Using Genetic Algorithms for Optimal Investment Allocation Decision in Defined Contribution Pension Schemes

Kerem Senel
(Corresponding Author)
Isik University
Istanbul, Turkey
ksenel@isikun.edu.tr

A. Bulent Pamukcu
Istanbul Commerce University
Istanbul, Turkey
abpamukcu@iticu.edu.tr

Serhat Yanik
Istanbul University
Istanbul, Turkey
syanik@istanbul.edu.tr

Abstract

In this paper, we demonstrate that genetic algorithms may provide an alternative solution for optimal investment allocation decision in defined contribution pension schemes. Most of the previous research papers attempt to solve the problem analytically. The problem with analytical solutions is that they make numerous restricting assumptions such as lognormal distributions, time-invariant covariance matrices, or short selling restrictions that are not (or, rather, that cannot be) incorporated into the model for the sake of mathematical tractability. Although some of these restricting assumptions can be relaxed, as previously demonstrated by relaxing the assumption of time-invariant covariance matrix, such improvements come at the expense of increased mathematical complexity. Genetic algorithms provide numerical solutions that are not bound by such restricting assumptions. For instance, asset returns can be simulated via a bootstrap method so that the genetic algorithm can work with any distribution and not just with lognormal distribution. Similarly, short selling restrictions can easily be incorporated in the genetic algorithm. This study focuses on the relative performance of

genetic algorithms in solving the asset allocation problem for defined contribution pension schemes. In particular, we compare the simulation results from a standard analytical model with results from a genetic algorithm for analyzing the effect of short selling restrictions. This comparison also sheds light on the degree of suboptimality due to restricting assumptions used in analytical models.

Keywords: Defined Contribution Pension Scheme, Genetic Algorithms, Simulation Approach, Life Insurance Mathematics, Stochastic Processes.

1. Introduction and Related Research

Optimal investment allocation decision in defined contribution pension schemes is quite a different problem compared to other portfolio management problems. The Markowitz portfolio theory, or the so called modern portfolio theory, attempts to minimize the variance of the portfolio return given an expected portfolio return or attempts to maximize the expected portfolio return given the variance of the portfolio return in a single period. In defined contribution pension schemes, the problem is defined as the minimization of a given cost function which reflects the difference between a target fund level and actual fund level for each year. Hence, the problem is attempted to be solved at each point in time given the target and actual fund levels for the previous period.

Vigna and Haberman (2001) apply dynamic programming techniques to find an optimal investment strategy for the scheme member. They consider a defined contribution pension scheme where the fund can be invested in two assets with different levels of risk. This paper simplifies the problem by assuming that the asset returns are independently distributed. Their results suggest the appropriateness of the lifestyle strategy, where the fund is predominantly invested in higher risk instruments, namely equities, when the member is young and it is gradually switched into lower risk instruments, namely bonds and cash, as the member approaches retirement.

Haberman and Vigna (2002) extend the analysis in their previous paper to n assets where the asset returns may be correlated. They again confirm that the optimal strategy for a risk-averse scheme member is the lifestyle strategy. They find that the member switches into lower risk

assets sooner as risk aversion increases and the switch gets realized later as the time to retirement gets longer. They find no evidence of a significant impact of correlation between asset returns on simulation results. They utilize three different risk measures, namely probability of failing the target, mean short fall, and the 5^{th} percentile to analyze the final net replacement ratio. These different risk measures provide different and contradictory results. Also, in this paper, they impose the constraint that short selling is not allowed. Hence, for the case of two assets, when the portfolio weight of a certain asset is greater than 1, it is truncated to 1, and the other portfolio weight which is de facto less than 0 (since sum of portfolio weights is equal to 1) is set equal to 0. This approach, which may yield suboptimal results, is adopted for the sake of mathematical tractability.

Senel and Tuncer (2003) test alternative investment strategies for defined contribution pension schemes when asset returns and salary growth rates are not independent. Their results suggest that the assumption of zero correlation among asset returns and salary growth rates may considerably affect the optimal portfolio selection process. Besides, the net replacement ratio, which is considered to be the single most important indicator for the success of a retirement fund, is overestimated for most cases with the assumption of zero correlation.

