Asset and Liability Management model

Notes

1. Deterministic Model

Common portfolio management models deal with the maximisation of the financial well being of an investor or institution. Passive portfolio management lacks flexibility and foresight, two elements of prominent significance in today's highly volatile financial markets. Furthermore, frameworks such as *Mean-Variance* and *Minimum Absolute Deviation* employed for the optimal fixed mix strategy, exhibit weaknesses in the estimation of accurate portfolio weights. On the other hand, active portfolio management is nowadays gaining momentum among practitioners. Active portfolio management breaks away from myopic static tactics and implies revisiting the initial investment strategy, and re-balancing of the portfolio positions as financial conditions change. Carino and Turner (1998) illustrate the superiority of the dynamic asset allocation framework, in the form of a stochastic programming application, in contrast to the static M-V.

Assets only however, do not influence and fully represent the financial situation of the investor. Integrated risk management demands the inclusion and modelling of liabilities and goals. Any financial planning strategy should not only try to achieve the best possible return. The decision should be such that the asset-classes-mix in the portfolio manages to obtain a growth capable of satisfying future obligations and goals with the best possible return. This is known as Asset and Liability Management.

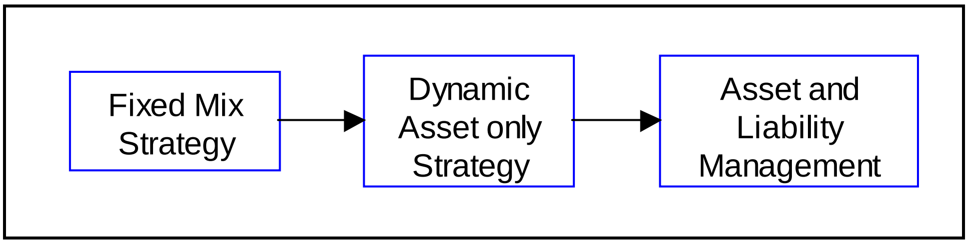


Figure 1: The Evolution of Strategic Risk Management

Asset and Liability Management becomes more effective when applied in conjunction with stochastic programming models with recourse. Major empirical ALM applications utilising stochastic programming include 'The CALM stochastic programming model for dynamic asset-liability management' (Consigli & Dempster 1998), and 'The Russell-Yasuda Kasai Model' (Carino et al. 1994).

Questions/Notes

Notes

1.1. *Conceptualisation*

The asset/liability management model considered in this course can be stated as follows:
An investor faces the problem of creating a portfolio allocating a set of assets belonging to a universe *I*. Each assets class is characterised by a price *P*. The goal of the investor is to maximise the portfolio wealth at the end of a predefined time horizon *T*. He needs to take into account future obligations (liabilities) *L*, and the fact that each trade has an associated transaction cost expressed by the fraction *g*. In each time period of the time horizon, and for each asset considered, the investor needs to decide:

- The amount of assets to buy
- The amount of assets to sell
- The amount of assets to hold

We can therefore define the indices, the problem parameters and the decision variables as set out below:

Table 1. Problem dimensions

Problem dimensions			
Index	**Notation**	**Description**	**Range**
assets	I	assets classes	i = 1 ... I; I=10
tp	T	Time periods	t = 1 ... T; T=4

Table 2. Problem parameters

Problem parameters		
Name	**Notation**	**Description**
Price	$P_{i\,t}$	Price of asset i at time period t
Liabilities	L_t	Liability at time period t
initialholdings	H_{i0}	Initial composition of the portfolio
Income	F_t	Funding in time period t
Tr	G	Transaction cost as % of trade value

Questions/Notes

Notes

Table 3. Decision variables

Problem decision variables		
Name	**Notation**	**Description**
amounthold	H_{it}	Quantity of assets i to hold in time period t
amountsell	S_{it}	Quantity of assets i to sell in time period t
amountbuy	B_{it}	Quantity of assets i to buy in time period t

1.2. Data modelling

In a deterministic version of the model, the investor may compute the expected return for each asset by looking at the historical data and assuming that the returns follow a normal distribution. The values obtained are showing in the next table.

Table 4. Expected asset prices

Asset Prices $P_{i\,t}$										
Time period	1	2	3	4	5	6	7	8	9	10
1	29.69	27.69	41.61	38.12	27.73	20.38	75.38	21.77	48.13	31.94
2	32.41	28.66	42.58	39.57	28.72	20.13	76.36	22.31	50.54	33.04
3	34.03	29.46	43.56	40.82	29.68	20.37	79.80	22.87	51.99	34.67
4	35.54	30.78	44.77	42.93	31.13	20.78	86.53	23.71	53.54	36.80

The investor assumes that the value of the future liabilities is known with certainty and that there will be no funding in the future, but only an initial capital of £150,000.

Table 5. Liabilities and funding

	Liabilities and Funding	
Time period	**Income**	**liability**
1	150000	0
2	0	1000
3	0	1200
4	0	1250

The portfolio is initially empty, therefore the initial holdings H_{i0} are 0 for each asset i, while the transaction cost is assumed to be g = 2.5% of the value of each trade.

Questions/Notes

Notes

1.3. *Algebraic model*

Asset holding constraints

During the planning horizon the portfolio is re-balanced at discrete points in time (beginning of each time period). The model buys the assets with the highest return expectation and sells the ones with poor performance. The *asset holding constraint* shows the evolution of the portfolio composition over time. It is divided into two parts. At time-period t=1, when the initial decision is taken, the portfolio either already consists of a known asset mix or the holdings in the portfolio are zero. For time-periods t>1 the amount of each individual asset held in the portfolio is associated to the holding amount for each asset during the previous time-period.

$$H_{it} = H_{i0} + B_{it} - S_{it} \qquad t = 1,\ i = 1\ldots I \qquad (1)$$

$$H_{it} = H_{it-1} + B_{it} - S_{it} \qquad t = 2 \ldots T,\ i = 1\ldots I \qquad (2)$$

Fund balance constraints:

Throughout the planning period cash inflows and cash outflows occur. The former is due to the assets selling or to profitable performance of the assets among with additional funding, which the investor's company might obtain. The later is due to company's payments and other liabilities, which have to be fulfilled as well as to the purchase of assets and the transaction costs associated with their trading (buying and selling). In other words, this constraint reflects the evolution of the cash balance of the investor over time.

$$(1-g)\sum_{i=1}^{I} P_{it} S_{it} - L_t + F_t = (1+g)\sum_{i=1}^{I} P_{it} B_{it} \qquad t = 1 \ldots T \qquad (3)$$

The concept above can be represented as in figure 2.

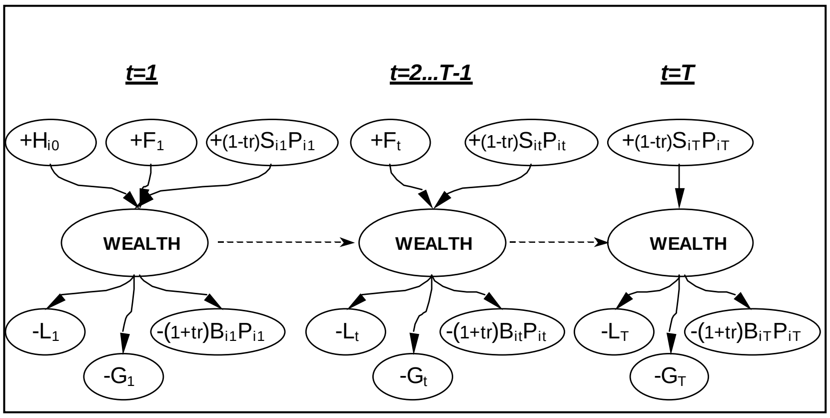


Figure 2: Evolution of investor's funding position through time.

Objective function

The goal of the investor is to maximise the terminal wealth of the portfolio. This can be expressed as in equation (4).

