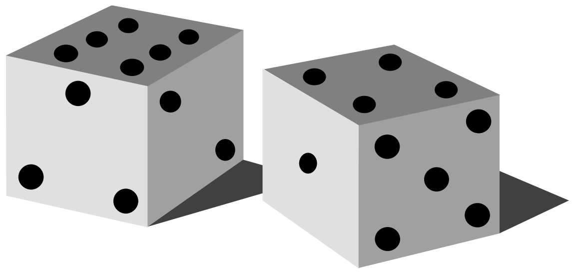


Retirement Planning using a Stochastic Dynamic Programming Approach

MS & E 358

Wednesday March 15th, 2000

SERGIO GARZA
RODRIGO BARRERA
FRANK SIEREVELD

Contents

SERGIO GARZA
RODRIGO BARRERA
FRANK SIEREVELD

Section 1

Summary

SERGIO GARZA
RODRIGO BARRERA
FRANK SIEREVELD

Dynamic programming approach delivered intuitive portfolio recommendations

During the course of this project we developed and tested various algorithms to determine an optimal allocation among four U.S. asset classes. We developed a dynamic program that computed this allocation given a goal several years ahead, and various descriptions of risk preference.

Our Goals

- Develop algorithms for investment decisions related to retirement planning
- Compare different utility functions, and ways to inherently model goals in these functions
- Verify assumptions such as lack of serial correlation between time periods

Our Approach

We considered four asset classes of various risk levels, roughly comprising the universe of investment opportunities in U.S. exchanges

- Equity
- Corporate Bonds
- Government Bonds
- Cash

We determined expected returns, and the variance-covariance matrix among these asset classes, and assumed that these parameters would hold over the entire period considered.

We subsequently assessed various types of utility functions, inherently capturing the investor's goal in the last period (in terms of net worth).

The dynamic program we implemented used backward recursion to step from the last period to the initial period over predefined wealth levels. At each point on this grid, corresponding to 10 wealth levels in each time period, the program incorporated a stochastic optimization over asset allocation. For each point we maximized expected utility given 1000 expected return scenarios, and added a fixed contribution corresponding to net annual income of the investor.

Recommendations

The approach delivers good results. Relative to a static portfolio optimization solution, we show that users can invest in higher yielding but more risky assets early on, and rebalance to a more diversified portfolio when approaching the goal. This dynamic approach is more realistic compared to a one-time static portfolio optimization – since it incorporates the flexibility that investors have to rebalance their portfolio over time. It furthermore also incorporates the opportunity that investors have for volatility pumping – the principle of buying low and selling high (works better for more volatile securities).

By using a piecewise linear utility function in the last period, discrete jumps in risk preference, relating to certain future (retirement) goals, can be modeled. This approach worked well – the recommended allocation varied according to wealth level relative to a retirement goal. Below the goal, the program recommended more risky securities, around the goal the recommended portfolio was more diversified, yet significantly above the goal the mix again rotated into a more equity oriented portfolio.

SERGIO GARZA
RODRIGO BARRERA
FRANK SIEREVELD

Section 2

Our Approach

SERGIO GARZA
RODRIGO BARRERA
FRANK SIEREVELD

We determined the optimal asset allocation for a person 15 years from retirement

The retirement planning problems tries to solve the question how to maximize your personal utility function given a monetary goal in a finite time horizon. Usually we are given a certain basket of assets that have different returns and risk. The returns and risk of each asset is uncertain, however we could calculate these parameters by using the historical information and assuming a probability distribution of outcomes.

Our main task is to estimate an optimal asset allocation through time to achieve our goal. In order to solved this problem we could take two possible positions. First, try to determine the optimal asset allocation at the beginning of the first period, which stay the same until we reach our time horizon. The second approach, which is more interesting, is to try to estimate the optimal asset allocation through time, having different policies depending on the state of wealth and a point in time.

In this report we are going to present an stochastic dynamic programming approach to calculate the optimal asset allocation by discounting the utility function through time assuming i.i.d normal distributions of the returns.