Tuncer and Senel (2004) develop a model for the optimal investment allocation decision in a defined contribution pension scheme whose funds are invested in n different assets with a time-varying covariance matrix. They employ a GARCH (1, 1) model to incorporate the dynamic nature of asset risk. By relaxing the assumption of a time-invariant covariance matrix for asset returns, they test for the effects of time-varying risk on investment allocation decisions. In general, they conclude that failing to recognize the time-varying risk structure of asset returns may lead to suboptimal investment allocation decisions. The degree of bias depends on the magnitude of GARCH effects. They base their conclusions on three different risk measures suggested by Haberman and Vigna (2002), i.e. probability of failing the target, mean shortfall, and 5th percentile of the simulated distribution of the net replacement ratio. In this study, they also consider the effect of correlations between asset returns on investment allocation decisions, which they have found to be potentially significant for the time-invariant case in Senel and Tuncer (2003). They observe that the case for the effect of correlations is stronger with the assumption of a time-varying risk structure.

Most of the previous research papers attempt to solve the problem analytically. The problem with analytical solutions is that they make numerous restricting assumptions such as lognormal distributions, time-invariant covariance matrices, or short selling restrictions that are not (or, rather, that cannot be) incorporated into the model for the sake of mathematical tractability. Although some of these restricting assumptions can be relaxed, as Tuncer and Senel (2004) have previously demonstrated by relaxing the assumption of time-invariant covariance matrix, such improvements come at the expense of increased mathematical complexity. Genetic algorithms provide numerical solutions that are not bound by such restricting assumptions. For instance, asset returns can be simulated via a bootstrap method so that the genetic algorithm can work with any distribution and not just with lognormal distribution. Similarly, short selling restrictions can easily be incorporated in the genetic algorithm. This study focuses on the relative performance of genetic algorithms in solving the asset allocation problem for defined contribution pension schemes. In particular, we compare the simulation results from a standard analytical model with results from a genetic algorithm for analyzing the effect of short selling restrictions. This comparison also sheds light on the degree of suboptimality due to restricting assumptions used in analytical models.

2. Genetic Algorithms

Genetic algorithms (or sometimes called evolutionary algorithms) were introduced by Holland (1975) as a method to perform randomized global search in a solution space. As McNelis (2005) indicates, the usefulness of classical optimization techniques such as Newton-based optimization (including backpropagation) and simulated annealing crucially depend on how good the initial parameter guess is. Hence, these techniques are vulnerable to the possibility of getting trapped at local minima. On the other hand, the likelihood of landing in a local minimum is greatly reduced with a genetic algorithm. The genetic algorithm does not require the approximation of Hessian matrices as in the case of Newton-based optimization. Hence, it is a statistical search process like simulated annealing. Yet, the term statistical does not suffice to describe the genetic algorithm, since the process is also evolutionary. Hence, the genetic algorithm is better termed as evolutionary stochastic search.

There are many different versions of the genetic algorithm which utilize different operators. The following algorithm due to McNelis (2005) is a brief description of just one version, which has also been used in the simulations of this study.

The method starts with N (population size) random solution vectors, the elements of which correspond to the input variables of the function to be optimized.[1] This phase of the algorithm is called "population creation".

In the "selection" phase, four solution vectors are randomly picked with replacement to form two pairs. Through a simple fitness tournament, the members of each pair are compared with respect to the fitness function, the function to be optimized, to select the winner of each pair. The winners, or the so called parents, are retained for breeding purposes. Though it is not always used, the "selection" phase has been demonstrated to speed up the convergence of the genetic algorithm.

In the "crossover" phase, the two parents breed two children according to a crossover method, which may be shuffle crossover, arithmetic crossover, or single-point crossover. For instance, in the shuffle crossover method, the corresponding elements of the parent vectors are swapped with a predetermined probability. The swap probability for each element is independent of the swap probabilities for the other elements.

In the "mutation" phase, each element of the two children is exposed to mutation with a small probability which decreases through time.

In the "election tournament" phase, the fitness of the parents and the children are evaluated and the two vectors with the best fitness score are elected for the next generation. The election operator is due to Arifovic (1996), who indicates that the election operator endogenously controls the realized rate of mutation in the genetic search process.

The phases "selection" through "election tournament" are repeated until the next generation is populated with N vectors. An optional phase after the next generation is born is called

[1] See, for instance, McNelis (2005) for a description of genetic algorithms.

