may choose to start the pay-outs as early as at the age of 55 and may choose a longer pay-out period if he/she so prefers. Naturally the payments will be adjusted to deal with a different pay-out scheme. The payments are adjusted to pay out the entire retrospective reserve when the contract expires (see Section 4.2 for an explanation of the retrospective reserve).

A customer may choose to have one or several beneficiaries of a contract. In the event of the insured dying they will be paid the current value of the contract, distributed over a number of years. A contract with this provision will naturally yield a lower return on the same investment, as the customers do not have the benefit of inheriting each other. Currently most customers have this option.

On the market there also exists a different kind of insurance, *capital insurance*. These contracts function as tax-shelters for the customers. They may be compared to a box into which the customers pay their money. Once inside the box, the customer has a choice of funds into which they may invest their money. Capital gains are not taxed, instead the current value of the box is taxed regardless of losses or gains. As LIVIA does not currently have this kind of contracts they are ignored in our study. In addition they are uninteresting from an asset liability modelling point of view as the customer makes all decisions and absorbs all the risk.

2.2 MODEL PARTS

In the following sections we will describe the parts needed to optimize the operation of a life insurance company. We will divide this description into three parts which are:

- The economy in which the company operates,

- The customer's actions, their level of investment in life insurance and death intensities,
- The actions of the company itself.

The surrounding economy model described in Chapter 3, models exogenous variables, that is, variables which are not affected by our actions. These include part of the asset prices, interest rates, inflation and so forth. As our partner company is a rather small life insurance company by Swedish standards, we feel that it is justified to assume that our actions will not affect the surrounding economy, and hence that our company will operate as a price-taker.

The customer model, given in Chapter 4, gives us the two reserves we need for our calculations. The prospective reserve is influenced by the customer's behaviour and external interest rates. The retrospective reserve is influenced by the customer's behaviour and the bonus rate of return set by the company. The customers are assumed to behave independently of both the the surrounding economy and the company's decisions.

Finally, the company model, give in Chapter 5, will model the company actions, which are influenced both by data from the surrounding economy and from the customer model.

Thus, information will flow from the model of the surrounding economy into the model of the company and into the model of the customer's behaviour, but not in the opposite direction. The customer model and the company model will both be dependent on each other.

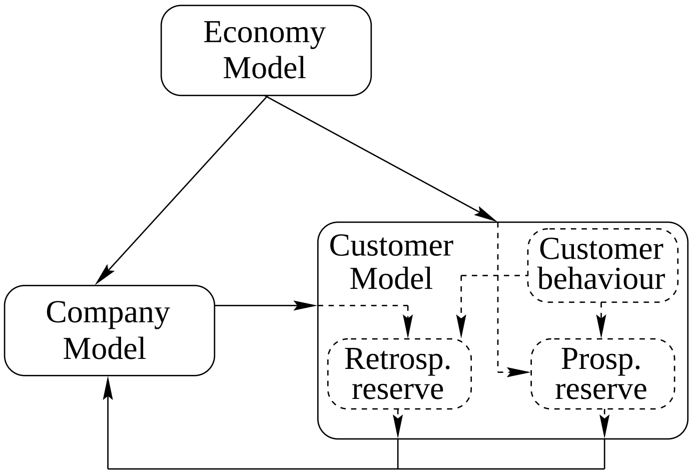

Figure 2.1: Information flow between sub-models.

Chapter 3

Model of the surrounding economy

The actions of our corporation will naturally be dependent on external influences beyond the control of its board. The main factors are interest rates, governing the reserve levels, and the price development of assets in which the company invests. Currently, we have assumed that the customer's behaviour are not influenced by the surrounding economy. The reserve levels are however still influenced via the different interest rates.

In order to be able to optimize the actions of the company we must know the company's beliefs of the future, given as a probability density for different outcomes of the uncertainties in the surrounding economy. We may for instance state that the company assumes that the yield on their stock-portfolio over the next 6 months is accurately described by a log-normal distribution of specified mean and variance. Naturally, these expectations may change over time and we would hence like to be able to switch the parameters of the model in a simple fashion. As the current model currently is rather crude, we would also like to be able to exchange the whole model in a simple fashion, if we decide that such a switch is necessary. As an example, we may decide that a log-normal distribution is not correct, and that we need to replace it with another distribution.

3.1 ASSET CLASSES

In order to make the model computationally tractable we must limit the number of assets in which we choose to invest our money. Currently this division is made into 5 classes; Swedish and foreign bonds, Swedish and foreign stock, and Swedish treasury bills. In addition to these classes LIVIA currently invests in Swedish real interest bonds and real estate. As the current policy of LIVIA is to not trade the two latter assets actively, these assets are excluded from our model. In order to make the assets at the start of our simulation match the given conditions at that time, we increase the holdings of other assets proportionally so the company has the same initial wealth, but distributed among the asset classes included in the model.

3.2 INTEREST RATES

The most fundamental set of external variables of our model are the interest rates. They influence both the reserve levels the company is required to hold, and the yield of bonds, in which a substantial part of the assets will be invested. In doing so they provide the primary link between the asset and liability side of our model. The interest rates we need are Swedish bonds and treasury bills, and the base rate given by FI (see further Section 4.2). In order not to bias our solution towards scenarios with high inflation and high nominal interest rates, we need to discount our model using the rate of inflation, which we hence need our model to produce. In the literature we encounter a vast number of interest rate models, having different tradeoffs between computational or analytical simplicity, and realism.

3.2.a ONE-FACTOR MODELS

A popular class of models for academic research are the one-factor models. As is apparent from the name these models use one explanatory factor to give the entire yield curve. Usually this factor is the instantaneous rate of return. Among these models we find models such as the Merton model, the Vasicheck model, the Cox-Ingersoll-Ross model and the Dothan model. These models all have the common form

$$dr = (\beta + \alpha r)dt + \sigma r^{\gamma} dz.$$

In other words they are stochastic processes driven by Wiener-processes (dz). All the constants ($\beta, \alpha, \sigma, \gamma$) are parameters specifying the model. If we assume $\alpha \neq 0$ we may write $r_0 = \beta/\alpha$ and get

$$dr = \alpha(r_0 - r)dt + \sigma r^{\gamma} dz$$

In this formulation we have a term driving the interest rate towards a long term value r_0. This will make the interest revert to a long term mean value, a property observable in real data. This also agrees with the idea that high economic activity causes a high demand of capital, driving the interest rates

Model	β	α	σ	γ
Merton [25]	free	0	free	0
Vasicek [30]	free	free	free	0
Cox-Ingersoll-Ross [18, 19]	free	free	free	$\frac{1}{2}$
Dothan	0	0	free	1
Brennan-Schwartz [10]	free	free	free	1

Table 3.1: Different one factor interest rate models.

up. High interest rates have a cooling effect on the economy driving the demand for capital, and thus the interest rates, down.

A number of different models have been studied in [16]. Which of the constants are allowed to vary will give us the exact model, as may be seen in Table 3.1.

Some of these models have undesirable long term properties, such as a positive probability for negative interest rates, which is not consistent with observations, and in others the expected value of the interest rate tends to infinity as the time tends to infinity. However, as we will use a limited time horizon with a reasonable starting value, bad long term behaviour alone is not enough to disqualify a model.

These models will generally allow us to obtain closed form expressions for the yield curve as a function of the interest rates, and a closed form expression for the probability density of $r(t + \Delta t)$ given $r(t)$. As any bond may be constructed as a linear combination of zero coupon bonds, we may price any bond given the yield curve.

3.2.b MULTI FACTOR MODELS

As one factor models have difficulties in correctly modelling the yield curve dynamics, focus has shifted to more complex models such as two factor models, multi factor models and infinite dimensional models. They more accurately describe the interest rate dynamics at the price of greater complexity. It is usually no longer possible to derive a closed form expression for the yield curve, instead we need to solve a partial differential equation to give us the yield curve. An example of a infinitely dimensional model is the Ho and Lee [25] model which describes the short rate dynamics as the Merton model above, but with a time dependent β.

3.2.c OUR MODEL

We have settled for a model similar to [11] as the one factor models tested (the Cox-Ingersoll-Ross and Brennan-Schwartz one factor model) give us a stronger correlation between short and long interest rates than what is evident from real world data. We will use interest rates of 6 month treasury bills for the short rate and 5 year government bonds for the long rate, as this is the data we have. We define the equations driving our interest rates as

$$
\begin{aligned}
dr_b =& \alpha_b(\widehat{r}_b - r_b)dt + \sigma_b r_b dz_b, \\
dr_s =& \alpha_s(r_b - s_s - r_s)dt + \sigma_s r_s dz_s,
\end{aligned}
$$

where

r_b	interest rate of 5 year government bonds,
r_s	interest rate of 6 month government treasury bills,
$\widehat{r}_b$	long term mean value of 5 year bond rate,
s_s	long term difference between bond rate and treasury bill rate,
α_b	mean reversion strength of bond interest rate,
α_s	mean reversion strength of treasury bill interest rate,
σ_b	variance factor for bond rate,
σ_s	variance factor for treasury bill rate,
ρ	correlation factor,
dz_b	driving standard Wiener process of bond interest rate,
dz_s	driving standard Wiener process of treasury bill interest rate,
dt	time-step.

As the Wiener-processes are correlated, we have

$$\mathsf{E}(dz_b dz_s) = \rho dt.$$

3.2.d PRICING BONDS

In order to price our bonds, we would like to solve the bond pricing equation, giving us an arbitrage free yield curve, which in turn would give us a price of any bond which is consistent with the chosen interest rate model. For further information see Appendix A. As this is computationally difficult, and not the main focus of this study, we make the same simplification as in [26] and assume that the yield on a 5-month treasury bill equals the yield of a 6-month treasury bill, and that the yield on a 5-year bond equals that of a bond with one month shorter maturity.

Assuming we deal only in zero-coupon bonds, we may then obtain the yield over a month as

$$y = \exp((t_1 \ln(1 + r_1) - (t_1 - \frac{1}{12}) \ln(1 + r_2)),$$

where t_1 is the maturity before the month, expressed in years, and r_1, r_2 are the interest rates before and after the month.

3.2.e IMPLEMENTATION OF THE INTEREST RATE MODEL

We have five years of data for both interest rates, obtained from the Swedish treasury. The interest rate for a five year bond is however not for a zero coupon bond, but we approximate it using the available rates, as the ambition primarily is to capture the general behaviour of the interest rates. If we discretize our time series, assuming a time step of one month, we get with M months of data

$$\begin{aligned} r_b^{t+1} - r_b^t &= \alpha_b(\widehat{r}_b - r_b^t) + \sigma_b r_b^t z_b^t, \quad t = 1, \dots, M-1, \\ r_s^{t+1} - r_s^t &= \alpha_s(r_b^t - s_s - r_s^t) + \sigma_s r_s^t z_s^t, \quad t = 1, \dots, M-1. \end{aligned}$$

We start out by estimating the process for the bond rates as it is independent of the treasury bill rates. From the above formula we get that r_b^{t+1} has the distribution $N(\alpha_b(\widehat{r}_b - r_b^t) + r_b^t, (\sigma_b r_b^t)^2)$. An individual sample r_b^{t+1} hence has the frequency function

$$\frac{1}{\sigma_b r_b^t \sqrt{2\pi}} \exp\left(-\frac{(r_b^{t+1} - (\alpha_b(\widehat{r}_b - r_b^t) + r_b^t))^2}{2(\sigma_b r_b^t)^2}\right).$$

If we sum the negative log likely-hood function for M samples we get

$$l(\sigma_b, \alpha_b, \widehat{r}_b) = \sum_{t=1}^{M-1} \log(\sigma_b r_b^t \sqrt{2\pi}) + \frac{(r_b^{t+1} - (\alpha_b(\widehat{r}_b - r_b^t) + r_b^t))^2}{2(\sigma_b r_b^t)^2}.$$

We may disregard constant terms and obtain the following function which we wish to minimize:

$$\widehat{l}(\sigma_b, s_b, \alpha_b) = (M-1)\log(\sigma_b) + \frac{1}{2\sigma_b^2} \sum_{t=1}^{M-1} \left(\frac{r_b^{t+1} - (\alpha_b(\widehat{r}_b - r_b^t) + r_b^t)}{r_b^t} \right)^2.$$

In order to find the constants for the treasury bill interest rate we do a similar simplification, giving us the following function to minimize:

$$\widehat{l}(\sigma_s, \alpha_s, s_s) = (M-1)\log(\sigma_s) + \frac{1}{2\sigma_s^2} \sum_{t=1}^{M-1} \left(\frac{r_s^{t+1} - (\alpha_s(r_b^t - s_s - r_s^t) + r_b^t)}{r_s^t} \right)^2.$$

Samples from the driving process are obtained as

$$\frac{r_b^{t+1} - r_b^t - \alpha_b(\widehat{r}_b - r_b^t)}{\sigma_b r_b^t} = z_b^t,$$

$$\frac{r_s^{t+1} - r_s^t - \alpha_s(r_b^t - s_s - r_s^t)}{\sigma_s r_s^t} = z_s^t.$$

These samples are checked for their correlation, in order to be able to create standard Wiener processes with the correct correlation when we use these parameters for simulation.

3.2.f EXPECTED VALUES

For future reference we will need the expected values of our discretized processes. We start by noting that due to the fact that our process is Markovian we have that

$$
\begin{aligned}
&\mathsf{E}[r_b^{t+n}|r_b^t] = \\
&\mathsf{E}[\mathsf{E}[r_b^{t+n}|r_b^{t+n-1}]|r_b^t] = \\
&\mathsf{E}[\mathsf{E}[\alpha_b\widehat{r}_b + (1-\alpha_b)r_b^{t+n-1} + \sigma_b r_b^{t+n-1} dz_b|r_b^{t+n-1}]|r_b^t] = \\
&\mathsf{E}[\alpha_b\widehat{r}_b + (1-\alpha_b)r_b^{t+n-1}|r_b^t] = \\
&\alpha_b\widehat{r}_b + (1-\alpha_b)\mathsf{E}[r_b^{t+n-1}|r_b^t].
\end{aligned}
$$

We may hence compute $\mathsf{E}[r_b^{t+n}|r_b^t]$ recursively.

In the same fashion we may compute the expected value of r_s and we get

$$
\begin{aligned}
&\mathsf{E}[r_s^{t+n}|r_s^t, r_b^t] = \\
&\mathsf{E}[\mathsf{E}[r_s^{t+n}|r_b^{t+n-1}, r_s^{t+n-1}]|r_b^t, r_s^t] = \\
&\mathsf{E}[\mathsf{E}[\alpha_s(r_b^{t+n-1} - s_s) + (1-\alpha_s)r_s^{t+n-1} + \\
&\sigma_s r_s^{t+n-1} dz_s|r_s^{t+n-1}, r_b^{t+n-1}]|r_b^t, r_s^t] = \\
&\mathsf{E}[\alpha_s(r_b^{t+n-1} - s_s) + (1-\alpha_s)r_s^{t+n-1}|r_s^t, r_b^t] = \\
&\alpha_s(\mathsf{E}[r_b^{t+n-1}|r_b^t] - s_s) + (1-\alpha_s)\mathsf{E}[r_s^{t+n-1}|r_s^t, r_b^t],
\end{aligned}
$$

which may be used to compute $\mathsf{E}[r_s^{t+n}|r_s^t, r_b^t]$ recursively.

3.2.g OTHER INTEREST RATES AND INFLATION

In addition to the above mentioned bond-rates, our model should provide us with the base rate given by FI, the rate of inflation, and a benchmark interest rate called the *state borrowing rate*, which is defined as the average yield on all Swedish bonds with a time to maturity exceeding 5 years. Currently the base rate in our model is defined as the bond rate minus 2%, or 2%, whichever is the highest. The inflation is specified as the treasury bill rate minus 1%, and as a proxy for the state interest rate, we take the bond yield of our model.

3.3 OTHER ASSETS

The other asset classes are considered to be jointly log-normal, that is, the price of asset class i is given by the process

$$\frac{dp_i}{p_i} = \alpha_i dt + \sigma dz_i,$$

and the driving Wiener processes are correlated between different assets.

3.3.a IMPLEMENTATION

As mentioned earlier our assets are Swedish bonds, Swedish stocks, foreign bonds, foreign stocks and Swedish treasury bills. In order to fit our model to the data, we have time-series of a number of benchmark funds.

Swedish government bonds are represented by the index ‘‘OM Benchmark Statsobl’’, Swedish Treasury bills by ‘‘OM Benchmark SSV’’, Swedish stock by ‘‘Findatas avkastningsindex’’, foreign stock by ‘‘Morgan Stanley World’’ and foreign bonds by ‘‘J P Morgan Global’’.

All benchmarks are denominated in SEK and all dividends are reinvested continuously into the benchmark portfolio.

The correlation between the assets are calculated by discretizing the data with monthly intervals and then transforming to get samples from a multivariate process which is approximately normal. These samples are used to estimate the covariances of the driving standard Brownian motions, as well as the drift terms α_i.

3.4 TIME SERIES GENERATION

The time series are generated in the following fashion. First, a sequence of interest rates are generated for Swedish treasury bills and bonds. These series are used to generate yields of investments in bonds and treasury bills. The time-series of yields are (slightly incorrectly) assumed to be log-normal, and they are used to generate the driving series. Second, conditionally on these driving series we generate driving series for the other asset classes, and hence prices for these. The reason for not making the connection directly between the interest rates and the other assets is that we have more and better data on bonds than on interest rates, and do hence have a better idea of the correlation between bonds and other assets than between interest-rates and other assets.

3.4.a DRIFT TERMS

The capital asset pricing model (CAP) is derived from Markowitz’ theory for portfolio optimization. In the CAP model we assume that there exists a risk free asset, that all investors are rational and that all investors share a common view of the probabilities for different yields of all investment opportunities. Under these assumptions all investors will divide their investments between a portfolio common to all investors, the market portfolio, and the risk free asset (the fractions invested in each depending of the risk tolerance of the investor). For a description of this model, see [32].