Questions/Notes

$$\max \sum_{i=1}^{I} P_{iT} H_{iT} \qquad (4)$$

Notes

The expression above can be used to calculate the market value of the portfolio for each time-period by substituting T with $t=1..T$.

1.4. Implementation

The formulation of the asset and liability management model can be expressed in AMPL as follows (file *alm_lp1.mod*), where the values for the data vectors are read from tables in a MS Access database.

```
#PARAMETERS: SCALARS
param NA:=10;
param NT:=4;
param tbuy := 1.025;
param tsell := 0.975;
param risklevel:=0.3;

#SETS
set assets := 1..NA;
set tp :=1..NT;

#PARAMETERS : VECTORS (read from database!)
param liabilities{tp};
param initialholdings{assets} := 0;
param income{tp};
param target{tp};
param price{assets, tp};

#VARIABLES
var   amounthold{t in tp,a in assets} >=0;
var   amountbuy{t in tp,a in assets} >=0;
var   amountsell{t in tp,a in assets} >=0;
var   marketvalue{t in tp} >=0 ;

#OBJECTIVE
maximize wealth : marketvalue[4];


#CONSTRAINTS
subject to

assetmarketvalue1:
      marketvalue[1]=sum{a in assets}
initialholdings[a]*price[a,1];

assetmarketvalue2{t in 2..NT}:
```

Questions/Notes

```
      marketvalue[t] = sum{a in assets}
amounthold[t,a]*price[a,t];

stockbalance1{a in assets}:
      amounthold[1,a]=initialholdings[a]+amountbuy[1,a]-
amountsell[1,a];

stockbalance2{a in assets,t in 2..NT }:
      amounthold[t,a]=amounthold[t-1,a]+amountbuy[t,a]-
amountsell[t,a];

fundbalance1{t in tp}:
      sum{a in assets} amountbuy[t,a]*price[a,t]*tbuy
       -sum{a in assets} amountsell[t,a]*price[a,t]*tsell=
        income[t]-liabilities[t];
```

1.5. Solution

The solution to this problem given by FortMP is:

```
AmplStudio Modeling System -Copyright (c) 2003 SM Software.
____________________________________________________________

MODEL.STATISTICS

Problem name                      :ALMDET
Model Filename                    :Y:\AMPLStudio\Bin\alm_det.mod
Data  Filename                    :Y:\AMPLStudio\Bin\alm_det.dat
Date                              :10:7:2004
Time                              :13:2

Constraints                       : 24          :  Nonzeros
S_Constraints                     : 23
Variables                         : 62          :  Nonzeros


SOLUTION.RESULT

Optimal_solution_found

'FortMP 3.2j: LP OPTIMAL SOLUTION, Objective = 99282.23931'


DECISION.VARIABLES

  Name                Activity     .uc      Reduced Cost
 ________________________________________________________
amounthold[1,1]      0.0000       Infinity      -0.0339
amounthold[1,2]      0.0000       Infinity     0.0000
amounthold[1,3]      0.0000       Infinity     0.0000
amounthold[1,4]      0.0000       Infinity     0.0000
amounthold[1,5]      10627.5573   Infinity     0.0000
amounthold[1,6]      0.0000       Infinity     0.0000
amounthold[1,7]      0.0000       Infinity     0.0000
amounthold[1,8]      0.0000       Infinity     0.0000
amounthold[1,9]      0.0000       Infinity     0.0000
```

Questions/Notes

Notes

amounthold[1,10]	0.0000	Infinity	0.0000
amounthold[2,1]	0.0000	Infinity	0.0000
amounthold[2,2]	0.0000	Infinity	0.0000
amounthold[2,3]	0.0000	Infinity	0.0000
amounthold[2,4]	0.0000	Infinity	0.0000
amounthold[2,5]	10518.8912	Infinity	0.0000
amounthold[2,6]	0.0000	Infinity	0.0000
amounthold[2,7]	0.0000	Infinity	-0.0930
amounthold[2,8]	0.0000	Infinity	0.0000
amounthold[2,9]	0.0000	Infinity	-0.0527
amounthold[2,10]	0.0000	Infinity	-0.0260
amountbuy[1,1]	0.0000	Infinity	-0.0138
amountbuy[1,2]	0.0000	Infinity	-0.5011
amountbuy[1,3]	0.0000	Infinity	-0.1033
amountbuy[1,4]	0.0000	Infinity	-0.1312
amountbuy[1,5]	10627.5573	Infinity	0.0000
amountbuy[1,6]	0.0000	Infinity	-0.0325
amountbuy[1,7]	0.0000	Infinity	-0.6935
amountbuy[1,8]	0.0000	Infinity	-0.0732
amountbuy[1,9]	0.0000	Infinity	0.0000
amountbuy[1,10]	0.0000	Infinity	0.0000
amountbuy[2,1]	0.0000	Infinity	-0.0120
amountbuy[2,2]	0.0000	Infinity	-0.5303
amountbuy[2,3]	0.0000	Infinity	-0.1256
amountbuy[2,4]	0.0000	Infinity	-0.2128
amountbuy[2,5]	0.0000	Infinity	-0.4840
amountbuy[2,6]	0.0000	Infinity	-0.1979
amountbuy[2,7]	0.0000	Infinity	-0.5957
amountbuy[2,8]	0.0000	Infinity	-0.3533
amountbuy[2,9]	0.0000	Infinity	-0.0226
amountbuy[2,10]	0.0000	Infinity	-0.0096
amountsell[1,1]	0.0000	Infinity	0.0000
amountsell[1,2]	0.0000	Infinity	-0.0278
amountsell[1,3]	0.0000	Infinity	-0.0213
amountsell[1,4]	0.0000	Infinity	-0.0776
amountsell[1,5]	0.0000	Infinity	-0.4604
amountsell[1,6]	0.0000	Infinity	-0.1574
amountsell[1,7]	0.0000	Infinity	0.0000
amountsell[1,8]	0.0000	Infinity	-0.2665
amountsell[1,9]	0.0000	Infinity	-0.0742
amountsell[1,10]	0.0000	Infinity	-0.0352
amountsell[2,1]	0.0000	Infinity	-0.0000
amountsell[2,2]	0.0000	Infinity	0.0000
amountsell[2,3]	0.0000	Infinity	0.0000
amountsell[2,4]	0.0000	Infinity	0.0000
amountsell[2,5]	108.6660	Infinity	0.0000
amountsell[2,6]	0.0000	Infinity	0.0000
amountsell[2,7]	0.0000	Infinity	-0.0930
amountsell[2,8]	0.0000	Infinity	-0.0000
amountsell[2,9]	0.0000	Infinity	-0.0527
amountsell[2,10]	0.0000	Infinity	-0.0260
marketvalue[1]	0.0000	0.0000	0.0000
marketvalue[2]	99282.2393	Infinity	0.0000

CONSTRAINTS

Name	Slack	body	dual

Notes

Questions/Notes

```
assetmarketvalue1    0.0000        0.0000        0.0000
assetmarketvalue2[2] 0.0000        0.0000        1.0000
stockbalance1[1]     0.0000        0.0000        0.2682
stockbalance1[2]     0.0000        0.0000        10.3400
stockbalance1[3]     0.0000        0.0000        2.4500
stockbalance1[4]     0.0000        0.0000        4.1500
stockbalance1[5]     0.0000        0.0000        9.4385
stockbalance1[6]     0.0000        0.0000        3.8600
stockbalance1[7]     0.0000        0.0000        13.5230
stockbalance1[8]     0.0000        0.0000        6.8900
stockbalance1[9]     0.0000        0.0000        1.5207
stockbalance1[10]    0.0000        0.0000        0.7209
stockbalance2[1,2]   0.0000        0.0000        0.2343
stockbalance2[2,2]   0.0000        0.0000        10.3400
stockbalance2[3,2]   0.0000        0.0000        2.4500
stockbalance2[4,2]   0.0000        0.0000        4.1500
stockbalance2[5,2]   -0.0000       0.0000        9.4385
stockbalance2[6,2]   0.0000        0.0000        3.8600
stockbalance2[7,2]   0.0000        0.0000        13.5230
stockbalance2[8,2]   0.0000        0.0000        6.8900
stockbalance2[9,2]   0.0000        0.0000        1.5207
stockbalance2[10,2]  0.0000        0.0000        0.7209
fundbalance1[1]      -0.0000       100000.0000   1.0031
fundbalance1[2]      0.0000        -1000.0000    1.0256
 ________________________________________________________

END
```