Problem Definition

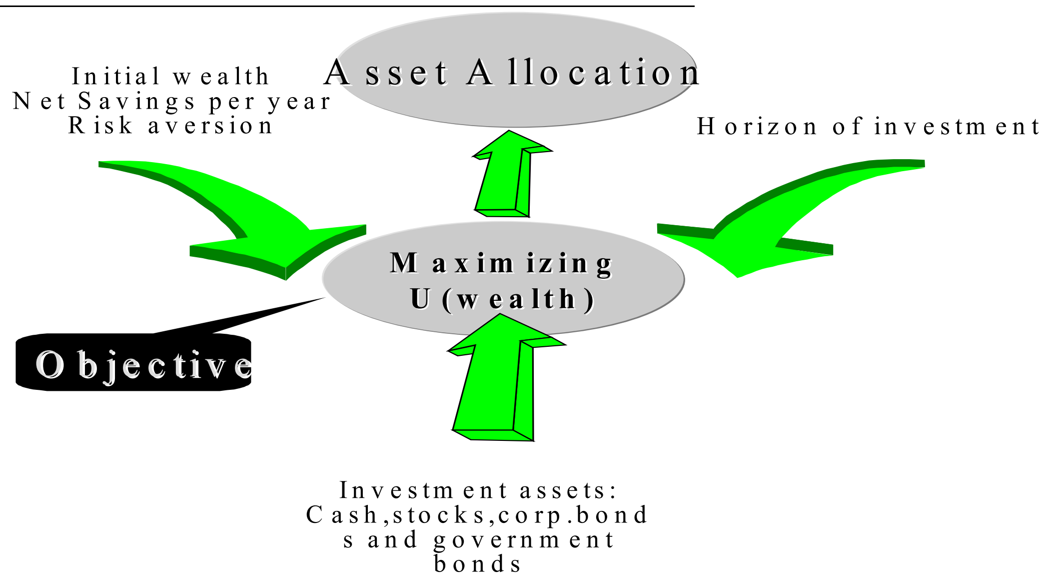


Figure 2-1 ***Problem overview***

In order to solve the retirement problem, we need to ask to the individual their attitude towards risk (risk aversion coefficient), their monetary goal, and their initial wealth and net savings per year. In order to answer each of this questions there are different techniques to help. For instance, in order to estimate the goal we need to consider the expected remaining life of the individual to calculate an annuity, which will be good enough for supporting his lifestyle for rest of his retirement years.

Sergio Garza
Rodrigo Barrera
Frank Siereveld

After obtaining these parameters, we use as a given a collection of US asset classes, which are uncertain through time and finally, we develop an algorithm that optimizes the maximum utility function of an individual by obtaining an optimal asset allocation among the four U.S. asset classes.

The problem can be captured in these bullet points:

- Initial Wealth & Net Annual Savings
- Utility Function (exponential, quadratic, etc.)
- Horizon of investment 15 years
- Goal after 15 years
- Possible Investment Assets:
  - Stocks, Corporate Bonds, Government Bonds and Cash

Assumptions

In order to make this problem tractable, even for the contemporary powerful optimization algorithms and processors, several simplifying assumptions had to be made.

- Distributions of returns do not exhibit serial correlation (otherwise the program should be path dependent)
- Utility of a user follows a predefined analytical shape (exponential, quadratic, piecewise linear)
- Number of wealth levels and time periods can be discretized on a 10 by 15.

Theoretical foundations based on multi period stochastic utility optimization

Overview

We used a backward recursive approach to determine the optimal allocation at the start of year 1

- Elicit a utility function for the last period just before retirement
- Determine the domain over which to run the dynamic program
- Generate a sample of returns for each period, identical for each wealth level
- For each time period
  - Maximize next period's Expected Utility given initial z_i wealth at time t, by varying the asset allocation
  - Fit a curve to assess the optimality (discounted utility) function to be used in next step, by minimizing least square errors between calculated and modeled utility
  - Use modeled utility curve in the previous period
- Assess optimal allocation at initial wealth level, in initial period

Domain Determination

- Determine the domain over which to run the dynamic program by assuming a best and worst case scenario
  - best case corresponds to a 100% investment in stocks yielding one standard deviation above expected return over 15 years
  - worst case corresponds to a 100% investment in stocks yielding one standard deviation below expected return over 15 years
  - the discrete wealth levels are equally spaced between these two scenarios

Simulation

In order to calculate normal distribution random variables for the possible returns and since there is some correlation between the asset classes we need to calculate the Cholesky factors. We used the following factors for the simulation:

	Stock	*CorpBonds*	*GvmtBonds*	*Cash*
Stocks	.154740	.046752	.026692	0.0
CorpBonds		.108694	.074614	0.0
GvmtBonds			.020603	0.0
Cash				0.0

We simulated 1000 expected returns for each period in time, independent on wealth level. We determined that the assumption of no serial correlation holds, and that hence the simulations can be carried out independent of each other.

Risk Aversion

We use different utility functions because of their different risk aversion characteristics.

Absolute risk aversion

A(w)=-u''(w)/u'(w)

If an individual chooses to invest x(w) dollars in a particular risky asset when has wealth w, then we have:

If dA(w)/dw<0 then dx(w)/dw>0

If dA(w)/dw=0 then dx(w)/dw=0

If dA(w)/dw>0 then dx(w)/dw<0

Absolute risk aversion measures how the dollar amount that an individual will invest in given risky assets changes as her/his wealth changes. If one exhibits increasing risk aversion it means that his/her demand for risky assets will decrease as wealth increases (opposite for decreasing).

Relative risk aversion

R(w)=w*A(w)=-w*u''(w)/u'(w) (1)

If dR(w)/dw<0 then d/dw (x(w)/w) >0

If dR(w)/dw=0 then d/dw (x(w)/w) =0

If dR(w)/dw>0 then d/dw (x(w)/w) <0

Relative risk aversion measures the percentage of wealth which a person will invest in a risky asset as his/hers wealth changes. Increasing relative risk aversion (with wealth level) implies that the percentage of wealth invested in risky assets will decrease as wealth increases. The opposite holds for decreasing relative risk aversion with wealth level.

Empirical research shows that most people exhibit constant or decreasing relative risk aversion, both imply decreasing absolute risk aversion.

