Mathematical
Sciences

Economics
and Finance

Electronic
and Computer
Engineering

CARISMA

Centre for the Analysis of Risk and
Optimisation
Modelling Applications

A multi-disciplinary research centre focussed on understanding,
modelling, quantification, management and control of RISK

Human
Sciences

Systems
Engineering

Centre for
Environmental Research

TECHNICAL REPORT

Stochastic Programming and Scenario Generation within a Simulation Framework: An Information Systems Perspective

Nico Di Domenica, George Birbilis,
Gautam Mitra, Patrick Valente.

CARISMA

Department of Mathematical Sciences and Department of
Economics and Finance
Brunel University, Uxbridge, Middlesex, UB8 3PH
www.carisma.brunel.ac.uk

BRUNEL
UNIVERSITY

CARISMA

Table of Contents

Abstract

Stochastic Programming brings together models of optimum resource allocation and models of randomness to create a robust decision making framework. The models of randomness with their finite, discrete realisations are called Scenario Generators. In a compendium report we have considered the modelling perspective of Scenario Generation and its integration within Stochastic Programming. In this paper we investigate the role of such a tool within the context of a combined information and decision support system. We analyse the roles of decision models and descriptive models, and also examine how these can be integrated with data marts of analytic organisational data and decision data. Recent developments in On-Line Analytical Processing (OLAP) tools and multidimensional data viewing are taken into consideration. We finally introduce illustrative examples of optimisation, simulation models and results analysis to explain our multifaceted view of modelling.

Keywords : Scenario Generation, Stochastic Programming, DSS , OLAP.

1 Introduction and Motivations

1.1 Role of Business Analytics within organisational IS

Two basic modelling paradigms come together in stochastic programming (SP). These are : (a) model of optimum resource allocation and (b) model of randomness respectively. It is well established that in the realm of OR/MS and its contribution to managerial decision making four categories of models are of interest. For a detailed discussion see Mitra (1988) :

- *Descriptive Models* as defined by a set of mathematical relations which simply predicts how a physical, industrial or a social system may behave.

- *Normative Models* constitute the basis for (quantitative) decision making by a superhuman following an entirely rational, that is, logically scrupulous set of arguments. Hence quantitative decision problems and idealised decision makers are postulated in order to define these models.

- *Prescriptive Models* involve systematic analysis of problems as carried out by normally intelligent persons who apply intuition and judgement. Two distinctive features of this approach are uncertainty analysis and preference (or value or utility) analysis.

- *Decision Models* are in some sense a derived category as they combine the concept underlying the normative models and prescriptive models.

Within the organisational context, the deployment of such models and their role in the IS infrastructure is recognised as "Business Analytics". Optimisation based Decision Support Systems (DSS) are assuming an increasingly important role in many industries and public organisations. Optimisation modelling and solution techniques have seen sustained developments from the early sixties until now. The main achievements have come from the improvements in model conceptualisations, solution algorithms and software techniques. Optimisation models are considered as critical components of an organisation's analytical information technology (IT) systems, as they are used to analyse and control critical business measures such as cost, profit, quality, and time. In the context of organisational information systems (IS) (Koutsoukis and Mitra, 2003) quantitative models constitute an important component within the information value chain (IVC). Typically, transactional (operational) data which is available in a firm is analysed and synthesised to create analytic data or information, which is the first link of the information value chain. The analytic data is stored in data marts, which are subsets of the overall collection of data within a firm and is called data warehouse.
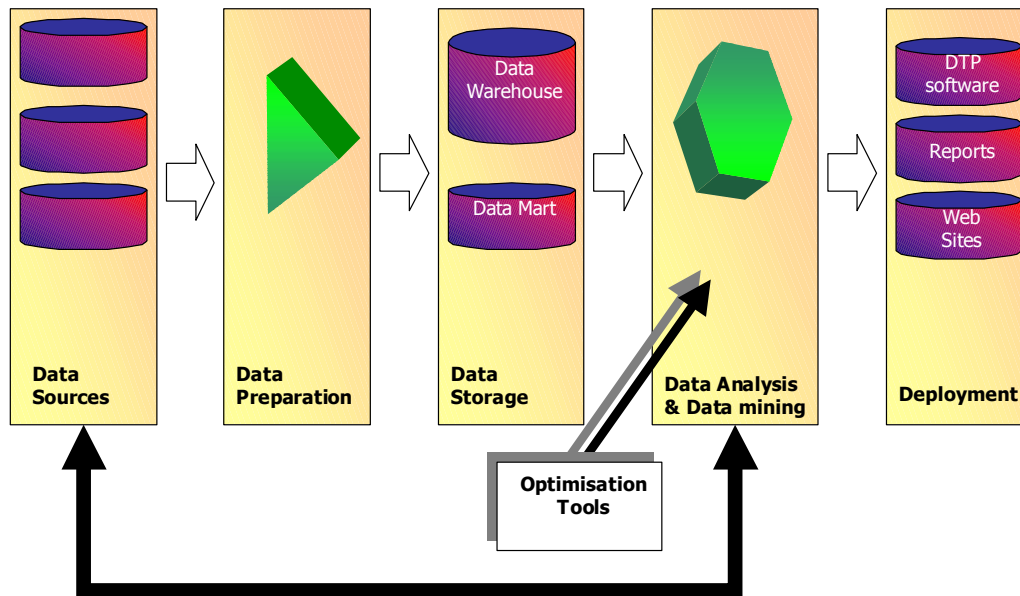
Figure 1. The information value chain

In this context, optimisation models and the tools which enable their formulation and solution are considered to be inference engines for knowledge discovery. They operate on the analytic data stored in the data marts to provide decision makers with decision data which is inferred from the models. Optimisation models can also be used within the more general framework of simulation, enabling decision makers and analysts to take advantage of descriptive models as well as prescriptive models. Another important aspect of the information value chain is that the decision data which is created by (the optimisation) models is added to the analytic data, and naturally becomes a part of the data mart. An example of such a process is the use of prescriptive models to obtain a set of optimum decisions, which are then used as input for a descriptive (simulation) model for risk analysis under alternative scenarios; together they make up a valuable data mart.

The ever growing number of industrial strength applications underlines the importance of optimisation within the information value chain. Beyond the well established application domains such as supply chain management and transportation, other fields are also benefiting from optimisation based applications. This is due to two factors: the advances in optimisation techniques and tools and the evolution of the IT and information system infrastructure, which greatly facilitate the integration of the OR tools with the firm's data sources. This seamless integration is essential for an effective use of optimisation as a Decision Support System within a firm.

1.2   Optimum decision making under uncertainty

Optimum decision making concerns a class of problems where it is necessary to make decisions to optimise one or more given objectives, subject to appropriate restrictions. Mathematical Programming (MP) models have made considerable contribution to optimum decision making, as they enable the modeller to capture the structure of the problem and to quantify the effects of the decisions in terms of the objectives of the problem owner also known as the decision maker.

The future, due to its very nature, is naturally linked with *uncertainty*. When using LP or IP for optimum decision making, modellers are required to analyse the available data in order to identify the parameters which are to be considered in the model. Aggregation and estimation intervene in this important phase of the LP/IP modelling process. In many situations the parameter values do not remain constant and are variable or "volatile".

The class of problems for which the assumption of a deterministic world (that is, model parameters are known exactly), is relaxed and often referred to as *optimum decision making under uncertainty*, since some, if not all, the model parameters may be uncertain.

In these cases the modeller needs to take into account the effects induced by uncertainty into the underlying optimisation models. An early approach in the investigation of these effects was the use of sensitivity analysis. Unfortunately, as shown in (Higle and Wallace, 2002) this approach shows a number of limitations, and may provide misleading conclusions in respect of the nature of the solutions. In general sensitivity analysis is not a suitable approach for understanding the effects of random behaviour of the model parameters.

In many real world problems, the uncertainty relating to one or more parameters can be modelled by means of probability distributions. In essence, every uncertain parameter is represented by a random variable over some canonical probability space; this in turn quantifies the uncertainty. *Stochastic Programming (SP)* enables modellers to incorporate this quantifiable uncertainty into an underlying optimisation model. Stochastic Programming models combine the paradigm of dynamic programming with modelling of random parameters, providing optimal decisions which hedge against future uncertainties.

The optimum decision model together with the constraints constitutes the core of the problem that has to be solved and varies with respect to the specific characteristics of each individual application. The second element is the model of randomness. Once the distribution is established, different scenarios that follow the underlying probability distribution are used to represent the uncertainty. A major issue in any application of stochastic programming is the representation of the underlying random data process.
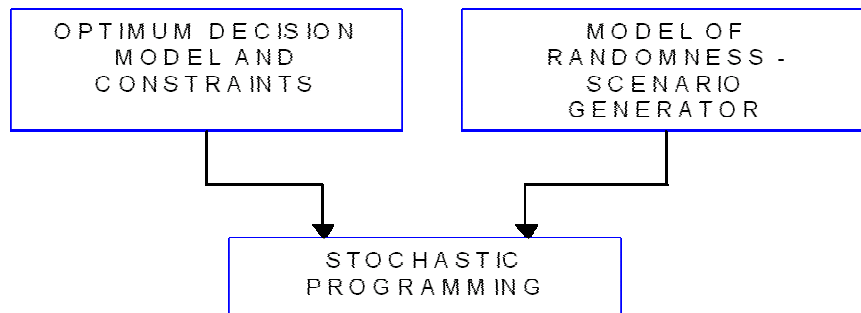

Figure 2: Breakdown of Stochastic Programming

1.3   Applications of Stochastic Programming

The successful application of Linear and Integer Programming models to decision problems in their turn have opened up the scope of applying Stochastic Programming models in decision making where time and uncertainty affect the decisions. Moreover advances in hardware as well as software techniques and solution methods have made SP a viable optimisation tool for decision making under uncertainty. Finance and Supply Chain planning are typical areas where deterministic optimisation models have proven to be extremely inadequate since the model parameters are by their very nature uncertain. Many problem domains in which SP is finding progressive applications are considered below.

- **Finance**: Basic concepts of Asset and Liability Management (ALM) models under uncertainty were developed by Kallberg et al. (1982) and Kusy and Ziemba (1986). Subsequently a number of substantial applications were developed. They include the Russell-Yasuda Kasai model by Carino et al. (1994). Mulvey (Mulvey, 1994) developed a framework for modelling financial planning problems based on multistage stochastic programs. Applications include risk management for institutions, banks, government entities, pension plans, and insurance companies. The approach also applies to individual investors who are interested in integrating investment choices with savings and borrowing strategies. A dynamic discrete-time structure addresses realistic financial issues. The resulting stochastic program is enormous by current computer standards, but it possesses a special structure that lends itself to parallel and distributed optimisation algorithms. Solving these stochastic programs presents a major challenge for the computational operations research and computer science community. The CALM stochastic programming model for dynamic asset-liability management (Dempster and Consigli, 1998) is also based on multistage stochastic programming. The CALM model was designed to deal with uncertainty affecting both assets (in either the portfolio or the market) and liabilities (in the form of scenario dependent payments or borrowing costs). The portfolio manager, given an initial wealth, seeks the maximization of terminal wealth at the horizon, with investment returns modelled as discrete state random vectors. ALM models were extended to capture the increasing complexity of the fixed income securities markets (Zenios, 1995). The author also found that the use of stochastic models is well justified given their superior performance over traditional immunization techniques. In Dempster et al. (2003) the authors introduced strategic DFA (Dynamic Financial Analysis) based on dynamic stochastic optimisation. This approach is described in terms of an illustrative case study of a joint university/industry project to create a decision support system for strategic asset liability management involving global asset classes and defined contribution pension plans. Although the application of the system is to fund design and risk management, the approach and techniques described are much more broadly applicable to strategic financial planning problems. De Lange et al. (2004) examine the rationale for financial reinsurance in the casualty insurance business. This concept refers to an investment strategy that uses the financial markets to hedge insurance risk. Applications of SP in finance are also discussed in (Vladimirou, 2004).

- **Supply chain:** Manufacturing flexibility has received much attention in recent years, especially after the advent of flexible and computer-controlled manufacturing systems; see Sethi and Sethi (1990) for extensive reviews. Much research has focused on determining optimal investment decisions on flexible and dedicated capacity at the strategic level, under the assumption of a single production period or a single point at which all uncertainty is resolved (e.g. Eppen et al. (1989)). Jordan and Turnquist (1983) study the benefits of partial manufacturing process flexibility in multi-plant multi-product settings and introduce the concept of chaining. Dominguez-Ballesteros (2002) proposes a two stage stochastic programming model for strategic and tactical supply chain planning. The work developed in SCHUMANN (SCHUMANN, 2000) provides a recent example of a prototype supply chain optimisation-based DSS. The software tool was built to aid supply chain planning and management in the automobile sector (Escudero et al., 1999) as well as in the pharmaceutical sector. In (Santoso et al. 2003), the authors investigate the use of SP for realistic scale supply chain network design. Poojari et al. (2004) studies a decision model for natural oil buying policy under uncertainty. Mirhassani et al (2000) and Messina et al. (1996) focus their studies on stochastic integer programming models for supply chain planning problems. An application of a stochastic mixed 0-1 approach to supply chain planning for helping to decide the production system topology under uncertainty is studied in Alonso-Ayuso et al. (2003).