If these assumptions are valid the trading will move the price of an asset i until it is valued so that its expected return $\bar{r}_i$ is given by

$$\bar{r}_i = r_f + \beta_i(r_m - r_f), \tag{3.1}$$

where r_m is the return on the market portfolio, and r_f is the return on the risk free investment. The value of β_i is given by the correlation between our asset and the market portfolio, and will be given by $\frac{\sigma_{mi}}{\sigma_m^2}$, where σ_{mi} is the covariance of asset i and the market portfolio, and σ_m^2 is the variance of the market portfolio.

Earlier in our model we assumed that the variances and covariances of our different asset classes were fixed. If we further assume that the market price of risk (the yield premium we get for investing in the market portfolio) is fixed we get that $r_m - r_f$ is fixed since the properties of r_m do not change. We then get $\bar{r}_i = r_f + c_i$ as the expected yield on individual assets since $\beta_i(r_m - r_f)$ is fixed.

It thus makes sense to make the expected yield of our assets be the expected yield of the safe asset to which we add a risk premium. As we do not have a risk-free asset in our model, we use the return rate on treasury bills as this is the safest investment we have at our disposal.

3.4.b FUTURE EXPECTATIONS

We have taken us some trouble to make the model mimic the past.

It is well worth stressing that this is only part of what we should do. Our main task is to have a model that is consistent with the views of the user. If the past behaviour of the world and the views of the user clashes, the user should prevail. Creating a model that mimics the past is however a good first step towards this goal. It is useful as it gives the users something to relate to in the specification of their expectations. In this work a good realism of the model is not absolutely necessary as we will not be testing against real data. It is however still important that the model exhibits a behaviour similar to that of the more accurate model needed if the system is to be used commercially.

Chapter 4

Customer model

4.1 MORTALITY MODEL

When dealing with the issue of life insurance, the expected lifetime of our customers has a significant impact on our calculations. Our model must hence be supplied with a set of assumptions about these expected lifetimes. In fact, having a clearly stated mortality model is required by FI. The model which LIVIA uses for the death intensity (the fraction of the population of a certain age dying each time interval) for a person of age x is given as

$$\mu(x) = \alpha + \beta \exp(\gamma(x - f)),$$

with α, β and γ being known constants. The constant f is 6 for women and 0 for men, that is, we assume that the death intensity of a woman is the same as that of man six years her junior.

Assuming this death intensity, what is the probability of a person living to be x years. This probability is found as $l(x)$ with definition

$$l(x) := \exp\left(-\int_{u=0}^{x} \mu(u)du\right),$$

and we denote this function the life function. We may now compute the probability of a person living to see the age of $x + s$ given that he/she has reached x years as $\frac{l(x+s)}{l(x)}$.

Since life insurance deals with payments in the event of people reaching certain ages, we wish to compute the expected present value of a payment of 1SEK to be paid if someone x years old reaches the age of $x+s$. Given the discount rate intensity δ, the present value of a certain payment will be $\exp(-\delta s)$. Combined with the death probability the present value of such a payment is $\frac{l(x+s)}{l(x)}\exp(-\delta s)$. Constructing the function $D(x) = l(x)e^{-\delta x}$ will allow us to write this present value as $\frac{D(x+s)}{D(x)}$.

If we instead of a fixed payment are interested in the present value of a stream of payments until the client dies, this may be obtained by simple integration. Assuming we wish to calculate the present value of a stream of 1SEK per year, continuously paid to a client after the age $x+s$ when the client is currently of age x, this may be computed as

$$\int_{x+s}^{\infty} \frac{D(u)}{D(x)} du = \frac{\int_{x+s}^{\infty} D(u)du}{D(x)}.$$

Creating the function $N(x) = \int_x^{\infty} D(u)du$ this may be conveniently written as $\frac{N(x+s)}{D(x)}$.

If the payments are to terminate when the client dies or when the client reaches $x+r$ years, whichever comes first, the present value will be computed as $\frac{N(x+s)-N(x+r)}{D(x)}$ for $s \leq r$.

In the cases where the insured has a pool of beneficiaries, who are to be paid the value of the insurance in case the insured dies, we set the death intensity to a small constant value. This small constant value is used to estimate the probability of the insured dying without any beneficiaries left to inherit them.

We now have the appropriate tools to take a closer look at the reserves required for guaranteeing the solvency of the corporation.

4.2 RESERVES

The life insurance business is governed by a number of laws and regulations. The purpose of these regulations is to lower the risks for the customers, and making sure that the companies do not promise what they can not keep. In line with this idea, we have the base rate. The base rate is the maximum return rate the company may use when discounting the payments promised to the customers when calculating the reserve requirements. It is also the minimum return the customers must be given on their money. This rate is given by Finansinspektionen, and according to their policy it should be influenced by the long term market rates. The authorities have however some freedom in how it sets this rate, it is not mechanically determined from the market rates in the same fashion as the state interest rate. As this rate is usually rather low (approximately the same level as the rate on treasury bills) the company usually has a surplus. This surplus should be divided among the customers, as we have assumed that dividends to the owners still are forbidden. This is done via the *bonus rate of return*. The bonus rate is the return on their money the customers are given in excess of the base rate of return. From now on we will however refer to the bonus rate of return as the total return the customer gets on their money, including the base rate. Each customer thus has what may be described as an ordinary bank account in the company, stating how big their share of the total assets of the company are. This account is increased continuously with the bonus rate of return, and whenever the customer makes a payment to the insurance company. The account is decreased whenever the company makes a payment to the insured. Two different reserves are required to make sure that the company will fulfil their obligations with respect to both the base rate and the bonus rate. These reserves are described below.

4.2.a PROSPECTIVE RESERVE

When a customer has a policy with the company he/she agrees to pay a monthly fee, and in return he/she will receive a stream of payments at a later date. In order to simplify calculations, we assume that this stream back to the consumers is continuous, and not made at discrete times. With respect to reserve calculations, each monthly payment by the customer may be viewed as a separate contract. We denote the payment intensity to the customer by o SEK/year and the payment made by the customer by p SEK, with x, r and s defined as previously.

As noted above, the present value of a stream of payment of size o SEK/year is be given by $o\frac{N(x+s)-N(x+r)}{D(x)}$. If we want the payment to correspond to this value, we have

$$p\frac{D(x)}{N(x+s)-N(x+r)} = o.$$

If we assume that we have I customers, we may add upp the present value of all these contracts to get the total porspective reserve as

$$V = \sum_{i=1}^{I} o_i \frac{N(x_i+s_i)-N(x_i+r_i)}{D(x_i)}.$$

The interest rate $\bar{\delta}$ used in the computation of N and D in Section 4.1 is $\bar{\delta} = \delta - c_{\text{run}} - c_{\text{tax}} - c_{\text{margin}}$ where δ is the base rate and the different constants are burdens for running costs, taxes, and the last constant is introduced to obtain a general safety margin.

If the base rate of return does not change, then the prospective reserve will increase with the rate $\bar{\delta}$ assuming we do not pay out any money. If the base rate increases, the present value of the future payments decreases, and hence the prospective reserve decreases, and vice versa. The total prospective reserve is simply the sum of the reserves for all customers.

4.2.b RETROSPECTIVE RESERVE

As we mentioned earlier each customer has an account in the company, measuring their share of the total assets. This account is increased continuously with the bonus rate of return. The retrospective reserve at time t_1 from one payment of size p at time t_0 hence is

$$V'_{t_1} = p \int_{t_0}^{t_1} \exp(r_b(t))dt,$$

with $r_b(t)$ being the return rate intensity. The total retrospective reserve is given by the sum of all payments for all customers. Worth noting is that the company is not bound by this bonus rate of return; in case of insolvency, the company may choose not to honour these allocations, and retroactively lower the bonus rate. This is however very rare, and would seriously disturb the relation to the customers.

4.2.c COMMISSION

LIVIA does not sell their own insurances; the contracts are sold by agents. These agents should get paid; currently they receive a lump sum per contract sold (different for different contracts), plus a fraction of each payment made. A part of this commission is covered directly by the customer via a fee paid with each payment. As only a part is covered, the amount used to calculate increases in the reserve differs from the amounts actually received from the customers, something we must and do take into account by using different payment sizes to estimate the increase in reserve levels and the money flowing into the company.

4.3 IMPLEMENTATION

We will run the model using data of LIVIA's stock of customers from November 1999 (the starting date for all our simulations). We assume that all our customers have a pool of beneficiaries. We further assume that all customers born the same year choose identical pay-out schemes. All of these assumptions may be changed easily; they are ad hoc assumptions made to avoid the work of estimating customer behaviour. In addition, we assume that we recruit new customers at a rate according to the simple function in Figure 4.1 below, and that all customers make a monthly payment of 500SEK. The number of new customers is a very crude approximation of historical data, but having such a function makes it easy to refine the model later. In order to have a more realistic model, the number of new customers should be influenced by the bonus rate of return, something we have avoided modelling. To construct such a model may prove to be hard, especially as the customers make choices between different life insurance companies. We would hence have to model the behaviour of the competition in order to be able to model the customers behaviour. A simple model backed by experience of the modeller may however still give a better result than the current approach.

In order to get an example of how these reserves may deveolp we use the above assumptions of customer behaviour. We combine this with the assumption that all customers choose a 5 year payment period starting at age 65.

We further assume that the base rate of return is constant at 3%, and that the bonus rate of return is constant at 7% Using these assumprions we get the development of our reserves given in Figure 4.2 over a 5 year period.

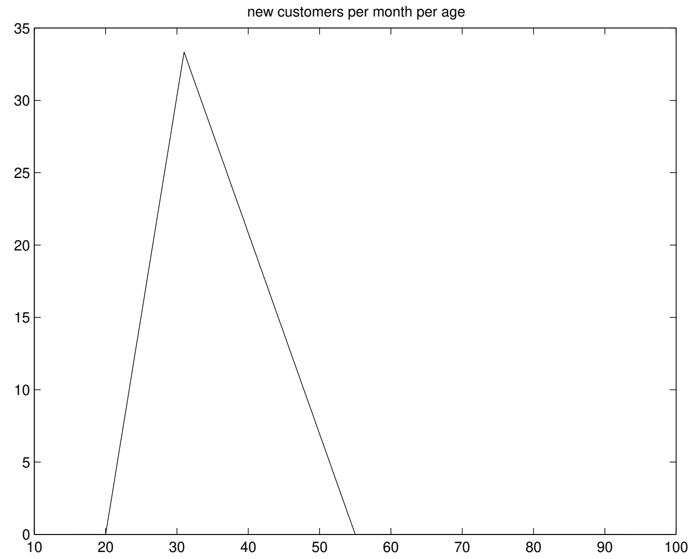

Figure 4.1: New customers per month per age-bracket.

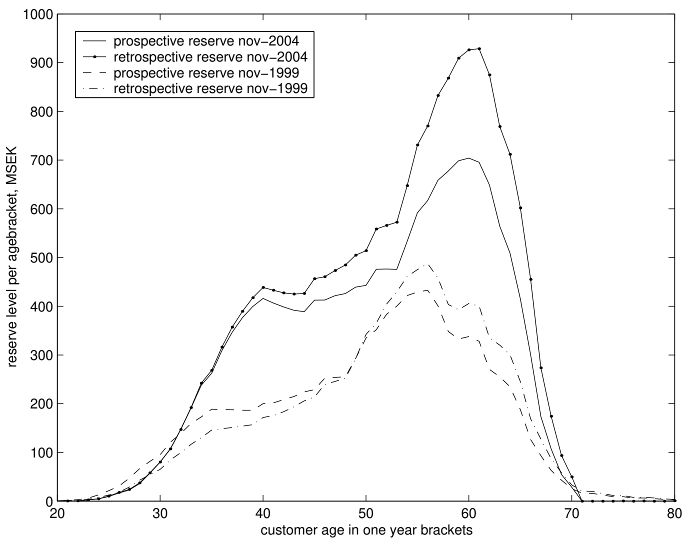

Figure 4.2: Reserves development over 5 years

Chapter 5

Company model

The central part of our optimization model is the company itself and its actions. We model the company's actions as a multi-stage stochastic program. In order to guide us regarding which decision to take we need to express the utility of a state of the company. In addition to expressing the utility of the decisions, we need to make sure that we do not break any regulatory constraints. Further we need to guarantee that the model is correct in so far that we correctly link the assets over different time-stages, not creating or destroying any assets in doing so.

5.1 THE GOAL OF THE OPTIMIZATION

As the owners of the company are not allowed to receive dividends, a reasonable view is that the owners gain secondary benefits from having customers. Hence we want large volumes of business in order to increase these secondary benefits, and hence we want to make the customers as happy as possible. Since the company is not allowed to pay dividends, it is reasonable to view all assets in the company as the customers money. We assume that more money makes the customers happier and thus the goal of our optimization is to maximize the average value of the assets controlled by the company. To the total assets we must add the money paid to the customers according to the insurance policies. Since we should estimate the real value the customers get, we should discount all payments using the rate of inflation. As we have assumed that the customers behaviour will not depend on our actions, we will not have to normalize with respect to the number of customers, as the number of customers and the amounts they have paid will be identical across different scenarios. If this was not true, we would have to weigh the happiness or monetary gain of each customer versus having a large number of customers, something which would introduce additional complexity into our model.

In addition to trying to maximise the total value of the assets, there are a number of unfavourable events which we wish to avoid. In order to be able to choose between increasing the expected return on our assets and avoiding unfavourable events, we will penalize unwanted outcomes.

Thus the utility function we will try to maximize will be the discounted sum of assets retained in the company to which we add the discounted payments made to the customers and we reduce this sum with penalties for unwanted events. We do not explicitly try to lower the variance of the return by using a concave utility function as this would make the problem nonlinear. We do however feel that the effects of adding this kind of risk aversion will be small compared to the effect of the penalties.

5.2 UNWANTED EVENTS

5.2.a PROSPECTIVE RESERVE

From the customer model in Section 4.2 we recall the definition of the prospective reserve. According to Swedish law [1] we must at all times own assets exceeding the prospective reserve. A failure to do so may result in the company being liquidated, and we hence have a strong incentive to avoid breaking this requirement. However, it is in a non-deterministic world not possible to force the probability of never breaking this rule to zero, regardless of what we are ready to sacrifice. We hence assign a large penalty to breaking this requirement, but do not explicitly forbid total asset levels below the prospective reserve.

As the requirement to cover the prospective reserve is the requirement which will cause the gravest consequences if broken, we add penalties for almost not meeting this condition. We add extra requirements to keep the total assets above a security factor times the prospective reserve (for instance 105% and 115%). Naturally, we have lower penalties for higher security factors. We have also added increasing penalties for increasing violations of this rule (security factors of less than 100%), as failing to do so leads to a rather eccentric behaviour of our solutions, see Chapter 7.

In addition to being able to cover this reserve at all times, Swedish law states regulations on how the assets may be invested. For instance, a maximum of 25% of the assets used to cover this reserve may be invested in stock, a maximum of 25% in real estate and so forth. This does not limit us to placing more than 25% in stock, but the additional means invested in stock may not be counted when we compare our total assets to the prospective reserve.

5.2.b RETROSPECTIVE RESERVE.

The retrospective reserve represents the total sum the company claims to have allocated to the customers. We say "claim" as this allocation is not final as mentioned in Section 4.2. In order to measure these claims we define the term *consolidation* as the value of the total assets divided by the retrospective reserve. Since the company should not allocate more money than it owns, we add penalties for having too low a consolidation. According to the regulations the company may violate this limit by small amounts for shorter periods of time. Should it however be severely violated over a longer period of time, the regulating authorities may step in and require the company to re-take some of the bonuses allocated to the customers, thus decreasing the retrospective reserve and increasing the consolidation. Although this will not force the liquidation of the company it is most unfortunate from a public relations point of view. In order to avoid this as far as possible, we add increasing penalties for dropping below different levels down to 95% of the retrospective reserve.

5.2.c BONUS RATE OF RETURN

In addition to the requirements forced upon us by laws and regulating authorities, LIVIA has the stated policy of trying to keep the bonus rate of return on a high and steady level. In order to achieve this we define three reference levels for the return rate and penalize whenever the rate drops below these levels. These levels are the base rate of return plus 4%, 2% and 0% respectively. Currently we do not implement any means of keeping the bonus rate of return stable over time, although this should be a goal of the optimization. In order to keep the relative importance of the interest rate constraints constant, the penalties are scaled with the retrospective reserve, in effect penalizing the amount of money the customers are not allocated.

5.3 ADDITIONAL MODELLING CONCERNS

In addition to the unwanted events mentioned above, there are a couple of other aspects of the modelling we need to address.

5.3.a TAXES

The company must pay taxes on the assets owned. Currently this rate is 15% of the state interest rate on our total assets. Happily enough from a modelling point of view all assets should be valued at market value, thus not forcing us to keep track of the price at which they were bought. In reality the taxes are determined by the state interest rate at the turn of the year. As we do not wish to keep track of which time of year the model is started, we simplify and assume that tax is payed continuously and is related to the current state interest rate.

5.3.b TRANSACTION COSTS

We have assumed that we incur a transaction cost whenever we trade an asset, and further that this transaction cost is proportional to the amount bought/sold. This cost is supposed to mimic both the actual brokerage cost, and the value lost due to spread (the difference between bidding and asking price on the market).

5.3.c DIRECT YIELD

In the model we allow for assets to give us direct yield (dividends from stock and interest payments from bonds). Currently this feature is included in the model for future reference, but all direct yield is set to 0. This is appropriate since we assumed all dividends to be reinvested when we fitted our model of the economy in Section 3.3.

5.3.d NONLINEARITY

Most of the model described above is linear. The only remaining terms are the retrospective reserve and the payments made to the customers, which are nonlinear functions of all previous bonus rates of return. If we linearize these we will end up with a linear model, something most desirable when we optimize. Note that these functions are only slightly nonlinear with respect to the bonus rate of return, as may be seen in Chapter 7

5.3.e TIME DISCRETISATION

In order to make the model implementable, we must restrict ourselves to a discrete number of occasions when we make our decisions. We introduce a discrete set of solution times, at which we may buy and sell assets, and alter the bonus rate of return.