Utility functions

Exponential

The key characteristics of the function are

- absolute risk aversion is constant for all wealth levels
- increasing relative risk aversion (in percentage terms the individual risks less as he is wealthier)

$$Max..u = (1/\omega) * \sum_{\omega} a - b[e^{-c*v^{\omega}}]$$

Equation 2-1

SERGIO GARZA
RODRIGO BARRERA
FRANK SIEREVELD

where:
a,b arbitrary constants
c risk tolerance
ν wealth level
ω scenario

Quadratic

Quadratic functions exhibit increasing absolute and relative risk aversion (we need to use w<1/a in order to keep utility a monotonic function of wealth. For a utility function of the shape u(w)=w-(aw^2)/2. The function we used

$$Max..u = (1/\omega) * \sum_{\omega} \left(-av^{2\omega} + bv^{\omega} + c \right)$$

Equation 2-2

where:
a,b,c arbitrary constants
ν wealth level
ω scenario

Two approaches were used with respect to the quadratic utility function. The first fitted a quadratic non decreasing curve to reach its maximum at the highest level of wealth at each period in time, though in theory wealth could reach infinitum, but for practical purposes a certain level of confidence could be assumed. The second used what we called a quad-*flat* function that reaches its maximum at the goal or its discounted value at the risk-free rate (in this case cash) and stays constant beyond this maximum. The intuition is that once you reach a certain level of wealth that lets you attain your goal, you would put everything into your risk free asset to avoid any risk of not accomplishing it. This function makes the problem a dual of the variance minimization problem.

Piecewise linear

A linear utility function exhibits constant absolute and relative risk aversion.

The advantages of using this type of piecewise linear function are

- Discrete goals can be modeled using a 'kink' in the curve
- No need for a potentially slower nonlinear optimizer
- No need for the least squares utility function modeling step each period – and a potentially more accurate characterization of the utility curve

$$Max..u = (1/\omega) * \sum_{k} \sum_{\omega} g_k * v_k^{\omega}$$

$$g_k = \frac{y_k - y_{k-1}}{v_k - v_{k-1}}$$

Equa

where:
k wealth segment
ν wealth level
ω scenario

SERGIO GARZA
RODRIGO BARRERA
FRANK SIEREVELD

In all cases the utility function were subject to the next constraints:

$$\sum_i x_i = 1$$

$$v^{\omega} = \sum_{\omega} (wealth_{t-1} + contribution) * R_i^{\omega} * X_i$$

$$X_i \geq 0$$

Where:
Xi asset allocation in the asset i
Ri return on asset i scenario ω

To simplify things we assessed a bilinear utility function at time 15 (retirement point) that mainly required 3 intuitive parameters: Goal amount, Pre-Goal Risk Aversion Slope, Post-Goal Risk Aversion Slope. Basically the questions that are to be asked to the individual are: *What is your goal?, How happy would you be to surpass it? And how dramatical would be not to achieve it?*. This bilinear function is plotted in the following chart with the alternative utility functions that could be used to fit the optimality function when doing the dynamic programming backward recursion.

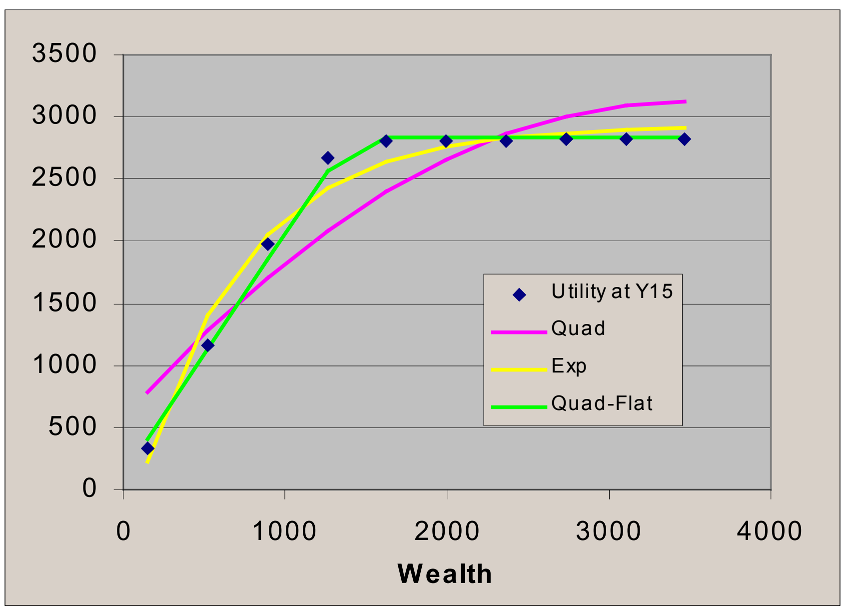


Figure 2-2 Utility functions used compared in our algorithm runs

An interesting aspect of the quadratic utility function is that it is non-monotonic. This is a-typical for utility functions because it essentially means that utility decreases above a certain level of wealth. On the other hand, some investors might become more risk averse after reaching their goal, and essentially want to invest in the asset class with the lowest expected return. If this behavior is not representative of a certain investor, the 'Quad-flat' utility function can be used instead.