- **Transportation**: Transportation and logistics represented some of the earliest applications of stochastic programming. Dantzig (1955) used fleet management (in an airline setting) as an early motivation for optimisation under uncertainty. Ermoliev et al. (1988) formulated the

planning of empty shipping containers as a stochastic program. There has been a rich history of research in fleet management in the context of the "car distribution problem" of railroads. Most of this work consists of deterministic linear programming models. Dejax and Crainic (1987) provide a thorough review of the research in fleet management at the time, covering both rail and intermodal container applications. Crainic et al. (1993) provide a general stochastic, dynamic model for container distribution. Jordan and Turnquist (1983) provide a stochastic formulation of the empty car distribution problem. In their model, a car could be assigned to at most one demand, and cars could not be repositioned more than once. This structure allowed the problem to be formulated as a non-linear programming problem. Powell (1986,1987) extended this methodological approach, using the trucking industry as the context, to multistage problems with reusable resources. This formulation involved forming deterministic decision variables which specified the percentage of the supply of trucks at a node that would be moved loaded or empty from one location to another. Powell (see Powell and Carvalho (1998)) solved the problem as a sequence of network problems, using linear approximations of the value of the future. This approach was computationally quite easy, and scaled to much harder problems. Powell et al. (2002) showed how the technique could be applied to the heterogeneous resource allocation problem, which is a more general problem than the multicommodity flow problem (the resource attribute space is much larger). Topaloglu and Powell (2001) applied similar techniques to stochastic multicommodity flow problems. This problem class introduces the additional complexity that multicommodity problems, combined with non-linear approximations of the future, produce sequences of (usually integer) multicommodity flow problems.

- **Telecommunication :** Telecommunications have a long tradition of application of advanced mathematical modelling methods. Stochastic optimisation is the methodology of choice for optimal decision support under uncertainty, see Ermoliev (1988), Kall and Wallace (1994), Birge and Louveaux (1997). While traditional design approach is centred around minimization of the network costs under technological and quality of service constraints, systematic application of stochastic programming techniques includes incorporation of modern tools like evaluation of real options. Comprehensive models which include pricing decisions and binary variables provide a motivation for further development of this methodology. There are several examples of stochastic optimisation models for design problems at the network level. For related examples see Bonatti and Gaivoronsky (1994), Sen et al. (1992), Tomasgard et al. (1998), Dempster et al. (2000), Andrade et al. (2002).

- **Environment** : Most of the environmental problems are characterised by two main features. On one side, environmental processes and resources are often characterised by irreversibility, i.e. their consumption, conversion, exploitation and degradation can be a one-way decision. On the other side, the uncertainty deriving from incomplete knowledge concerning natural processes and the effects of human actions on those processes, call for a particular attention in choosing the modelling techniques to be deployed. The theoretical framework for decision making in environmental applications in the presence of uncertainty is the quasi-option value approach, are discussed by Arrow and fisher (1974) and subsequently investigated, in its different features, in papers by Conrad (1980), Hanemann (1989) and Coggins and Ramezani (1998), among others. Clarke and Reed (1990) focus on optimal strategies for gradual conversion, involving stochastic dynamic programming. Bosetti et al. (2002) propose a multistage stochastic programming approach to land allocation while Gassmann (1987) adopts SP for forestry problems.

In addition to the application of SP techniques to specific problems, researchers have focused on the perfecting and improving algorithms for the solution of such problems, (Birge and Wets, 1986), (Laporte and Loveaux 1993), (Higle and Sen., 2004). Solution methods for SP with integer variables have been studied and proposed in (Van deVlerk 2001), (Escudero et al., 1999), (Schultz, 2003) and Cerisola and Ramos (2002).

1.4　Modelling Systems, Database and Constituents

Whereas in section 1.2, we have highlighted the importance of optimisation within an organisational IS here we consider a modelling perspective. In the different stages of the optimisation process, various constituents undertake different roles and interact with each other. These constituents and their interaction are set out in Figure 3.
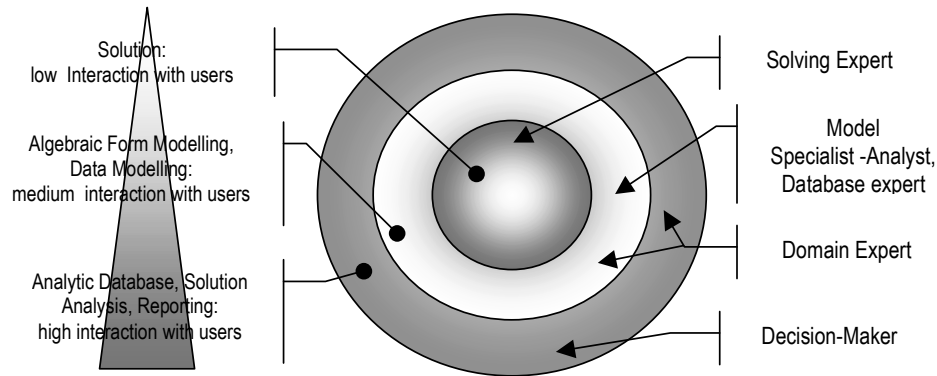

Figure 3　Optimisation Process and Constituents

The *technical experts* (*modelling experts*, *database experts*, *solver experts*, and *domain experts*, depending on their expertise) have access to all system components including the underlying models, databases, and can change the solver strategy. The different technical experts collaborate to create a domain-specific application by integrating the different tools for the different stages of the process. The *decision maker(s)* are the constituents that utilise the (application) system. Typically the decision makers have access to the data and solution analysis routines available in the database, and use the underlying models and data as a black box. These are only interested in the solution values, which are expressed in terms of decisions. Hereafter, they analyse and report the results that are in the analytic database. Accordingly, there is a need to have integrated optimisation tools for all the constituents involved. These tools must be capable of providing a set of features that ideally allow the different experts and problem owner(s) to complete the life cycle of the LP, MIP, IP, QP, SP based optimisation application. The traditional view of a mathematical programming based modelling and solving environment is shown in Figure 4.
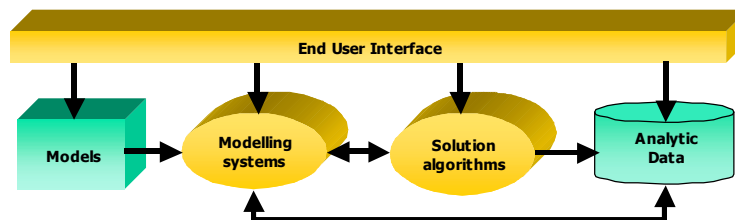

Figure 4 . Traditional view of an optimisation-based application

The architecture is centred on the (algebraic) modelling systems, which process one or more algebraic models and the analytical data in order to generate a model instance. The instance is then passed onto the solvers (which implement one or more of solution algorithms). These optimise the model instance and return the values of the optimum decisions and objective (decision data) to the modelling system, which in turn stores them into the database. The components are typically controlled by a Graphical User Interface (GUI).

1.5 Guided Tour.

The rest of this paper is organised in the following way. We provide a short introduction to Stochastic Programming in section 2 which also includes a list of the available software tools for SP. This section is complemented by an explanation of relevant model definitions and stochastic measures in Appendix A and by a review of a few well established scenario generation models in Appendix B. In section 3 we first give an overview of organisational information system and position the roles of analytic database and multidimensional database including On-Line Analytical Processing (OLAP) views. The connection of IS to our decision modelling and description modelling methods is essential, as this enables the embedding of models within business analytics, business processes or business rules as appropriate. In section 4 we provide a high level conceptual architecture of the integrated system. We consider the role of scenario generation and scenario trees and how these relate to the SP decision model. The use of Algebraic Modelling Languages (AMLs) to formulate optimisation models and the extensions of AMLs to construct SP models are discussed and the integrated system is introduced in a summary form. In section 5 we illustrate our combined modelling framework of decision making followed by evaluation through simulation using an illustrative case study. We describe an Asset Liability Management (ALM) model. For the purpose of generating scenarios we outline a multistage extension of Moment Matching approximation. We then consider the results of the decision model obtained by computing the solutions of the Expected Value, Here-and-Now (for both two-stage and multistage) formulation of the ALM problem. These first stage decisions are then fixed to create simulation runs for the Wait and See model and the frequency distribution of the objective solution values are computed. We discuss the in-sample stability of a scenario generation method, and present, stochastic measures and risk measures (VaR and CVaR) are presented as obtained by the simulation. Our conclusions are presented and discussed in section 6.

2 Stochastic Programming

2.1 Background to SP

Stochastic Programming is now a well established mathematical programming paradigm which explicitly incorporate uncertainty in the form of probability distributions of some parameters. These models can be further categorised taking into account the way in which such uncertainty is expressed and dealt with in the underlying linear optimisation model. The classification of stochastic programming problems leads to a working taxonomy shown in figure 5.
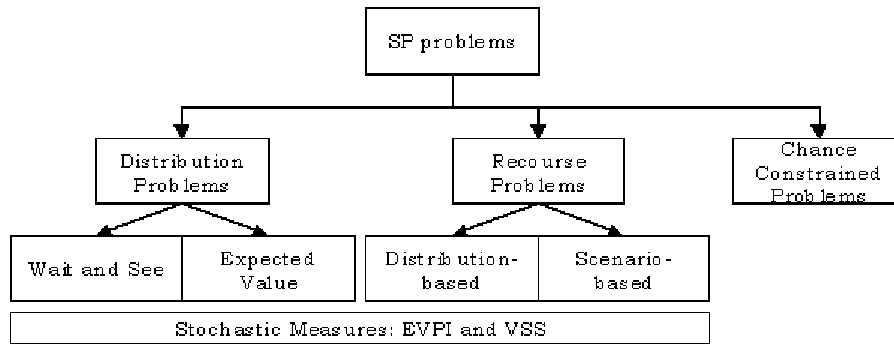
Figure 5 Taxonomy of SP problems.

These problem classes can be illustrated by first considering the linear programming problem:

$$Z \;=\; \min cx$$
$$subject\ to \quad Ax = b$$
$$x \geq 0$$
$$where \quad A \in R^{m \times n}; c, x \in R^{n}; b \in R^{m}$$

Let *(Ω, ℑ, P)* denote a (discrete) probability space where $\xi(\omega)$, $\omega \in \Omega$ denote the realizations of the uncertain parameters. Let the realizations of *A, b, c* for a given event ω be defined as:

$$\xi(\omega) \ or \ \xi_{\omega} = (A, b, c)_{\omega}$$

The associated probabilities of these realizations are often denoted as $p(\xi(\omega)) \ or \ p_{\xi(\omega)}$. For notational convenience these probabilities are denoted simply as $p(\omega)$.

Distribution Problems

The optimisation problems which provide the distribution of the objective function value for different realisations of the random parameters and also for the expected value of such parameters are broadly known as the distribution problems.

Expected value problem

The Expected Value (EV) model is constructed by replacing the random parameters by their expected values. Such an EV model is thus a linear program, as the uncertainty is dealt with before it is introduced into the underlying linear optimisation model. It is common practice to formulate and solve the EV problem in order to gain some insight into the decision problem.

Wait and see problems

Wait and See (WS) problems assume that the decision-maker is somehow able to wait until the uncertainty is resolved before implementing the optimal decisions. This approach therefore relies upon perfect information about the future. Because of its very assumptions such solution cannot be implemented and is known as the "passive approach". Wait and see models are often used to analyse the probability distribution of the objective value, and consist of a family of LP models, each associated with an individual scenario.

Recourse Problems.

Here and Now Problems

Assuming Z as solution of the objective functions, simple (single stage) stochastic programming model can be formulated as follows:

$$Z_{HN} = \min E[c(\omega)x]$$
$$where \quad x \in F$$

$$and \qquad F = \bigcap_{\omega \in \Omega} F^{\omega}$$

The optimal objective function value Z_{HN} denotes the minimum expected costs of the stochastic optimisation problem. The optimal solution $x^* \in F$ hedges against all possible events $\omega \in \Omega$ that may occur in the future.

A description of the Here and Now problems can be found in Appendix A, together with the uncertainty measures : value of the stochastic solution (VSS) and the expected value of perfect information (EVPI) (CARISMA SP workshop notes (2000), Infanger (1994)).

2.2   SP Software Tools

The Algebraic Modelling Languages (AML) have played an important role in the acceptance of mathematical programming techniques as an aid to decision making. AMLs are declarative languages which enable practitioners to rapidly build structured and scalable optimisation models. Modern systems based on algebraic modelling languages support the formulation and implementation of Linear Programming (LP), Mixed Integer Programming (MIP), Quadratic Programming (QP), and to some extent Non-Linear Programming (NLP) models. These systems are readily connected to linear or non-linear optimisers for the solution of the models under investigation, and are able to interact with corporate data warehouses and data marts stored in relational, object oriented or in other emerging standards. Until recently, however, the investigation of stochastic programming models could not take advantage of comparable tools. In fact, the practical exploitation of SP presents various difficulties, which affect the whole process of

modelling, instantiation, solution and analysis of the results of SP problems. Recently there has been considerable progress in the development and application of SP:

- the links between the structure of stochastic programming models and the models of randomness which provide the quantification of the uncertainty and their joint adoption for a given problem are better understood.

- integrated modelling environments have been developed which combine of scenario generators, modelling systems based on extended algebraic modelling languages, solution algorithms and analysis tools in a single software environment support the creation and investigation of stochastic programming problems.