"elitism" which requires the best member of the previous generation to replace the worst member of the new generation if the former is fitter than the latter.

The genetic algorithm terminates after a prespecified maximum number of generations, when it hopefully converges to a solution; i.e., no improvement is observed in the fitness score of the best member of each generation for several generations.

The downside to the genetic algorithm is that it can be extremely slow. This phenomenon is an example of the well-known curse of dimensionality in nonlinear optimization. One solution to overcome this problem could be the employment of a hybrid approach, where optimization starts with the genetic algorithm and continues with the gradient-descent or simulated annealing methods. Another possible solution can be the use of parallel algorithms with the advent of parallel processors. In particular, the repetition of "selection" through "election tournament" for the formation of next generation is very suitable for the utilization of a parallel algorithm.

Shapiro (2002) reviews genetic algorithms and soft computing techniques in general in the field of insurance. He mentions a number of studies with genetic algorithms, such as Frick et al. (1996), Lee and Kim (1999), Wendt (1995), and Jackson (1997). The studies by Wendt (1995) and Jackson (1997) investigate asset allocation. Wendt (1995) compares the portfolio efficient frontier of a genetic algorithm to that of a sophisticated non-linear optimizer, whereas Jackson (1997) investigates the performance of a genetic algorithm as a function of discontinuities in the search space.

3. The Model

In our defined contribution pension scheme, funds can be invested in n assets with different levels of risk. Contributions, which are assumed to be a fixed percentage of salary, are paid at the beginning of each period. The only decrement from accumulated funds is assumed to be retirement. Taxation is not taken into account; i.e., contributions and investment income are assumed to be exempt from tax.

3.1. Derivation of Optimal Portfolio Weights with the Analytical Model

We use the model developed by Tuncer and Senel (2004). Following Haberman and Vigna (2002), they define a disutility or cost function that penalizes deviations from targets for level of funds as follows:

$$C_t = (F_t - f_t)^2 + \alpha(F_t - f_t), \; t = 0, 1, ..., N, \tag{1}$$

C_t : Cost incurred at the end of period t
F_t : Target level for accumulated funds at the end of period t
f_t : Actual level of accumulated funds at the end of period t
α : Risk aversion parameter

where $\alpha = 0$. For the above disutility function, risk aversion is decreasing with increasing risk aversion parameter α. Hence, greater values of α are associated with less risk-averse individuals. The target level for each period is given a priori.

Hence, the optimization problem can be stated as

$$\min \; E(C_{t+1} | I_t) \; . \tag{2}$$

The actual level of funds at time t + 1 can be expressed as

$$f_{t+1} = (f_t + c)(\mathbf{y}'\mathbf{W}), \tag{3}$$

or,

$$f_{t+1} = (f_t + c)(\mathbf{W}'\mathbf{y}), \tag{4}$$

where c is the contribution rate, **y** is the vector of portfolio weights, and **W** is the vector of real (gross) asset returns. Here, the real salary growth rate is assumed to be zero and the real salary level is set at 1. Therefore, the annual contribution by the individual is assumed to be

equal to a constant contribution rate, c, throughout the whole period. Although **y** and **W** are dependent on time, the subscript t is dropped for convenience.

y and **W** can explicitly be written as

$$\mathbf{y}' = [y_{1t} \quad y_{2t} \quad \dots \quad y_{nt}], \tag{5}$$

and,

$$\mathbf{W}' = [W_{1t} \quad W_{2t} \quad \dots \quad W_{nt}],\ W_{it} = e^{X_{it}}, \qquad X_{it} \sim N(\mu_i,\ \sigma_{it}^2), \tag{6}$$

where X_{it} is the real force of interest for the i-th asset in period t, which is assumed to be constant throughout the period. The asset returns are assumed to be lognormally distributed and the usual assumption of time-invariant covariance matrix of asset returns is relaxed. If equation (3) is rewritten explicitly,

$$f_{t+1} = (f_t + c)[y_{1t} \quad y_{2t} \quad \dots \quad y_{nt}] \begin{bmatrix} e^{\mu_1 + V_{1t}} \\ e^{\mu_2 + V_{2t}} \\ \dots \\ e^{\mu_n + V_{nt}} \end{bmatrix}, \qquad V_{it} \sim N(0,\ \sigma_{it}^2), \tag{7}$$

or,

$$f_{t+1} = (f_t + c)\left(y_{1t} e^{\mu_1} e^{V_{1t}} + y_{2t} e^{\mu_2} e^{V_{2t}} + \dots + y_{nt} e^{\mu_n} e^{V_{nt}}\right). \tag{8}$$