Section 3

Results

Quadratic utility function

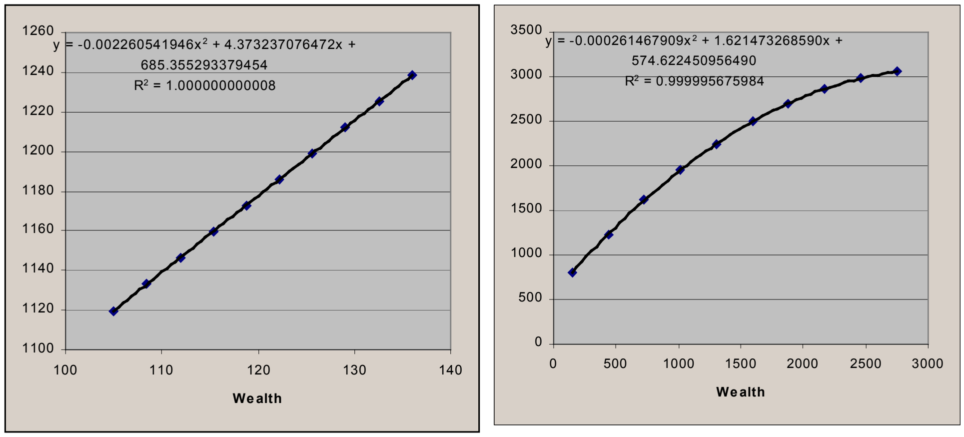


Figure 3-1 Change in shape of quadratic utility function

The quadratic function displays certain convexity towards the goal, indicating a reduced propensity to take risks when getting closer to the goal in terms of wealth level and timing. Though this degree of diversification is rather dramatic and occurs very closely to the final year and to the set goal. Another important issue is that after reaching the goal and accumulating a certain surplus, the optimal allocation yields again towards investing solely in stocks until reaching the maximum wealth point. At initial stages the utility resembles a linear function. In the way this problem was set, the goal was initially far away from the starting amount. As a result the linear nature of the utility function at early stages forced the asset allocation into only stocks. In this case stocks resulted dramatically the best asset to invest when certain risk could be afforded. Below graph at period 13 indicates that indeed the dynamic program assumes that the user moves into less risky assets at higher wealth levels

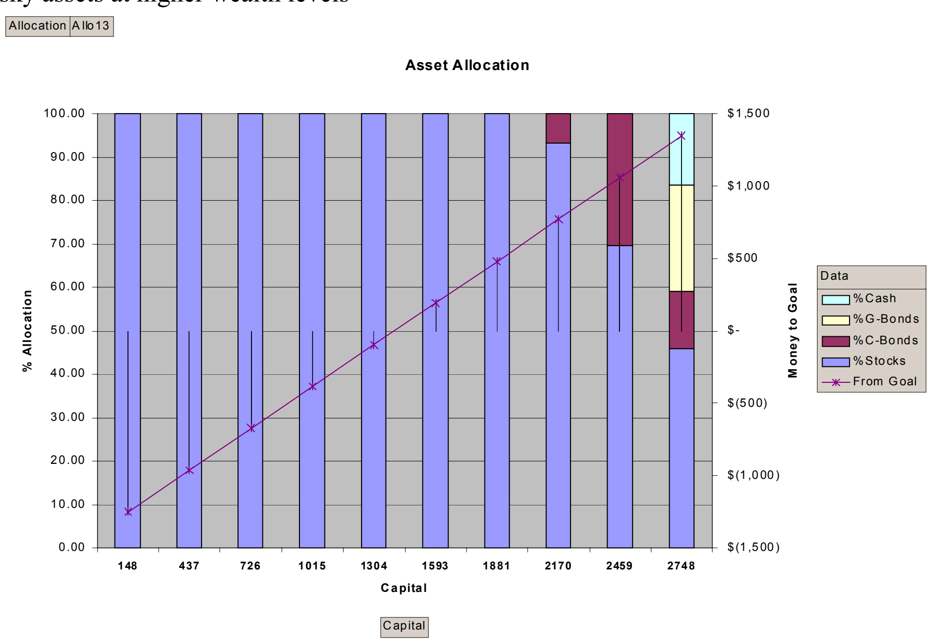


Figure 3-2 Optimal allocation in period 13

Quadratic '*flat*' utility function

This hybrid utility function has the differentiating factor of pushing the investor into the most safe asset class after meeting the goal or its discounted representation. Witness below graph which depicts the recommended allocation 5 periods before the goal: the investor mainly holds cash if above the goal.