SP Software Tools

Name	Affiliation	System Name	Type
JJ Bisshop, et al.	Paragon Decision Tech.	AIMMS	Modelling System
A Meeraus, et al.	GAMS	GAMS	Modelling System
B Kristjansson	Maximal Software	MPL	Modelling System
R Fourer, et al.	Northwestern University	AMPL	Modelling System
MAH Dempster, et al.	Cambridge University	STOCHGEN	Modelling System
E Fragniere, et al.	University of Geneva	SETSTOCH	Modelling System
A King, et al.	IBM/COIN-OR	OSL/SE, SMI	Solver
HI Gassmann, et al.	Dalhousie University	MSLiP	Solver
G Infanger et. Al.	Stanford University	DECIS	Solver
P Kall, et al.	University of Zürich	SLP-IOR	Modelling System / Solver
G Mitra, et al.	Brunel University	SPInE	Modelling System / Solver

Table 1 SP Software tools

3 Information Systems Components

3.1 Transactional Data, Analytic Data and Information Value Chain

In an organisational context, the role of the information system has become well established. In general data across the organisation is classified in two groups, namely transactional data and analytic data. Typically the transactional database system is optimised for performance and efficiency rather than supporting analytical features. In contrast the analytical database is carefully prepared for exploitation and decision support. (Berson and Smith 1997).

Analytic data is usually stored in data warehouses; these repositories of subject oriented integrated and non volatile information are aimed at supporting the knowledge analyst to make better and faster decisions. The data in the warehouse are modelled as a multidimensional item; this approach facilitates the query engines to aggregate data across many dimensions in order to detect trends. We take a particular perspective of data and infrastructure in our modelling system. This view is called the information value chain (IVC) paradigm that commences with the analysis and synthesis of transactional data to create analytic data or information, which is the first step in the information value chain.

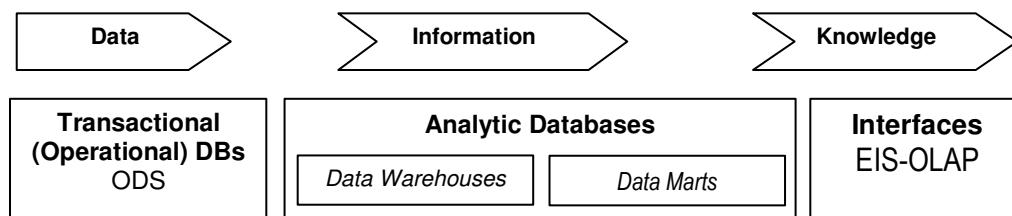| Data	Information	Knowledge
Transactional (Operational) DBs ODS	**Analytic Databases** *Data Warehouses* *Data Marts*	**Interfaces** EIS-OLAP

Figure 6: Information Value Chain.

Quantitative models constitute an important component within the IVC (Koutsoukis and Mitra., 2003). Optimisation models are used within the framework of simulation, enabling the analysts to take advantage of descriptive models as well as prescriptive models. An important aspect of IVC is that the decision data which is created by the optimisation models is itself analytic data, and can therefore be fed into the chain again. As an example of the importance of IVC we can observe the use of prescriptive (optimisation) models to obtain a set of optimum decisions, which are then used as input for a descriptive (simulation) model for risk analysis under alternative scenarios.

The term *decision database* denotes the collection of data repositories that are used for decision support applications. We use the term decision database to denote data marts that are used to store the results of our model-driven investigations. For instance a decision-maker or an end-user is typically interested in working with the optimisation results, in order to proceed with the decision making process. In this case, it is easily seen that the decision making process is very much related to the 'new' information, namely the optimisation results. Hence, a decision database requires some of form medium-term storage for the inferred information generated by the optimisation models.

Data Modelling, Decision Modelling, and Model Investigation, are logical steps, which play a leading role both in the interaction of information systems and decision technologies; taken together they lead to business analytics. It is easily seen that Data Modelling and Decision modelling closely interact with each other. The following list describes these logical constructs and their relationship to knowledge.

- Data Modelling refers to the 'structured' internal representation and external presentation of recorded facts. Broadly speaking this provides the decision-maker with information about their decision problem.

- Decision Modelling is the development of a model, or a range of models that captures the structure as well as the decisions in respect of a given problem. These models are used to evaluate possible decisions (actions) in a given problem domain, and the probable outcomes of these actions.

- Model Analysis and Investigation refers to the instantiation of the model with data, and the evaluation of the model parameters as well as the results in order to gain confidence and insight into the model.

Typically, data modelling involves defining relationships between data items leading to a relational data model, or identifying categories that are then used to define multidimensional tables, leading to a multidimensional data model. Decision modelling involves the development of models that are used for decision-making. Model analysis and investigation is often a descriptive analysis of the results obtained which is applied to gain insight, or knowledge with respect to a given decision problem. In particular, during the analysis phase, a series of 'what-if' investigations are used to change parameters and, therefore, to evaluate the understanding of the problem by the decision-maker. A generic representation of the decision problem is also desirable and this is achieved through the interaction between data modelling and symbolic modelling. In MP models there is a direct mapping between the sets and indices that form the decision model and the structure of the data used to instantiate the decision model. This close coupling and physical separation of data modelling and decision modelling leads to a number of distinct advantages for the implementation and use of decision support systems:

- Large amounts of data are managed and maintained by database systems, which are built for this purpose, instead of MP modelling systems.

- Decision models can be variously instantiated with the use of different data marts.

- The information used to develop the decision support system is available in databases.

- Database systems, and data marts in particular, are easily connected to a range of analytical tools which can facilitate the analysis of optimisation results. Therefore analysis of optimisation results is easier to achieve by using dedicated data marts, and standard database technology.

The Decision Database

It is easily seen that for any given decision support system, data models, symbolic models, and algorithmic tools interact. In many optimisation based DSS there exists an inherently strong coupling of data models, symbolic models and algorithms, also called an inference engine (Shapiro, 1993). In general, one can consider the decision database as a data mart, in which the data models provide information about the problem at hand, decision models describe the decision problem, or the known relationships between the known information, and the algorithms generate the set of optimal, or near optimal "solutions" for the problem at hand. In stochastic programming applications, the decision database may consist of three main types of information: *deterministic information* (or information invariable to the uncertainty of the future), s*tochastic information* (possible characteristics of future uncertainty and the information that is subject to that uncertainty), as well as the *solutions* (possible courses of action in the uncertain future). Further, it will include different solutions from different solving methods for a particular problem (output results, e.g., 'wait-and-see' scenario analysis, and 'here-and-now' solutions). To summarise an optimisation-based DSS is an application-specific information systems, consisting of Database Tools, Algebraic Modelling Tools, and Algorithmic ('solving') tools, with additional analytical tools which are often used to provide further insight into the problem at hand. These additional analytical tools may consist, for

example, of (Relational) Online Analytical Processing (R)OLAP tools which can help an analyst achieve a fast and descriptive overview of the data, in order to study the problem, and make appropriate reports. In Section 4.4 we revisit some of these aspects to illustrate how this relates to our combined SP and simulation framework.

3.2   OLAP: Multidimensional View and Datacube

Since the inception of the term OLAP (Codd et al., 1993), the technology is recognized as a promising approach for the analysis and navigation of data warehouses and multidimensional data. OLAP systems enable powerful decision support based on multidimensional analysis of large amounts of summary data commonly drawn from a number of different transactional databases. OLAP's multidimensional data model and data aggregation techniques organize and summarize large amounts of data to facilitate quick evaluation using online analysis and graphical tools. In this section we will discuss the data operators that we use to generate our set of numeric summary tables corresponding to a given aggregation hierarchy.

The OLAP data are organized in multidimensional cubes containing measured values that are characterized by a number of hierarchical dimensions. The typical operations on data cubes include roll-up (increasing the level of aggregation), drill-down (decreasing the level of aggregation or increasing detail) along one or more dimension hierarchies, slice-and-dice (selection and projection), and pivot (re-orienting the multidimensional view of data). For a detail discussion on these operations, see (Codd et al. 1993) and (Koutsoukis and Mitra, 2003).

Data warehouses and related OLAP technologies continue to receive strong interest from the research community as well as from industry, since OLAP tools present their users with a multidimensional perspective of the data and facilitate the writing of reports involving aggregations along the various dimensions of the data set.

In order to build effective decision support systems in the presence of large data warehouses and complex underlying queries, it is very important to present and process data effectively. Moreover, decision support tools need to connect alternative modelling paradigms, artificial intelligence methods, and database technologies with multidimensional data technologies such as OLAP. These disciplines and enabling technologies together are applied to achieve one or more data mining tasks.

The diagram below shows how OLAP views are integrated in a decision support system. The architecture logic is based on the IVC: from the Transactional Databases to the Analytical Databases, and from Data to information leading to new knowledge, an OLAP viewer is used to present and analyse raw data, analytic data and decision data.
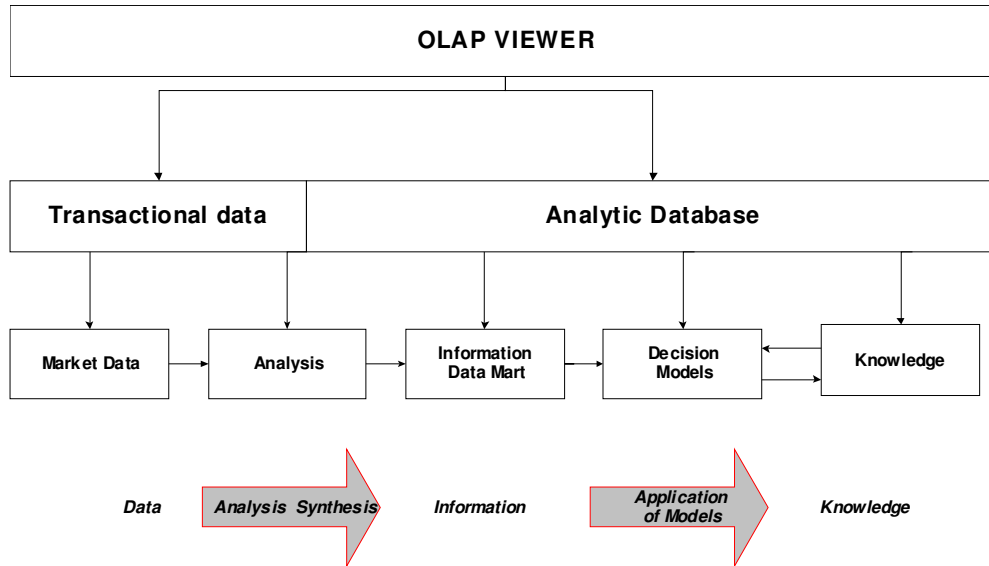
Figure 7 OLAP and Information Value Chain

The multidimensional approach offers a number of advantages over traditional types of database management systems (DBMSs), including automatic application of the pre-specified aggregation functions (automatic aggregation) (Rafanelli and Shoshani, 1990), visual querying, and good query performance due to the use of pre- aggregation (Gupta et al., 1995, Pedersen and Jensen, 1999). Additionally, the dimensional approach is a natural fit for data analysis problems. To be able to capture the complex data found, the data model for the OLAP system is able to handle irregular dimension hierarchies that do not fit the balanced-tree hierarchies (Pedersen and Jensen, 1999). In general, OLAP cubes are a multi-dimensional representation of data, in which each attribute of a given data entity is considered as a separate dimension. For our purpose an OLAP data cube consists of a lattice of cuboids each of which represents a certain level of hierarchy. The Cube's aggregate functions compute statistics for a given set of values within each cuboid. Table 2 shows alternative aggregate functions, which can be grouped into three categories namely: distributive, algebraic, and holistic (Gray, 1995).

	Aggregation Function		
Data Type	Distributive Function	Algebraic Function	Holistic Function
Set of Numbers	Count, Min, Max, Sum	Average, Standard Deviation	Median, Rank, Most Frequent,

Table 2 Aggregate Functions for OLAP data cube

In a traditional SQL relational database, aggregate functions and the *GROUP BY* operator function only produce one out of N aggregates at a time. An OLAP data cube is an aggregate operator which computes all N aggregates in one shot.