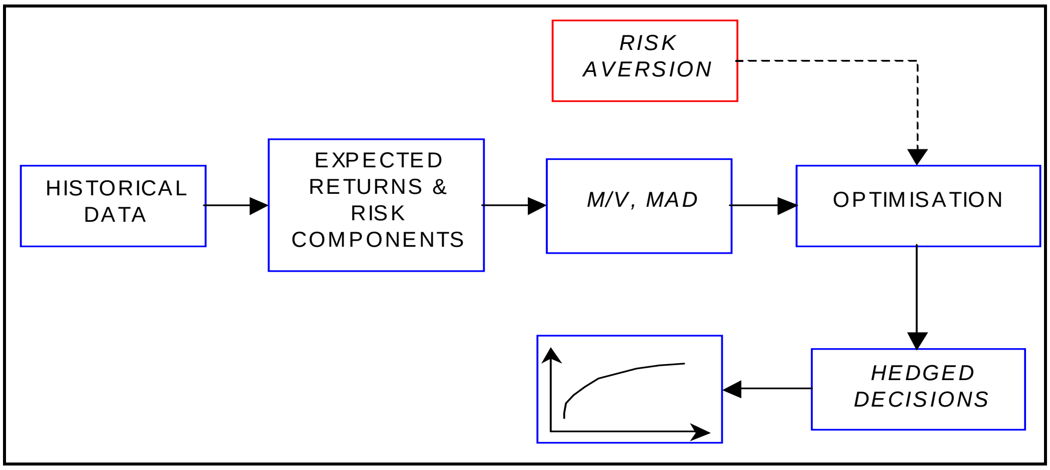
Notes

Questions/Notes

Notes

2. Introducing uncertainty

The deterministic models previously investigated provide optimal decisions assuming the model parameters are known with certainty. However, it has been shown how a wrong estimation of these parameters (the future asset prices in our asset and liability management model) can lead to a non-optimal decision, which translates into a loss of profits. The uncertainty in the asset prices has to be considered and introduced in the model in order to obtain a robust solution.
In the deterministic case, the investor makes assumptions on the expected returns based directly on historical data. Additionally, the required components for the underlying measure of risk are calculated. The next step, as shown in figure below, is to select the appropriate model, which is usually related with the investor's standpoint towards risk. Finally, the risk profile of the investor and the underlying optimisation model will result in the optimal hedged decision.



Modelling Architecture - The deterministic case

A first refinement of the framework illustrated above is to introduce a more sophisticated method for the estimation of the asset prices. Empirical evidence demonstrates that the normal approximation employed earlier does not reflect the reality, as the returns of the financial assets exhibit fat tails (rare events do occur in the markets!), and the volatility, described as the standard deviation, is not itself constant. The sub-models of randomness, introduced in stochastic programming, do not impose such restrictions. This results in more accurate representation of the behaviour of assets.
The (optimisation) decision model is therefore coupled with a model of randomness. The modelling of randomness utilises the set of available past data with the aim of building sub-models for each individual stochastic parameter. These sub-models are then employed to generate a set of scenarios that

Questions/Notes

Notes

encapsulate the modeller's perception about the future. On the other hand, a deterministic model representing the institution's or investor's particular needs and constraints has to be modelled initially. This is represented in a linear form and includes a risk measure usually in the form of a penalty function. The linear programming representation coupled with the model of randomness implied by the scenarios provides the instrument for dealing with uncertainty. However, once scenarios have been calculated, they have to be incorporated into the optimisation model. The easiest way to do so is to create the Expected Value Problem.

2.1. Problem

Using the scenario generator illustrated in the earlier lecture, we have obtained 64 possible future outcomes over a time horizon of 4 trimesters for the assets taken into account in our ALM model. We assume that each scenario can occur with the same probability P= 1/64. These values are stored in a table a MS Access database

The optimisation model that we consider is a refinement of the ALM model earlier investigated. The introduction of the scenarios requires the addition to the model of a new index *scenarios,* ranging from 1 to S=64. The prices are now indexed over this dimension. We also introduce a constraint to control the *downside risk*. This is measured as the deviation of our portfolio wealth from a predefined target A_t.
Downside risk, as opposed to the mean variance framework, penalises only under-performance. The investor, according to this approach, is not interested in avoiding investment opportunities with very high levels of return but is only concerned in investments that will drive the portfolio to smaller returns. As a result of the non-parametric nature of the downside risk, financial derivatives such as option and future contracts can be added to the list of asset classes, which the investor can include as constituents of his portfolio. A risk aversion parameter is introduced to calculate the trade off between risk and return.
As explained in the previous lecture, the expected value approach introduces the uncertainty represented by the set of scenarios, by substituting the random parameters with their expected value. In our case, the price of the assets is the only random parameters, therefore:

$$eprice_{it} = \sum_{s=1}^{Sc} price_{its} p_s \qquad \forall t, \forall i \qquad (1)$$

Where:

- $eprice_{it}$ is the expected price of asset i in time period t
- $price_{its}$ is the price of asset I in time period t under scenario s
- p_s is the probability of scenario s to occur ($0<=p<=1$ and $\Sigma p=1$).

Questions/Notes

Notes

2.2. Algebraic formulation

The algebraic model for the Expected Value problem consists in the following:

Non-anticipativity constraints

$$H_{its} = H_{its+1} \qquad t = 1,\ i = 1\ldots I, s = 1...Sc-1 \qquad (12)$$

$$B_{its} = B_{its+1} \qquad t = 1,\ i = 1\ldots I, s = 1...Sc-1 \qquad (13)$$

$$S_{its} = S_{its+1} \qquad t = 1,\ i = 1\ldots I, s = 1...Sc-1 \qquad (14)$$

Asset holding constraints

$$H_{it} = H_{i0} + B_{it} - S_{it} \qquad t = 1,\ i = 1\ldots I \qquad (2)$$

$$H_{it} = H_{it-1} + B_{it} - S_{it} \qquad t = 2...T,\ i = 1\ldots I \qquad (3)$$

Fund balance constraint:

$$(1-g)\sum_{i=1}^{I} eprice_{it}S_{it} - L_t + F_t = (1+g)\sum_{i=1}^{I} eprice_{it}B_{it} \qquad t = 1...T \qquad (4)$$

Downside Risk Constraint

$$\frac{A_t - \sum_{\mathrm{i}=1}^{\mathrm{I}} eprice_{\mathrm{it}}H_{it}}{A_t} \le \mathrm{R}_t \qquad t > 1 \qquad (5)$$

where:

- A_t is the predefined target for time period t
- R_t is expressed as a positive fraction and specifies the maximum deviation from the target accepted by the investor

Objective function

$$\max \sum_{\mathrm{i}=1}^{\mathrm{I}} eprice_{\mathrm{iT}}H_{iT} \qquad (6)$$

2.3. AMPL formulation

The expected value problem can be formulated in AMPL as follows:

```
#PARAMETERS: SCALARS
param NT:=2;
param NS:=64;
param NA:=10;
param tbuy := 1.025;
param tsell := 0.975;
param risklevel:=0.3;

#SETS
set assets := 1..NA;
```