Tuncer and Senel (2004) define

$$\mathbf{y_T}' = [y_{1t} e^{\mu_1} \quad y_{2t} e^{\mu_2} \quad \dots \quad y_{nt} e^{\mu_n}], \tag{9}$$

and,

$$\mathbf{W_T}' = [e^{V_{1t}} \quad e^{V_{2t}} \quad \dots \quad e^{V_{nt}}], \tag{10}$$

where $\mathbf{y_T}$ is the transformed vector of portfolio weights and $\mathbf{W_T}$ is the transformed vector of (gross) asset returns. Similar to $\mathbf{y}$ and $\mathbf{W}$, $\mathbf{y_T}$ and $\mathbf{W_T}$ are also dependent on time, but the subscript t is dropped for convenience. Then, equation (8) can be rewritten as

$$f_{t+1} = (f_t + c)\left(\mathbf{y_T}'\mathbf{W_T}\right), \tag{11}$$

or,

$$f_{t+1} = (f_t + c)\left(\mathbf{W_T}'\mathbf{y_T}\right). \tag{12}$$

If the constraint that the portfolio weights should add up to 1 is imposed upon, the optimization problem can be stated as

$$\begin{aligned} &\min\ E(C_{t+1}|I_t) \\ &\text{subject to} \quad \boldsymbol{\mu}'\mathbf{y_T} = 1, \end{aligned} \tag{13}$$

where $\boldsymbol{\mu}$ is a vector comprised of all $e^{-\mu_i}$'s. $\boldsymbol{\mu}$ can explicitly be written as

$$\boldsymbol{\mu}' = \left[e^{-\mu_1} \quad e^{-\mu_2} \quad \dots \quad e^{-\mu_n}\right]. \tag{14}$$

In order to solve the optimization problem in equation (13), a Lagrange multiplier, κ, is associated with the above constraint and the Lagrangian

$$E(C_{t+1}|I_t) - \kappa(\boldsymbol{\mu}'\mathbf{y_T} - 1) \tag{15}$$

is formed.

Now, the optimization problem in equation (13) becomes

$$\min\ \left[E(C_{t+1}|I_t) - \kappa(\boldsymbol{\mu}'\mathbf{y_T} - 1)\right]. \tag{16}$$

Tuncer and Senel (2004) derive $\mathbf{y_T}$, the transformed optimal portfolio weights that minimize the expected cost function at t + 1, as

$$\mathbf{y_T} = \frac{(2F_{t+1} + \alpha)(f_t + c)\mathbf{H}^{-1}E(\mathbf{W_T}|I_t) + \kappa\mathbf{H}^{-1}\boldsymbol{\mu}}{2(f_t + c)^2}, \tag{17}$$

where,

$$\mathbf{H} = E\left(\mathbf{W_T}\mathbf{W_T}'\middle|I_t\right) \text{ and} \tag{18}$$

$$\kappa = \frac{2(f_t + c)^2 - (2F_{t+1} + \alpha)(f_t + c)\boldsymbol{\mu}'\mathbf{H}^{-1}E(\mathbf{W_T}|I_t)}{\boldsymbol{\mu}'\mathbf{H}^{-1}\boldsymbol{\mu}}. \tag{19}$$

The analytical model doesn't take into account the short selling restriction. Hence the portfolio weights to be calculated using $\mathbf{y_T}$ are truncated to the range [0, 1].

This is a straightforward procedure for two assets. If one of the portfolio weights is less than 0, the other portfolio weight must be greater than 1. Furthermore, the difference between the latter and 1 must be equal to the difference between the former and 0, since the sum of the portfolio weights is equal to 1. Hence, if one of the portfolio weights is less than 0, it is set equal to 0 and the other portfolio weight, which is greater than 1, is set equal to 1 if we have only two assets.