5.4 MATHEMATICAL MODEL

Now we are ready to formulate the mathematical model of our problem. The difference between decision variables and penalty variables are that we control the former while the latter only are used to sum up our violations of the constrains. As this is a general specification of our model, we do not specify in which assets we may trade, at which times we may do so, or what the penalties or other constants actually are. All this data need to be supplied whenever we create an instance of the model. In conjunction with the specification of the constraints and the utility function, we will give a short description of what the different equations represent.

5.4.a NOTATION

In order to define our model we need a number of sets over which we define our constraints, parameters and variables.

T	Time horizon.
$t = 0..T$	Decision stages.
ξ^t	Random variables revealed up to decision stage t.
I	Set of asset classes.
K	Set of capital cover rules (i.e, "max 25% in stock" etc.).
$I_k \subset I, k \in K$	Subset of asset classes affected by a capital cover type rule.
Q	Set of penalties, prospective reserve.
L	Set of penalties, consolidation.
A	Set of penalties, bonus rate of return.

5.4.b DECISION VARIABLES

Here we list the variables which we change to control the actions of the company.

$x_i^t(\xi^t)$	Amount of asset i held at time t.
$y_{+i}^t(\xi^t)$	Amount of asset i bought at time t.
$y_{-i}^t(\xi^t)$	Amount of asset i sold at time t.
$\widehat{x}_i^t(\xi^t)$	Amount of asset i used to cover the prospective reserve at t.

$r^t(\xi^t)$	Bonus rate of return at time t.

5.4.c SHORTHAND VARIABLES

These variables are added for notational simplicity and to make the constraint matrix sparser. In addition we need these variables when using the fixed mix strategy, as described in Chapter 8

$x^t_{\text{tot}}(\xi^t)$	Total value of assets at time t.

5.4.d PENALTY VARIABLES

These variables are not a part of a decision, the are used only to compute the penalties for a specific solution.

$z^t_p(\xi^t)$	Violation of prospective reserve cover type constraint at time t.
$z^t_q(\xi^t)$	Violation of prospective cover reserve $q \in Q$ at time t.
$z^t_a(\xi^t)$	Violation of low return rate class $a \in A$ at time t.
$z^t_k(\xi^t)$	Violation of maximum consolidation at time t.
$z^t_l(\xi^t)$	Violation of minimum consolidation rule $l \in L$ at time t.

5.4.e FUNCTIONS

$V^t(r^0, \ldots, r^{t-1})$	Retrospective reserve as a function of previous return rates at time t.
$P^t_{\text{out}}(r^0, \ldots, r^{t-1})$	Payments to customers from time $t-1$ to time t as a function of previous return rates.

5.4.f PARAMETERS

$S^t(\xi^t)$	Prospective reserve.

$\eta_i^t(\xi^t)$	Price development of asset class i from time $t-1$ to t.
$\rho_i^t(\xi^t)$	Direct return of asset class i form time $t-1$ to t.
γ_i	Transaction cost for asset class i.
P_{in}^t	Premium inflow between time $t-1$ and t.
c_k	Maximum prospective cover from assets of set I_k.
s_q	Penalty for violating reserve cover rule q.
s_p	Penalty for violating reserve cover type rule.
s_a	Penalty for low return class a.
s_k	Penalty for high consolidation.
s_l	Penalty for low consolidation rule l.
f_q	Security factor for prospective cover rule q.
$r_{\text{ref}}^t(\xi^t)$	Reference interest rate.
$\Delta r_{\text{ref},a}$	Offset from reference interest rate for penalty class a.
$\kappa_{\max}$	Maximum consolidation.
$\kappa_{\min,l}$	Minimum consolidation, rule l.
$\theta(\xi^t)$	Tax level times period length (that is, the actual taxes paid during the period).
$d^t(\xi^t)$	Discount factor.
x_i^{-1}	Assets owned before first time step (initial value).

5.4.g UTILITY FUNCTION

We sum up the expected terminal wealth of our company (5.1c) combined with the money already paid to the customers (5.1b). We deduct penalties for not covering the prospective reserve with the correct assets (5.1d), for not covering the prospective reserve at different levels (5.1h), for having too low bonus rate of return (5.1e), for having too high consolidation (5.1f) and for having too low consolidation (5.1g). Everything is discounted using the discount factor $d^t(\xi)$ in order to compensate for not using real values in the model.

$$
\begin{aligned}
&w(x,z,r) := && \text{(5.1a)}\\
&\sum_{t=1}^{T} \mathsf{E}[d^t(\xi^t)P^t_{\text{out}}(x^0,\dots,x^t)] && \text{(5.1b)}\\
&+ \mathsf{E}[d^{\text{T}}(\xi^t)x^T_{\text{tot}}(\xi^{\text{T}})] && \text{(5.1c)}\\
&- \sum_{t=0}^{T} \mathsf{E}[d^t(\xi^t)s_p z^t_p(\xi^t)] && \text{(5.1d)}\\
&- \sum_{t=0}^{T} \mathsf{E}[d^t(\xi^t)\sum_{a\in A} s_a z^t_a(\xi^t)V^t(r^0,\dots,r^{t-1})] && \text{(5.1e)}\\
&- \sum_{t=0}^{T} \mathsf{E}[d^t(\xi^t)s_k z_k(\xi^t)] && \text{(5.1f)}\\
&- \sum_{t=0}^{T} \mathsf{E}[d^t(\xi^t)\sum_{l\in L} s_l z_l(\xi^t)] && \text{(5.1g)}\\
&- \sum_{t=0}^{T} \mathsf{E}[d^t(\xi^t)\sum_{q\in Q} s_q z_q(\xi^t)]. && \text{(5.1h)}
\end{aligned}
$$

5.4.h PENALTY DEFINING CONSTRAINTS

Requirements to cover the prospective reserve

We must not use more of an asset to cover the prospective reserve than we do actually own:

$$\widehat{x}^t_i(\xi^t) \le x^t_i(\xi^t), \quad \forall t = 0,\dots,T, \quad \forall i \in I. \tag{5.2}$$

Prospective reserve cover type rule

We must cover the prospective reserve with assets of the correct type, or pay a penalty:

$$\sum_{i\in I} \widehat{x}^t_i(\xi^t) + z^t_p(\xi^t) \ge S^t(\xi^t), \quad \forall t = 0,\dots,T. \tag{5.3}$$

We must not use more than the allowed amount of an asset class to cover the prospective reserve.

$$\sum_{i\in I_k} \widehat{x}_i^t(\xi^t) \leq c_k S^t(\xi^t), \quad \forall t = 0,\ldots,T, \quad \forall k \in K. \tag{5.4}$$

Prospective reserve cover rules

We should penalize not covering the prospective reserve with a certain safety-margin. Note that these constraints do not use the variables $\widehat{x}$ as we do not wish to restrict the asset-types simply because we enforce a safety-margin.

$$x_{\text{tot}}^t(\xi^t) + z_q(\xi^t) \geq f_q S^t(\xi^t), \quad \forall t = 0,\ldots,T, \quad \forall q \in Q. \tag{5.5}$$

Retrospective cover rules

We should penalize a too high consolidation (5.6b) and a too low consolidation (5.6a).

$$x_{\text{tot}}^t(\xi^t) + z_l(\xi^t) \geq \kappa_{\min,l} V^t(r^0,\ldots,r^{t-1}), \quad \forall t = 0,\ldots,T, \quad \forall l \in L, \tag{5.6a}$$

$$x_{\text{tot}}^t(\xi^t) - z_k(\xi^t) \leq \kappa_{\max} V^t(r^0,\ldots,r^{t-1}), \quad \forall t = 0,\ldots,T. \tag{5.6b}$$

Interest rate level

We should penalize a low level of the bonus rate of return:

$$r^t(\xi^t) + z_a^t(\xi^t) \geq r_{ref}^t(\xi^t) + \Delta r_{\text{ref},a}, \quad \forall t = 0,\ldots,T, \forall a \in A. \tag{5.7}$$

5.4.i LINKING CONSTRAINTS

Aggregation

We make sure that the total value of assets equals the sum of the parts:

$$x_{\text{tot}}^t(\xi^t) = \sum_{i\in I} x_i^t(\xi^t), \quad \forall t = 0,\ldots,T. \tag{5.8}$$

Time linking

What we own of an asset at a certain time should equal what we owned the previous period, times the price development of the asset, to which we should add the value bought and deduct the value sold.

$$x_i^t(\xi^t) = y_{+i}^t(\xi^t) - y_{-i}^t(\xi^t) + \eta_i^t x_i^{t-1}(\xi^{t-1}) \quad \forall t = 0, \dots, T. \tag{5.9}$$

Mass conservation

We should make sure that we do neither invent nor destroy any money. Hence the net amount we pay in tax (5.10c) and payments to customers (5.10b) should equal what we get from other customers (5.10a), net gains made from selling assets (5.10d) and the received direct yield (5.10e),

$$
\begin{aligned}
& P_{\text{in}}(\xi^t) && \text{(5.10a)}\\
& - P_{\text{out}}(r^0, \dots, r^{t-1}) && \text{(5.10b)}\\
& - \theta^t(\xi^t) \sum_{i \in I} x_i^t(\xi^t) && \text{(5.10c)}\\
& + \sum_{i \in I} \big(y_{-i}^t(\xi^t)(1-\gamma_i) - y_{+i}^t(\xi^t)(1+\gamma_i)\big) && \text{(5.10d)}\\
& + \sum_{i \in I} x_i^{t-1} \rho_i^t(\xi^t) && \text{(5.10e)}\\
& = 0 \quad \forall t = 0, \dots, T.
\end{aligned}
$$

5.4.j NON-NEGATIVITY

As we may not short sell assets we must never own negative amounts of our assets,

$$x_i^t(\xi^t) \geq 0, \quad \forall t = 0, \dots, T, \quad \forall i \in I. \tag{5.11}$$

The penalties should only apply when we violate a requirement:

$$z_p^t \geq 0, \quad \forall t = 0, \ldots, T, \tag{5.12a}$$
$$z_q^t \geq 0, \quad \forall t = 0, \ldots, T, \forall q \in Q, \tag{5.12b}$$
$$z_k^t \geq 0, \quad \forall t = 0, \ldots, T, \forall k \in K, \tag{5.12c}$$
$$z_l^t \geq 0, \quad \forall t = 0, \ldots, T, \forall l \in L, \tag{5.12d}$$
$$z_a^t \geq 0, \quad \forall t = 0, \ldots, T, \forall a \in A. \tag{5.12e}$$

We may now express the full model as follows:

maximize (5.1) subject to (5.2)–(5.12).

5.5 LINEARIZATION

As mentioned earlier, we need to linearize the nonlinear parts of our model in order to make it easier to solve.

5.5.a PARAMETERS

We need to add the following parameters.

$\bar{r}^t(\xi^t)$	Assumed bonus rate of return at time t.
$\bar{V}^t(\xi^t)$	Assumed retrospective reserve at time t (calculated using previous assumed bonus rates $\bar{r}^t(\xi^t)$).
$\bar{P}^t_{\text{out}}(\xi^t)$	Assumed payments to customers from time $t-1$ to time t (calculated using previous assumed bonus rates).
$\frac{\bar{V}^t(\xi^t)}{\bar{r}^\tau(\xi^\tau)}$	Partial derivative of the assumed retrospective reserve w.r.t the bonus rate at time τ.
$\frac{\bar{P}^t_{\text{out}}(\xi^t)}{\bar{r}^\tau(\xi^\tau)}$	Partial derivative of the assumed payments with respect to the bonus rate at time τ.
$\Delta r_{\max}$	Maximum deviation from assumed value.

5.5.b VARIABLES

Further, we need to add the following variables:

$P_{\text{out}}^t(\xi^t)$	Taylor expansion of degree 1 of payments to customers from time $t-1$ to time t around the point $\bar{P}_{\text{out}}^t(\xi^t)$.
$V^t(\xi^t)$	Taylor expansion of degree 1 of retrospective reserve around the point $\bar{V}^t(\xi^t)$.

We replace $V^t(r^0, \dots, r^{t-1})$ with $V^t(\xi^t)$ in (5.6) and add the constraints

$$V^t(\xi^t) = \bar{V}^t(\xi^t) + \sum_{\tau=0}^{t} \frac{\partial \bar{V}^t(\xi^t)}{\partial \bar{r}^\tau(\xi^\tau)} (r^\tau(\xi^\tau) - \bar{r}^\tau(\xi^\tau)), \quad \forall t = 0, \dots, T. \tag{5.13}$$

In the same fashion we replace $P_{\text{out}}^t(r^0, \dots, r^{t-1})$ by $P_{\text{out}}^t(\xi^t)$ in (5.1) and add the constraints

$$P_{\text{out}}^t(\xi^t) = \bar{P}_{\text{out}}^t(\xi^t) + \sum_{\tau=0}^{t} \frac{\partial \bar{P}_{\text{out}}^t(\xi^t)}{\partial \bar{r}^\tau(\xi^\tau)} (r^\tau(\xi^\tau) - \bar{r}^\tau(\xi^\tau)), \quad \forall t \in 0, \dots, T. \tag{5.14}$$

In order to keep the errors in our expansions low, we introduce box-constraints on the bonus rate of return:

$$-\Delta r_{\max} \le r^t(\xi^t) - \bar{r}^t(\xi^t) \le \Delta r_{\max}, \quad \forall t \in 0, \dots, T. \tag{5.15}$$

With these replacements made, we may express the linearized problem as the problem to maximize (5.1) subject to (5.2)–(5.15).

We may now solve our original problem using sequential linear programming. Between iterations we update the values of the assumed bonus rates and the values of $P_{\text{out}}^t(\xi^t)$ and $V^t(\xi^t)$ are recalculated using the customer model. As the constraints we use are only slightly nonlinear, we may assume that sequential linear programming will converge for our problem, something which is confirmed from the tests made.

Chapter 6

Implementation

6.1 ALGEBRAIC MODELLING LANGUAGES

The formulation of an optimization problem is greatly facilitated by the use of an *algebraic modelling language* (AML). An AML allows the user to formulate a model using a more natural syntax as compared to explicitly writing the different matrices and vectors involved in formulating the problem. Most modelling languages do however not yet explicitly support stochastic programming, although drafts are sometimes prepared to deal with this problem. An example of a proposed extension to AMPL may be found in [22]. There are some attempts based on existing AMLs where one MPS-file is written for each scenario, and these are condensed to the SMPS-format (a de facto standard for stochastic programming problems, see [4]) outside the modelling kernel, a technique used in [17] as well as in earlier versions of the SPINE system [34]. Other systems use the AML solely for defining the core problem (a problem where all random variables have one possible outcome) and supply the stochastic data separately. Such a system is SLP-IOR which is described in [27]. One may also express the entire scenario tree inside the modelling language explicitly, however the additional model parts needed to express non-anticipativity tend to make the model less clear to read and understand. In addition, when wanting to experiment with different solution techniques it is important that we may extract information of the problem structure in a simple manner. This may not always be done if we express the deterministic problem in a modelling language, as we have no way of extracting the period structure from the final optimization problem. In order to fulfil the needs of this project an existing language with open source, PLAM, has been modified to accommodate stochastic programming problems. A summary of the capabilities of PLAM may be found in [2].

In the modifications of PLAM, we have made it possible to declare some parameters as being random. In addition, all variables and random parameters may be assigned a period when their values become known. From the modelling environment we output a standard linear program of the same type as (1.11), and in the same fashion, the problem is divided into time-stages.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{t=0}^{T} c_t^{\mathrm{T}} x_t, \\
\text{subject to} \quad & W_0 x_0 = h_0, \\
& \sum_{k=0}^{t-1} A_{t,k} x_k + W_t x_t = h_t, \quad t = 1, \ldots, T, \\
& x_t \geq 0, \quad t = 0, \ldots, T.
\end{aligned}
$$

Some elements of these matrices and vectors will however be algebraic expressions of our random parameters; for instance, we may have

$$W_1(5,7) = \text{direct\_return\_stock\_1*deflator\_1}.$$

This means that the value at position (5,7) in the matrix W_1 is dependent on the random variables direct_return_stock_1 and deflator_1. When instantiating the problem the parameters direct_return_stock_1 and deflator_1 will be filled with the correct realisations of these values depending on which scenario we are in. We do thus separate the random data from the rest of the problem, and the scenario tree will consist only of the outcomes of random variables which are needed in our formulation.

6.2 SCENARIO GENERATION

The goal of the scenario generation process is to generate a good representation of all the possible outcomes. Assuming that our model may, through simulations, provide us with samples from the modelled distribution, we may generate a tree by simply conditionally generating random scenarios. This will however at least locally cause the statistical properties of the tree to greatly deviate from the statistical properties of the model, although we may use conditional sampling to reduce the variance.

We hope that a more structured approach to scenario tree generation will provide better results. If we have a model with a simple structure, (for instance all our random variables are jointly normal) we may manually create approximations to the probability function. An example of this is [28] in which Keefer examines the error in a utility function when approximating continuous random variables with different discrete approximations. Since our scenario generation method is a bit more complicated, and we wish to retain the possibility of adding even more complex schemes, we choose a method similar to the one used by Høyland in [26] in which a nonlinear optimization problem is solved to obtain a scenario tree which is fitted to the statistical properties of the model. The model in Chapter 3 is used to give us the statistical properties of the one period distribution. The properties in question are the four lowest moments of the random variables and their correlation. If the model is complex enough to prevent the derivation of these properties directly they may be obtained through simulation. To this model the statistical properties of the descendant nodes are fitted. This is done by using the nonlinear optimization package donlp2 by Peter Spellucci [39]. The decision variables are the Swedish treasury bill rate, the Swedish bond rate and the yields on the asset classes not determined directly by these interest rates (Swedish and foreign stock, and foreign bonds), as well as the probabilities of the scenarios.