The ability of OLAP to support multidimensional data, makes it naturally suitable as a viewing and analysis tool for decision support systems which are based on multistage stochastic programming models. Indeed, as discussed in section 2, the uncertain parameters of these models as well as the event-dependent optimum decisions are usually represented by at least two dimensions, namely the time and the scenarios. As an example, consider the problem of generating and representing a number of possible future prices of a set of assets for a given number of time periods. In section 4

we describe how these values (analytic data) are computed by processing transactional (in our case: market) data using *scenario generators*. A relational table which contains this data should have at least the following fields, where the first three taken together define a primary key:

- Asset
- Period
- Scenario
- Price

In other words, we have a price for each triplet (Asset, Period, Scenario). Using an OLAP Cube, the data can be aggregated (rolled-up), or disaggregated (drilled-down) to obtain progressively summarised information or increasing detailed information respectively. This is illustrated in Figure 8 below:

0 - D Data Cube

Asset	Period	Scenario	Price
ALL	ALL	ALL	5.15

1 - D Data Cube

Asset	Period	Scenario	Price
1	ALL	ALL	0.24
2	ALL	ALL	10.162069
3	ALL	ALL	2.4108309
4	ALL	ALL	4.0341134
5	ALL	ALL	9.3705672
6	ALL	ALL	3.7087038
7	ALL	ALL	13.445068
8	ALL	ALL	6.0159124
9	ALL	ALL	1.4141097
10	ALL	ALL	0.6728426

2 - D Data Cube

Asset	Period	Scenario	Price
1	1	ALL	0.21
1	2	ALL	0.25
1	3	ALL	0.25
2	1	ALL	9.84
2	2	ALL	10.32
2	3	ALL	10.32
…	…	…	…
10	1	ALL	0.67
10	2	ALL	0.67
10	3	ALL	0.67

3 - D Data Cube

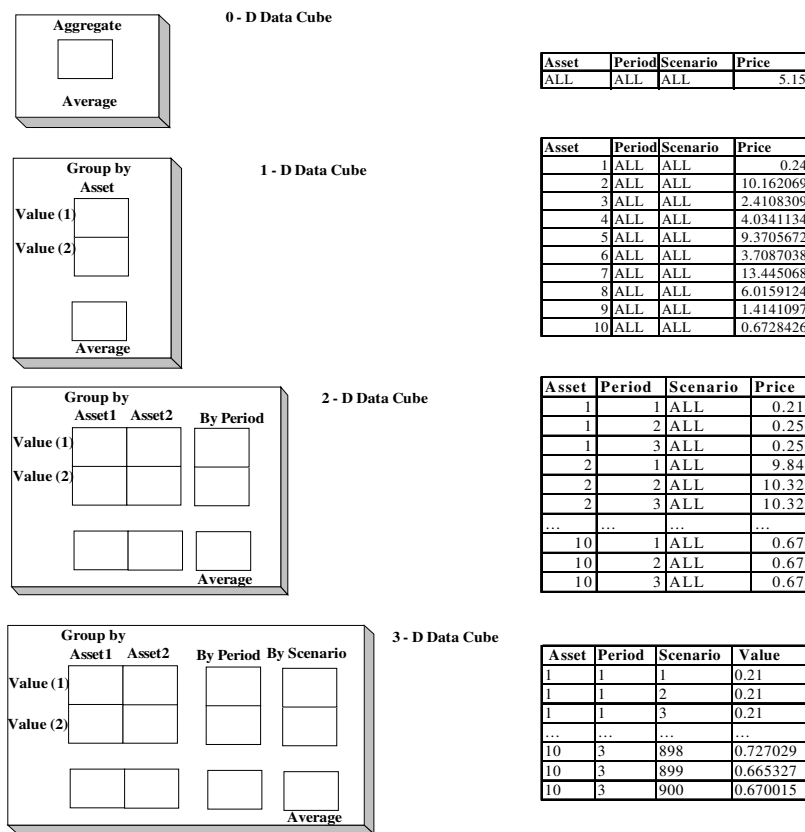| Asset	Period	Scenario	Value
1	1	1	0.21
1	1	2	0.21
1	1	3	0.21
…	…	…	…
10	3	898	0.727029
10	3	899	0.665327
10	3	900	0.670015

Figure 8 OLAP cuboids

The 0-D cube is obtained by rolling up all three dimensions Asset, Period and Scenario. Using Average as the aggregate function, we obtain a single scalar which is the "average price of all period, of all scenarios and for all assets". By drilling down this cube along one dimension, we obtain 1-D cubes (one per dimension) which in our example contain:

- The average asset price for each asset calculated over all periods and scenarios
- The average asset price for each period calculated over all assets and scenarios
- The average asset price for each scenario calculated over all assets and periods

By applying the drill down procedure to a 1D cuboid, we obtain a 2D cuboid and so on, until we expand all dimensions and obtain the full detailed data (in this case, a 3D cuboid). Besides roll-up and drill down, the slice and dice operations also enable OLAP users to create cuboids of smaller dimensions. Slicing into one dimension is very much like drilling one level down into that dimension but the number of entries displayed is limited to that specified in the slice command. A dice operation is like a slice on more than one dimension. Slicing and dicing therefore lead to a view in which only a subset of the data is shown. When the data is represented in 1D or 2D cuboids, it can also be visualised using 2D or 3D graphs. For instance, the 2D cuboid obtained by slicing into the asset dimension (selecting the data pertaining to one single asset) is shown in Figure 9 below. The 3D graph represents the price for a given asset in all time periods and for all scenarios. Some of the scenario data paths are highlighted.
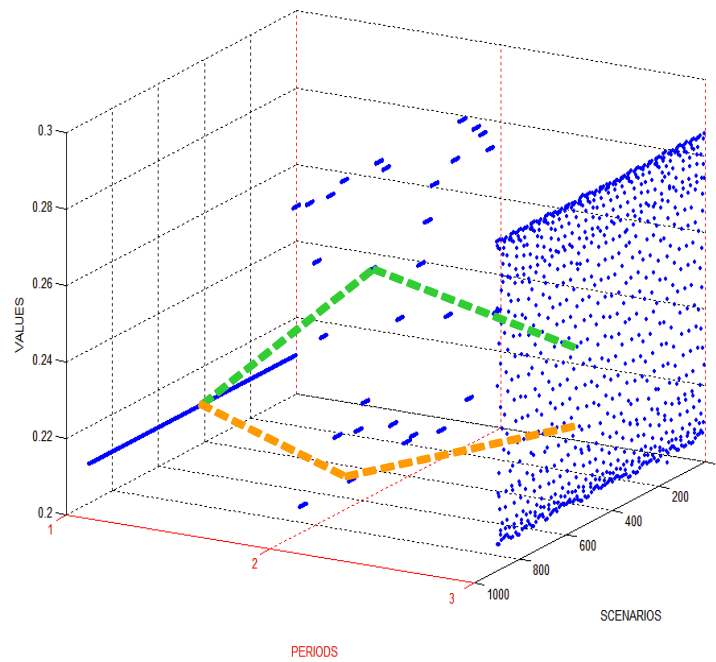

Figure 9 OLAP Datacube of asset price "datapaths"

A 2D graph with multiple series can also be used to display a 2D cuboid. For instance, the figure below shows the transactional (market) data used to generate the scenarios used in the previous examples.
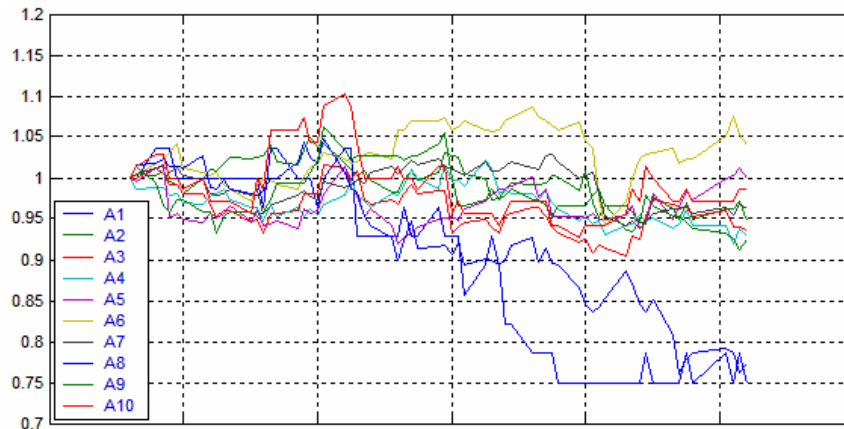

Figure 10: Time Series View: 10 Assets for the 65 time periods:

19

4 Integrating Scenario Generation and SP Software Tools

4.1 Scenario Trees.

Two stage stochastic programs with recourse (see Section 2) when stated explicitly take the form of large linear programs of a special structure which capture the realisations of the random parameters. In the two-stage case, let represent ξ the vector of all random parameters in a model. The probability distribution of ξ is assumed to be discrete with a finite number of realisations ξ_k. The probability of each realisation is defined as :

$$p_k = P(\xi_k) \text{ for k = 1..K } \text{ where } p_k \geq 0 \text{ and } \sum_{k=1}^{K} p_k = 1$$

For multistage Stochastic Programming one assumes that the random vector ξ follow a stochastic process ξ_t over the planning horizon. If the process is assumed to be discrete, with Probability $P(\xi_t)$ the uncertainty can be represented through a multilevel event tree which defines the possible sequence of realizations also known as data paths. In general these are called scenarios, over the whole planning horizon, see Figure 12.
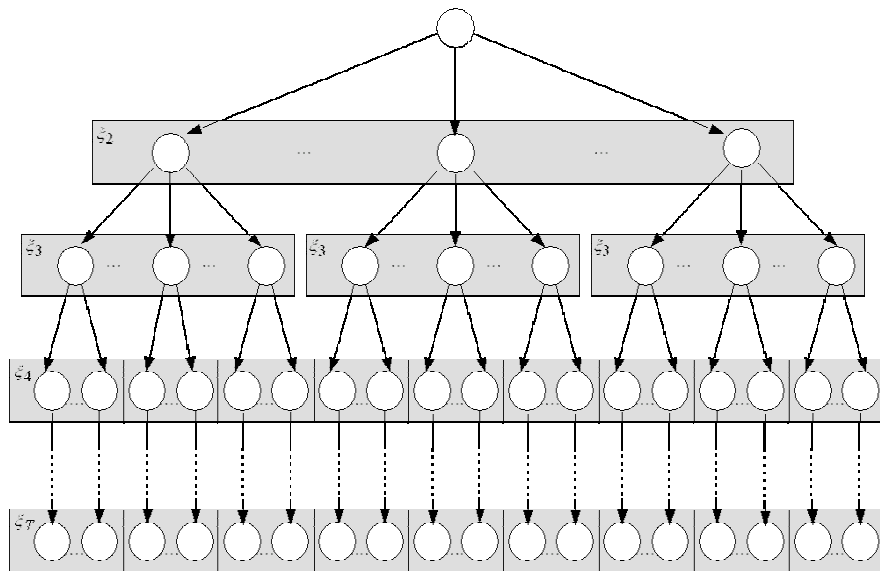

Figure 11 Multistage Tree

Levels in the tree are associated to decision stages. In particular if we denote N_t the set of nodes at the t the level, then each node $n \in N_t$, represents a particular realization sequence of the data process and it can be thought as a particular state of the system at a given time. A probability π_n can be associated with each node n at level t such that:

$$\pi_n = p\{\xi_t \mid \xi_{t-1} \mid ... \mid \xi_2\} \quad \sum_{n \in N_t} \pi_n = 1 \quad \pi_n > 0 \quad t=2..T$$

Hence, arcs in the tree represent the probability distribution of ξ_t

The random data representing the uncertainty of the future are expressed in stochastic programming by a scenario tree. In the two stage stochastic programmes the structure of the tree encapsulates the first and second stage phases. The beginning of the planning horizon is represented by a sole root node and extends linearly to nodes until the end of the first stage. Each level in the first stage is represented by a single node since the states of the world during the first stage time periods are known with certainty. Moving to the second stage the tree branches into nodes only at level t=k+1 as shown in the figure below. From each of these nodes discrete flat scenarios commence with nodes at each time period an optimum decision has to be taken until level t=T. This means that the scenario tree is nothing else but a fan of individual scenarios $\omega_s = \{\omega_{1,s}, ..., \omega_{T,s}\}$ which occur with probabilities $p_s = P(\omega_{k,s}) \ \forall \ s$.
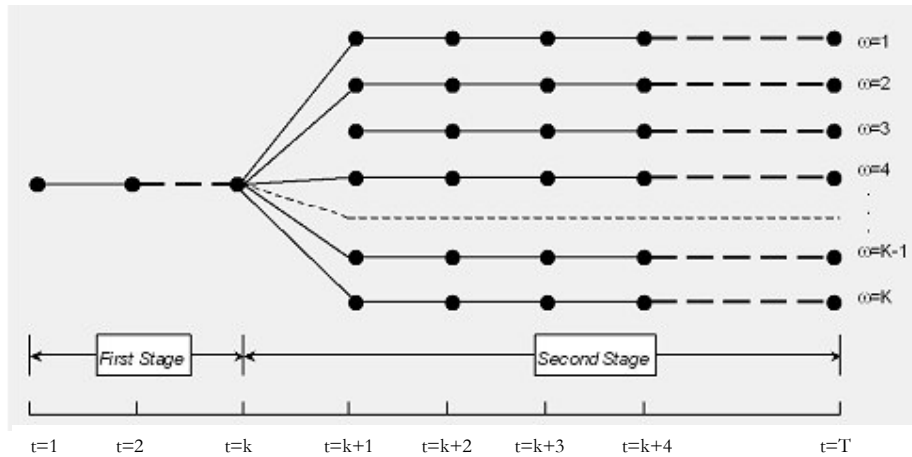

Figure 12: Scenario Tree for Two-stage Stochastic Programming

A representative scenario tree corresponding to the multistage stochastic programming formulation can be visualised as a tree starting similarly with the previous case with a sole root node at time 0 and branches into a finite number of nodes at level 1. This branching continues for all stages of the problem until level T. It is worth pointing out that although a node can have a finite number of descendants, a descendant could only have one immediate predecessor. A multistage scenario tree is considered to be balanced if the number of all descendants of the nodes at all stages 0<t<T is equal.
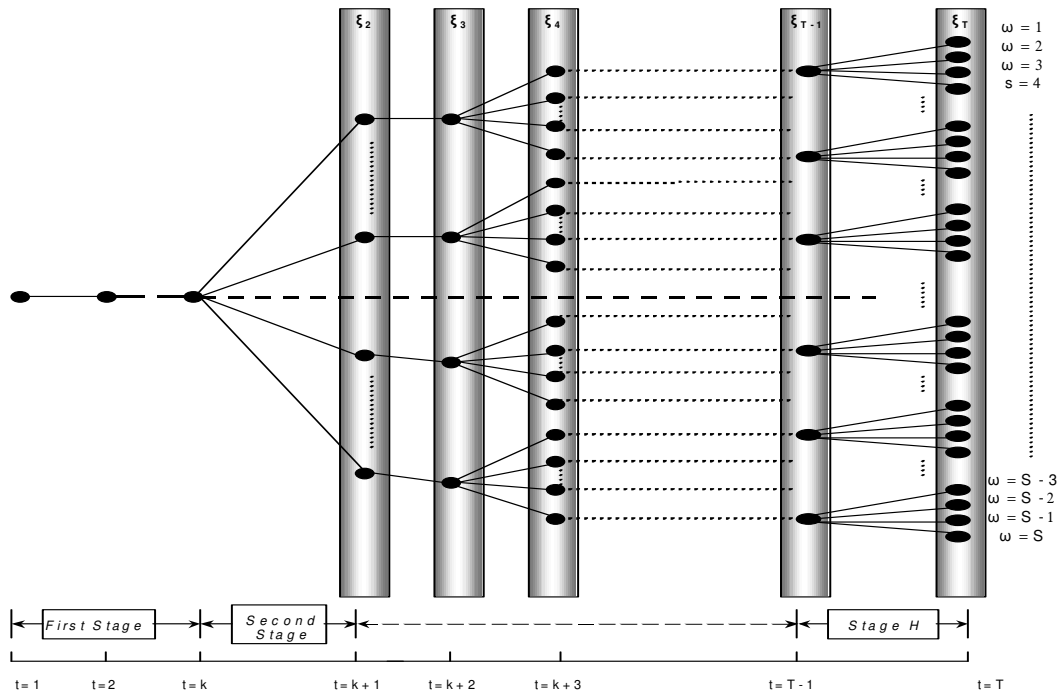
Figure 13 : Multistage tree