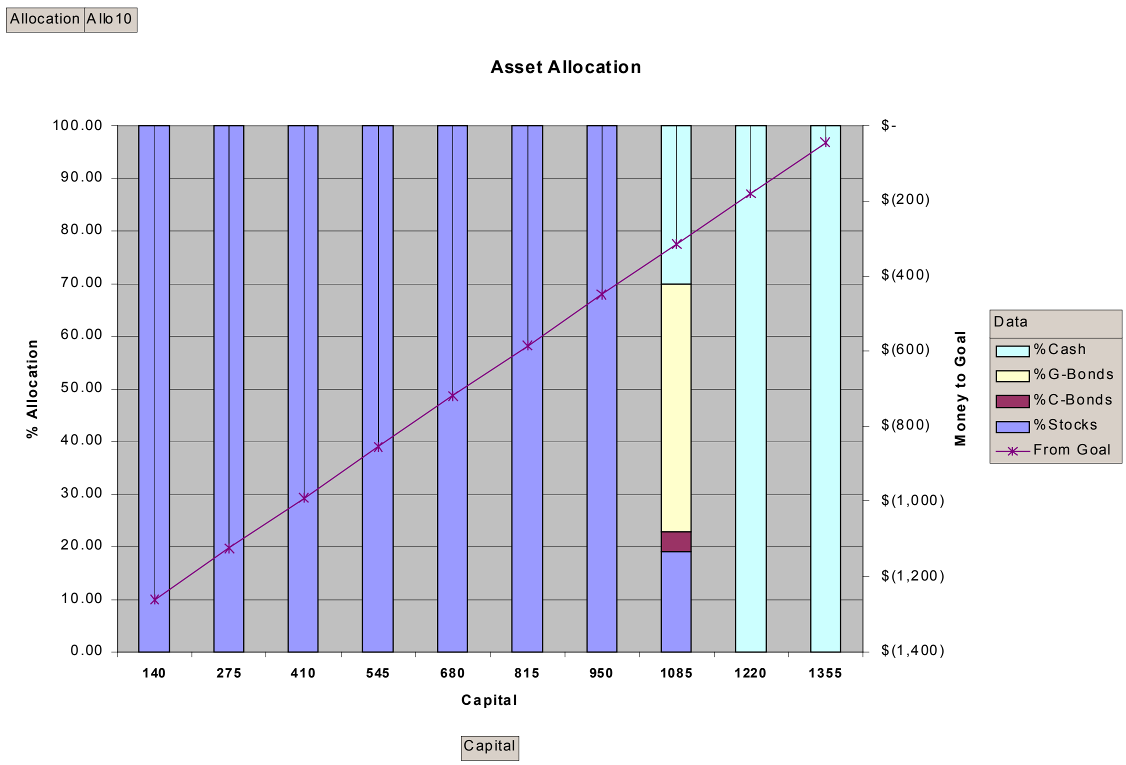


Figure 3-3 Optimal allocation in period 10, two years before reaching retirement

In this utility function the policy for allocating assets is extreme, the only point the portfolio is diversified is right at the money, but when out of the money the policy demands a pure stock portfolio, and when in the money it demands a pure cash one.

SERGIO GARZA
RODRIGO BARRERA
FRANK SIEREVELD

Exponential utility function

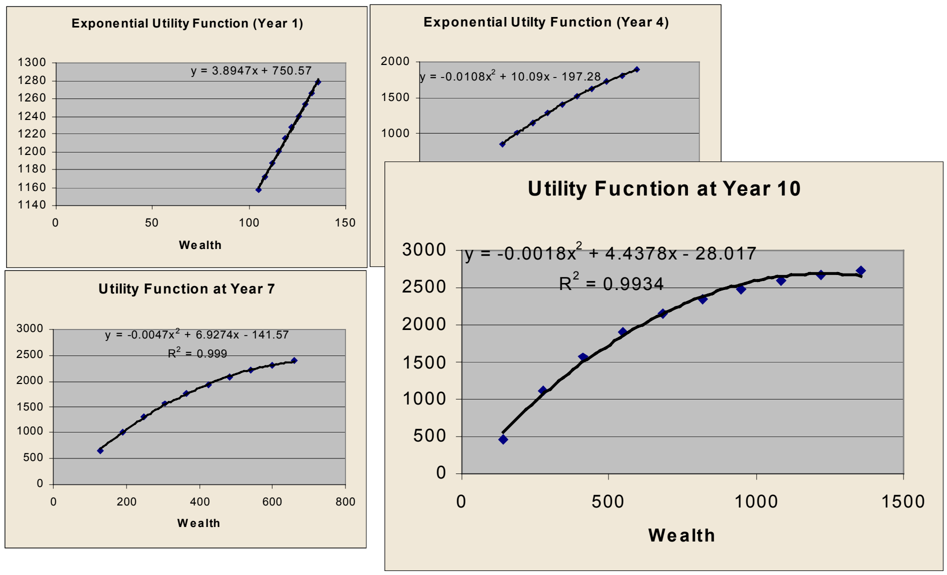


Figure 3-4 Change in shape of the exponential utility function

It can be ascertained from the change in shape of the exponential utility function over time that indeed the investor becomes more risk seeking while moving away from the goal. This is indicated by the change in slope, which is steeper in earlier periods. In comparison to the quadratic utility, this function results in a smoother diversification when approaching the goal. It also has the same effect as in the quadratic case where, after reaching a certain surplus above the goal, it starts risking again. As mentioned in the theoretic introduction, the main reason of this different behavior in allocation comes from absolute and relative risk aversion changes with respect to wealth. Even in the case of the exponential utility, the fact that there is an increasing relative risk aversion could not quite represent the typical investors behavior.