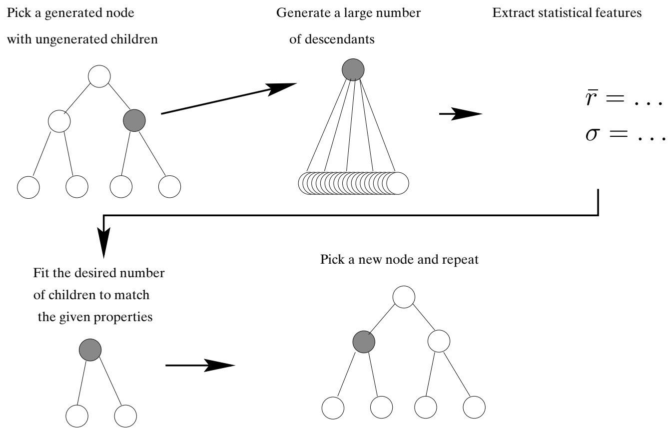

Figure 6.1: Tree generation.

We use random outcomes as our initial solution, and the goal is to minimize the deviation from the specified properties.

If we formalize this we end up with the problem solved for each node.

N	number of child nodes.
M	number of independent variables.
x_i^j	independent variable i in child node j.

p^j	Probability of node j.
$\bar{x}_i$	Target value for average of variable i.
σ_{ik}	Target value for correlation of variable i and variable k.
γ_i	Target value for third order moment of variable i.
δ_i	Target value for fourth order moment of variable i.
$p_{\min}$	Minimum probability for child node.

$$\text{minimize} \tag{6.1a}$$

$$\sum_{i=1}^{M}\left(\frac{\sum_{j=1}^{N}(x_i^j)p^j}{\bar{x}_i}-1\right)^2 \tag{6.1b}$$

$$+\sum_{i=1}^{M}\sum_{k=1}^{M}\left(\frac{\sum_{j=1}^{N}(x_i^j-\bar{x}_i)(x_k^j-\bar{x}_k)p^j}{\sigma_{ik}^2}-1\right)^2, \tag{6.1c}$$

$$+\sum_{i=1}^{M}\left(\frac{\sum_{j=1}^{N}(x_i^j-\bar{x}_i)^3p^j}{\gamma_i^3}-1\right)^2 \tag{6.1d}$$

$$+\sum_{i=1}^{M}\left(\frac{\sum_{j=1}^{N}(x_i^j-\bar{x}_i)^4p^j}{\delta_i^4}-1\right)^2 \tag{6.1e}$$

$$\text{subject to}\quad 1=\sum_{j=1}^{N}p^j, \tag{6.1f}$$

$$p^j \geq p_{\min},\quad j=1,\ldots,N. \tag{6.1g}$$

6.2.a ANTITHETIC SAMPLING

When we made our model of the surrounding economy in Chapter 3 we use Wiener processes to drive our simulations. As a wiener process is symmetric, a certain realization dz will be just as probable as $-dz$. By using both dz and $-dz$ we will get a better estimate of the means of our process compared to using entirely unrelated driving processes. The use of antithetic sampling and other methods to reduce the variability of solutions in stochastic programming has been studied by Higle in [24]

6.2.b ALTERATION OF SIMULATED VALUES

The simulation approach to obtaining the stochastic properties of the model has a drawback; we will get errors in the properties. As the model is rather sensitive to changes in the expected value of our variables, we have calculated these analytically (see Section 3.2) and replaced the simulated averages with the correct ones.

6.2.c ARBITRAGE

Even if the model constructed is in itself free of arbitrage (although we do not claim that the model used here is) the discretization of the model may result in the created tree containing arbitrage opportunities. Usually we may not utilize these possibilities directly, as we are restricted from short selling assets. However Klaasen [29] has shown that the presence of such opportunities may significantly bias the solution of the stochastic program even when short selling restrictions are present.

Our remedy to this problem is quite simple. We simply generate the descendants of a node, check for arbitrage, and if arbitrage is present, generate the descendants again. We hence must implement a check for arbitrage. An arbitrage portfolio is a portfolio that

- is self financing (has value 0 at time t),
- has a positive expected value at some future time $t + \Delta t$,
- almost surely has a nonnegative value at this future time.

We check for arbitrage through solving the following linear program.

M	Number of asset classes.
N	Number of child states.
y_i^j	Yield of asset class i between time t and $t + \Delta t$ in scenario j.
x_i	Amount invested in asset class i.
p^j	Probability of scenario j.

$$\text{maximize} \quad z = \sum_{i=1}^{M} \sum_{j=1}^{N} y_i^j p^j x_i, \tag{6.2a}$$

$$\text{Subject to} \quad \sum_{i=1}^{M} y_i^j x_i \geq 0, \quad \forall j = 1, \ldots, N, \tag{6.2b}$$

$$x_i \geq -1, \quad \forall i = 1, \ldots, M. \tag{6.2c}$$

The objective function represents the expected profit of our investment, and the constraints (6.2b) guarantee that we do not loose any money in any scenario. If this problem has a solution with $z > 0$ we know that arbitrage opportunities exists, since we may make a profit without risk.

6.3 THE TOTAL SYSTEM

When we pull all the components together we end up with the system seen in Figure 6.2. Currently the different subsystems must be run by hand, or via custom written shell scripts, but in the future we hope to integrate it all into a more easy to use package.

We specify a model and scenario independent data as input to PLAM, which creates a template LP. An initial state of the economy (i.e, interest rates) is given to the economy model. With this starting stage, the economy model is used to generate a fitted scenario tree using the method described earlier in this chapter. The scenario tree and the current state of the company (assets owned) are given to the solver which produces a solution which is optimal with respect to the given scenario-tree. The customer model is included in both the scenario generation program (to give the prospective reserve) and in the solver (to give the retrospective reserve). Thus the actions within the box labelled "Template model generation" are performed once, where as the rest of the actions are performed once per application of the model. When we run the model over several time steps, the solution from the first stage will become the company state for the next stage and so forth.

6.4 A POSSIBLE PITFALL

Above we assume that we have model a which may give us one realization of the outcome of our external variables from a distribution which we believe to be true. When we optimize the variables to fit a certain distribution, we may however not be sure that the resulting outcomes are probable realizations of our process. A more appropriate solution to the selection of our representative scenarios would be to require that they should in fact be generated by the model, and the optimization of a certain node would then consist of finding the subset of the correct cardinality of a given larger set of realizations that do the best job of describing the properties of the tree, possibly optimising over the probabilities of the different outcomes. However, this would alter our problem from being a nonlinear constrained optimization problem to being a mixed integer constrained nonlinear problem, something which would be considerably harder to solve. Hence we have to settle for the current approach.

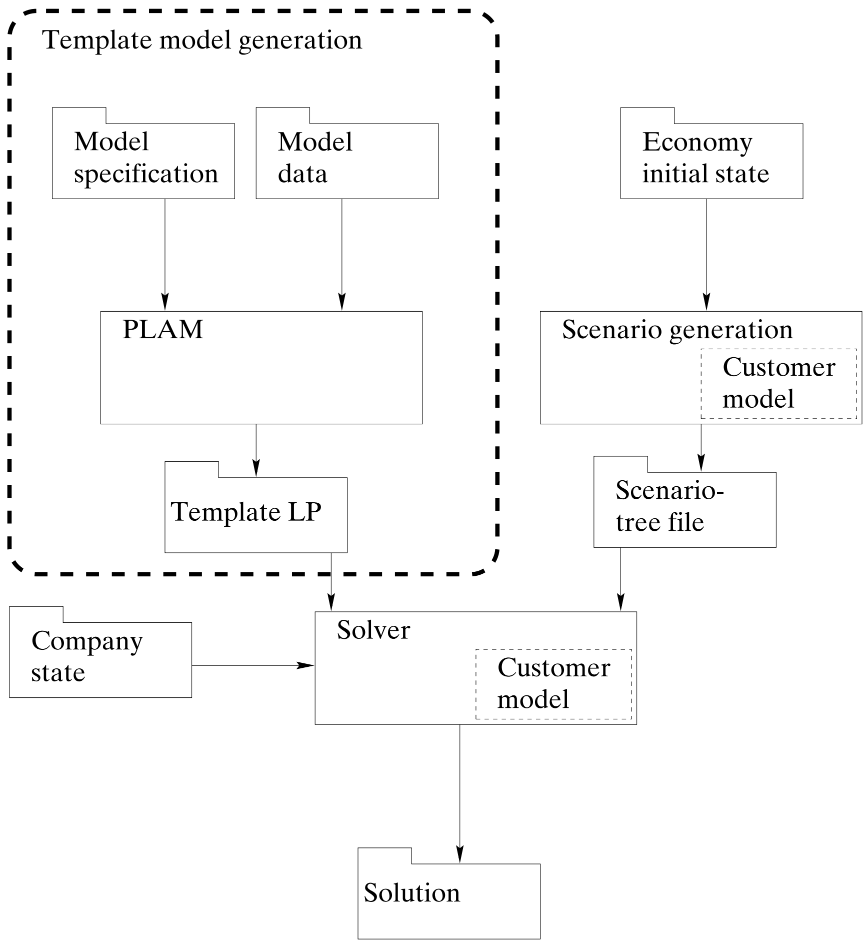

Figure 6.2: The entire system used to generate a decision.

Chapter 7

Properties

In this chapter we take a brief look at our problem, trying to describe some aspects of its behaviour.

We look at the level of stochasticity in the problem, that is, we try to find how important the random components are. What will happen if we remove all random components? What will happen if we become clairvoyant, knowing the random outcomes in advance?

We further look at the structure of the constraints of the extensive form, how much we are affected by the nonlinearities mentioned in Chapter 5 and why we need to have progressively increasing penalties for violations of the reserve requirements.

7.1 STOCHASTICITY

We try to measure the level of stochasticity in our problem, how much the presence of random components affect our solution. In order to do this we define a number of terms. *The wait and see solution*, WS, is the expected value of our objective function if we have perfect information. We get the value of the wait and see solution, WS, as the expected value of the optimal value of the problem

$$
\begin{aligned}
\text{minimize} \quad & f(\overrightarrow{x}_T(\xi_T), \xi_T), && \text{(7.1a)}\\
\text{subject to} \quad & x_0 \in X_0(\xi_0), && \text{(7.1b)}\\
& x_t \in X_t(\overrightarrow{x}_{t-1}(\xi_t), \xi_t), \quad \forall t = 1, \dots, T. && \text{(7.1c)}
\end{aligned}
$$

Note that this problem contains no expectations as we know the outcome of all random varibles.

The naïve alternative to using the full stochastic formulation is to replace all random data ξ_t with their expected values $\bar{\xi}_t = \mathsf{E}(\xi_t)$. Doing this yields the *mean value problem*:

$$
\begin{aligned}
\text{minimize} \quad & f(\overrightarrow{x}_T(\xi_T), \bar{\xi}_T), && \text{(7.2a)}\\
\text{subject to} \quad & x_0 \in X_0(\bar{\xi}_0), && \text{(7.2b)}\\
& x_t \in X_t(\overrightarrow{x}_{t-1}(\bar{\xi}_t), \bar{\xi}_t), \quad \forall t = 1, \dots, T, && \text{(7.2c)}
\end{aligned}
$$

and we denote its optimal value EV.

Finally, as we are interested in how well the solution of the mean value problem fares when we apply it to the full random problem, we define the *expected value of the mean value solution*, EEV. In order to get this value we simply apply the first stage decision from the mean value problem to the full stochastic problem and solve the corresponding recourse problems. Thus, if the optimal first stage solution of the mean value problem is $\bar{x}_0$, we get the value EEV as the optimal value of the problem

$$
\begin{aligned}
\text{minimize} \quad & \mathsf{E}\left(f(\overrightarrow{x}_T(\xi_T), \xi_T)\right), && \text{(7.3a)}\\
\text{subject to} \quad & x_0 \in X_0(\xi_0), && \text{(7.3b)}\\
& x_t \in X_t(\overrightarrow{x}_{t-1}(\xi_t), \xi_t), \quad \forall t = 1, \dots, T, && \text{(7.3c)}\\
& x_0 = \bar{x}_0. && \text{(7.3d)}
\end{aligned}
$$

For further reference we denote the value of the recourse problem (1.7) by RP.

Having defined these expressions, we may now define two other numbers, the *expected value of perfect information*, EVPI which is calculated as

$$
\text{EVPI} = \text{WS} - \text{RP}, \tag{7.4}
$$

and as the name suggests it is the expected value of perfect information of all future random outcomes. We define the relative EVPI as

$$
\text{REVPI} = \frac{\text{EVPI}}{\text{RP}}. \tag{7.5}
$$

WAS	17.979885
RP	16.069245
EV	16.344863
EEV	16.068268
EVPI	1.91064
Relative EVPI	0.1189
VSS	0.000977

Table 7.1: Indicator values, original assets.

We may further define the *value of the stochastic solution*, VSS, as

$$\mathrm{VSS} = \mathrm{RP} - \mathrm{EEV}, \tag{7.6}$$

which is the improvement in optimal solution value from using a stochastic formulation of our problem.

We may now use these values to determine a measure of the stochasticity of our problem. In Chapter 8 we define a test-case starting in November 1999 having three periods. If we look at this problem from the start of our simulations we get values according to Table 7.1, where all values are in billion SEK.

Most worth noting is that we have a very low VSS. It would hence seem that using SP in our problem is a waste of time. This is true under the economic circumstances prevailing at the start of our simulations. The reason for this is simple, at the start of the simulations, we have a good safety-margin versus all our reserves (we have a consolidation of about 119%). As we have a good margin, we have a smaller need to hedge against unwanted outcomes as we will probably not break any reserve rules even if the outcome of our investment is unfavourable. Due to this fact the stochastic programming approach will suggest the same portfolio as the average problem does, investing just enough in Swedish bonds to fulfil the reserve requirements, and investing the rest in Swedish stock which has the highest expected yield. Hence the hedging considerations inherent in the stochastic programming approach will not affect our problem, and the only contribution to VSS comes from the models setting the bonus rate of return differently.

In order to see if the system becomes more usable under less favourable conditions, we use the same initial conditions, but remove some money from the company. We remove enough money to bring the consolidation down to 101%; the results may be viewed in Table 7.2. Now the asset mix changes between the average case and the stochastic programming case, as the proximity to the requirements forces us to hedge against unfavourable

WAS	15.793091
RP	13.350385
EV	14.717167
EEV	13.209668
EVPI	2.4427
Relative EVPI	0.1830
VSS	0.1407

Table 7.2: Indicator values, reduced assets.

outcomes. We see that for this case the EVPI increases, perfect information is more profitable as it will let us avoid breaking the requirements, and hence avoid penalties. We also see that the value of the stochastic solution increases. Hence stochastic programming may pay when we need to hedge against unfavourable outcomes more than what is implied by the owning restriction forced upon us by law.

7.2 PROBLEM STRUCTURE

In order to see what the problem matrix structure looks like, we instantiate a problem with a scenario-tree of the form given in Figure 1.1. In Figure 7.1 we see the resulting deterministic equivalent constraint matrix (the matrix $\widehat{A}$ in Figure 1.2).

As we may see from Figure 7.1 we clearly have the block structure from Figure 1.2. Worth noting is that we almost have fixed recourse, which may be seen in Figure 7.1. There is only one single random entry in the recourse matrices [$W_t(\xi_t)$ in (1.11), the diagonal blocks in the figure]; this entry is the tax payments which vary between the scenarios, as they are dependent on the interest rates which are in turn scenario-dependent.

7.3 NONLINEARITY

As mentioned earlier, the retrospective reserve will be a mildly non-linear function of the bonus rate of return. In order to give an idea of the magnitude of this nonlinearity, and of its impact on the problem, we plot the correct retrospective reserve and a fitted linear function. We plot the retrospective reserve after 24 months, when the bonus rate has one constant value for 6 months, and another constant value for the last 18 months. The period lengths are chosen to correspond to the two first stages of our test-runs in Chapter 8. In Figure 7.2 we see the true value and the fitted plane, and in Figure 7.2 the difference between a fitted plane and the true values. Note the scales in the figures.

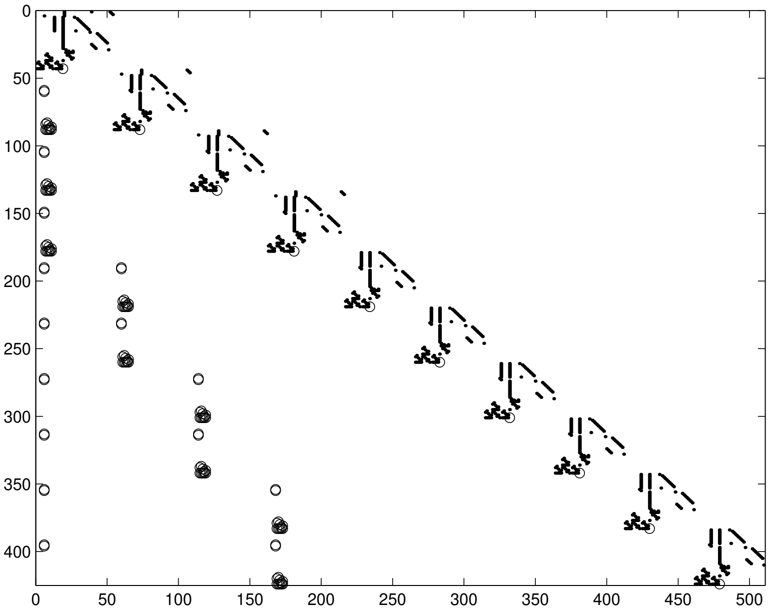

Figure 7.1: Problem matrix. Random entries are circles.

As may be seen in the figures, the difference is very small for moderate ranges of the bonus rate of return (less than 1% of the total retrospective reserve). As we may on occasion get extremely high bonus rates of return (more or less corresponding to one-time bonus returns to the customers), these nonlinearities still may influence the solution. In addition, solving the mildly nonlinear problem using sequential linear programming (see Chapter 5) as opposed to solving the linearized version once carries only a small computational penalty; as we may reuse older solutions the increase in the total solution time is around 10–15% for our basic test-case from Chapter 8.