Questions/Notes

```
set tp :=1..NT;

#SCENARIO
set scen:=1..NS;


#RANDOM PARAMETERS
param price{assets, tp, scen};

#PROBABILITIES
param Prob{scen}:=1/NS;

#PARAMETERS : VECTORS (read from database!)
param liabilities{tp};
param initialholdings{assets} := 0;
param income{tp};
param target{tp};


#STAGES
suffix stage LOCAL;

#VARIABLES
var   amounthold{t in tp,a in assets,s in scen} >=0;# , suffix
stage t;
var   amountbuy{t in tp,a in assets,s in scen} >=0;#, suffix
stage t;
var   amountsell{t in tp,a in assets,s in scen} >=0;#, suffix
stage t;
var   marketvalue{t in tp,s in scen} >=0 ;#        , suffix
stage t;


#OBJECTIVE
maximize wealth : sum{s in scen} Prob[s]*marketvalue[2,s];

#CONSTRAINTS
subject to

assetmarketvalue1{s in scen}:
      marketvalue[1,s]=sum{a in assets}
initialholdings[a]*price[a,1,s];

assetmarketvalue2{t in 2..NT,s in scen}:
      marketvalue[t,s] = sum{a in assets}
amounthold[t,a,s]*price[a,t,s];

stockbalance1{a in assets,s in scen}:
      amounthold[1,a,s]=initialholdings[a]+amountbuy[1,a,s]-
amountsell[1,a,s];

stockbalance2{a in assets,t in 2..NT, s in scen}:
      amounthold[t,a,s]=amounthold[t-1,a,s]+amountbuy[t,a,s]-
amountsell[t,a,s];
```

Questions/Notes

Notes

Notes

```
fundbalance1{t in tp,s in scen}:
      sum{a in assets} amountbuy[t,a,s]*price[a,t,s]*tbuy
       -sum{a in assets} amountsell[t,a,s]*price[a,t,s]*tsell=
        income[t]-liabilities[t];

zeta{ t in 2..NT,s in scen}: target[t]-
marketvalue[t,s]<=risklevel*target[t];

#NON ANTICIPATIVITY CONSTRAINTS
na1{t in tp,a in assets,s in 2..NS}: amounthold[1,a,s]=
amounthold[1,a,s-1];
na2{t in tp,a in assets,s in 2..NS}: amountbuy[1,a,s]=
amountbuy[1,a,s-1];
na3{t in tp,a in assets,s in 2..NS}: amountsell[1,a,s]=
amountsell[1,a,s-1];
```

2.4. Solutions

The solution for the Expected Value Problem is:

```
            AmplStudio Modeling System -Copyright (c) 2003 SM Software.
            ____________________________________________________________

      MODEL.STATISTICS

      Problem name                  :M
      Model Filename                :Y:/Simulazione/almtwo.mod
      Data  Filename                :Y:/Simulazione/almtwo.dat
      Date                          :10:8:2004
      Time                          :17:31

      Constraints                   : 5380               :  Nonzeros
      S_Constraints                 : 5252
      Variables                     : 3968               :  Nonzeros


      SOLUTION.RESULT

      Optimal_solution_found

      'FortMP 3.2j: LP OPTIMAL SOLUTION, Objective = 119368.6897'


      DECISION.VARIABLES

        Name                     Activity              .uc                  Reduced
Cost
       _________________________________________________________
      amounthold[1,1,1]      464576.0743    Infinity       0.0000
      amounthold[1,1,2]      464576.0743    Infinity       0.0000
      amounthold[1,1,3]      464576.0743    Infinity       0.0000
      amounthold[1,1,4]      464576.0743    Infinity       0.0000
      amounthold[1,1,5]      464576.0743    Infinity       0.0000
      amounthold[1,1,6]      464576.0743    Infinity       0.0000

      . . . .
```

Questions/Notes

Notes

amounthold[2,10,57]	0.0000	Infinity	0.0000
amounthold[2,10,58]	0.0000	Infinity	-0.0000
amounthold[2,10,59]	0.0000	Infinity	0.0000
amounthold[2,10,60]	0.0000	Infinity	0.0000
amounthold[2,10,61]	0.0000	Infinity	-0.0000
amounthold[2,10,62]	0.0000	Infinity	0.0000
amounthold[2,10,63]	0.0000	Infinity	0.0000
amounthold[2,10,64]	0.0000	Infinity	0.0000
amountbuy[1,1,1]	464576.0743	Infinity	0.0000
amountbuy[1,1,2]	464576.0743	Infinity	0.0000
amountbuy[1,1,3]	464576.0743	Infinity	0.0000
amountbuy[1,1,4]	464576.0743	Infinity	0.0000
amountbuy[1,1,5]	464576.0743	Infinity	0.0000
amountbuy[1,1,6]	464576.0743	Infinity	0.0000

. . . .

amountbuy[2,10,46]	0.0000	Infinity	-0.0005
amountbuy[2,10,47]	0.0000	Infinity	-0.0005
amountbuy[2,10,48]	0.0000	Infinity	-0.0005
amountbuy[2,10,49]	0.0000	Infinity	-0.0005
amountbuy[2,10,50]	0.0000	Infinity	-0.0005
amountbuy[2,10,51]	0.0000	Infinity	-0.0005
amountbuy[2,10,52]	0.0000	Infinity	-0.0005
amountbuy[2,10,53]	0.0000	Infinity	-0.0005
amountbuy[2,10,54]	0.0000	Infinity	-0.0005
amountbuy[2,10,55]	0.0000	Infinity	-0.0005
amountbuy[2,10,56]	0.0000	Infinity	-0.0005
amountbuy[2,10,57]	0.0000	Infinity	-0.0005
amountbuy[2,10,58]	0.0000	Infinity	-0.0005
amountbuy[2,10,59]	0.0000	Infinity	-0.0005
amountbuy[2,10,60]	0.0000	Infinity	-0.0005
amountbuy[2,10,61]	0.0000	Infinity	-0.0005
amountbuy[2,10,62]	0.0000	Infinity	-0.0005
amountbuy[2,10,63]	0.0000	Infinity	-0.0005
amountbuy[2,10,64]	0.0000	Infinity	-0.0005
amountsell[1,1,1]	0.0000	Infinity	0.0000
amountsell[1,1,2]	0.0000	Infinity	0.0000
amountsell[1,1,3]	0.0000	Infinity	0.0000
amountsell[1,1,4]	0.0000	Infinity	0.0000
amountsell[1,1,5]	0.0000	Infinity	0.0000
amountsell[1,1,6]	0.0000	Infinity	0.0000

. . . .

amountsell[2,10,59]	0.0000	Infinity	0.0000
amountsell[2,10,60]	0.0000	Infinity	0.0000
amountsell[2,10,61]	0.0000	Infinity	-0.0000
amountsell[2,10,62]	0.0000	Infinity	0.0000
amountsell[2,10,63]	0.0000	Infinity	0.0000
amountsell[2,10,64]	0.0000	Infinity	0.0000
marketvalue[1,1]	0.0000	0.0000	0.0000
marketvalue[1,2]	0.0000	0.0000	0.0000
marketvalue[1,3]	0.0000	0.0000	0.0000
marketvalue[1,4]	0.0000	0.0000	0.0000
marketvalue[1,5]	0.0000	0.0000	0.0000
marketvalue[1,6]	0.0000	0.0000	0.0000
marketvalue[1,7]	0.0000	0.0000	0.0000
marketvalue[1,8]	0.0000	0.0000	0.0000

. . . .

marketvalue[2,52]	119158.7957	Infinity	0.0000
marketvalue[2,53]	119092.8259	Infinity	0.0000
marketvalue[2,54]	119196.8909	Infinity	0.0000
marketvalue[2,55]	119266.5773	Infinity	0.0000
marketvalue[2,56]	119150.4333	Infinity	0.0000