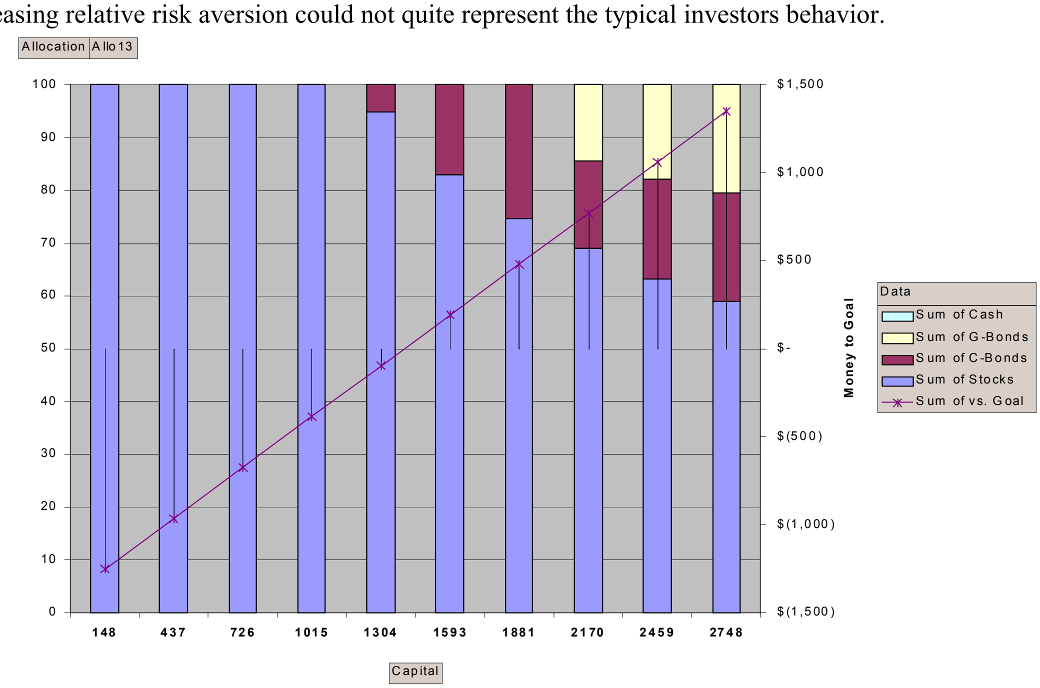


Figure 3-5 Optimal allocation in period 13, two years before reaching retirement

SERGIO GARZA
RODRIGO BARRERA
FRANK SIEREVELD

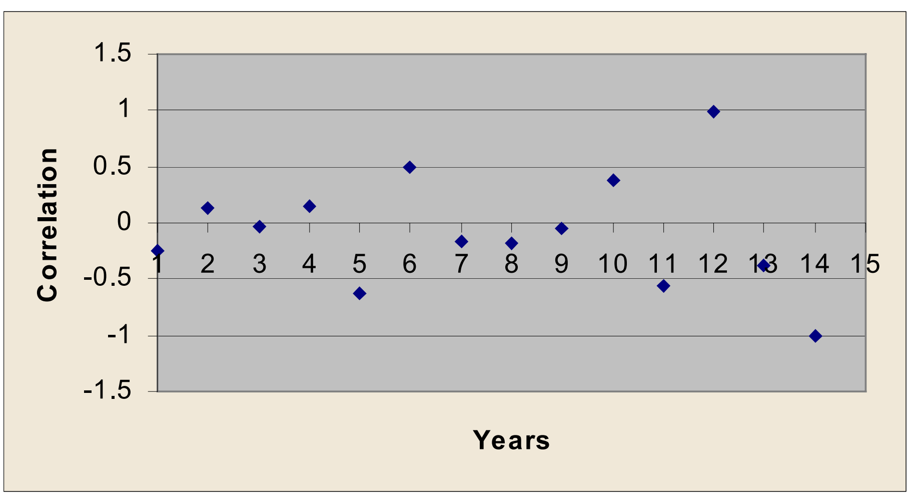
Serial Correlation Analysis

To prove that the Dynamic Programming approach is valid, a serial correlation test was conducted. The results express the impact of assuming that **yearly** returns are independent through time. The following table depicts the methodology used, in which a correlation analysis was made of a series of yearly returns lagged *n* periods.

Year	Return S&P	lag-1	lag-2	lag-3	lag-4	lag-5	lag-6	lag-7	lag-8	lag-9	lag-10	lag-11	lag-12	lag-13	lag-14
1999	21.04	28.58	33.36	22.96	37.58	1.32	10.08	7.62	30.47	-3.1	31.69	16.61	5.26	18.68	31.75
1998	28.58	33.36	22.96	37.58	1.32	10.08	7.62	30.47	-3.1	31.69	16.61	5.26	18.68	31.75	6.27
1997	33.36	22.96	37.58	1.32	10.08	7.62	30.47	-3.1	31.69	16.61	5.26	18.68	31.75	6.27	
1996	22.96	37.58	1.32	10.08	7.62	30.47	-3.1	31.69	16.61	5.26	18.68	31.75	6.27		
1995	37.58	1.32	10.08	7.62	30.47	-3.1	31.69	16.61	5.26	18.68	31.75	6.27			
1994	1.32	10.08	7.62	30.47	-3.1	31.69	16.61	5.26	18.68	31.75	6.27				
1993	10.08	7.62	30.47	-3.1	31.69	16.61	5.26	18.68	31.75	6.27					
1992	7.62	30.47	-3.1	31.69	16.61	5.26	18.68	31.75	6.27						
1991	30.47	-3.1	31.69	16.61	5.26	18.68	31.75	6.27							
1990	-3.1	31.69	16.61	5.26	18.68	31.75	6.27								
1989	31.69	16.61	5.26	18.68	31.75	6.27									
1988	16.61	5.26	18.68	31.75	6.27										
1987	5.26	18.68	31.75	6.27											
1986	18.68	31.75	6.27												
1985	31.75	6.27													
1984	6.27														
		-0.25	0.132	-0.03	0.141	-0.62	0.494	-0.16	-0.18	-0.05	0.384	-0.56	0.991	-0.37	-1

Figure 3-6 Correlation of returns

The graph below confirms that there is limited serial correlation in returns of the S&P 500 between 1984 and 1999. This result is to be expected if one relies on market efficiency: any correlation would quickly disappear through arbitrage (which impacts the price of a security, such that there is no information in historical returns). In general, in this kind of analysis the points that are closer in time are the most representative. You may see that in this case these points are very close to the *zero correlation line*. In consequence for this case, Dynamic Programming is a valid approach.



Section 4

Recommendations

SERGIO GARZA
RODRIGO BARRERA
FRANK SIEREVELD

Conclusions

The results of our dynamic program were surprisingly relevant portfolio decision making by the typical individual investor. Current methodologies do not incorporate the flexibility that investors have to rebalance their portfolio over time. Most financial advisors either use portfolio theory to recommended a diversification strategy, or just use a cookie cutter, back of the envelope approach. Our methodology, if molded into an easy to use framework, has the potential to revolutionize retirement planning. Several refinements of the algorithms will be necessary to accomplish this goal (see next page).