7.4 THE NEED FOR PENALIZING ILLEGAL STATES

In Subsection 5.2.a we mentioned the need to use progressively steeper penalties for larger violations of the requirements. This need is not apparent as we have stated that failing to meet this reserve is a disastrous event which should more or less stop our simulation. In Chapter 8 we make 220 simulated runs using the model over 5 years. The example used to illustrate the need for penalizing illegal states is the worst outcome of all these runs. Figure 7.4 shows an earlier version of the model running

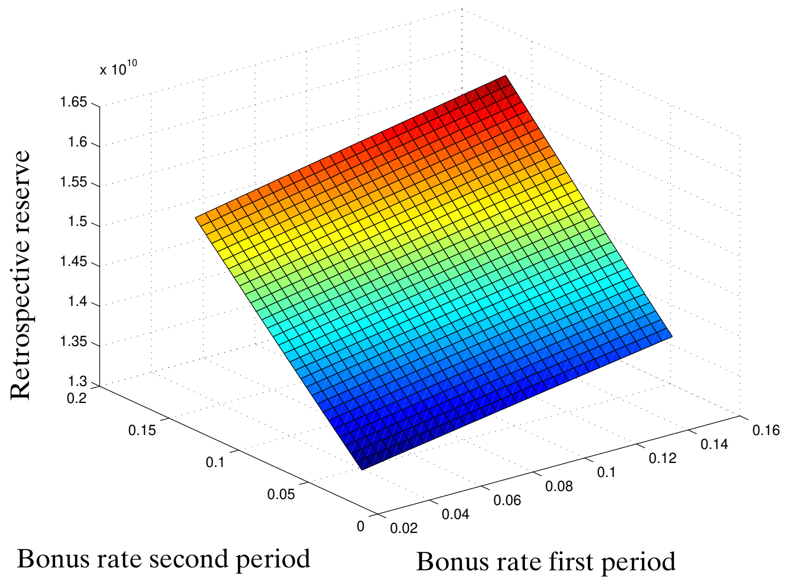

Figure 7.2: Retrospective reserve, true value and fitted plane.

the worst-case scenario, where we would be forced into liquidation after approximately 4 years (when the total assets falls below the prospective reserve). As may be seen, the model actually trades foreign stock for Swedish stock after this date, increasing the expected yield but increasing the risk as well, a behaviour which may seem strange. The reason is quite natural. After breaking the prospective reserve requirement, we do not any longer face progressively increasing penalties for larger violations. Hence the optimization ceases to try to hedge against low returns, and tries to maximize the expected yield by increasing the holdings of riskier assets with higher expected return. If we instead view the model where penalties increase progressively even when we are already breaking the requirement, the model behaves in a similar fashion before and after we start breaking the requirements.

If we use progressively steeper penalties even when we would be forced into liquidation, we get the results shown in Figure 7.5. Here, we get a consistent behaviour both before and after we would be forced into liquidation.

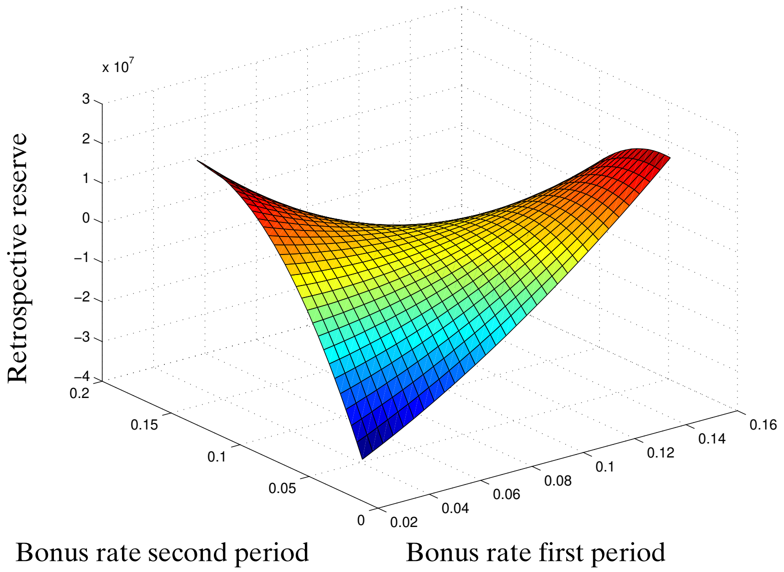

Figure 7.3: Retrospective reserve, difference between true value and fitted plane

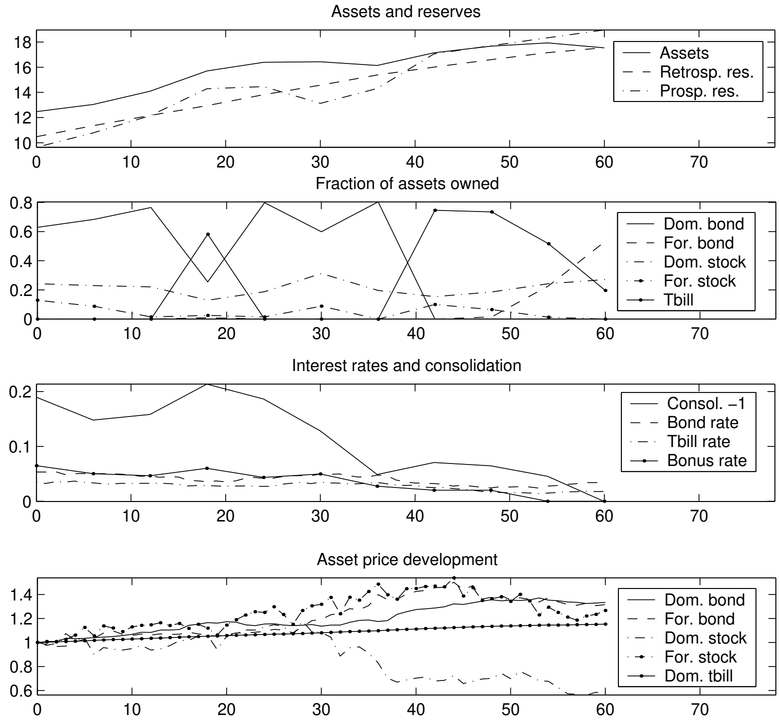

Figure 7.4: A sample run of an earlier version of the model.

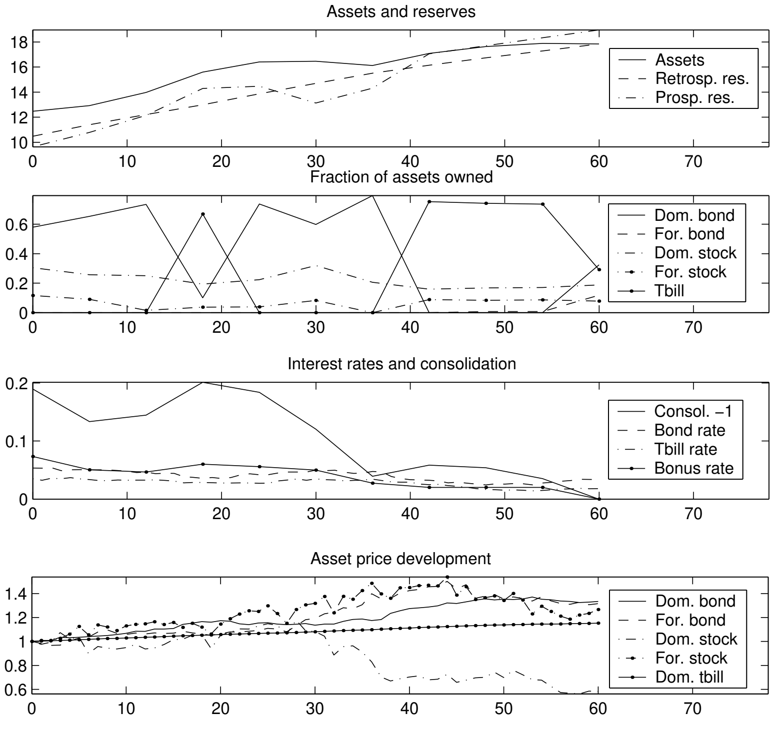

Figure 7.5: A sample run of the final model.

Chapter 8

Numerical experiments

Naturally, we wish to determine if our proposed model may give us useful advice on how to run the company. In order to do this we need a clearly defined testing procedure. The optimal value from the optimization is for this purpose useless since it will neither reflect how well the model performs if it is periodically restarted, nor how well the solution behaves when we use a different random seed.

8.1 TEST PROCEDURE

In order to run a scenario using our stochastic programming environment, we must define the size of the tree. In order not to get excessive simulation times, we have chosen a three-stage (that is, two decision stages and one recourse stage) tree which branches into 30 scenarios in the first time-stage, and each of those scenarios branches into 8 different scenarios in the second stage. The length of the first stage is 6 months and the length of the second is 18 months. This gives us a deterministic equivalent model of 240 scenarios with 12830 variables and 10629 constraints. This kind of problem size is by no means cutting edge. The size of the tree was chosen to give a reasonably good description of our probabilities, while making it possible to run 220 simulations over night on 5–6 workstations (Ultra sparc 10 with 128 or 256 MB memory). When we increased the problem size in test-case 2 by adding an extra stage, the size of the problems increased to 102433 constraints and 120234 variables. These larger LPs took approximately one hour each to solve. As we solve 1320 LPs per test case, this size of problem is impractical for extensive experimentiation.

The tests are carried out as follows:

First we use the model of the surrounding economy to generate a set of 220 test scenarios with a duration of 60 months, and with a time resolution of six months. All scenarios will have a common starting point, the state of the world and the company in November 1999. For the first test scenario we generate a scenario tree of the specified size using the data from November 1999. The tree is fed into the solver and we optimize to find the best solution with respect to the given tree. The first stage solution is extracted and we use data from the test scenario (*not* the generated scenario tree) to project this solution forward in time 6 months. The state of the world after 6 months according to the scenario is used to generate another tree. We solve again, using the solution from the last step as the initial state of our company. This process is repeated untill we reach the end of the test scenario, when we compute the merit function value for this scenario. This process is repeated for each test scenario. A schematic figure of the testing system is give in Figure 8.1.

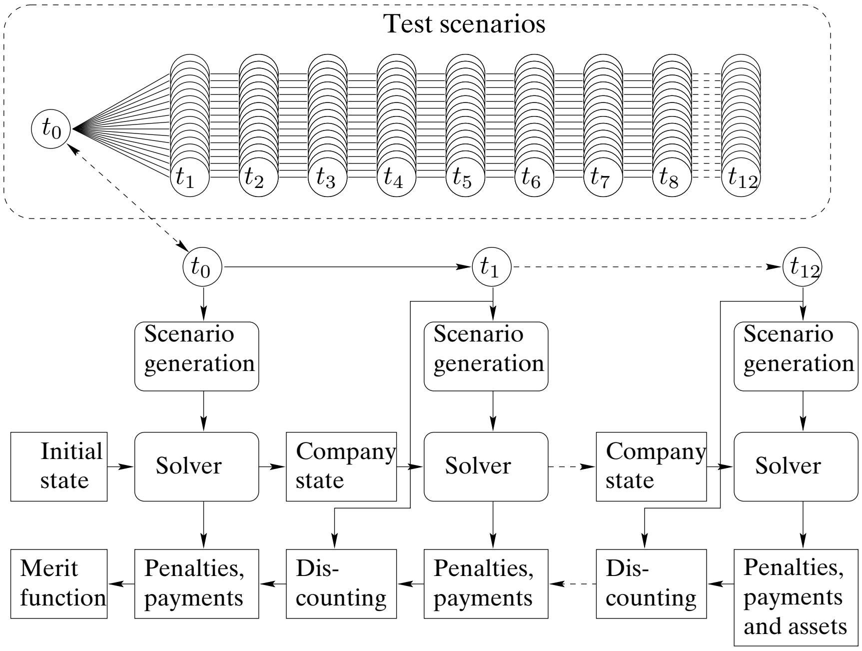

Figure 8.1: Testing system.

8.1.a MERIT FUNCTION

As a merit function we will use the discounted value of the company's total assets at the end of the simulation, to which we add the total discounted payments made to our customers, and we deduct the total penalties applied at each decision point (reduced in the manner described below).

If we look at the objective function for the stochastic linear program, our penalties are applied twice, once after 6 months, and once after 24 months for all penalties except the low return penalties, which are applied directly and after 6 months. Thus the penalties are applied twice over a 24 month period. When we do our time stepping scheme, we apply the penalties once every six months, or four times in a 24 month period. As the optimization assumes that the penalties should be applied twice, but when we evaluate sequentially over a scenario, they are applied four times, we need to scale the penalties by a factor of 2 when evaluating over a test scenario in order to make the testing procedure and the optimization match.

8.1.b BENCHMARK STRATEGY

As the current strategy of LIVIA is to choose a fixed mix of assets too keep, or rather to set bounds in which the asset proportions may vary, and as the idea of constantly re-balancing the portfolio to a given mix of assets is a popular strategy, we choose a *fixed mix* strategy as our benchmark. This approach is simply to decide on a mix of assets, and assume we keep this mix. The implementation is rather straightforward; we use the same stochastic linear program as earlier, but we add extra constraints to keep the assets at a constant mix. These extra constraints are specified as

$$x_j^t(\xi^t) - \nu_j x_{\text{tot}}^t(\xi^t) = 0, \quad \forall j \in I \setminus \{\text{domestic bonds}\}, t = 0, \dots, T, \quad (8.1)$$

where x_j^t is the amount invested in asset class i at time t and ν_j is the prescribed fraction which we wish to invest in asset class i. We avoid introducing redundant equations by not having a constraint for domestic bonds. Domestic bonds are chosen as it is the asset class we are most likely to own. In the optimization we have constraints forcing the sum of our fractions to be slightly less than 1, thus forcing us always to invest a small amount in domestic bonds, in order to avoid numerical problems.

We may now evaluate a fixed mix by solving the problem to maximize (5.1) subject to (5.2) – (5.15), (8.1). We may view the optimal value of this problem as a function of ν and we denote this function by $z(\nu)$. The problem solved to find the best fixed mix strategy is hence

$$\text{minimize} \quad z(\nu), \tag{8.2}$$

$$\text{subject to} \quad \sum_{j \in I \setminus \{\text{domestic bonds}\}} \nu_j \leq 1 - \delta, \tag{8.3}$$

$$\nu_j \geq 0, \forall j \in I \setminus \{\text{domestic bonds}\}. \tag{8.4}$$

Here δ is chosen small (some magnitudes above the machine precision) to avoid numerical problems.

As we have very few degrees of freedom left in our model, all we get from solving the problem is an objective value for the given fixed mix, and a way to set the bonus rate of return. In order to find the optimal asset mix we employ a simple gradient search method. In order to do so we need a gradient of z with respect to ν.

From the optimal dual variables associated with (8.1) we can in certain cases get a gradient of z with respect to ν. In order to derive this gradient we assume that $x^t_{\text{tot}}(\xi^t) > 0, \forall t = 1, \ldots, T$, and hence that these variables are basic variables.

We look at a standard linear program

$$\begin{aligned} \text{minimize} \quad & c^{\mathrm{T}} x, \\ \text{subject to} \quad & Ax = b, \\ & x \geq 0. \end{aligned}$$

We know that for an optimal basic solution with corresponding basis B we will have the optimal value $z^* = c_B^{\mathrm{T}} B^{-1} b$ (with c_B^{T} being the cost coefficients for the basic variables). We further know that the columns of B are linearly independent, and hence that B is invertible. The reason for using this expression for the optimal solution value is that we wish to examine the effect on the objective value of small changes in the constraint matrix. If we disturb B by a small amount ΔB we may expand the inverse as

$$(B + \Delta B)^{-1} = \sum_{k=0}^{\infty} (-1)^k B^{-1} (\Delta B B^{-1})^k.$$

If we have $||\Delta B B^{-1}|| < 1$ we know that the right hand side will converge, and thus that this inverse exists. If we choose the disturbance ΔB small enough, so that $||\Delta B B^{-1}|| \ll 1$ we may disregard higher order terms and we have

$$(B + \Delta B)^{-1} \approx B^{-1} + B^{-1}\Delta B B^{-1}.$$

We write $z(\nu)$ for the optimal value of our LP when we have the prescribed fractions ν. We now change the coefficient ν_i by a small disturbance ϵ to get $\bar{\nu} := \nu + \epsilon e_i$, where e_i is the ith coordinate vector. This will change the coefficients of the basic matrix, as we know that the variables $x^t_{\text{tot}}(\xi^t)$ are basic. We denote this change by $\Delta B(\epsilon)$.

If we assume that $z(\nu)$ is a non-degenerate basic solution, we know that we may choose ϵ small enough to make the current optimal basis stay optimal. We further choose ϵ so small that $||\Delta B(\epsilon) B^{-1}|| \ll 1$ to justify neglecting higher order terms. The optimal value with the disturbance may now be expressed as

$$z(\nu + \epsilon e_i) := c_B^{\mathrm{T}}(B + \Delta B(\epsilon))^{-1} b,$$

and hence

$$z(\nu + \epsilon e_i) \approx c_B^{\mathrm{T}}(B^{-1} + B^{-1}\Delta B(\epsilon) B^{-1}) b.$$

This gives us further that

$$z(\nu + \epsilon e_i) - z(\nu) \approx c_B^{\mathrm{T}}(B^{-1}\Delta B(\epsilon) B^{-1}) b.$$

We may now identify the value of the different parts on the right hand side. We see that $c_B^{\mathrm{T}} B^{-1}$ corresponds to the value of the dual variables of our problem and $B^{-1} b$ corresponds to the value of the basic variables. We denote the dual variables by π and further denote the single dual variable associated with constraint (8.1) for outcome ξ^t_k of stage k by $\pi(\xi^t_k)$.

If we examine exactly where our changes occur in the basis matrix, we note that the constant ν_i occurs in the columns for $x^t_{\text{tot}}(\xi^t)$ and in the rows of the constraints (8.1).

Performing the matrix multiplication, we get

$$z(\nu + \epsilon e_i) - z(\nu) \approx \sum_{t=0}^{T} \sum_{k=1}^{n_t} \pi(\xi^t_k) \epsilon x^t_{\text{tot}}(\xi^t_k).$$

Letting ϵ go to 0 now gives us

$$\frac{\partial z(\nu)}{\partial \nu_i} = \lim_{\epsilon \to 0} \frac{z(\nu + \epsilon e_i) - z(\nu)}{\epsilon} = \sum_{t=0}^{T} \sum_{k=1}^{n_t} \pi(\xi_k^t) \epsilon x_{\text{tot}}^t(\xi_k^t).$$

From these partial derivatives we get a gradient of the objective function. We may now use this gradient to optimize the fixed mix strategy.