Questions/Notes

Notes

```
marketvalue[2,57]       119193.6389    Infinity       0.0000
marketvalue[2,58]       119192.2452    Infinity       0.0000
marketvalue[2,59]       119259.1441    Infinity       0.0000
marketvalue[2,60]       119182.4891    Infinity       0.0000
marketvalue[2,61]       119277.7272    Infinity       0.0000
marketvalue[2,62]       119183.4182    Infinity       0.0000
marketvalue[2,63]       119270.2939    Infinity       0.0000
marketvalue[2,64]       119188.0640    Infinity       0.0000
 ________________________________________________________

CONSTRAINTS

  Name                  Slack                 body          dual
 ________________________________________________________

assetmarketvalue1[1]    0.0000         0.0000         0.0000
assetmarketvalue1[2]    0.0000         0.0000         0.0000
assetmarketvalue1[3]    0.0000         0.0000         0.0000
assetmarketvalue1[4]    0.0000         0.0000         0.0000

. . . .

assetmarketvalue2[2,60]        0.0000         0.0000         0.0156
assetmarketvalue2[2,61]        -0.0000        -0.0000        0.0156
assetmarketvalue2[2,62]        0.0000         0.0000         0.0156
assetmarketvalue2[2,63]        -0.0000        0.0000         0.0156
assetmarketvalue2[2,64]        -0.0000        -0.0000        0.0156
stockbalance1[1,1]      0.0000         0.0000         0.0000
stockbalance1[1,2]      0.0000         0.0000         0.0000
stockbalance1[1,3]      0.0000         0.0000         0.0000

. . . .

stockbalance1[10,17]    0.0000         0.0000         0.0000
stockbalance1[10,18]    0.0000         0.0000         0.0000
stockbalance1[10,19]    0.0000         0.0000         0.0000
stockbalance1[10,20]    0.0000         0.0000         0.0000
stockbalance1[10,21]    0.0000         0.0000         0.0000

. . . .

stockbalance1[10,60]    0.0000         0.0000         0.0000
stockbalance1[10,61]    0.0000         0.0000         0.0000
stockbalance1[10,62]    0.0000         0.0000         0.0000
stockbalance1[10,63]    0.0000         0.0000         0.0000
stockbalance1[10,64]    0.0000         0.0000         0.8268
stockbalance2[1,2,1]    -0.0000        0.0000         0.0040
stockbalance2[1,2,2]    -0.0000        -0.0000        0.0040
stockbalance2[1,2,3]    -0.0000        0.0000         0.0040
stockbalance2[1,2,4]    -0.0000        -0.0000        0.0040


. . . .

stockbalance2[10,2,6]   0.0000         0.0000         0.0107
stockbalance2[10,2,7]   0.0000         0.0000         0.0107
stockbalance2[10,2,8]   0.0000         0.0000         0.0107
stockbalance2[10,2,9]   0.0000         0.0000         0.0107
stockbalance2[10,2,10]  0.0000         0.0000         0.0107
stockbalance2[10,2,11]  0.0000         0.0000         0.0107
stockbalance2[10,2,12]  0.0000         0.0000         0.0107

. . . .

stockbalance2[10,2,63]  0.0000         0.0000         0.0107
stockbalance2[10,2,64]  0.0000         0.0000         0.0107
fundbalance1[1,1]       0.0000         100000.0000    0.0000
```

Questions/Notes

Notes

```
fundbalance1[1,2]       0.0000         100000.0000    0.0000
fundbalance1[1,3]       0.0000         100000.0000    0.0000
fundbalance1[1,4]       0.0000         100000.0000    0.0000

. . . .

fundbalance1[2,59]     -0.0000        -1000.0000     0.0160
fundbalance1[2,60]     -0.0000        -1000.0000     0.0160
fundbalance1[2,61]     -0.0000        -1000.0000     0.0160
fundbalance1[2,62]     0.0000         -1000.0000     0.0160
fundbalance1[2,63]     0.0000         -1000.0000     0.0160
fundbalance1[2,64]     0.0000         -1000.0000     0.0160
   zeta[2,1]    49366.4612     -119366.4612   0.0000
   zeta[2,2]    49238.2382     -119238.2382   0.0000
   zeta[2,3]    49232.1987     -119232.1987   0.0000
   zeta[2,4]    49248.9234     -119248.9234   0.0000

   . . . .

  zeta[2,61]    49277.7272     -119277.7272   0.0000
  zeta[2,62]    49183.4182     -119183.4182   0.0000
  zeta[2,63]    49270.2939     -119270.2939   0.0000
  zeta[2,64]    49188.0640     -119188.0640   0.0000
  na1[1,1,2]    0.0000         0.0000         -0.0040
  na1[1,1,3]    0.0000         0.0000         0.0000
  na1[1,1,4]    0.0000         0.0000         0.0000
  na1[1,1,5]    0.0000         0.0000         0.0000
  na1[1,1,6]    0.0000         0.0000         0.0000

  . . . .

  na2[1,1,7]    0.0000         0.0000         0.0000
  na2[1,1,8]    0.0000         0.0000         0.0000

  . . . .

  na3[1,1,9]    0.0000         0.0000         0.0000

  na3[2,10,64]  0.0000         0.0000         -0.0403
________________________________________________________

END
```

Questions/Notes

Notes

3. Two stage stochastic program

In an asset and liability management problem the investor has to make a contingent decision facing future uncertainty. The models so far investigated do not consider this aspect in a satisfactory way. Scenarios are dealt with by either grouping them together (Expected Value model) or by consider them completely independently (Wait and See model). The stochastic programming models (also know as Here and Now), take into account the decisional sequence illustrated in figure 3.

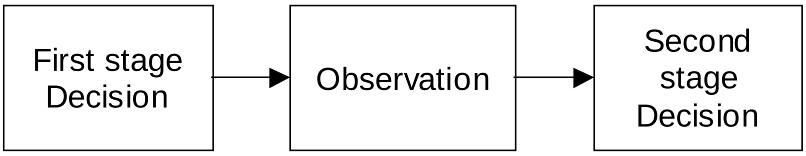


Figure 3. Decisional sequence in SP

In our case, the investor tries to make a "good" asset allocation in the first time period. Subsequently, after having observed the actual price of the assets (i.e. which scenario has manifested), he takes corrective actions (portfolio re-balancing). The concept is illustrated in figure 4, where the scenarios tree for the EV, WS and HN models are shown.

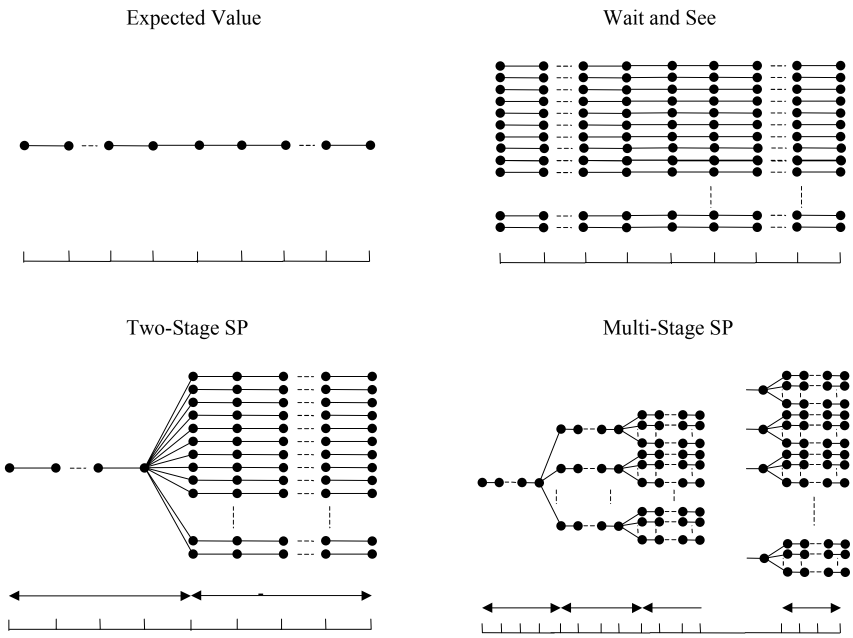


Figure 4. Scenario trees.