To summarize our conclusions in three succinct bullet points:

- Asset Allocation Behavior throughout time will be changing as we approach our Goal:
  - *Out of the money* mostly we invest in stocks
  - *At the money* we diversify
  - *In the money* in most cases we Risk Again
- Allocation differences between the various utility curves are explained by:
  - Absolute and relative risk aversion characteristics
  - Embedded king in the bilinear utility curve
- In long term horizons serial correlation is not present. We can therefore conclude that the time-independence assumption in our dynamic programming approach is valid.

SERGIO GARZA
RODRIGO BARRERA
FRANK SIEREVELD

Various avenues for further research

Although the results of this project were promising, there are several issues that need to be ironed out before this type of stochastic dynamic program becomes practical. The areas where further research is necessary are captured in below 'wish list' for version II of our dynamic programming algorithms:

- Customize recommendations – the average investor usually has a current portfolio of stocks which h/she does not want (or should) sell on the spot, various asset class preferences, and also specific tax issues, which all put complex constraints on the optimization algorithms. Investors further will have more complex goals and risk aversion characteristics, than assumed in this project.
- Optimize processing speed – above mentioned customized results need to be obtained on a standard PC (or web-based server) in a short enough time to keep the user interested in the recommendations
- Include liabilities and other asset classes– besides looking at price fluctuations in the asset classes we considered, one should also include the key liabilities (mortgage, loans, etc), and other key asset classes (real estate, precious metals), to get a complete picture of net worth.
- Include other risk factors – security returns are just one of many risk factors in a persons life. Even from a financial perspective, one should incorporate real estate market risk, interest and inflation risk, and income risk (what if you get fired). This in order to properly map the distribution of net worth over time, and make appropriate asset allocation decisions based on these distributions.

We have outlined several avenues of further research, with the goal to accomplish this wish list, in below paragraphs:

Sensitivity Analysis

Perform an analysis decreasing the resolution of the algorithm, until the results become unrealistic. The objective of this exercise is to

- Minimize the number of time periods,
- Minimize the number of wealth levels, and
- Minimize the number of return scenarios

such that processing speed is optimized.

Mapping goals using utility functions

A key issue is solving the utility curve assessment problem. What is the true shape for a specific investor, in line with his propensity for taking risk.

- How to map multiple goals in the utility function (house, college, retirement)
- How to map 'soft' and 'hard' goals (nice to have versus must have – corresponding to the investor specific subsistence level).

Some users want to maximize returns, but at least meet their goals, while other users want to minimize the probability of not meeting their goals. Additional research is required to incorporate these preferences into the algorithm

SERGIO GARZA
RODRIGO BARRERA
FRANK SIEREVELD

Risk in level of income and liabilities

How to incorporate other important risks into the simulation:

- Interest rates and inflation (refinance mortgage?)
- Risk of getting fired (versus chance of becoming financially independent)
- Risk of major devaluation in mortgage prices

Other user preferences

- Certain views/preferences on asset classes
- Certain buys/keeps for securities

A key issue here is to incorporate rebalancing costs.

Taxed versus tax-deferred accounts

Retirement planning cannot be considered serious without treating the allocation problem between taxed and tax-deferred accounts.