In the above calculations we have assumed that we have a non-degenerate optimal basis. If we do not have this assumption we do not get a correct gradient and may not obtain a descent direction. We use a simple descent method to optimize this problem to find the optimal mix. Although we are sure neither that the optimal basis is non-degenerate nor that the problem solved is convex, tests show that we converge to approximately the same point regardless of the choice of starting point.

To solve a large scale linear programming problem just to evaluate our objective function is naturally horribly inefficient, a fact we ignore as we are not overly concerned with computational efficiency at this point. The main advantage of using the LP formulation for function evaluation is that we get an equivalent way of determining the bonus rate of return for our two strategies.

8.2 QUESTIONS

After the tests we would like to be able to answer the following questions:

1. Does the stochastic solution approach outperform the fixed mix approach?
2. Does the size of the tree used significantly increase the performance?
3. Does optimization of the tree improve the results?
4. If we remove the check for arbitrage, will the performance of our model decrease? (See Subsection 6.2.c.)

8.3 NUMERICAL RESULTS

8.3.a STOCHASTIC PROGRAM VERSUS FIXED MIX

In order to answer whether the stochastic programming approach will outperform the fixed mix strategy, we run our 220 scenarios, solving both

Method	Mean (BSEK)	Std (BSEK)
Stochastic linear program	20.2640	5.0582
Fixed mix	20.0598	3.9614
difference	0.2042	1.5871

Table 8.1: Stochastic program versus fixed mix.

Method	Mean (BSEK)	Std (BSEK)
Four stage tree	20.3896	4.7381
Three stage tree	20.3038	5.0382
Difference	0.0858	0.6494

Table 8.2: Tree size.

for the fixed mix approach and the stochastic linear programming approach. The mean value of our merit function at the end of the simulation is given in Table 8.1.

If we look at the difference between the two methods (SLP-Fixed mix) on a scenario to scenario basis, the mean of this difference is 0.2042 in favour of the stochastic programming approach, and the standard deviation of the difference is 1.5871. Since we have a rather large number of simulations it is a fair approximation to assume that the average value of the difference is normally distributed with standard deviation $\frac{1.5871}{\sqrt{220}}$. Since we have $P(N(0, \frac{1.5871}{\sqrt{220}}) > 0.2042) = 0.0282$, we may reject the hypothesis that the SLP approach and the fixed mix approach are equivalent for our test-case.

8.3.b TREE SIZE

In order to see if a larger tree will give us a better result we increase the size of the tree using an extra stage with a length of 24 months. The resulting problem will have 102433 rows and 120234 columns. In this case we do not compare with the original three-stage run made so far. Instead we prune the four-stage tree to three stages and compare with the results obtained using the smaller tree. Hence the two cases we compare share identical information for three stages. The results from this run may be viewed in Table 8.2

Using the same normal approximation as above we get $P(N(0, \frac{0.6494}{\sqrt{220}}) > 0.0858) = 0.0250$, and we may reject the hypothesis that a larger tree does not improve the performance.

Method	Mean (BSEK)	Std (BSEK)
Optimized scenarios	20.2621	5.0473
Random scenarios	19.3156	6.1849
difference	0.9484	4.3753

Table 8.3: Fitting of tree.

Method	Mean (BSEK)	Std (BSEK)
SLP without arbitrage	20.2621	5.0473
SLP with arbitrage	20.3344	5.0488
Difference (per scenario)	0.0704	1.1095

Table 8.4: comparison arbitrage/no arbitrage.

8.3.c FITTING OF TREE

In order to test if our optimization of the tree does actually improve the performance of our model, we use sampling to generate trees of equal size via direct sampling from our model of the surrounding economy. The samples are generated using antithetic sampling, see Subsection 6.2.a. The results of this test-case is given in Table 8.3.

Testing as in the previous example we have $P(N(0, \frac{4.3753}{\sqrt{220}}) > 0.9484) = 0.0007$. We may hence reject the hypothesis that the methods are equivalent.

8.3.d ARBITRAGE

Here we run the same scenarios as in question 1, but in the scenario generation process we turn off the feature guaranteeing our scenario trees to be free from arbitrage. The results of this experiment is given in Table 8.4.

Contrary to intuition and the results in [29] where the presence of arbitrage decreased the performance of a SLP-based decision method, the method with arbitrage slightly outperforms the method where the trees are arbitrage-free. If we use the same normal assumption as above, we get $P(N(0, \frac{1.1095}{\sqrt{220}}) > 0.0704) = 0.1733$. We may hence not draw any statistically significant conclusions from this experiment.

8.3.e STABILITY OF THE SOLUTION

At the first stage in each simulation, all trees generated are generated from the same starting conditions. The only difference between our different

Method	SLP opt.	fix-mix opt.	slp rand. gen.
domestic bonds	7.3469	8.6856	5.6653
foreign bonds	0.0333	0.0014	2.0502
domestic stock	4.0873	3.3065	3.6849
foreign stock	1.001	0.4749	0.9890
domestic treasury-bills	0	0	0.0770

Table 8.5: Average values of first time investment.

Method	SLP opt.	fix-mix opt.	slp rand. gen.
domestic bonds	0.4368	0.1417	3.2236
foreign bonds	0.1508	0.0150	3.0097
domestic stock	0.7884	0.3680	2.0015
foreign stock	0.7360	0.3487	1.6972
domestic treasury-bills	0	0	0.5706

Table 8.6: Standard deviation of first time investment.

scenarios is hence the randomness included in the scenario generation process. If we were able to describe the possible outcomes in a representative way using our scenario-trees, there would be no difference in the first stage decision between two scenarios generated from the same prerequisites.

If we look at the initial asset mix (Table 8.5) and its standard deviation (Table 8.6), we see a much greater variance for the stochastic programming approach. This is only natural as the stochastic programming method has greater freedom to exploit the deficiencies in the scenario-tree. We have an even more varying starting solution for the random generation method, not surprisingly since the random generation procedure will give different expected yields for our assets with different random seeds.

In order to further test the impact of the random component of the tree generation procedure, we run one single test scenario 220 times, the only difference between the runs being the random seeds used to generate the trees. The scenario used is the test scenario which had the median merit function vaule in the first test case. When we run this single test case multiple times, we get an avreage merit function value of 20.7177 BSEK and an standard deviation of 0.7492 BSEK. This test shows us that the use of different seeds give a large variation in the utility function value. Hence we may conclude that the current tree-size and the current method of specifying the tree does not unambiguously represent the probability-function given from the economy model. If it had done so, different random seeds would not cause any difference in merit function value.

Chapter 9

Specialized solution methods

As may be noted from Figure 1.1 a linear stochastic programming problem has a great deal of structure which we may utilize to improve the efficiency of our calculations. A large number of different methods for doing this have been proposed.

In order to illustrate the principles of some decomposition techniques, we will use the two-stage stochastic linear program with recourse [a simplification of (1.11)]. In order to reduce the number of indices used, we denote the first stage variables by x and the second stage variables by y. The problem in question is

$$\begin{aligned}
\text{minimize} \quad & c_0^{\mathrm{T}}x + \mathsf{E}[c_1^{\mathrm{T}}y(\xi)], && (9.1a)\\
\text{subject to} \quad & W_0 x = h_0, && (9.1b)\\
& A_{0,1}(\xi)x + W_1(\xi)y(\xi) = h_1(\xi), && (9.1c)\\
& x, y(\xi) \geq 0, && (9.1d)
\end{aligned}$$

and as before we have the deterministic equivalent

$$\begin{aligned}
\text{minimize} \quad & c_0^{\mathrm{T}}x + \mathcal{Q}(x), && (9.2a)\\
\text{subject to} \quad & W_0 x = h_0, && (9.2b)\\
& x \geq 0, && (9.2c)
\end{aligned}$$

with $\mathcal{Q}(x) = \mathsf{E}_\xi(Q(x,\xi))$. The value of $Q(x,\xi)$ is computed as

$$
\begin{aligned}
&Q(x,\xi) = \\
&\underset{y}{\text{minimize}} \quad c_1(\xi)^{\mathrm{T}} y, \\
&\text{subject to} \quad A_{0,1}(\xi)x + W_1(\xi)y = h_1(\xi), \\
&\qquad\qquad\quad\; y \geq 0.
\end{aligned}
$$

with the convention of $Q(x,\xi) = \infty$ if the problem is infeasible. If we further assume that we have a finite number n of possible outcomes of our random variables of probability $\rho^i, i = 1, \ldots, n$, we may write the extensive form as

$$
\begin{aligned}
\text{minimize} \quad & c_0^{\mathrm{T}} x + \sum_{i=1}^{n} \rho^i {c_1^i}^{\mathrm{T}} y^i, \\
\text{subject to} \quad & W_0 x = h_0, \\
& A_{1,0}^i x + W_1^i y^i = h_1^i, \quad i = 1, \ldots, n, \\
& x \geq 0, y^i \geq 0, \quad i = 0, \ldots, n.
\end{aligned}
$$

9.1 DECOMPOSITION METHODS

As the name suggests these methods try to solve the problem by dividing it into subproblems. These methods may be divided into primal and dual methods. The primal methods split the problem according to time-stages, causing the later stages to decompose. In dual methods, we express the non-anticipativity constraints explicitly and relax these, causing the problem to decompose into one problem for each scenario when we optimize the dual problem.

9.1.a L-SHAPED METHOD

The L-shaped method of Van Slykes and Wets [42] is an extension of the Benders method [3], where the extensions deal with the feasibility of the second stage problem. This method tries to build a model of both the second stage feasibility set S and the value of the recourse problem, $\mathcal{Q}(x)$. We recall the definition of the second stage feasibility set from Chapter 1, $S = \{x | \mathcal{Q}(x) < \infty\}$.

We create a master problem

$$\begin{array}{ll} \text{minimize} & c_0^{\mathrm{T}}x + \theta, \\ \text{subject to} & W_0 x = h_0, \\ & D_l x \geq d_l, \quad i = 1, \ldots, r(k), \\ & E_l x + \theta \geq e_l, \quad i = 1, \ldots, s(k), \end{array}$$

which is solved to generate x^k and θ^k.

The vectors D_l and constants d_l represent feasibility cuts, and the vectors E_l and constants e_l represent optimality cuts. A feasibility cut is a cut with the property that, $Dx \geq d$ on S. An optimality cut is a supporting hyper plane of the function $\mathcal{Q}(\cdot)$.

The algorithm alternates between solving a master problem and evaluating $\mathcal{Q}(x)$, and works as follows:

0: Set $k := 0, s(k) := 0, r(k) := 0$.

1: Solve the master problem in order to get x^k. If $s(k) = 0$ we remove θ from the objective function. If the problem is infeasible we stop knowing the two-stage problem is infeasible. We assume that we get a finite solution x^k; if we get a direction of unboundedness we deal with it according to the description further below.

2: Evaluate $\mathcal{Q}(x^k)$. If $\theta^k = \mathcal{Q}(x^k)$, we have found an optimal solution and we quit. If $\mathcal{Q}(x^k) = -\infty$ we know that the problem is unbounded and we quit.

3: If we have $\mathcal{Q}(x^k) = \infty$ we generate a hyper plane $(E^{s(k)+1}, e^{s(k)+1})$ with the property that $E^{s(k+1)}x^k < e, E^{s(k)+1}x \geq e^{s(k)+1}, \forall x \in S$. We set $s(k+1) := s(k) + 1, r(k+1) := r(k)$, $k := k+1$ and go to step 1.

4: If we have $-\infty < \mathcal{Q}(x^k)$ we generate a supporting hyper plane to the function $\mathcal{Q}(\cdot)$, $(D^{r(k)+1}, d^{r(k)+1})$, such that $\mathcal{Q}(x) \geq d^{r(k)+1} - D^{r(k)+1}x, \forall x$ and $\mathcal{Q} = d^{r(k)+1} - D^{r(k)+1}x^k$. We set $s(k+1) := s(k), r(k+1) := r(k) + 1$, $k := k+1$ and go to step 1.

Details on the algorithm The function $\mathcal{Q}(x^k)$ is evaluated by solving the subproblems

$$
\begin{aligned}
\text{minimize} \quad & Q^i(x^k) = {c_1^i}^{\mathrm{T}} y^i, && (9.6a)\\
\text{subject to} \quad & W_1^i y^i = h_1^i - A_{1,0}^i x^k, && (9.6b)\\
& y^i \geq 0. && (9.6c)
\end{aligned}
$$

for all i, and computing $\mathcal{Q}(x) = \sum_{i=1}^n \rho^i Q^i(x)$. Let ν^i be the dual optimal solution to subproblem i. Duality theory gives us that $Q^i(x^k) = (\nu^i)^{\mathrm{T}}(h_1^i - A_{1,0}^i x^k), Q^i(x) \geq (\nu^i)^{\mathrm{T}}(h_1^i - A_{1,0}^i x), \forall x$. Summation gives $\mathcal{Q}(x^k) = \sum_{i=1}^n \rho^i (\nu^i)^{\mathrm{T}}(h_1^i - A_{1,0}^i x^k)$ and $\mathcal{Q}(x) \geq \sum_{i=1}^n \rho^i (\nu^i)^{\mathrm{T}}(h_1^i - A_{1,0}^i x), \forall x$. We may now identify our optimality cut for step 4 above as $E := \sum_{i=1}^n (\nu^i)^{\mathrm{T}} \rho^i A_{1,0}, e := \sum_{i=1}^n (\nu^i)^{\mathrm{T}} \rho^i h_1$.

Should one of the subproblems be infeasible, we may obtain a feasibility cut as follows:

Solve the problem

$$
\begin{aligned}
\text{minimize} \quad & w(x^k) = \bar{1}^{\mathrm{T}}(u + v), && (9.7a)\\
\text{subject to} \quad & W_1^i y^i + u^{\mathrm{T}} - v^{\mathrm{T}} = h_1^i - A_{1,0}^i x^k, && (9.7b)\\
& y^i, u, v \geq 0. && (9.7c)
\end{aligned}
$$

letting u and v be vectors of the appropriate length, and $\bar{1} = (1, \ldots, 1)^{\mathrm{T}}$.

This problem will have a solution with a positive value, as (9.6) is assumed to be infeasible. We name the dual optimal solution of this problem μ. Duality theory gives us that $w(x) \geq \mu^{\mathrm{T}}(h_1^i - A_{1,0}^i x), \forall x$. Thus any first stage solution for which this subproblem is feasible must fulfil $0 = w(x) \geq \mu^{\mathrm{T}}(h_1^i - A_{1,0}^i x)$ and hence that $\mu^{\mathrm{T}} A_{1,0}^i x \geq \mu^{\mathrm{T}} h_1^i$. Now we identify the feasibility cut as $D := \mu^{\mathrm{T}} A_{1,0}, d := \mu^{\mathrm{T}} h_1$.

Unbounded master problem solution If we encounter an unbounded solution when we solve the master problem there exists a ray $x = \bar{x} + \lambda \widehat{x}, \lambda \geq 0$ along which the objective function decreases towards $-\infty$. If $\bar{x}$ alone causes the second stage to be infeasible (that is if $\mathcal{Q}(\bar{x}) = -\infty$) we generate a cut for this point as previously. If $\mathcal{Q}(\bar{x}) > -\infty$, we solve the problem

$$
\begin{aligned}
\text{minimize} \quad & {c_1^i}^{\mathrm{T}} y^i, \\
\text{subject to} \quad & W_1^i y^i = -A_{1,0}^i \widehat{x}, \\
& y^i \geq 0.
\end{aligned}
$$

If it has a feasible solution for all outcomes, the ray is feasible in the second stage, as we know that both $h^i - A_{1,0}^i \bar{x}$ and $-A_{1,0}^i \widehat{x}$ belong to $\mathrm{pos}(W_1^i)$. If $c_0^{\mathrm{T}} x + \sum_{i=1}^n \rho^i c_1^i y < 0$ we conclude that the problem is unbounded, as the combined first and second stage objective function value will go to $-\infty$ as λ goes to ∞.

If $c_0^{\mathrm{T}} x + \sum_{i=1}^n c_1^i y \geq 0$ we add the cut $E = \sum_{i=1}^n (\nu^i)^{\mathrm{T}} A_{1,0}^i$, $e = \sum_{i=1}^n (\nu^i)^{\mathrm{T}} h_1^i$ as previously. We do not know that this is a strict supporting hyper plane, we do however know that $\mathcal{Q}(x) \geq e - Ex$, and that this hyper plane is parallel to a supporting hyper plane for $Q(\bar{x} + \lambda \widehat{x})$ as λ tends to ∞.

The case remaining to be dealt with is the case of the ray being infeasible for some subproblem for some λ.

We do now solve the problem

$$
\begin{aligned}
\text{minimize} \quad & w(x^k) = \bar{1}^{\mathrm{T}}(u + v), && \text{(9.9a)} \\
\text{subject to} \quad & W_1^i y^i + u^{\mathrm{T}} - v^{\mathrm{T}} = -A_{1,0}^i \widehat{x}^k, && \text{(9.9b)} \\
& y^i, u, v \geq 0. && \text{(9.9c)}
\end{aligned}
$$

which will have a positive solution. We denote the dual optimal solution as μ. We know that the optimal value of (9.9) is positive and hence that $\mu^{\mathrm{T}}(-A_{1,0}^i \widehat{x}^k) > 0$. Thus for large λ we will have $\mu^{\mathrm{T}}(h_1^i - A_{1,0}^i(\bar{x}^k + \lambda \widehat{x}^k)) > 0$ and if we set $D = \sum_{i=1}^n (\nu^i)^{\mathrm{T}} A_{1,0}^i$, $d = \sum_{i=1}^n (\nu^i)^{\mathrm{T}} h_1^i$ we have found a hyper plane separating S and $\bar{x}^k + \lambda \widehat{x}^k$ for large λ.