Questions/Notes

Notes

3.1. Algebraic formulation

Asset holding constraints

$$H_{its} = H_{i0} + B_{its} - S_{its} \qquad t = 1, i = 1\ldots I, s = 1...Sc \qquad (15)$$

Fund balance constraint

$$(1-g)\sum_{i=1}^{I} price_{its} S_{its} - L_t + F_t = (1+g)\sum_{i=1}^{I} price_{its} B_{its} \quad t = 1...T, \forall s \qquad (16)$$

Downside Risk Constraint

$$\frac{A_t - \sum_{i=1}^{I} price_{its} H_{its}}{A_t} \leq R_t \qquad t > 1, \forall s \qquad (17)$$

where:

- A_t is the predefined target for time period t
- R_t is the maximum deviation from the target accepted by the investor

Objective function

$$\max \sum_{s=1}^{Sc} p_s \sum_{i=1}^{I} price_{iT} H_{iT} \qquad (18)$$

This objective function denotes the expected value of the final portfolio wealth.

3.2. SAMPL formulation

In the AMPL formulation of the model above, we define a new decision variable called *finalwealth*, which will contain the final wealth of the portfolio under the different scenarios.

```
#PARAMETERS: SCALARS
param NT:=2;
param NS:=64;
param NA:=10;
param tbuy := 1.025;
param tsell := 0.975;
param risklevel:=0.3;

#SETS
set assets := 1..NA;
set tp :=1..NT;

#SCENARIO
scenarioset scen:=1..NS;
```

Questions/Notes

```
#TREE
tree theTree:= twostage{2};


#RANDOM PARAMETERS
random param price{assets, tp, scen};

#PROBABILITIES
probability param Prob{scen}:=1/NS;

#PARAMETERS : VECTORS (read from database!)
param liabilities{tp};
param initialholdings{assets} := 0;
param income{tp};
param target{tp};


#STAGES
suffix stage LOCAL;

#VARIABLES
var   amounthold{t in tp,a in assets,s in scen} >=0;
var   amountbuy{t in tp,a in assets,s in scen} >=0;
var   amountsell{t in tp,a in assets,s in scen} >=0;
var   marketvalue{t in tp,s in scen} >=0 ;

#STAGING INFORMATION
let {t in tp,a in assets,s in scen} amounthold[t,a,s].stage := if
(t=1) then 1 else 2;
let {t in tp,a in assets,s in scen} amountbuy[t,a,s].stage := if
(t=1) then 1 else 2;
let {t in tp,a in assets,s in scen} amountsell[t,a,s].stage := if
(t=1) then 1 else 2;
let {t in tp,s in scen} marketvalue[t,s].stage := if (t=1) then 1
else 2;


#OBJECTIVE
maximize wealth : sum{s in scen} Prob[s]*marketvalue[2,s];

#CONSTRAINTS
subject to

assetmarketvalue1{s in scen}:
      marketvalue[1,s]=sum{a in assets}
initialholdings[a]*price[a,1,s];

assetmarketvalue2{t in 2..NT,s in scen}:
      marketvalue[t,s] = sum{a in assets}
amounthold[t,a,s]*price[a,t,s];

stockbalance1{a in assets,s in scen}:
      amounthold[1,a,s]=initialholdings[a]+amountbuy[1,a,s]-
amountsell[1,a,s];
```

Notes

Questions/Notes

Notes

```
stockbalance2{a in assets,t in 2..NT, s in scen}:
      amounthold[t,a,s]=amounthold[t-1,a,s]+amountbuy[t,a,s]-
amountsell[t,a,s];

fundbalance1{t in tp,s in scen}:
      sum{a in assets} amountbuy[t,a,s]*price[a,t,s]*tbuy
       -sum{a in assets} amountsell[t,a,s]*price[a,t,s]*tsell=
        income[t]-liabilities[t];

zeta{ t in 2..NT,s in scen}: target[t]-
marketvalue[t,s]<=risklevel*target[t];
```

3.3. *Solutions*

```
HN = DETEQE
Obj  119360.937105
STATUS = 3: Optimal LP solution
Variables
 Index  Stage   Scen        Value        Rscos          Lob          Upb
     1      1      1 464576.074332            0            0       1e+035
     2      1      1            0 1.19864580982            0       1e+035
     3      1      1            0 0.346474245539            0       1e+035
     4      1      1            0 0.716391943786            0       1e+035
     5      1      1            0            0            0       1e+035
     6      1      1            0 0.871266577724            0       1e+035
     7      1      1            0 2.58569816887            0       1e+035
     8      1      1            0            0            0       1e+035
     9      1      1            0 0.222358743043            0       1e+035
    10      1      1            0 0.144452608902            0       1e+035
    11      1      1 464576.074332            0            0       1e+035
    12      1      1            0            0            0       1e+035
    13      1      1            0            0            0       1e+035
    14      1      1            0            0            0       1e+035
    15      1      1            0 0.577253642022            0       1e+035
    16      1      1            0            0            0       1e+035
    17      1      1            0            0            0       1e+035
    18      1      1            0            0            0       1e+035
    19      1      1            0            0            0       1e+035
    20      1      1            0            0            0       1e+035
    21      1      1            0 0.0126405907012            0       1e+035
    22      1      1            0 0.592301964286            0       1e+035
    23      1      1            0 0.142658095057            0       1e+035
    24      1      1            0 0.235957693089            0       1e+035
    25      1      1            0            0            0       1e+035
    26      1      1            0 0.224520968169            0       1e+035
    27      1      1            0 0.793347549724            0       1e+035
    28      1      1            0 0.320830230655            0       1e+035
    29      1      1            0 0.0818628731127            0       1e+035
    30      1      1            0 0.0403295036658            0       1e+035
    31      1      1            0            0            0       1e+035
Constraints
 Index  Stage   Scen       Rowact       SPrice          Lhs          Rhs
     2      1      1            0            0            0            0
     3      1      1            0 0.259132109375            0            0
     4      1      1            0 12.1421902679            0            0
     5      1      1            0 2.92449094866            0            0
     6      1      1            0 4.83713270833            0            0
     7      1      1            0 11.2564460194            0            0
     8      1      1            0 4.60267984747            0            0
     9      1      1            0 16.2636247693            0            0
```

Questions/Notes

```
    10       1      1             0 6.57701972842            0            0
    11       1      1             0 1.67818889881            0            0
    12       1      1             0 0.826754825149            0            0
    13       1      1        100000 1.20386578107       100000       100000
END
```

Notes

Questions/Notes

Notes

4. SAMPL model for the ALM problem

In this workshop we will implement the Asset/Liability Management using the SAMPL language extension supported by SPInE. The SAMPL/SPInE environment will be used to edit the model and generate the SMPS representation for this model and we will export the results to an Access database, where the solution will be analysed.