The decision on the number of the stages, time periods and the branching scheme is a very important initial step for a "good" representation of the uncertainty. The distinction between stages and time periods is essential because it is always possible in a specific application the number of the time periods to be greater than the corresponding nodes. Thus between two nodes, more than two time periods may exist. As a result in the two-stage stochastic program the stages are fixed to two but the decision on the number of the time periods for the recourse actions remains to be identified. In general, stages and time periods relate to points in time, that is, (in the planning horizon) that are important for the decision maker. The expiration date of an option contract or the dates a pension fund has to pay the contributions to the participants are examples of such situations. For a portfolio manager who wishes to rebalance his portfolio quarterly and has a planning horizon of twelve months, the problem translates into a multistage stochastic program with four decision stages each made of three time periods.

4.2   Scenario Generation

As discussed in the previous section, the data for a stochastic optimisation model is provided in the form of scenario trees; these are created using scenario generation methods which may be very specific to the domain of application (see section 2).

A major focus of scenario generation is to create a tree-structure of scenarios that "best" approximates a given underlying distribution of the random parameters. The criterion for choosing the "best" approximation is usually based on some measure which quantifies the distance of the generated scenario tree from the underlying distribution.

In general, a scenario generation procedure (for multistage problems) involves some or all the following steps:

- Assumption of a model which explains the behaviour of the random parameters (for instance, econometric models for interest rates, etc.)

22

- Estimation/Calibration of parameters for the chosen model which uses historical data/subjective view.
- Generation of data trajectories paths according to the chosen model or discretisation of the distributions using approximation of statistical properties.
- Conditional sampling of the trajectories so that a scenario tree with the desired properties can be constructed

In many cases, practitioners also apply some reduction techniques to the resulting scenario tree to provide model instances which can be realistically optimised by the available computational resources. Table 3 shows the most common techniques which may be adopted for the different steps defined above.

Purpose	Methods
Generation of data trajectories	**Econometric Models and Time series:** o Autoregressive Models : AR(p) o Moving Average Models : MA(q) o Autoregressive Moving Average Models : ARMA(p,q) o Generalised Autoregressive Conditional Heteroscedasticity : GARCH (p,q) o Vector Auto Regressive models: VAR o Bayesian VAR o Reduced Rank Regression **Diffusion Processes :** o Wiener Processes (Brownian Motion) o Generalised Wiener Processes (Brownian motion with drift) **Other methods:** o Neural networks
Discretisation	**Statistical Approximation** o Property Matching (Høyland and Wallace, 2001) o Moment Matching (Høyland et al, 2003) o Non parametric methods **Sampling** o Random sampling o Stratified Sampling o Bootstrapping
Tree construction and Conditional Sampling	o Optimal discretisation (Pflug and Hochreiter, 2003) o Barycentric Approximation (Frauendorfer 1996) o Sequential Clustering (Dupacova et. al, 2001)
Reduction	o Scenario Reduction (Dupacova et al., 2003)
Internal Sampling	o Stochastic Decomposition (Higle and Sen 1996) o Sochastic Quasi-gradient (Ermoliev 1988) o EVPI-based Importance Sampling (Dempster and Thomson 1999)

Table 3 Techniques used in scenario generation

As an alternative to generating scenarios which are then used to instantiate large deterministic equivalent models, Internal Sampling techniques are methods for solving stochastic programming problems which sample from the underlying distributions during the solution procedure.

4.3 Integrating Scenarios and Decision Models

We introduced the concept of scenarios and noted that the scenario tree within SP serves two different purposes: (i) define the model uncertainty, (ii) specify the algebraic structure of the decision variables and constraints.

A scenario generator φ captures in a procedural form a domain-specific model of randomness. In particular it uses historical information, an event tree structure and some other specification parameters. We can thus separate the main groups of parameters as H : History, τ : Event Tree, θ : Remaining Parameters. The set of scenarios Ξ is then seen as the collection of scenarios which are output by the generation procedure:

$$\varphi(H, \tau, \theta) \Rightarrow \Xi$$

In the algebraic form of the SP model we also need to specify the 'variable and constraint' tree structure, which we label as τ'. Thus using the extended AML we provide a specification of τ' in the SP model through the *tree* declaration.

For consistency, of course, we need the two trees to be congruent. In other words we need to ensure that the event tree structure τ used by the special purpose scenario generator is 'compatible' with the τ' specified in the SP model (Valente and Mitra. 2001). The requirement for scenario generator parameter passing and tree consistency conditions are illustrated in Figure 15. When a special purpose scenario generator is connected to SPInE, the two trees τ and τ' are compared for consistency. The scenario generator then creates the set of scenarios and the associated probabilities $p(\omega)$.

Alternatively, the modelling system can "import" the tree structure and allocate variables and constraints accordingly. This automatically avoids the problem of tree consistency.

Figure 14 SG Overview Scheme
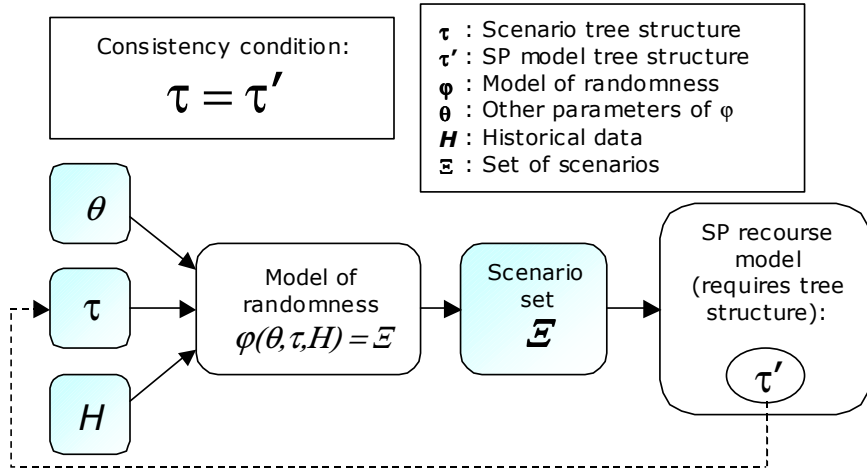
The issues involved in the integration of scenario generators with a modelling system such as SPInE can be summarised as follows:

- **Data consistency** : The data exchange between SG and SP Modelling System should be consistent and also should provide a well defined structure of the data tree. The types and the dimension of the information contained should be compact and functional.

- **Data communication**: a common format for the representation of the data trees has to be defined. A standard method of generation and data flows between the objects has to be identified as well. Independently from the data used the organisation of the information should keep the same format in a way that make the transmission easy and quick to analyse.

- **Data viewing, Modelling system and SG interoperability**: The objects involved in the optimisation system should be independent, but at the same time should have a solid data communication interface: this enables for instance the interoperability of different SGs. Data viewing plays an important role in the solution process: data analysis and data reliability are the main objectives to achieve.

- **Standard interface**: The data interface for the presentation of the scenarios to the modelling system is based on ODBC connections. This allows the scenario generator to store the output in virtually any type of database (including text files). The flexible interface with scenario generators and the ability to create in-sample scenarios for SP model optimisation and out-of-sample scenarios for simulation make the connection to external generators a valuable feature of stochastic modelling systems.

Taking into account the above requirements we are creating a library of scenario generators based on the techniques illustrated in Table 3. The availability of such a library makes it possible to formulate within the same modelling environment stochastic programming models which cover different application domains (Finance, Supply Chain, Energy). A well defined interface of SG libary enables other practitioners to add their own scenario generators to the system, leading to a very versatile platform for the investigation of SP problems (see also Di Domenica et al. 2004).

4.4   System overview

We discuss the development of a modelling and analysis environment which combines MDDB, OLAP, procedural languages and declarative modelling languages and supports the modeller in the automatic generation of multi-period stochastic models and the browsing of data and solutions across different time stages and over different scenarios. The system is designed to integrate Databases, OLAP viewers, Modelling Systems which include Scenario Generation, algebraic description of SPs, and SP Optimisation Solvers and simulation within a single application system. The system components are also referred to objects which are designed and specified to a common standard such that, where necessary, these can be replaced by alternative components. For example we use FortMP solver but the solver object is defined in such a way that it can be easily replaced by other leading industry solvers such as CPLEX or XPRESS.
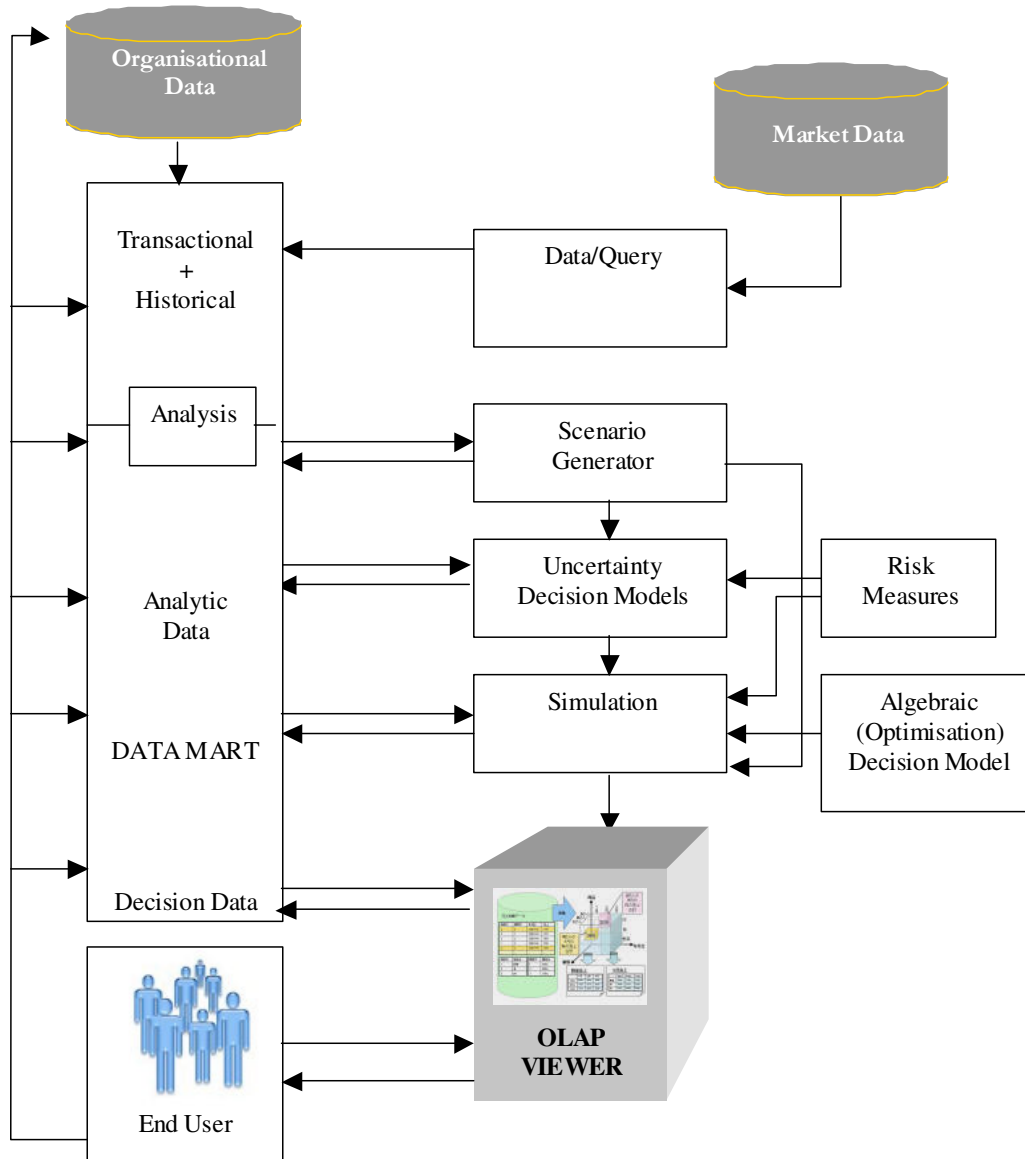
Figure 15 System Overview

Data flow through the system and interoperability present considerable design challenges. In the diagram above the relations and data flows of the different objects in the system are illustrated. The database is organised in three main sections: transactional data, analytic model data, decision data (see section 3). Scenario Generator and the software tools (SP Optimisation and OLAP) have all a specific role within the system. Initially the transactional data are analysed to create analytic data which can be viewed by the user through OLAP technology: a suitable interface is created to support the query and the analysis. The historical data is processed into an analytic data mart which is used by the Scenario Generator. The resulting scenario tree data is combined with the algebraic model description to instantiate an SP decision model. The SP solver processes the machine readable instance of the SP model created in external SMPS or in internal SIR format and the results are then put back into the analytic decision database. This step is followed by the results evaluation through simulation runs or other diagnostic tests.