Appendix I

GAMS

Source Code

```
* Dynamic Retirement Planning

***********************************************************************
* Define and declare all variables

set j assets
/
Stocks
CorpBonds
GvmtBonds
Cash
/;

alias (j, jj);

set t time
/1*15/;

set i samples
/1*1000/;

set k wealth levels
/1*10/;

alias (k,wlvl);

parameter iter(wlvl)
/
1       1.0
2       2.0
3       3.0
4       4.0
5       5.0
6       6.0
7       7.0
8       8.0
9       9.0
10      10.0/;

parameter rbar(j) mean returns
/
Stocks      1.105
CorpBonds   1.085
GvmtBonds   1.07
Cash        1.04
/;

parameter rstd(j) standard deviation of returns
/
Stocks       15.474
CorpBonds    11.866
GvmtBonds    8.188
Cash         0
/;

parameter jettl(wlvl) exp utility stored;

Table rcorr(j, jj) Correlation matrix of returns
                      Stocks      CorpBonds       GvmtBonds       Cash
Stocks                 1.0         0.394            0.326         0.0
CorpBonds                          1.0              0.966         0.0
GvmtBonds                                           1.0           0.0
Cash                                                              1.0;

rcorr(j, jj)$(ord(jj) lt ord(j)) = rcorr(jj, j);

Table rcov(j,jj) variance-covariance matrix of returns
```

```
                      Stocks      CorpBonds        GvmtBonds      Cash
Stocks                15.474      4.675             2.669        0.0
CorpBonds                         11.866            7.909        0.0
GvmtBonds                                           8.188        0.0
Cash                                                             0.0;

rcov(j, jj)$(ord(jj) lt ord(j)) = rcov(jj, j);

table chol(j,jj) cholesky factors matrix
                 Stocks   CorpBonds     GvmtBonds       Cash
Stocks           .154740
CorpBonds        .046752  .108694
GvmtBonds        .026692  .074614       .020603
Cash                                                         ;

Parameter vector(j);
Parameter r(i,j);

Parameter at(t);
Parameter bt(t);
Parameter ct(t);

Parameter anext;
Parameter bnext;
Parameter cnext;

SCALAR cont      contribution of income pa   /10/;
SCALAR KC        capital at time 0           /100/;
SCALAR a         parameter in utility        /0/;
SCALAR b         parameter in utility        /5000/;
SCALAR c         parameter in utility        /-.005/;

SCALAR period ;
SCALAR index ;
SCALAR wealth    used in the loop;
SCALAR umdl;
SCALAR maxwealth;
SCALAR maxret                                /1.26/;
SCALAR minwealth;
SCALAR minret                                /-1.05/;
SCALAR topcone;
SCALAR bottomcone;

************************************************************************
* initialize utility function

anext=a;
bnext=b;
cnext=c;

************************************************************************
* Carry out simulation and read results into matrix r(i,j) with i samples and j asset
classes
* assume that exp returns, stdevs, and covars remain same over time
* later repeat simulation by period (inside GAMS using predetermined cholesky factors)

LOOP(i,
       vector(j)=normal(0,1);
       r(i,j) = rbar(j)+ sum(jj, chol(j,jj)*vector(jj));
  );

************************************************************************
* Determine the 'cone'
* Go up/down one stdev each period - and divide up in 10 wealth points
* read in to cone(k,t) with t (15) periods, and k (10) wealthpoints

* below code is not used anymore, cone is calculated on the fly
Table cone(k,t) wealthcone (wealth at end of period)
  1   2   3   4   5   6   7   8   9   10  11  12  13  14  15
1 105 110 114 119 123 127 130 134 137 140 143 146 149 152 154
2 108 118 128 140 154 170 189 212 241 275 318 372 438 521 624
3 112 126 142 161 185 213 248 291 344 410 493 597 727 889 1094
4 115 134 156 183 215 256 306 369 447 545 668 822 1015 1258 1564
```

```
5 119 142 169 204 246 299 365 448 551 680 843 1047 1304 1627 2034
6 122 150 183 225 277 342 424 526 654 815 1018 1273 1593 1996 2504
7 126 157 197 246 308 386 483 605 758 950 1193 1498 1882 2365 2974
8 129 165 211 268 339 429 541 683 861 1085 1368 1723 2171 2734 3444
9 133 173 225 289 370 472 600 761 965 1220 1543 1948 2459 3103 3914
10136 181 238 310 401 515 659 840 1068 1355 1718 2174 2748 3472 4384;

Parameter localcone(k);

************************************************************************
* Asset allocation optimizer

VARIABLES
x(j)     fraction of portfolio invested in asset j in wealthpoint k
je       maximum expected utility function at wealthpoint k and time t;

POSITIVE VARIABLE x(j);
FREE VARIABLE je;

EQUATIONS
  exputil         Objective function
  budget          Portfolio adds up to one;

  exputil..   je =E= sum(i, anext-
bnext*exp(sum(j,(wealth+cont)*x(j)*r(i,j))*cnext))/card(i);
  budget..    sum(j, x(j)) =E= 1;

Model retirement /ALL/;

file output /dp5output.lst/;
put output;

************************************************************************
* Utility curve from least squares on expected utility at 10 wealthpoints

Variables
        au          constant
        bu          factor
        cu          risk tol
        error       least squares error              ;

Negative variable cu;

Equations
         erroreq           objective function          ;

* changed '15' below, check if period is correct (or period +1)
erroreq..   error =E= sum(k, sqr((au-bu*exp(localcone(k)*cu)-jettl(k))))  ;

Model leastsquares /all/;


************************************************************************
* Start loop over time periods
************************************************************************
* Setup and optimize the expected bilinear/exponential utility function valid at the
end of year 15
* Loop over t (15) periods, and k (10) wealthpoints in each period
* Sum over 1000 scenarios i using asset class returns r(i,j)

period = 15;

WHILE((period gt 0),

* determine cone
maxwealth = KC;
topcone = 0.0;
minwealth = KC;
bottomcone = 0.0;
index = 0;
WHILE (index le period,
     topcone = maxwealth*maxret + cont;
     maxwealth = topcone;
```

```
     bottomcone = minwealth*minret + cont;
     minwealth = bottomcone;
     index = index + 1;
     );
LOOP (wlvl, localcone(wlvl) = bottomcone + (topcone-bottomcone)/9*(iter(wlvl)-1););

* old stuff:
* maxwealth = KC*maxret**period + cont*period;
* minwealth = KC*minret**period + cont*period;
* For (index= 1 to 10,
*     localcone(iter) = minwealth + (maxwealth-minwealth)/9*(iter-1););
* localcone(k) = cone(k, '2')

put //'period',period/;
put 'wealth',@20,'expected utility',@40,'asset allocation'/;

* determine optimal asset alloc for each wealth level this period
* check if period is correct (or period +1)
LOOP (wlvl, wealth=localcone(wlvl);
Solve retirement using NLP maximizing je;
jettl(wlvl)=je.l; put wealth,@20,je.l,@40;
loop(j, put x.l(j)); put /);

* fit exponential utility function
solve leastsquares using nlp minimizing error       ;

* carry utility curve parameters to next iteration
anext=au.l;
bnext=bu.l;
cnext=cu.l;

* output model of utility function
put /'utility function fit'/;
put 'exp utility',@20,'model utility',@40,'a',@50,'b',@60,'c'/;
* changed '15' below, check if period correct (or period + 1)
LOOP (wlvl, umdl=au.l-bu.l*exp(localcone(wlvl)*cu.l); put
jettl(wlvl),@20,umdl,@40,au.l,@50,bu.l,@60,cu.l/);

period = period - 1;

);

***********************************************************************
* End loop over time periods

* Print asset alloc x(j,k) for K0 at start of year 1
```