9.1.b L-SHAPED METHOD, MULTI-CUT VERSION

The multi-cut version works similar to the ordinary version. The difference is that in the ordinary L-shaped method we build up a model of $\mathcal{Q}(\cdot)$, while in the multi-cut version we build separate models of each $Q^i(\cdot)$.

Thus the master problem is formulated as

$$
\begin{aligned}
\text{minimize} \quad & c_0^{\mathrm{T}} x + \sum_{i=1}^{n} \theta^i, && (9.10a)\\
\text{subject to} \quad & W_0 x = h_0, && (9.10b)\\
& D_l x \geq d^j, \quad l = 1, \dots, r(k), && (9.10c)\\
& E_l^i x + \theta^i \geq e_l^i, \quad i = 1, \dots, n, l = 1, \dots, s(k,i). && (9.10d)
\end{aligned}
$$

with one θ^i for each outcome. The feasibility cuts are formed as earlier, and the optimality cuts are formed as $E^i = \nu^{\mathrm{T}} A_{1,0}^i \rho^i, e_i = \nu^{\mathrm{T}} h_1^i \rho^i$. We only add optimality cuts for the scenarios fulfilling $\rho^i Q^i(x) > \theta^i$, that is, where we do actually get new information.

The advantage of the multi-cut version of the L-Shaped method is that each solved master problem generates up to n cuts, thus providing us with more information that the single-cut version. The flip side of this is of course a more rapid growth in the number of cuts in the master problem.

9.1.c L-SHAPED METHOD, MULTISTAGE VERSION

The multistage version of the L-shaped method may be applied to stochastic linear programs having a staircase structure, which has be shown by Birge [5]. By a staircase structure we mean that $A(\xi_j)_{t,j} = 0$ for $j < t-1$. Worth noting is that any stochastic linear programming problem of the form of (1.15) may be transfered to a staircase structure by duplicating variables.

The method works identically to the two stage L-shaped method. In each stage t except the last stage we build a model of $\mathcal{Q}_{t+1}(x_{t+1})$ in the same manner as described above.

9.1.d L-SHAPED METHOD, REGULARIZED

The regularized version of the L-shaped algorithm adds a regularizing term to the objective function [37]. This change brings with it two advantages. It prevents the method from taking large steps in the beginning of the algorithm, thus avoiding searching regions which may be clearly infeasible or non-optimal when we take the second stage problem into consideration. In addition, the regularized version allows us to delete non-binding cuts from the problem without compromising the convergence of the algorithm. Deleting cuts keeps the master problem from growing to unmanageable sizes when the number of scenarios is large compared to the number of first-stage variables.

In this method the master problem (9.10) is augmented with a regularizing term, and we get the problem

$$
\begin{aligned}
\text{minimize} \quad & c_0^{\mathrm{T}} x^{k+1} + \sum_{i=1}^{n} \theta_i^{k+1} + \frac{1}{2} ||\bar{x}^k - x^{k+1}||^2, \\
\text{subject to} \quad & W_0 x^{k+1} = h_0, \\
& D_l x^{k+1} \geq d_l, \quad l = 1, \ldots, r(k), \\
& E_l^i x^{k+1} + \theta^{i,k} \geq e_l^i, i = 1, \ldots, n, l = 1, \ldots, s(k, l).
\end{aligned}
$$

We define $F(x) := c_0^{\mathrm{T}} x + \mathcal{Q}(x)$.

The method works as follows.

0: Find a feasible initial point $\bar{x}^0$. Set $k := 0, r(k) := 0, s(k, i) := 1, i = 1, \ldots, n$. Set the optimality-cuts, E_l^i, e_l^i to the cuts generated in this feasible point for $l = 1$. Let the set of feasibility-cuts be empty as in the previous method. Set $\gamma \in (0, 1)$.

1: Solve the augmented master problem (9.11). Make sure that the set of active constraints are linearly independent.

2: If $c_0^{\mathrm{T}} x^{k+1} + \sum_{i=1}^{n} \theta^{i,k+1} = F(x^k)$ we know that we have found the optimum point and we quit.

3: Solve the subproblem (9.6) for each scenario. If the scenario is feasible and $Q^i(x^{k+1}) > \theta^{i,k+1}$, add an optimality-cut as above. If it is not feasible, find and add a feasibility-cut as above.

4: If at least one subproblem was infeasible, set $\bar{x}^{k+1} = \bar{x}^k$ and go to step 7.

5. If $F(x^{k+1}) = c_0^{\mathrm{T}} x^{k+1} + \sum_{i=1}^{n} \theta^{i,k+1}$ set $\bar{x}^{k+1} = x^{k+1}$ and goto step 7.

6: If $F(x^{k+1}) \leq \gamma F(x^k) + (1 - \gamma)(c_0^{\mathrm{T}} x^{k+1} + \sum_{i=1}^{n} \theta^i)$ and we have n+N active constraints, set $\bar{x}^{k+1} = x^{k+1}$, else set $\bar{x}^{k+1} = \bar{x}^k$.

7: Delete constraints which were inactive in step 1 (hence no new constraints from step 3) from the optimality-cuts and feasibility-cuts, so that at most $2n + N$ members remain (N being the length of the vector x). Goto step 1.

9.1.e PROGRESSIVE HEDGING

Where the different versions of the L-shaped method work by decomposing the stages, solving different stages separately, the progressive hedging method works via decomposing the scenarios from each other. In order to do this we formulate the split variable version of the problem (9.1).

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} \rho^i c_0^{\mathrm{T}} x^i + \sum_{i=1}^{n} \rho^i {c_1^i}^{\mathrm{T}} y^i, && (9.12a)\\
\text{subject to} \quad & W_0 x^i = h_0, \quad i = 0, \dots, n, && (9.12b)\\
& A_{1,0}^i x^i + W_1^i y^i = h_1^i, \quad i = 0, \dots, n, && (9.12c)\\
& x^i = \sum_{j=1}^{n} \rho^j x^j, \quad i = 0, \dots, n, && (9.12d)\\
& x^i, y^i \geq 0, \quad i = 0, \dots, n. && (9.12e)
\end{aligned}$$

Here we have different first stage decision variables for each scenario, we enforce the non-anticipativity explicitly via (9.12d). We simplify notation by writing $\bar{x} = \sum_{i=1}^{n} \rho^i x^i$.

In order to solve this problem we relax the constraint (9.12d) with Lagrange multipliers π and we form the augmented Lagrangian problem

$$\begin{aligned}
\max_{\pi} \min_{x^i, y^i} \quad & \sum_{i=1}^{n} \rho^i c_0^{\mathrm{T}} x^i + \sum_{i=1}^{n} \rho^i {c_1^i}^{\mathrm{T}} y^i + \sum_{i=1}^{n} (\pi^i)^{\mathrm{T}} (x^i - \bar{x}) + \frac{r}{2} ||x^i - \bar{x}||^2, \\
& && (9.13a)\\
\text{subject to} \quad & W_0 x^i = h_0, && (9.13b)\\
& A_{1,0}^i x + W_1^i y^i = h_1^i, \quad i = 1, \dots, n, && (9.13c)\\
& x^i, y^i \geq 0, \quad i = 0, \dots, n, && (9.13d)
\end{aligned}$$

where we have added the regularizing term $\frac{r}{2}||x^i - \bar{x}||^2$ to the objective. This problem may be solved using an ascent method. We have however not yet decomposed the different scenarios as they are are coupled via $\bar{x}$. The progressive hedging algorithm solves this by using older values for $\bar{x}$. Using this assumption the Lagrangean function from (9.13) will separate into the subproblems

$$\min_{x^i,y^i} \quad \rho^i c_0^{\mathrm{T}} x^{i,k} + \rho^i {c_1^i}^{\mathrm{T}} y^{i,k} + (\pi^{i,k-1})^{\mathrm{T}}(x^{i,k} - \bar{x}^{k-1}) + \frac{r}{2}||x^{i,k} - \bar{x}^{k-1}||^2, \tag{9.14a}$$

$$\text{subject to} \quad W_0 x^{i,k} = h_0, \quad i = 1,\dots,n, \tag{9.14b}$$

$$A_{1,0}^i x^{i,k} + W_1^i y^i = h_1^i, \quad i = 1,\dots,n, \tag{9.14c}$$

$$x^{i,k}, y^{i,k} \geq 0, \quad i = 1,\dots,n. \tag{9.14d}$$

The algorithm works as follows:

0: Assume we have some nonanticipative first stage solution $\bar{x}^0$ with feasible second stages, some initial multiplier $\pi^{i,0}$ and $r > 0$. Let $k = 0$.

1: Solve the problems (9.14) for each $i = 1,\dots,n$ to obtain $x^{i,k+1}$. Set $\bar{x}^{k+1} = \sum_{i=1}^n \rho^i x^{i,k+1}$.

2: Update the multipliers as $\pi^{i,k+1} = \pi^{i,k} + r(x^{i,k+1} - \bar{x}^{k+1})$. If $\bar{x}^{i,k} = \bar{x}^{i,k+1}$ and $\pi^{i,k} = \pi^{i,k+1}$ stop, the solution is optimal. Otherwise set $k = k+1$ and go to 1.

Thus the algorithm works by pulling the stage 1 solutions of different scenarios together. The algorithm may be extended to a multi stage setting in a straight forward way by relaxing the non-anticipativity constraints of each stage.

9.2 NON-DECOMPOSITION METHODS

The structure of the extensive form may be exploited not only by decomposition methods, but also when we try to solve the extensive form directly. Some effort has gone into this by using specialized basis factorisation methods for the simplex method.

Others have exploited this structure when solving via interior point methods. When solving via interior point methods, the main work is to factorize the matrix ADA^{T}, with A being the constraint matrix of the extensive form and D being a diagonal matrix. Lustig, Mulvey and Carpenter [33] has employed a split variable formulation, which will increase the size of A but make the combined matrix sparser, thus reducing the workload.

The system ADA^{T} is generated when we try to solve the KKT-system in an interior point method. Recently, new methods have been developed to solve this system recursively. Steinbach [40, 41] has developed a method where leaves of the scenario tree are systematically eliminated until we may solve for the first stage variables. The values of the first stage variables are used to propagate the solution back down the tree. Thus instead of having to factor one large system we factor one system for each node in the tree. The size of these smaller systems is determined by the number of variables and constraints in each stage. A similar method has been developed in [7, 8] where dynamic programming is used to find Newton-steps for an interior point algorithm.

9.3 APPLICATION TO OUR PROBLEM

The progressive hedging method has proven to be rather slow for linear problems; its main advantage is that it works equally well for non-linear problems. Hence this method is not so interesting for our model, but may prove interesting if we are to introduce nonlinear constraints directly into the problem in the future. More interesting are the L-shaped method and the recursive interior point methods. The principal advantage of the regularized L-shaped method is the ability to remove cuts without jeopardizing the convergence of the algorithm. As we in this study have a small number of subproblems compared to the number of variables, this advantage is not critical. Thus we find it unlikely that employing the regularized version of the L-shaped method would perform better than the ordinary L-shaped method. The recursive methods perform best for small nodal problems. Assuming the number of constraints in each nodal problem is n we have to factor a $n \times n$ matrix requiring in the order of n^3 operations (assuming dense treatment of the matrices). If the nodal problems double in size we get an eightfold increase in the computational effort. If we at the same time reduce the number of nodal problems with a factor of 2 to keep the total problem size constant, we would get an increase in the solution-time by a factor 4. In adition we expect the numerical problems due to ill-conditioning of the involved matrices to become worse when the matrix size increases. The method's performance should however not decrease due to an increasing number of stages. In the L-shaped method we have to communicate information between the time-stages using cuts, which seems to indicate that the L-shaped method will perform worse with increasing number of stages. Thus the L-shaped method should have a comparative advantage when we have few stages with large nodal problems.

The problem in this work has approximately 50 variables and constraints per stage, significantly more than in [40, 41] and [7, 8]. Further, we currently have relatively few stages. This suggests that the L-shaped method may prove to be a good choice for our problem, especially as we may easily reformulate the problem to have fixed recourse in the last stage.

We do however plan to add quadratic penalties to the model, making the recursive methods more appealing. In addition, quadratic penalties would reduce the number of penalty variables as we would not need several penalties to get progressively steeper penalties.

Chapter 10

Conclusions and further work

To conclude, we have built a asset liability system which with the current model of a surrounding economy outperforms a fixed mix strategy. We have not found any evidence that arbitrage possibilities in the scenarios provided to the system will skew the solutions to make the system perform worse. Increasing the size of the problem solved will give us better solutions, although we feel that the marginal improvement of adding extra stages will decrease, as these extra stages will have less and less impact on the first stage solution.

We have further found that different scenario-trees having similar statistical properties will give significantly different solutions. We may hence conclude that we have not captured enough of the properties of the model in our scenario-trees to get consistent results. If we had specified the scenario-trees in a satisfactory way, two different scenario-trees generated to have the same statistical properties would produce the same proposed strategy. We must however stress that removing randomness from the scenario generation process is not the answer, such a solution will only hide our problems, as we will get solutions with a consistent bias. The current approach will at least let us estimate the size of this error.

10.1 FURTHER WORK

10.1.a BETTER MODEL OF SURROUNDING ECONOMY

The current model of the surrounding economy is rather crude, specifically the coupling between bond prices and interest rates should be improved by actually solving the bond-pricing equation. We further need to correct the fitted model to better correspond to an analysts expectation of the future, instead of simply fitting historical data. If this is not possible, we should do a more thorough fitting of our model to historical data, using longer time-series. We further need to improve how the base rate is set in the model, which would require more information on how FI sets this rate in the real world.

10.1.b BETTER PENALTIES

Currently the penalties are not specified in a systematic way. The penalties should be made to reflect the risk-tolerance of the board. This may be done in a number of ways, we may ask a number of questions of the type found in the description of utility in Chapter 1, or we may use an iterative process where the a number of scenarios are run repeatedly, and after each run the user may change the penalties to trade expected return for lower probability of breaking requirements or vice versa.

10.1.c LINEAR QUADRATIC MODEL

As we have found that many penalties at different levels are needed to make the model perform well, we would like to extend the model to a stochastic programming problem with linear constraints and linear and quadratic penalty terms. This would reduce the number of penalties and constraints and hence make the problem easier to solve. If we were to utilize an interior point algorithm, such a change would probably not significantly increase the computational burden of our problem.

10.1.d DECOMPOSITION SOLVERS

We would like to try out different types of decomposition solvers, as described in section Chapter 9. The primary candidates for implementation would be the multi-stage L-shaped method and a recursive interior point solver. Using a decomposition solver would probably make it possible to extend the model using more time-stages, both increasing the resolution in time and the horizon of the model.

10.1.e POLICY OPTIMIZATION

The solution of large scale LP-s for this kind of problem has a major drawback. In a scenario-tree, the decisions in one branch is more or less independent of the decisions in other branches of the tree. Even if it is possible that different nodes in a tree end up in approximately the same state of the world, they do not share information. This forces us to introduce great redundancy in the tree, as the descendants of each node must provide a more or less complete description of all possible outcomes of our stochastic processes. A quick and dirty remedy to this problem would be to enforce artificial trading restrictions. If we add constraints on how much of our assets may be traded in between stages, we will let nodes at later stages communicate via nodes higher up in the tree. In the most extreme case of not allowing any trade at all, we will in effect combine several stages into one, and hence reduce our tree to a tree with fewer stages, but more branches at each stage.

The fixed mix approach used is a simple way of sharing information between the different branches of the scenario tree, as we use the same asset mix at all nodes. It is however a very crude policy. What we wish the stochastic programming problem to do is to define a mapping from a state of the world to a decision. All the generation of scenario-trees and solving of stochastic problems amounts to nothing more than this. An alternative to using the current approach would be to construct a class of parametrized policies which map from the current state of the world into the set of possible actions. We would then optimize the parameters of this class in the same fashion as we did with the fixed mix approach. As a simple class of such policies, we may let the fractions of assets owned after trading be linearly dependent on the consolidation of our company.

Optimizing such a strategy over a tree will make sure that information is shared between nodes which are siblings, that is nodes with a common parent node. If the siblings share information, we do not need a good description of all possible outcomes in each node, as the lateral coupling will prevent us from using only local information.

The technique of optimizing a policy is used by Mulvey, Gould and Morgan in [35]. Optimizing policies of decisions instead of optimizing the decisions directly will in most cases lead to nonlinear non-convex optimization problems, problems which are hard to solve. The technique of optimizing policies governing decisions instead of optimizing the decisions directly seem to be less prevalent than the stochastic linear programming approach in the literature. A reason for this may be that policy optimization has less structure and is more complex than stochastic linear programming over a scenario tree, and hence seem less tempting to algorithmically inclined academic researchers.

In order to implement policy-based decision strategies efficiently we would need to implement a better way of obtaining the value of a policy. The current method of solving an LP is far too inefficient. Further, with the current way of defining penalties, the value of a policy over a scenario will be a non-smooth function, forcing us to use techniques to deal with this. We currently do some work in this area, with one student implementing a policy-based system for his masters thesis.

10.1.f COMPANIES WITHIN THE COMPANY

In the current model, we only trade at specified points in time. After the trade is done, we do not alter our assets until the next decision point. During this time the price development of our assets may drive the mix owned far from the mix chosen at the previous decision point. This reduces the length we may have on our stages, as rebalancing is necessary to keep the solution efficient. A possible remedy to this problem would be to create fictitious sub-companies within our company. Each of these sub companies would follow a predefined strategy as similar to the ideas presented above. The optimization model would then allocate the companies assets to these sub companies, investing x% of the assets in strategy A, y% in strategy B and so forth. Once the optimal solution if found, we may create the resulting linear combination of strategies and add them as a new sub-company. Iteratively doing this would allow us to find a good policy without having to resort to computationally more difficult nonlinear programming.

Appendix A

Bond pricing

In the original Brennan-Schwarts model [11], we have two state variables giving us the entire yield curve of bonds. These are r, the instantaneous rate of return and l, the return on a consol bond (a bond of infinite maturity). The stochastic processes defining the model are

$$\begin{aligned} dr =& \beta_1(r,l,t)dt + \eta_1(r,l,t)dz_1 \\ dl =& \beta_2(r,l,t)dt + \eta_2(r,l,t)dz_2 \end{aligned}$$

with t being time, dz_1, dz_2 being two Wiener processes correlated by a factor ρ, and β and η are general functions.