The problem becomes much more complicated for more than two assets. If the above procedure is applied, then the sum of the portfolio weights will not be equal to 1. For instance, assume that we have three assets for which the portfolio weights are –0.5, 0.3, and 1.2. If we set the portfolio weights for the first and third assets to be equal to 0 and 1, respectively, then the sum of the portfolio weights is going to be equal to 1.3. Therefore, we need a more complicated procedure in such a case.

We propose the following procedure for the case of three assets:

i. Find the maximum distance between the portfolio weights and the violated boundaries for those portfolio weights outside the range [0, 1].

ii. If the portfolio weight corresponding to the maximum distance is greater than 1:
   - Set that portfolio weight to be equal to 1.
   - Set the other two portfolio weights, which are de facto negative, to be equal to 0.

   If the portfolio weight corresponding to the maximum distance is less than 0:
   - Set that portfolio weight to be equal to 0.
   - Reduce the other portfolio weights, which are de facto positive, so that they are proportional to their original sizes and the total reduction is equal to the maximum distance.

If this procedure is applied to the above mentioned example, the curtailed portfolio weights become 0, 0.175, and 0.825.

3.2. Derivation of Optimal Portfolio Weights with the Genetic Algorithm

In the derivation of optimal portfolio weights with the genetic algorithm, we use two different variants of the genetic algorithm, namely the real value encoded and the binary value encoded.

In the real value encoded genetic algorithm, the solution vector is comprised of N-1 elements which correspond to the portfolio weights of N-1 assets. The portfolio weight of the N-th asset is dependent on other portfolio weights since the sum of all portfolio weights should be equal to 1. With this type of genetic algorithm, there is the possibility of attaining portfolio weights outside the range [0, 1] due to the mutation operator. To prevent this, we use a special technique called mirroring, which corresponds to taking the mirror image of the solution outside the range with respect to the violated boundary until the solution lands in the search space.

In the binary value encoded genetic algorithm, each variable is represented with a string of 1's and 0's. To achieve this, the search space corresponding to the variable is discretized according to some power of 2. For instance, if the search space is divided into $2^n - 1$ equal intervals, then n bits are enough to represent the variable. This type of encoding ensures that

the solution remains within the search space, since any combination of 1's and 0's will correspond to a valid solution.

First, we define the cost function in (1) to be the fitness function to be minimized. We use a population size of 10 for both the real value encoded and the binary value encoded genetic algorithms.

In the "population creation" phase of the real value encoded genetic algorithm, the elements of the solution vectors are randomly generated from a uniform distribution in the range [0, 1].[2] Yet, random number generation is repeated until the sum of the N-1 portfolio weights for each solution vector remains below 1, so that the portfolio weight of the N-th asset is ensured to be positive. For the binary value encoded genetic algorithm, we use 10 bits to represent each portfolio weight. This corresponds to an interval size of $1/(2^{10} - 1)$ or 0.0978% between discrete values, which seems to provide more than enough precision for portfolio weights. Similar to the real value encoded genetic algorithm, random number generation is repeated until the sum of the N-1 portfolio weights for each solution vector remains below 1 to ensure that the portfolio weight of the N-th asset is positive.

The "crossover" phases of both the real value encoded and the binary value encoded genetic algorithms potentially create "bad" children that violate the short selling restriction for the N-th asset. In other words, the sum of the N-1 portfolio weights may exceed 1 for some solution vectors. Yet, the elimination of such "bad children" is postponed until the "election tournament" phase. Our genetic algorithms use the shuffle crossover method with a 50% probability of shuffle.

As mentioned before, in the "mutation" phase of the real value encoded genetic algorithm, we use a special technique called mirroring to prevent the possibility of attaining portfolio weights outside the range [0, 1]. For the binary value encoded genetic algorithm, again as mentioned before, this is not a problem since all solution vectors de facto belong to the search space. Similar to their "crossover" phases, the "mutation" phases of both the real value encoded and the binary value encoded genetic algorithms potentially create "bad" children

[2] The general framework for the genetic algorithm we employ is due to McNelis (2005).

that violate the short selling restriction for the N-th asset. Again, the elimination of such "bad children" is postponed until the "election tournament" phase.

The probability of mutation is given by

$$\text{Probability of Mutation} = .15 + \frac{.33}{\text{Generation Number}},^{3} \tag{20}$$

which translates into a lower probability of mutation which decreases through time.