4.1. Stochastic information

According to the SAMPL syntax, we can now identify the information about the random nature of our problem:

The scenario index has to be introduced into the model, as the price is a random parameter indexed over this dimension:

```
#SCENARIO
scenarioset scen:=1..NS;
```

Probabilities

The scenarios in our example follow a discrete uniform probability distribution. We can express this with the following:

```
#PROBABILITIES
probability param Prob{scen}:=1/NS;
```

Tree

The scenarios tree considered for this problem is a multistage tree, where the random data paths become independent after the first time period. The *tree* section is therefore:

```
#TREE
tree theTree:=multibranch{8,4,2};
```

Random Data

The only random parameter in our model is the price of the assets over time. The price data parameter is scenario-dependent, and this is reflected in the *random data* section as follows:

Questions/Notes

Notes

```
#PARAMETERS: SCALARS
param NT:=4;
param NS:=64;
param NA:=10;
param tbuy := 1.025;
param tsell := 0.975;
param risklevel:=0.3;

#SETS
set assets := 1..NA;
set tp :=1..NT;

#TREE
tree theTree:=multibranch{8,4,2};

#SCENARIO
scenarioset scen:=1..NS;

#RANDOM PARAMETERS
random param price{assets, tp, scen};

#PROBABILITIES
probability param Prob{scen}:=1/NS;

#PARAMETERS : VECTORS (read from database!)
param liabilities{tp};
param initialholdings{assets} := 0;
param income{tp};
param target{tp};

#STAGES
suffix stage LOCAL;

#VARIABLES
var   amounthold{t in tp,a in assets,s in scen} >=0;
var   amountbuy{t in tp,a in assets,s in scen} >=0;
var   amountsell{t in tp,a in assets,s in scen} >=0;
var   marketvalue{t in tp,s in scen} >=0 ;

#STAGING INFORMATION
let {t in tp,a in assets,s in scen} amounthold[t,a,s].stage :=t;
let {t in tp,a in assets,s in scen} amountbuy[t,a,s].stage :=t;
let {t in tp,a in assets,s in scen} amountsell[t,a,s].stage :=t;
let {t in tp,s in scen} marketvalue[t,s].stage :=t;

#OBJECTIVE
maximize wealth : sum{s in scen} Prob[s]*marketvalue[4,s];

#CONSTRAINTS
subject to
```

Questions/Notes

```
assetmarketvalue1{s in scen}:
      marketvalue[1,s]=sum{a in assets}
initialholdings[a]*price[a,1,s];

assetmarketvalue2{t in 2..NT,s in scen}:
      marketvalue[t,s] = sum{a in assets}
amounthold[t,a,s]*price[a,t,s];

stockbalance1{a in assets,s in scen}:
      amounthold[1,a,s]=initialholdings[a]+amountbuy[1,a,s]-
amountsell[1,a,s];

stockbalance2{a in assets,t in 2..NT, s in scen}:
      amounthold[t,a,s]=amounthold[t-1,a,s]+amountbuy[t,a,s]-
amountsell[t,a,s];

fundbalance1{t in tp,s in scen}:
      sum{a in assets} amountbuy[t,a,s]*price[a,t,s]*tbuy
       -sum{a in assets} amountsell[t,a,s]*price[a,t,s]*tsell=
        income[t]-liabilities[t];

zeta{ t in 2..NT,s in scen}: target[t]-
marketvalue[t,s]<=risklevel*target[t];
```

4.2. Solution

```
HN = DETEQE
Obj  125019.728131
STATUS = 3: Optimal LP solution
Variables
```

Index	Stage	Scen	Value	Rscos	Lob	Upb
1	1	1	464576.074332	0	0	1e+035
2	1	1	0	0	0	1e+035
3	1	1	0	0	0	1e+035
4	1	1	0	1.00089800669	0	1e+035
5	1	1	0	0	0	1e+035
6	1	1	0	0	0	1e+035
7	1	1	0	0	0	1e+035
8	1	1	0	0	0	1e+035
9	1	1	0	0.345660007653	0	1e+035
10	1	1	0	0	0	1e+035
11	1	1	464576.074332	0	0	1e+035
12	1	1	0	0.631694081173	0	1e+035
13	1	1	0	0.152145830526	0	1e+035
14	1	1	0	0	0	1e+035
15	1	1	0	0.615644942932	0	1e+035
16	1	1	0	0.239453142559	0	1e+035
17	1	1	0	0.846110568075	0	1e+035
18	1	1	0	0	0	1e+035
19	1	1	0	0	0	1e+035
20	1	1	0	0	0	1e+035
21	1	1	0	0.0134812761226	0	1e+035
22	1	1	0	0	0	1e+035
23	1	1	0	0	0	1e+035
24	1	1	0	0.251650487622	0	1e+035
25	1	1	0	0	0	1e+035
26	1	1	0	0	0	1e+035

Questions/Notes

Notes

```
    27      1      1            0            0            0       1e+035
    28      1      1            0 0.342167627302            0       1e+035
    29      1      1            0 0.0873073120321            0       1e+035
    30      1      1            0 0.0430116904864            0       1e+035
    31      1      1            0            0            0       1e+035
Constraints
 Index  Stage   Scen       Rowact       SPrice          Lhs          Rhs
     2      1      1            0            0            0            0
     3      1      1            0 0.276366160513            0            0
     4      1      1            0 12.3180345829            0            0
     5      1      1            0 2.96684369527            0            0
     6      1      1            0 5.15883499625            0            0
     7      1      1            0 12.0050763872            0            0
     8      1      1            0 4.66933627989            0            0
     9      1      1            0 16.4991560775            0            0
    10      1      1            0 7.01443635969            0            0
    11      1      1            0 1.78979989666            0            0
    12      1      1            0 0.881739654971            0            0
    13      1      1       100000 1.28393105929       100000       100000
END
```

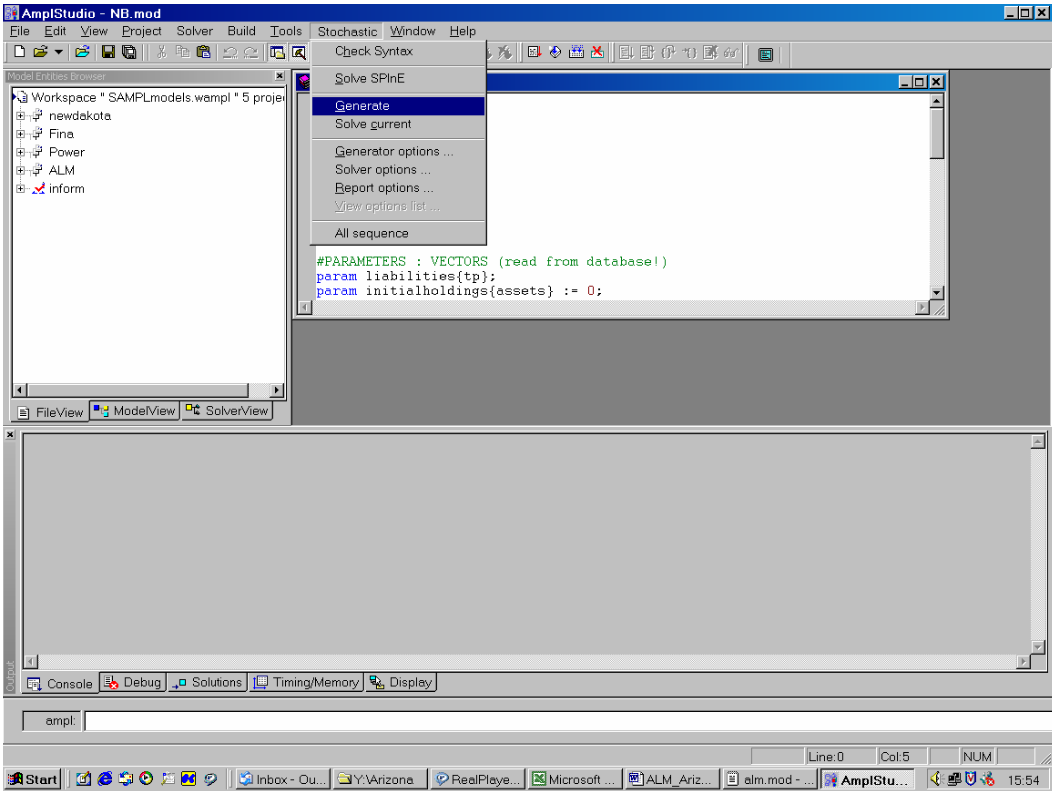
Notes

Questions/Notes

Notes

5. Generate the SMPS instance

The SAMPL/SPInE system allows the user to generate a compact representation of the problem in a standard SMPS format. In order to do so, the user can use the *Generate SMPS* command from the *Stochastic* menu.