SG is used to create scenarios for optimisation as well as to create scenarios for simulation. Simulation is very useful and can be displayed for different purposes including :

- Evaluation of the stability (robustness) of a SP solution
- Evaluation of the stability of the SG itself (Pflug and Hochreiter, 2003)
- Quantification of the risk associated with a given decision.

The system we developed shows that the integration between these objects is possible. In the next section a case study explains step by step the solution process and stability tests.

There is another user perspective of the software environment which comprises of four interacting functional modules and the resulting system architecture is displayed in figure below.
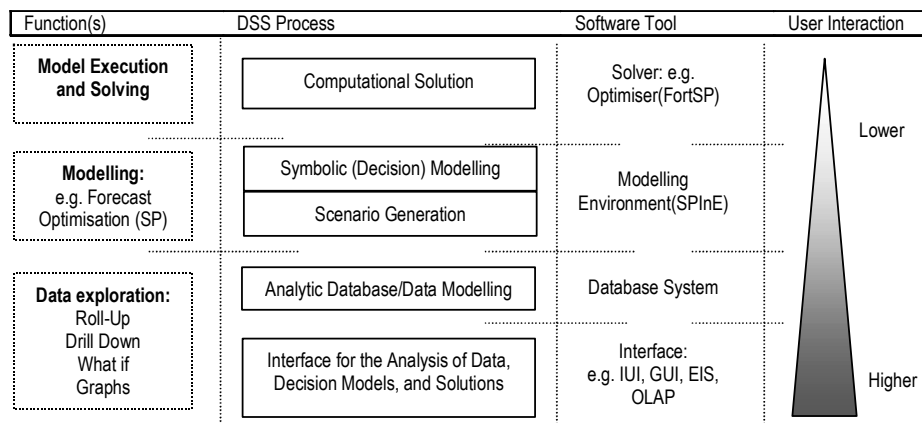
Also see (Kutsoukis et al. 2001)

Figure 16 DSS, Software and User Interaction

The functionalities of the four modules are explained below.

External Interface: This module enables the user to query the system.

Information System :

- *Database System*: it represents the repository of input data for the optimisation and evaluation models. The DB system also maintains and analyses the solutions of the SP models. More specifically the DB system is split into two parts, namely:

  a) *Data Tables*: contain all the input data tables which determine the instances of the problem generated by the Modelling System. Similarly, the output data tables contain the solutions supplied by the SP Optimisation System; all these tables can be viewed through OLAP technology in the IS system. In addition to model data they include optimisation results and simulation results

  b) *Procedures:* In order to support SP modellers in defining the models and scenarios, tools such as non-linear solver for parameter estimation and simulators for sampling are included. Furthermore, it contains the following procedures for the communication between different modules:
     o Procedures for generating scenarios.

- o Procedures for building the deterministic equivalent model;
- o Procedures for analysing, browsing and representing the results of the Optimisation Modules.

Modelling System: We use SPInE (Stochastic Programming Integrated Environment) (Messina and Mitra 1997; Valente et al. 2004) as modelling tool for SP. In SPInE the user can build his own library of models which can be used for studying and analysing different instances of the problem. SPInE brings together the Scenario Generator output tree and the algebraic model. The SP model can be created in external SMPS format, or internal SIR format and an XML realisation is under consideration. SPInE aims to facilitate model construction, model investigation, solution reporting and overall browsing of a stochastic programming application.

SP Optimisation System: The system provides computational solution of the SP problems generated by the Modelling System. It includes presolving procedures and optimisation algorithms for the solution of large scale stochastic LP/ILP problems. The SP-optimisation suite uses the embedded solver FortSP (Mitra et al. 2004) to process a range of SP models.

5   A Case Study

5.1   The experimental set up

We present a case study involving a small prototype model; the purpose of the case study is to illustrate the combined paradigm of data modelling, scenario generation, decision making under uncertainty via SP modelling and result evaluation using SPInE.

We describe an Asset Liability Management (ALM) model in section 5.2; in section 5.3 we outline a multistage extension of Moment Matching approximation, which is used for generating scenarios for a two-stage and multistage SP model. We then consider the results of the decision model obtained by computing the solutions of the Expected Value, Here-and-Now (for both two-stage and multistage formulation of the ALM problem). These first stage decisions are then fixed to create simulation runs for the Wait and See model and the frequency distribution of the objective solution values are computed. The analysis of results and the evaluation step through simulation cover the following aspects. In section 5.4 we present the stochastic measures, in section 5.5 we discuss the "In sample stability" and in section 5.6 the risk measures (Var and CVar) are presented.

5.2   The Decision Model

The finance industry, rather belatedly, has embraced the Markowitz mean-variance model (Markowitz 1952; see also Jobst et al. 2001) for portfolio planning and asset/liability management applications. A major criticism of this approach (Luenberger 1997) is that it is a single period, static model. Carino and Turner (Carino and Turner 1997) illustrate the superiority of Stochastic Programming dynamic asset allocation models, over the mean-variance approach. In fact, active portfolio management breaks away from myopic static decisions and implies revisiting the strategy, and re-balancing of the portfolio positions as financial conditions change. We use an ALM multistage Stochastic Programming model with downside risk constraints (Kyriakis 2001).

Algebraic formulation

The ALM problem:

An investor faces the problem of creating a portfolio allocating assets out of a universe of I assets. Each asset is characterised by a price, which is (the only) random variable. The possible future prices are represented by an event tree. The goal of the investor is to maximise the portfolio wealth at the end of the time horizon T. He needs to take into account future obligations (liabilities). Asset buying and selling decisions are made, and each trade has an associated transaction cost. The deviation of the portfolio value from a predefined target is taken as measure of the risk. In each time stage the investor can decide the amount of assets to buy, sell and hold in the portfolio.

We formulate this problem as a two-stage as well as a multistage stochastic program with recourse, (Messina et al. 1997):

Sets and indices:

T	denotes the number of time period in the time horizon
Assets	is the set of assets in our universe, where $|Assets| = I$
Scenarios	is the set of scenarios, where $|Scenarios| = Sc$
$t = 1..T$	denote time periods,
$i = 1..I$	denote an asset,
$s = 1..Sc$	indicates a scenario.

Parameters:

$price_{its}$ $i \in$ *Assets*, t=*1..T*, s $\in$ *Scenarios* is the price of asset i in period t, for scenario s

p_s s $\in$ *Scenarios* is the weight (probability) associated to scenario s

$L_t \geq 0$ t=*1..T* is the expected liability at time period t

$F_t \geq 0$ t=*1..T* is the funding available in time period t

$A_t > 0$ t=*1..T* is the predefined target for time period t

$H0_i \geq 0$ i $\in$ *Assets* is the initial composition of the portfolio

$R \geq 0$ is the maximum deviation from the target accepted by the investor (in fraction)

$g \geq 0$ is the transaction cost rate

Decision variables:

$H_{its} \geq 0$ i $\in$ *Assets* t=*1..T*, s $\in$ *Scenarios* is the amount of assets of type i held in time period t under scenario s

$B_{its} \geq 0$ i $\in$ *Assets* t=*1..T*, s $\in$ *Scenarios* is the amount of assets of type i bought in time period t under scenario s

$S_{its} \geq 0$ i $\in$ *Assets*, t=*1..T*, s $\in$ *Scenarios* is the amount of assets of type i sold in time period t under scenario s

Objective function:

Maximise the expected value of the final portfolio wealth:

$$\max \sum_{s=1}^{Sc} p_s \sum_{i=1}^{I} price_{iT} H_{iT}$$

Subject to:

Asset holding constraints:

$$H_{its} = HO_i + B_{its} - S_{its} \qquad\qquad t = 1, i = 1..I, s = 1..Sc$$

$$H_{its} = H_{it-1s} + B_{its} - S_{its} \qquad\qquad t = 2..T, i = 1..I, s = 1..Sc$$

Fund Balance constraints:

$$(1-g)\sum_{i=1}^{I} price_{its} S_{its} - L_t + F_t = (1+g)\sum_{i=1}^{I} price_{its} B_{its} \qquad t = 1..T, s = 1..Sc$$

Downside risk constraints:

$$A_t - \sum_{i=1}^{I} price_{its} H_{its} \leq A_t R \qquad\qquad t = 2..T, s = 1..Sc$$

To complete the formulation we add to this a set of non-anticipativity constraints, which depend on the event tree structure.

5.3 Scenario Generation

Høyland et al. (2003) have proposed the Moment Matching method, in which a discrete distribution is constructed to fit the first four marginal moments of the probability distribution of a given random vector, while maintaining the correlations between the vector's elements. The authors assume that the correlation matrix and the first four marginal moments (Mean, Variance, Skewness, Kurtosis) are supplied by the decision maker. The structure of the algorithm is presented below in a summary form: (i) Generate n discrete univariate random variables, each satisfying a specification for the first four moments. (ii) Transform them so that the resulting random vector is consistent with the given correlation matrix. The transformation will distort the marginal moments of higher than second order. Hence, it is needed to start out with a different set of higher moments, so that the right ones can be obtained. The procedure would lead to the desired values for the correlations and the marginal moments if the generated univariate random variables were independent. This is, however, true only when the number of outcomes goes to infinity and all the scenarios are equally probable. With a limited number of outcomes, and possibly distinct probabilities, the marginal moments and the correlations will therefore not fully match the specifications.

We extend the Moment Matching method to create multistage scenario trees. According to the chosen scenario tree structure, a number of samples for the next stage is generated using the Moment Matching code provided by Høyland et al. (2003). Each new sample is then added to the historical data, the moments recalculated and a set of new samples generated and added as a sub-tree of the current node. The process is applied recursively for the number of stages required. The data manipulation procedures use a hierarchical data structure and the procedures for input/output are designed in a way that OLAP viewer (see section 3) can easily browse.

5.4 Optimisation and Stochastic Measures

Given the ALM model and a set of 2500 scenarios for the asset prices generated using the method described in section 5.3, we use SPInE to generate and solve the three related models Here-and-Now (HN), Wait-and-See (WS) and Expected Value (EV). Using the decisions obtained for EV, SPInE also computes the Expectation of the Expected Value (EEV) problem over all the scenarios. The optimum objective function value for these models are set out below :

- **WS**: 67189.4
- **HN**: 61354.4
- **EV**: 54846
- **EEV**: 60567.2

These are then used to compute the stochastic measures Expected Value of Perfect Information (EVPI) and Value of Stochastic Solution (VSS). These measures are designed to indicate whether or not randomness has much impact on the optimum decisions computed for the given problem :

- **EVPI =** (WS-HN) = 5834.91774172
- **VSS** = (HN-EEV) = 787.229642443

These results, however, are based at the assumption that the scenario tree, which is used adequately and accurately represents the model's uncertainty. Simulation is used to verify this assumption and test the reliability of the scenario generation method in terms of stability. Other approaches such as the contamination technique described in (Dupacova et. al 2000) are also used to perform an analysis of the robustness of the optimal value of a stochastic programming model.

5.5   Simulation: Stability of the scenario generation

Stability

For a scenario-generation method there are two desirable aspects which need consideration. Since the first stage Here-and-Now decisions are influenced by the randomness of the parameter, the first aspect is in-sample stability; that is, if we generate several trees (with the same input) and solve the stochastic optimisation problem with these trees, we should get nearly the same optimal values. The second aspect is that the scenario tree should not introduce any bias, compared to the true solution (out-of-sample stability).

In- sample stability

In sample stability (see Kaut and Wallace, 2003) therefore assures the "relative robustness" of a scenario generation method. The main purpose of a scenario generator is to provide a discrete approximation of a given stochastic process (the model of randomness). If the number of scenarios is too small, however, the corresponding scenario tree may not be able to represent the underlying random process appropriately, and different scenario trees of the same size may lead to very different solutions. The in-sample stability is measured as the distance of the objective functions obtained by solving the same problem with different scenario trees (of the same size) generated with the same method. In other words, the scenario generation method is compared with itself. In our case study we decided to adopt a three-stage tree with fifty scenarios in each stage. Subsequently we carried out one hundred simulation runs in order to measure the stability. In each simulation, we generate a scenario tree and use SPInE to solve the relating model instance. The distribution of the objectives obtained by the simulation is used to extract the following measures of stability.

Stability measured by :	Value
Min	57344.52844
Max	60055.20384
Range	2710.675393
Mean	59419.04298
Stdev	492.7932547
Relative Max Deviation	4.56%
Relative Mean Deviation	0.83%

Table 4 Stability Table

- **Min :** Min represents the minimum objective value of all the simulation runs (in our case 100)
- **Max :** Max represents the maximum objective value of all the simulation runs
- **Range :** The value "Range" is the difference between Max and Min and represent the maximum spread between all the runs
- **Mean :** We simply computed the Mean of all the objective values.
- **Stdev :** Stdev is the Standard Deviation of all the objective values
- **Relative Max Deviation :** this measure is expressed by the fraction between the Range and the Mean. In other words $RMD = \dfrac{Range}{Mean}$
- **Relative Mean Deviation :** this measure is expressed by the fraction between the Standard Deviation and the Mean. In other words $RMnD = \dfrac{Stdev}{Mean}$

We observe that the value given by the RMnD is less than 1%, therefore can assume that the scenario generation method used in this study is stable. Use of larger samples in the simulation runs, will provide more reliable values of stability measures with a smaller error interval.