The price of a bond of maturity τ with continuous coupon c may now be expressed as a function of r and l, and we denote this price by $B(r,l,\tau,c)$. We let subscripts denote partial derivatives so that $B_l := \frac{\partial B(r,l,\tau)}{\partial l}$.

In order to find the relative price change of this bond over a short interval of time we apply Ito's [36] lemma and obtain

$$\frac{dB + cdt}{B} = \mu dt + \frac{B_r}{B}\eta_1 dz_1 + \frac{B_l}{B}\eta_2 dz_2,$$

with μ defined by

$$\mu = \frac{B_r\beta_1 + B_l\beta_2 + B_{rr}\eta_1^2/2 + B_{ll}\eta_2^2/2 + B_{rl}\rho\eta_1\eta_2 - B_\tau + c}{B}.$$

The absence of arbitrage now gives us that

$$\mu - r = \lambda_1\frac{B_r}{B}\eta_1 + \lambda_2\frac{B_l}{B}\eta_2,$$

where $\lambda_{1,2}$ is the market price of instantaneous rate risk and consol rate risk respectively.

The price of a consol bond with a continuous coupon payment of $c = 1$ is known to be l^{-1} which is differentiable for $l \neq 0$. Substitution of the partial derivatives of this price into the equations above then give us

$$\lambda_2 = \frac{-\eta_2}{l} + \frac{\beta_2 - l^2 + r_l}{\eta_2}.$$

Substitution of this expression for the market price of consol bond rate risk into the previous equations will give us

$$\begin{aligned}&B_{rr}\eta_1^2/2 + B_{rl}\rho\eta_1\eta_2 + B_{ll}\eta_2^2/2 + B_r(\beta_1 - \lambda_1\eta_1) + \\ &B_l(\eta_2^2/l + l^2 - rl) - \beta_\tau + c - Br = 0.\end{aligned}$$

This partial differential equation is the bond pricing equation for our problem, which is what we should solve to find the bond price. The price must also fulfil suitable boundary conditions, such as having face value at maturity.

Appendix B

PLAM example

In order to facilitate the formulation of stochastic linear programming problems, we have made alterations to an existing modelling language with open source, PLAM, by Barth and Bockmayr [2].

In order to demonstrate how the language works, we present a short example taken from [6].

In this example we are to invest money in two different assets to meet a known goal in the future. The outcomes of our investments are unknown, and we may reinvest our money over time. The problem is divided into three periods (four stages). In the first three stages we invest or reinvest our money, and in the fourth we pay a penalty for not meeting the requirement, or get a reward for exceeding the target. We assume that we may invest our money in stocks and bonds. After each investment period the world may develop favourably or unfavourably. In the favourable case, one SEK invested in stocks will yield 1.25 SEK, and one SEK invested in bonds will yield 1.14 SEK. In the unfavourable case, the yields will be 1.06 for stocks and 1.12 for bonds.

Example model

With the following notation

I	Set of investments (stock,bond),
N	Number of investment occasions,
$T = 0..N-1$	Set of time-steps,
G	Initial wealth,
H	Target wealth at end of horizon,
b	Premium for exceeding target,
p	Penalty for failing to reach target,
$x(i,t)$	Amount invested in asset i at time t,
$s(i,t)$	Yield of asset i between time t and t+1,
h	Shortfall,
u	Surplus,

this problem may be formulated as

$$\begin{aligned} \text{maximize} \quad & ub - hp, \\ \text{Subject to} \quad & G = \sum_{i \in I} x(i,0), \\ & \sum_{i \in I} x(i,t) = \sum_{i \in I} s(i,t-1)x(i,t-1), \quad \forall t \in T \setminus \{0\}, \\ & \sum_{i \in I} s(i,N-1)x(i,N-1) + h - u = H. \end{aligned}$$

Model formulation in PLAM

When we express this in PLAM the syntax will be:

```
:-plam.

set investments.
```

```
set periods.
set sucperiods.
set invperiods.

param first_period.
param last_period.

param initwealth.
param target.
param reward.
param penalty.

param return:[investments,sucperiods].

variable above_target:- >=0.
variable below_target:- >=0.
variable amtinvest:[investments,invperiods]:- >=0.

period(above_target,P):-last_period(P).
period(below_target,P):-last_period(P).

s_period(A,A):-periods(A).

objective max:profit:-
        reward*above_target - penalty*below_target.

subject_to budget:-
        sum((investments(I),first_period(P)),amtinvest(I,P))=initwealth.

subject_to balance:-
        forall((invperiods(P),not(first_period(P)),PP is P-1),
             sum(investments(I),
             return(I,P)*amtinvest(I,PP)-amtinvest(I,P))=0).

subject_to scenario_value:-
        sum((investments(I),last_period(P),PP is P-1),
                amtinvest(I,PP)*return(I,P))
            -above_target+below_target=target.
```

The syntax may seem strange at a first glance, a consequence of all statements being valid prolog-statements. It does however quite readily relate to other modelling languages. Some shortcomings of PLAM become apparent. One may not use derived sets directly in parameter and variable definitions, which makes it necessary to introduce the sets `invperiods` and `sucperiods`. Further, we may only index with variables, not with expressions. This is why we use the variable `pp` above instead of simply writing `amtinvest(I,P-1)`. All statements above are normal PLAM statements described in [2] except the clauses

```
period(above_target,P):-last_period(P).
period(below_target,P):-last_period(P).
s_period(A,A):-periods(A).
```

The period clauses state that the variables `above_target` and `below_target` belong to the last time period.

A variable or parameter that does not have an explicit period statement associated with it may inherit its period property from the indexing. The clause

```
s_period(A,A):-periods(A).
```

states that a variable or parameter indexed by a member of the set periods will be given the value of that member as its period, i.e., a variable indexed by the number 0 will belong to period 0 (regardless of which set 0 belongs to, a flaw in the program). This makes sense only if the members of the set periods are integers. As a result the variable `amtinvest(stock,0)` will belong to period 0 and so forth. The explicit designation of the period overrides periods inherited by indexing. If a period is not supplied in any way a variable is assumed to belong to the first period.

Data in PLAM

All nonrandom data and set definitions are kept in a separate data file.

The syntax of the data file for our example is

```
:-plam.

investments(I):-member(I,[stock,bond]).

periods(A):-member(A,[0,1,2,3]).

first_period(0).
last_period(3).
```

```
invperiods(P):-periods(P),not(last_period(P)).
sucperiods(P):-periods(P),not(first_period(P)).

initwealth(55).
target(80).

reward(1).
penalty(4).

return(stock,1,ret_stock_1).
return(stock,2,ret_stock_2).
return(stock,3,ret_stock_3).

return(bond,1,ret_bond_1).
return(bond,2,ret_bond_2).
return(bond,3,ret_bond_3).

mark_random(ret_bond_1,1.13).
mark_random(ret_bond_2,1.13).
mark_random(ret_bond_3,1.13).

mark_random(ret_stock_1,1.155).
mark_random(ret_stock_2,1.155).
mark_random(ret_stock_3,1.155).
```

The statement `initwealth(55)` simply states that the parameter initwealth has the value 55. The statement `return(bond,1,ret_bond_1)` states that the parameter return(bond,1) has the random value ret_bond_1, and the statement `mark_random(ret_bond_1,1.13)` tells us that a typical value of this random variable is 1.13. These values will be used when generating the core problem. In the stochastic problem this value will be replaced by the random outcomes.

In order to define the random values, we write a file giving data for each node in the scenario-tree.

```
node(1,0,1,2,[
]).

node(2,1,0.5,2,[
ret_bond_1-(1.14),
ret_stock_1-(1.25)
]).
```

```
node(3,2,0.25,2,[
ret_bond_2-(1.14),
ret_stock_2-(1.25)
]).

node(4,3,0.125,0,[
ret_bond_3-(1.14),
ret_stock_3-(1.25)
]).

node(5,3,0.125,0,[
ret_bond_3-(1.12),
ret_stock_3-(1.06)
]).

node(6,2,0.25,2,[
ret_bond_2-(1.12),
ret_stock_2-(1.06)
]).
.
.
.
.
.

node(15,3,0.125,0,[
ret_bond_2-(1.12),
ret_stock_2-(1.06)
]).
```

Each node in a scenario tree is listed separately. If we look at the first line of each node definition, the first number is the ordering number of the node. This number will not influence the output in any way, it is only used to trace errors to specific positions in the data-file. The second number indicates the stage of the node. The third number is the (unconditional) probability of ending up in a node. The fourth is the number of children of each node. The ordering of the nodes is a depth first search of the scenario tree. How the nodes are ordered may be viewed in Figure B.1.

The program may be used in two modes. In the first mode, writing the command

```
write_smps(outname,ranfilename)
```

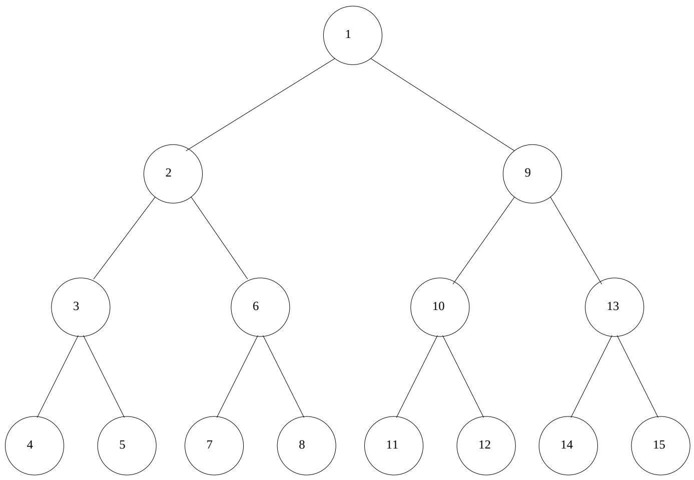

Figure B.1: Tree structure from the example

will produce a set of SMPS-files (see [4]) describing the problem. Variable and constraints will have alias names (`X1, X2,` ... and `C1, C2,` ...) as the SMPS format has a maximum of eight characters per name, a limit quickly exceeded since we need to include the indexing in the variable names. A separate file (`outname.alias`) is written connecting these aliases to the true names.

The second command

```
write_problem(probname).
```

will not use any random data. Instead it will give algebraic expressions for all matrix and vector elements describing the problem. These may be used later on for custom made solutions where problems are generated either via supplying random data from databases or other sources, or when the random data is generated on the fly from random number generators. This may be beneficial if the problem is so large that we may not keep it in the working memory of the computer at all times. It will then be enough to save the random seed used to create a subtree, a technique used in [23].

Bibliography

[1] Försäkringsrörelselag 1982:713 (in swedish).

[2] BARTH, P. & BOCKMAYR, A. Modelling mixed-integer optimization problems in constraint logic programming. Technical report, Max-Planck-Institut für Informatik, Saarbrücken, Germany, 1995.

[3] BENDERS, J. Partitioning procedures for solving mixed variables programming problems. *Numeriche Mathematik* , 238–252, 1962.

[4] BIRGE, J., DEMPSTER, M., GASSMAN, H., GUNN, E., KING, A., & WALLACE, S. A standard input format for multiperiod stochastic linear programs. *COAL Newsletter* , 1–19, 1987.

[5] BIRGE, J. R. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research* **33**(5), 989–1007, 1985.

[6] BIRGE, J. R. & LOUVEAUX, F. *Introduction to Stochastic Programming*. Springer-Verlag, Berlin, 1997.

[7] BLOMWALL, J. A Riccati-based primal interior point solver for multistage stochastic programming. Technical Report LiTH-MAT-R-2000-28, Department of Mathematics, Linköping University, Linköping, Sweden, 2000.

[8] BLOMWALL, J. A Riccati-based primal interior point solver for multistage stochastic programming - extensions. Technical Report LiTH-MAT-R-2000-30, Department of Mathematics, Linköping University, Linköping, Sweden, 2000.

[9] BOK, J.-K., LEE, H., & PARK, S. Robust investment model for long range capacity expansion of chemical processing networks under uncertain demand forecast scenarios. *Computers and Chemical Engineering* **22**(7-8), 1037–1049, 1998.

[10] BRENNAN, M. J. & SCHWARTZ, E. S. Analyzing convertible bonds. *Journal of Financial and Quantitative Analysis* **15**(4), 907–929, 1980.

[11] BRENNAN, M. J. & SCHWARTZ, E. S. An equilibrium model of bond pricing and a test of market efficiency. *Journal of Financial and Quantitative Analysis* **17**(3), 301–329, September 1982.

[12] CARØE, C. & SCHULTZ, R. A two-stage stochastic program for unit commitment under uncertainty in a hydro-thermal system. Technical Report 13, DFG-Schwerpunktprogramm "Echtzeit-Optimierung grosser Systeme", 1998.

[13] CARIÑO, D. R., MYERS, D. H., & ZIEMBA, W. T. Concepts, technical issues, and uses of the Russel-Yasuda Kasai financial planning model. *Operations Research* **46**(4), 450–462, 1998.

[14] CARIÑO, D. R. & ZIEMBA, W. T. Formulation of the Russell-Yasuda Kasai financial planning model. *Oper. Res.* **46**(4), 433–449, 1998.

[15] CARIÑO, D. R., ZIEMBA, W. T., KENT, T., MYERS, D. H., STACEY, C., SYLVANUS, M., TURNER, A. L., & WATANABE, K. The Russel-Yasuda Kasai model: An asset/liability model for a Japanese insurance company using multistage stochastic programming. *Interfaces* **24**(1), 29–49, 1994.

[16] CHANG, K., KAROLY, A., LONGSTAFF, F., & SANDERS, A. An empirical comparison of alternative models of the short-term interest rate. *The Journal of Finance* **47**(3), 1209–1225, 1992.

[17] CONSIGLI, G. & DEMPSTER, M. A. H. *Worldwide Asset and Liability Modeling*, chapter 19. The CALM Stochastic Programming Model for Dynamic Asset-Liability Management, pages 464–500. Cambridge University Press, 1998.

[18] COX, J., INGERSOLL, J., & ROSS, S. An intertemporal general equilibrium model of asset prices. *Econometrica* **53**(2), 363–384, 1985.

[19] COX, J., INGERSOLL, J., & ROSS, S. A theory of the term structure of interest rates. *Econometrica* **53**(2), 385–407, 1985.

[20] DERT, C. L. *Worldwide Asset and Liability Modeling*, chapter 20. A Dynamic Model for Asset Liability Management for Defined Benefit Pension Funds, pages 501–536. Cambridge University Press, 1998.

[21] EPPEN, G., KIPP, M., & SCHRAGE, L. A scenario approach to capacity planning. *Operations Research* **37**(4), 517–527, 1989.

[22] FOURER, R. & GAY, D. M. Stochastic programming using AMPL. Presented at ISMP 97, http://www.ampl.com/cm/cs/what/ampl/NEW/FUTURE/stoch.html.

[23] GONDZIO, J. & KOUWENBERG, R. High performance computing for asset liability management. Technical report, Department of Mathematics and Statistics, The University of Edinburgh, 1999. MS-99-004.

[24] HIGLE, J. L. Variance reduction and objective function evaluation in stochastic linear programs. *INFORMS Journal on Computing* **10**(2), 236–247, 1998.

[25] HULL, J. *Options Futures and Other Derivative Securities*. Prentice Hall, 1993.

[26] HØYLAND, K. *Asset Liability Management for a Life Insurance Company: A Stochastic Programming Approach*. PhD thesis, Department of Economics and Technology Management, Norwegian University of Science and Technology, Trondheim, Norway, 1998.

[27] KALL, P. & JANOS, M. SLP-IOR: An interactive model management system for stochastic linear programs. *Mathematical Programming* **75**, 221–240, 1996.

[28] KEEFER, D. Certainty equivalents for three-point discrete-distribution approximations. *Management Science* **40**, 760–773, 1994.

[29] KLAASSEN, P. Discretized reality and spurious profits in stochastic programming models for asset liability management. *European Journal of Operational Research* , 374–392, 1997.

[30] KLEBANER, F. *Introduction to Stochastic Calculus with Applications*. Imperial College Press, 1998.

[31] KOUWENBERG, R. Scenario generation and stochastic programming models for asset liability management. To appear in European Journal of Operational Research, 1998.

[32] LUENBERGER, D. *Investment Science*, chapter 7. The capital asset pricing model, pages 173–190. Oxford University Press, 1998.

[33] LUSTIG, I. J., MULVEY, J. M., & CARPENTER, T. J. Formulating two-stage stochastic programs for interior point methods. *Operations Research* **39**(5), 757–770, 1991.

[34] MESSINA, E. & MITRA, G. Modelling and analysis of multistage stochastic programming problems, a software environment. *European Journal of Operational Research* , 343–359, 1997.

[35] MULVEY, J. M., GOULD, G., & MORGAN, C. An asset and liability management system for Towers Perrin-Tillinghast. *Interfaces* **30**, 96–114, 2000.

[36] REBONATO, R. *Interest-rate Option Models*. Wiley & Sons, 1996.

[37] RUSZCZYNSJI, A. A regualrised decomposition method for minimizing a sum of polyhedral functions. *Mathematical Programming* **35**, 309–333, 1986.

[38] SEN, S., DOVERSPIKE, R., & COSARES, S. Network planning with random demand. *Telecommunication systems* , 11–30, 1993.

[39] SPELLUCCI, P. DONLP2 users guide. Technical report, Department of Mathematics, Technical University at Darmstadt, 1995.

[40] STEINBACH, M. C. Recursive direct algorithms for multistage stochastic programs in financial engineering. Technical report, ZIB, 1998.

[41] STEINBACH, M. C. Recursive direct optimization and successive refinement in multistage stochastic programs. Technical report, ZIB, 1998.

[42] VAN SLYKE, R. M. & WETS, R. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM journal on applied mathematics* , 638–663, 1969.

[43] VON NEUMANN, J. & MORGENSTERN, O. *Theory of Games and Economic Behaviour*. Princeton University Press, third edition, 1953.