Once the generation is over, the following files will be created in the model's directory:

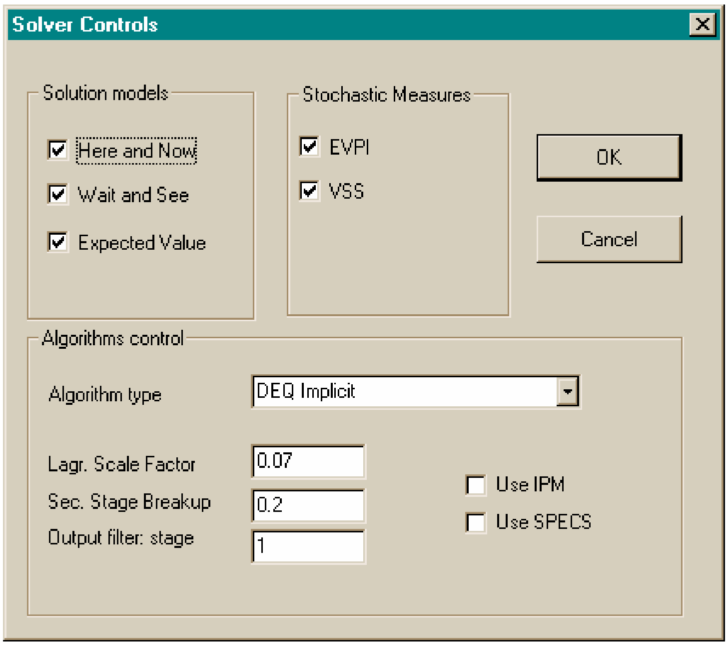
```
spg.log
alm.dic
alm.per
alm.cor
alm.tim
alm.sto
```

6. Solving the model

The solver of SPInE accepts models in standard SMPS format, and is capable of solve the three related models EV, WS and HN associated to the problem. The solver is also able of determining the stochastic indices EVPI and VSS.

Questions/Notes

The solver can be controlled using the *Solver Options…*dialog box in the *Stochastic* menu of SAMPL/SPInE.



The solutions can be obtained by running the *Solve SPInE* command.

This will create, among others, a file called SpineSol.sol, which will contain the optimal decisions for the three EV, HN and WS. A user friendly report of the solutions for the selected models can be viewed using the *View* menu.

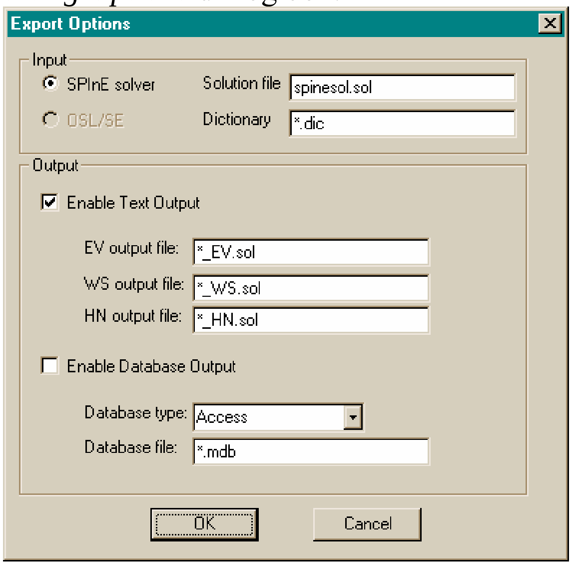
Notes

Questions/Notes

Notes

7. Report the results

The solution vectors can be exported to databases, using the *Reporting* capabilities of SAMPL/SPInE. In the current version, the user can use MPL' s standard *EXPORT TO* statements to export the Expected Value solutions. If the user solves the Here and Now model, then it is also possible to export in the same way the optimal Here and Now decisions. This is useful if the user is interested in analysing the first stage optimal decisions.
If the user decides to export in a single run EV, WS and HN decisions for all stages, SPInE provides direct database connections to Access and Excel. In this case, a completely new database will be created, accordingly to the options selected in the *Reporting Options* dialog box.



Questions/Notes

Notes

8. Analyse the results

Once the results have been exported to a database, the user can take advantage of the tools provided by the DBMS and perform advance analysis. As an example, we can create a graph of the final wealth values for the two-stage model ALM by opening the table *HN_wealth* in the database *alm.mdb*. Using the MS Access command *Analyze it with Excel* under *Tools/Office Links*, we can open the table in a new Excel workbook, select the data values related to the 4th time period and create the graph as shown in figure 2:

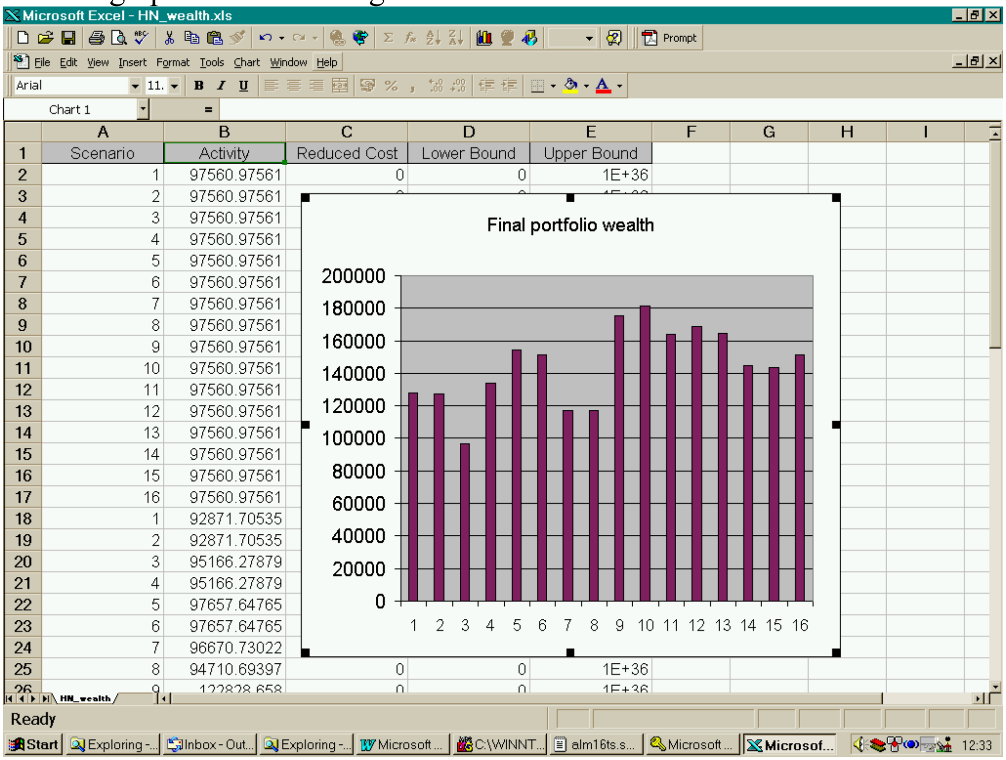


Figure 2. Analysis in Excel.

This is only one of the many tools which can be exploited by SPInE for the solution analysis: in fact the user can link his/her own analyser to the environment, making SPInE a flexible and versatile application.

Questions/Notes

Notes

9. Analysis and understanding of the solutions

The table below summarises the results given by the three models considered in this module:

Table 2. Results

Approach	Objective value
Expected Value	Z_{EV} =128051.943202
Two-stage SP	Z_{HN} =119360.937105
Multi-stage SP	Z_{HN} =125019.728131

EVPI = Z_{ws}- Z_{hn} = 6951,96874563 ,
VSS = Z_{hn}- Z_{eev} = INF

Questions/Notes