5.6   Simulation: Computation of risk measures

Simulation is also used to evaluate the outcome of a given decision. Another set of simulation runs are therefore undertaken to analyse the risk profile (measured in terms of VaR, CvaR and variance) of three sets of decisions for our ALM model; these decisions are obtained by solving: (i) The HN optimum decision (multistage), (ii) The HN optimum decision (twostage), (iii)The EV optimum decision.

We use a scenario tree with 2.500 scenarios to compute the three solutions above. Subsequently we generate a larger tree with 10.000 scenarios to simulate their performance and obtain the distributions of the objective function values for each set of stage decisions.

In Table 5 we set out the results obtained by the second set of simulations runs which use 10.000 scenarios:

	EV	HN (MS)	HN (TS)
VaR	44537.35	48370.01	48358.62
CvaR	41419.95	45054.09	45065.22
Variance	113806893.41	52421392.19	52374316.20
Mean	60798.51	61413.73	61413.96

Table 5 Simulation Results

- **EV :** Expected Value Problem
- **HN (MS) :** Here-and-Now Problem Multistage Scenario tree
- **HN (TS) :** Here-and-Now Problem Two-stage Scenario tree

The Table 10 displays the summary of each run: The distribution of the objective function values for the three sets of decisions are shown in figure 18.
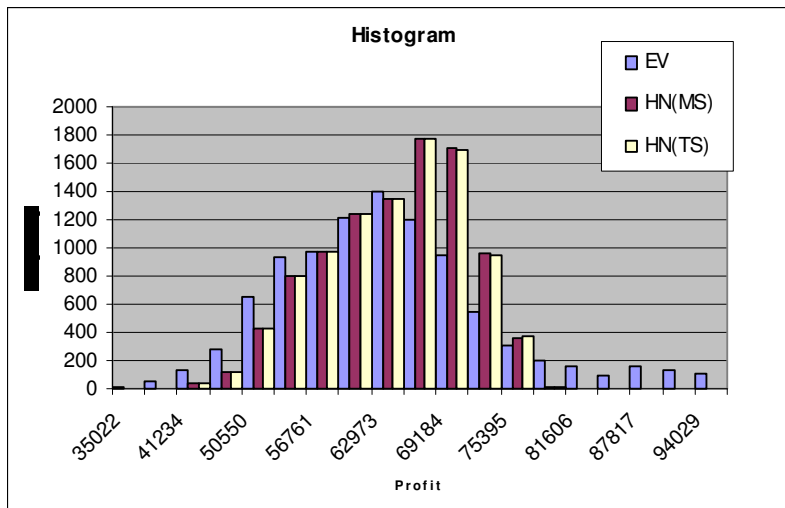

Figure 17 EV, HN(MS) and HN(TS)

It is easily seen that the SP approach provides solutions which are much more robust than those obtained by deterministic EV approach. Indeed the variance is almost half, while in terms of VaR and Cvar, the HN solutions outperform the EV solution by about 10%.

6   Discussions and Conclusions

The success of Linear Programming and Mixed Integer Programming has in turn fuelled considerable interest in the study of Stochastic Programming and more recently Stochastic Mixed Integer Programming. (SMIP). SP has wide ranging applications in situations where uncertainty and risk are taken into consideration in the planning process.

A natural evolution of the SP and SMIP models are to bring together (optimum) decision making with simulation evaluation. Our future research focus is therefore to develop modelling and solution environments (Valente et al 2004, Poojari et al 2004) which enable the problem owners to apply a "decision engine" with which they can study "hedged" decisions and "risk" decisions under conditions of uncertainty. For instance see Dempster (2003) and Ziemba et al. (1998)

There is thus considerable interest and growing adoption of SP as an important decision making tool. In this paper we have taken an integrated view of SP with data and information modelling as well as simulation modelling. Although we use well established modelling and solution components in our system, our approach is highly innovative, since we bring a completely new perspective to the entire modelling process.

Our case study is carefully chosen and underpins our belief that **ex ante** decision modelling should be closely coupled with **ex post** simulation modelling as this supports the decision maker (problem owner) to gain more confidence in the modelling process. The connection to data and information modelling is equally important, since this enables the problem owner to extend systems into an embedded application of "business analytics".

The system architecture described in this paper is "desktop" oriented and uses well defined internal system components which incorporates modelling language, scenario generation, data interchange.

The emerging trend is towards open architecture of web services (Valente et al. 2004). The advantage of this approach is that "best of breed" established components developed by different researchers and developers can be connected together to rapidly create proof of concept quality applications.

7 References

1. Alonso-Ayuso A., Escudero L.F., Garín A., Ortuño M.T. and Pérez G., An Approach for Strategic Supply Chain Planning under Uncertainty based on Stochastic 0-1 Programming, Journal of Global Optimization, 26:97-124, 2003.

2. Arrow K.J., Fisher A.C.: *Environmental Preservation, Uncertainty, and Irreversibility* Quarterly Journal of Economics, vol.88, no 1, May 1974, pgs.312-319 (1974)

3. Andrade C., Lisser A., Maculan N. et Plateau G. Simulation on the Integer Capacity Planning under Uncertain Demand Problem in Telecommunication Networks. In EUROSIM 2002, Delft - The Netherlands, pages In CD, (2002)

4. Berson A., Smith S.J., Data Warehousing, Data Mining, and OLAP, McGraw-Hill, (1997).

5. Birge JR., Louveaux F. Introduction to Stochastic Programming (Springer Series in Operations Research) by John R. (1997)

6. Birge J.R. and Wets R.J-B: Designing approximation schemes for stochastic optimisation problems, Mathematical programming study 27, 54-102. (1986).

7. Bisschop J, AIMMS Manual. www.aimms.com (2001)

8. Bonatti, M. and Gaivoronsky, A.A. Worst case analysis of ATM sources with application to access engineering of broadband multiservice networks. Proc. International Teletraffic Congress (ITC-14), 559-569 (1994)

9. Bosetti V., Messina E. and Valente P. Optimization technologies and environmental applications IMA Journal pp. 167-185, (2002)

10. Carino D. R.., Kent T., Myers D. H., Stacy C., Sylvanus M., Turner A. L., Watanabe K., and Ziemba W. T., "The Russell-Yasuda Model: An asset/liability model for a Japanese insurance company using multistage stochastic programming", *Interfaces*, January-February (1994)

11. Cariño D. R. and Turner A. L: Multistage Planning for Asset allocation, World Wide Asset and Liability Modeling (W. T. Ziemba and J. M. Mulvey, eds.), Cambridge University Press, (1997).

12. CARISMA Stochastic Programming workshop notes (2000).
http://www.carisma.brunel.ac.uk

13. Cerisola S., Ramos A., Benders Decomposition for Mixed-Integer Hydrothermal Problems by Lagrangean Relaxation Proceedings 14th Power Systems Computation Conference (PSCC '02) ISBN 84-89673-25-X Session 05, Paper 6, Pages 1-8 Seville, Spain (2002)

14. Clarke, H.R. and Reed, W.J., Land Development and Wilderness Conservation Policies under Uncertainty: A synthesis, Natural Resources Modelling, 4. 11-37. (1990)

15. Codd E.F., Codd S.B., Salley C.T., Providing On-Line Analytical Processing to User-Analysts: An IT Mandate, White Paper, E.F. Codd and Associates, (1993)

16. Coggins, J. S. and Ramezani, C.A.. "An Arbitrage-Free Approach to Quasi-Option Value". *Journal of Environmental Economics and Management.* 35, 103-125. (1998)

17. Conrad J.. "Quasi-option value and the expected value of information". Quarterly Journal of Economics. 94, 813-820. (1980)

18. Consigli G, Dempster M. The CALM stochastic programming model for dynamic asset/ liability management. In: W.T. Ziemba and M.J. Mulvey (eds). Worldwide asset and Liability Modeling (Cambridge University Press) (1998).

19. Crainic, T., Gendreau, M. and Dejax, P., 'Dynamic stochastic models for the allocation of empty containers', Operations Research 41, 102–126. (1993)

20. Dantzig, G., 'Linear programming under uncertainty', Management Science 1, 197–206. (1955)

21. Dejax, P. and Crainic, T., 'A review of empty flows and fleet management models in freight transportation', Transportation Science 21, 227–247. (1987)

22. De Lange P., Fleten S. E., Gaivoronski A.A., Modeling financial reinsurance in the casualty insurance business viastochastic programming in: Financial decision models in a dynamical setting. Editors G. Mitra, S.Zenios. Journal of Economics Dynamics and Control, vol 28, 991-1012. (2004)

23. Demarest M., Building the Data Mart, DBMS, (1994).

24. Dempster. M. STOCHGEN Manual. http://www-cfr.jims.cam.ac.uk/

25. Dempster M. and Consigli G. World Wide Asset and Liability Modelling. J. M. Mulvey and W. T. Ziemba, eds. Cambridge: Cambridge University Press 464-500 (1998)

26. Dempster, M.A.H., Hicks-Pedron, N., Medova, E., Scott, J. and Sembos, A. "Planning logistics operations in the oil industry." *Journal of the Operational Research Society*, 51(11): 1271-1288 (2000)

27. Dempster M., Germano, Medova and Villaverde. Global asset liability management. British Actuarial Journal 9. 137-216. (2003)

28. Dempster, M, and Thompson R. T. EVPI-based importance sampling solution procedures for multistage stochastic linear programmes on parallel MIMD architectures. Annals of Operations Research 90(1): 161-184 (1999)

29. Di Domenica N., Mendi M., Mitra G., Valente P., "A Review of Scenario Generation for Stochastic Programming (SP) and Simulation: A Modelling Perspective". CARISMA report CTR 31 in preparation (2004)

30. Dominguez-Ballesteros, B., Mitra G., "Modelling and Solving Environments for Mathematical Programming (MP): A status review and new directions." Journal of the Operational Research Society 53(10): 1072-1092. (2002)

31. Dupacova J, Bertocchi M., Moriggia V. "Sensitivity of bond portfolio with respect to random movements in yield curve: a simulation study", Annals of Operations Research 99, 267-286. (2000)

32. Dupacova J., Consigli G., Wallace S.W., "Scenarios for multistage stochastic programs", Annals of Operations Research 100, 25-53, (2001).

33. Dupacova J., Growe-Kuska N., Romisch W., "Scenario reduction in stochastic programming: An approach using probability metrics", Math. Programming A95, 493-511, (2003)

34. Dupacova, J. "Portfolio optimization via stochastic programming: Methods of output analisys", Mathematical Methods of Operational Research, 50, 245-270, (1999)

35. Eppen G.R., Martin R.K and Schrage L., "A Scenario Approach to Capacity Planning" Operations Research 17:4, pp. 517-527 (1989)

36. Ermoliev Y. Stochastic quasigradient methods. In Y. Ermoliev and R.J.-B. Wets editors. Numerical Techniques for Stochastic Optimization. pages 141- 185. Springer-Verlag. Berlin (1988)

37. Escudero L., Galindo E., Garcia G., Gomez E. and Sabau V. Schumann modeling framework for supply chain management under uncertainty. European Journal of Operational Research. Vol. 119. 14-34. (1999).

38. Fantauzzi F.; Gaivoronski A.; Messina E. Decomposition methods for network optimization problems in the presence of uncertainty. Lecture notes in economics and mathematical systems. 450: Network optimization. Red. Pardalos, Panos M.; Hearn, Donald W.; Hager, William W. Springer, 234-248 1997 ISBN 3-540-62541-0 ISSN 0075-8442 0

39. Fourer. R. AMPL manual. www.ampl.com

40. Fragniere E. SETSTOCH.
http://www.unige.ch/hec/logilab/templeet/template/papiers/papier30spistoch.pdf

41. Frauendorfer K., Barycentric Scenario Trees in Convex Multistage Stochastic Programming, Mathematical Programming 75, 277-293, (1996)

42. Gassmann. H. MSLiP. http://www.mgmt.dal.ca/sba/profs/hgassmann/

43. Gassmann, H. Optimal Harvest of a forest in the presence of uncertainty. Working PaperG. School of Business Administration Dalhousie University, Halifax, Canada. 59. (1987),

44. Godfrey, G. and Powell, W. B., An adaptive, dynamic programming algorithm for stochastic resource allocation problems I: Single period travel times, Transportation Science 36(1), 21–39. 23, 28 (2002)

45. Godfrey, G. A. and Powell, W. B., 'An adaptive, distribution-free approximation for the newsvendor problem with censored demands, with applications to inventory and distribution problems', Management Science 47(8), 1101–1112. (2001)

46. Gray J, Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals, MS Technical Report, (1995)

47. Gupta A., Harinarayan V., and Quass D.. Aggregate-Query Processing in Data Warehousing Environments. In Proceedings of 21th International Conference on Very Large Data Bases, pp. 358-369, (1995).

48. Hanemann, W.M. "Information and the concept of option value". Journal of Environmental Economics and Management. 16. 23-37. (1989)

49. Hearst M.A., "Modern Information Retrieval - Chapter I0: User Interfaces and Visualization", Addison Wesley Longman, (1999).

50. Higle J, Wallace S. Managing risk in the new power business - a sequel. IEEE Computer Applications in Power April 12-19 (2002).

51. Higle, J., Sen S. Stochastic decomposition: a statistical method for large scale stochastic linear programming. Dordrecht, Boston (1996).

52. Higle J, Sen S. Multistage Stochastic Programs: Duality and Its Implications, to appear Annals of Operations Research (2004).

53. Høyland K, Kaut M, Wallace S., A Heuristic for Moment-Matching Scenario Generation, Computational Optimization and Applications vol. 23, no. 2-3, Feb/Mar 2003, pp. 169-185 (2003)

54. Høyland K. and Wallace S.W., Generating scenario trees for multi-stage decision problems. Management Science 47 295—307, (2001).

55. Infanger. G. DECIS Manual. www.gams.com/solvers/decis.pdf

56. Infanger. G. Planning under uncertainty-Solving large scale stochastic linear programs. Boyd and Fraser Publishing company. (1994)

57. Inmon W H, "The Operational Data Store." INFODB, Vol. 9, No. 1, (1995).

58. Jobst N., Horniman M.H., Lucas C. and Mitra G., Computational Aspects of Alternative Portfolio Selection Models in the Presence of discrete Asset Choice Constraints, Quantitative Finance Vol. 1. (2001)

59. Jordan, W. and Turnquist, M., 'A stochastic dynamic network model for railroad car distribution', Transportation Science 17, 123–145. (1983)

60. Kall P. SLP-IOR. http://www.unizh.ch/ior/Pages/Deutsch/Mitglieder/Kall/bib/ka-may-92a.pdf

61. Kall, P. and Wallace, S. *Stochastic Programming.* Wiley, Chichester. (1994)

62. Kallberg, J. G., White R. W., Ziemba. W. T.. Short term financial planning under uncertainty. *Management Sci.* **28** 670-682. (1982)

63. Kaut M. and Wallace S., Evaluation of scenario-generation methods for stochastic programming. Posted as SPEPS 14-2003.

64. Kimball R., The Data Warehouse Toolkit, Wiley, (1996).

65. King A. IBM OSL. http://www-306.ibm.com/software/data/bi/osl/

66. Koutsoukis, N., Mitra, G., and Lucas, C. Adapting on-line analytical processing for decision modelling: The interaction of information and decision technologies. Decision Support Systems 26, 1–30. (1999)

67. Koutsoukis N, Mitra G :Decision Modelling and Information Systems: The Information Value Chain, Kluwer Academic Pub (2003)

68. Kristjansson. B. MPL Manual. www.maximal-software.com

69. Kusy M.I. and Ziemba WT ``A Bank Asset and Liability Management Model,'' Operations Research, 34(3), pp. 356-376 (1986)

70. Kyriakis T.: A Stochastic Programming approach to asset and liability management, Ph.D Thesis. (2001)

71. Laporte G., Louveaux F. V., The integer L-shaped methods for stochastic integer programs with complete recourse, Operations Research Letters, 13, pp. 133-142. (1993)

72. Luenberger D.G.: Investment Science, Oxford University Press, New York. (1997)

73. Markowitz, H. M., "Portfolio Selection", *Journal of Finance*, 7, p. 77–91. (1952)

74. Meeraus A, GAMS Manual. www.gams.com

75. Messina E., Lucas C. and Mitra G.: Risk and Return analysis of a multiperiod strategic planning problem, in L. Thomas and A. Christers (eds.) Stochastic Modelling in Innovative Manufacturing, Springer Verlag,(81-96). (1996)

76. Messina E., Mitra G., "Modelling and analysis of multistage stochastic programming problems: a software environment" *European Journal of Operational Research*, Vol. 101, pp. 343-359, (1997).

77. Mirhassani SA., Lucas C., Mitra G., Messina E., Poojari C.. Computational solutions to capacity planning under uncertainty, in Parallel Computing Journal (26), 511-538. (2000)

78. Mitra G., Models for Decision Making: An Overview of Problems, Tools and Major Issues, Mitra G. (Editor), Mathematical Models for Decision Support, NATO ASI Series, Springer-Verlag, (1988)

79. Mitra G, Poojari C., Sen S.. Strategic and Tactical planning models for supply chain: an application of stochastic mixed integer programming. To appear in Handbook of Discrete programming and Modelling. Edit. HP Williams. (2004)

80. Mulvey J. Financial planing via multi-stage stochastic programs. In John R. Birge and Katta G. Murty, editors, *Mathematical Programming: State of the Art 1994*, (1994)

81. OSP Project. Optimisation Service Provider. EU CRAFT Project. http://www.osp-craft.com (2003)

82. Pedersen T. B. and Jensen C. S. Multidimensional Data Modelling for Complex Data. In Proceedings of the Fifteenth International Conference on Data Engineering, pp. 336-345, (1999).

83. Pflug G., Hochreiter. R. Scenario generation for multi-stage decision models: an approach based on multidimensional facility location. Technical Report 2003-01, Department of Statistics and Decision Support Systems, University of Vienna, (2003)

84. Pirbhai M., Mitra G., Kyriakis T.. Asset Liability Management using stochastic programming. In: Asset and Liability Management Tools. A Handbook for best practice. Edt. B. Scherer. 95-308. (2003)

85. Poojari C., Lucas C., Mitra. G. A Decision Model for Natural Oil Buying Policy under Uncertainty in: Proceding of Industrial Mat. Conference. Editors: M. Joshi and A. Pani, Springer Verlag (2004).

86. Poojari C., Mitra G., Ellison F., Sen S. Computational investigations of algorithms for processing two-stage Stochastic linear programs. To be submitted INFORMS Journal of Computing (2004)

87. Powell, W. B., 'A stochastic model of the dynamic vehicle allocation problem', Transportation Science 20, 117–129. 28 (1986)

88. Powell, W. B., 'An operational planning model for the dynamic vehicle allocation problem with uncertain demands', Transportation Research 21B, 217–232. 28 (1987)

89. Powell, W. B., A comparative review of alternative algorithms for the dynamic vehicle allocation problem, in B. Golden and A. Assad, eds, 'Vehicle Routing: Methods and Studies', North Holland, Amsterdam, pp. 249–292. 28 (1988)

90. Powell, W. B. and Carvalho T., 'Dynamic control of logistics queuing network for largescale fleet management', Transportation Science 32(2), 90–109. 28 (1998)

91. Powell, W. B., Ruszczynski, A. and Topaloglu, H., Learning algorithms for separable approximations of stochastic optimization problems, Mathematics of Operations Research (to appear) (2002)

92. Powell, W. B., Shapiro, J. A. and Simao, H. P., 'An adaptive dynamic programming algorithm for the heterogeneous resource allocation problem', Transportation Science 36(2), 231–249. (2002)

93. Rafanelli M. and Shoshani A.. STORM: A Statistical Object Representation Model. In Proceedings of the Fifth Conference on Statistical and Scientific Database Management, pp. 14-29, (1990).

94. Römisch W., Stability of Stochastic Programming Problems, in: Stochastic Programming (A. Ruszczynski and A. Shapiro eds.), Handbooks in Operations Research and Management Science Vol. 10, Elsevier, Amsterdam, 483-554. (2003)

95. Santoso, Ahmed, Goetschalckx, and Shapiro. "A Stochastic Programming Approach for Supply Chain Network Design under Uncertainty," submitted for publication, (2003).

96. Schultz R., Integers in stochastic programming, working paper, Institute of Mathematics Gerhard-Mercator University, Duisberg, Germany, to appear in Mathematical Programming. (2003)

97. Schumann. Mitra, Lucas, Poojari and Dominguez-Ballesteros. http://www.supply-chain-optimise.com

98. Sen S., Doverspike R., and Cosares S.,. ``Network Planning with Random Demand," Dept. of Systems and Industrial Engineering, University of Arizona, Tucson, AZ, 85721 (1992)

99. Sethi, A. K. and Sethi S. P. Flexibility in Manufacturing: A Survey. The International Journal of Flexible Manufacturing Systems 2, pp. 289–328. (1990)

100. Shapiro J. F., The Decision Database, Sloan Working Paper WP#3570-93-MSA, (1993).

101. Tomasgard, A., Audestad, J.A., Dye, S., Stougie, L., Van der Vlerk, M., Wallace, S.W. "Aspects on modelling in distributed telecommunication networks." *Annals of Operations Research*, v82,: 161–184 (1998)

102. Topaloglu, H. and Powell, W. B., An algorithm for approximating piecewise linear functions from sample gradients, Technical report, Princeton University, Department of Operations Research and Financial Engineering. (2001)

103. Valente P., Mitra G. SPInE Manual. www.optirisk-systems.com

104. Valente, P., Mitra G. and Poojari. C. "A Stochastic Programming Integrated Environment (SPInE)." in S. W. Wallace and W. T. Ziemba (Editors): to appear in MPS/SIAM Series on Optimisation: Applications of Stochastic Programming (2004)

105. Van der Vlerk M., Stochastic programming with simple integer recourse, in: C. Floudas and P. M. Pardalos (eds.), Encyclopedia of Optimization, Volume V, pages 343-346. Kluwer Academic Publishers, (2001).

106. Vladimirou H. (ed.), "Financial Modeling", Kluwer Academic Publishers (to appear) (2004).

107. Zenios, S. : .Asset and liability management under uncertainty for fixed income securities., Annals of Operations Research. (1995)

108. Ziemba, W.T., Mulvey, J.M., Eds., Asset and Liability Management from a Global Perspective, Cambridge University Press, 665 pages, (1998)

Appendix A: SP Model Definition and Stochastic Measures

Here and Now Problems

The formulation of the classical two-stage SP model with recourse is as follows:

$$Z = \min cx + Q(x)$$

$$subject \quad to \quad Ax = b$$

$$x \geq 0$$

where :

$$Q(x) = \min E_w[q(w)y(w)]$$

$$subject \quad to \quad D(w)y(w) = h(w) + B(w)x$$

$$y(w) \geq 0$$

$$w \in \Omega$$

The matrix A and the vector b are known with certainty. The function $Q(x, \omega)$ is a non-linear term which is referred to as the recourse function. The technology matrix $B(\omega)$, the recourse matrix $D(\omega)$, the right-hand side vector $h(\omega)$, and the vector of objective function coefficients $q(\omega)$ of this linear program may be random. For a given first stage decision x, the corresponding recourse actions $y(\omega)$ are obtained by solving the sub-problem associated with the recourse function $Q(x)$.

As the future unfolds in several sequential steps and subsequent recourse actions are taken, one deals with the generalisation of the two-stage recourse problem, known as multistage stochastic programming problem with recourse. A decision made in stage t should take into account all future realisations of the random parameters and such decision only affects the remaining decisions in stages t+1 … T. In Stochastic Programming this concept is known as non-anticipativity. The general formulation of a multistage recourse problem is set out in the equations below:

$$Z_{HN} = \min_{x_1} \left\{ c_1 x_1 + E_{\xi_2}\left[\min_{x_2} c_2 x_2 + E_{\xi_3|\xi_2}\left[\min_{x_3} c_3 x_3 + ... + E_{\xi_T|\xi_{T-1}|...|\xi_2} \min_{x_T} c_T x_T \right]\right]\right\}$$

subject to:

$$
\begin{aligned}
A_{11}x_1 & & & = b_1 \\
A_{21}x_1 + & A_{22}x_2 & & = b_2 \\
A_{31}x_1 + & A_{32}x_2 + & A_{33}x_3 & = b_3 \\
\vdots & & \ddots \qquad\qquad \vdots & \\
A_{T1}x_1 + & A_{T2}x_2 + & A_{T3}x_3 + \quad ... \quad + A_{TT}x_T & = b_T \\
\end{aligned}
$$

$$\ell_t \leq x_t \leq u_t;$$

where: $t = 1,...,T$ represents the stages in the planning horizon and the vectors:

$$\xi_t = (b_t, c_t, A_{t1},...A_{tT}) \quad \forall t \in [2,...,T]$$

are random vectors on a probability space $(\Omega, \mathfrak{J}, P)$.

Stochastic Measures

It can be shown that the three objective function values Z_{EEV}, Z_{HN}, Z_{WS} are connected by the following ordered relationship:

$$Z_{WS} \leq Z_{HN} \leq Z_{EEV}$$

The inequality:

$$Z_{HN} \leq Z_{EEV}$$

can be argued in the following way: any feasible solution of the average value approximation is already considered in the Here and Now model, therefore the optimal Here and Now objective must be better.

The value of the stochastic solution (VSS)

The difference between these two solutions defines the **Value of the Stochastic Solution** (VSS):

$$VSS = Z_{EEV} - Z_{HN}$$

This is a measure of how much can be saved by implementing the (computationally expensive) Here and Now solution as opposed to the deterministic expected value solution. The practical computation of VSS is strictly related to the approach used in the computation of Z_{EEV}.

The expected value of perfect information (EVPI)

Another important index is represented by the **Expected Value of Perfect Information** (EVPI):

$$EVPI = Z_{HN} - Z_{WS}$$

This property of stochastic optimisation problems is interpreted as the expected value of the amount the decision maker is willing to pay to have perfect information (i.e. knowledge) about the future scenarios. A relatively small EVPI indicates that better forecasts will not lead to much improvement; a relatively large EVPI means that incomplete information about the future may prove costly.

Bounds on EVPI and VSS

Some useful bounds on the EVPI and VSS are presented below:

$$0 \leq EVPI \leq Z_{HN} - Z_{EV} \leq Z_{EEV} - Z_{EV}$$

$$0 \leq VSS \leq Z_{EEV} - Z_{EV}$$

These can help in estimating the relative benefit of implementing the computationally costly Stochastic Programming solution, as opposed to approximate solutions obtained by processing the Expected Value LP problem.