For the real value encoded genetic algorithm, the nonuniform mutation operator due to Michalewicz (1996) is utilized:

$$y_{\text{after mutation}} = y_{\text{before mutation}} + s\left[1 - r_2^{\left(1 - \frac{\text{Generation Number}}{\text{Maximum Number of Generations}}\right)^2}\right], \text{ if } r_1 > .5, \tag{21}$$

$$y_{\text{after mutation}} = y_{\text{before mutation}} - s\left[1 - r_2^{\left(1 - \frac{\text{Generation Number}}{\text{Maximum Number of Generations}}\right)^2}\right], \text{ if } r_1 \leq .5, \tag{22}$$

where y is a portfolio weight, r_1 and r_2 are random numbers from a uniform distribution in the range [0, 1], and s is a random number from a standard normal distribution. The mutation operator is nonuniform, since the odds of a mutated portfolio weight that is substantially different from the original portfolio weight become smaller and smaller as "Generation Number" gets closer to "Maximum Number of Generations" (or, as time passes). For the binary value encoded genetic algorithm, the mutation simply inverts the bit; i.e., a "1" becomes "0" and a "0" becomes "1".

We set the maximum number of generations to be equal to 20.

[3] Due to McNelis (2005).

3.3. Conversion of Final Fund into a Whole Life Annuity Due

We assume that the final fund (the actual level of accumulated funds at retirement), f_{final}, is converted into a whole life annuity due. Using the retiree's expected mortality[4] and the return of the low-risk asset as the discount rate, the annual retirement income is given by

$$\text{Annual Retirement Income} = \frac{f_{final}}{\ddot{a}_x}, \tag{23}$$

where $\ddot{a}_x$ is the actuarial present value of a whole life annuity of 1 payable at the beginning of each year (starting immediately after retirement) as long as the retiree who is at the age of x at retirement survives. The actuarial present value of a whole life annuity of 1 payable at the beginning of each year, $\ddot{a}_x$, is given by

$$\ddot{a}_x = \sum_{k=0}^{\omega} {}_k p_x \nu^k, \tag{24}$$

where ω is the maximum age, ${}_k p_x$ is the probability that a retiree who is at the age of x (at retirement) survives for k years, and ν is the discount factor that uses the return of the low-risk asset. ν is given by

$$\nu = E(e^{-X}), \quad X \sim N(\mu_{\text{low-risk asset}}, \sigma^2_{\text{low-risk asset}}), \text{ or,} \tag{25}$$

$$\nu = e^{-\mu_{\text{low-risk asset}} + 0.5\sigma^2_{\text{low-risk asset}}}, \tag{26}$$

where $\mu_{\text{low-risk asset}}$ and $\sigma^2_{\text{low-risk asset}}$ are the mean and variance of the low-risk asset.

Since the final fund is assumed to be converted into an annuity, the retiree will be more concerned with the net replacement ratio than the final fund. The net replacement ratio is defined as the ratio of retirement income to final salary. Since the real salary growth rate is assumed to be zero and the real salary level is set at 1, the net replacement ratio is given by

[4] For our simulations, we have used the 1983 US GATT (unisex) mortality table.

$$\text{Net Replacement Ratio} = \frac{\text{Annual Retirement Income}}{\text{Final Salary}} = \frac{f_{final} / \ddot{a}_x}{1} = \frac{f_{final}}{\ddot{a}_x}. \quad (27)$$

4. Simulations

4.1. Experiment Setup

In our simulations, we consider two different setups. In the first setup, there are two assets, one of them being the high-risk asset that may be associated with equity investment and the other being the low-risk asset that may be associated with fixed-income investment. In the second setup, there are three assets. These assets can similarly be classified as high-risk, medium-risk, and low-risk assets. Table 1 summarizes the parameter values that are common to both of these setups. Tables 2a and 2b provide the mean and covariance matrices of the asset returns for the two-asset and three-asset cases, respectively.

Table 1
Parameter Values for Simulations

Parameter	Parameter Value
Entry Age	25
Duration of Employment	30
Retirement Age	55
Contribution Range (c)	8%
Risk Aversion (α)	2

Table 2a
Mean and Covariance Matrices for the Two-Asset Case

Mean of the Log Return for the Low-Risk Asset	5.0%
Mean of the Log Return for the High-Risk Asset	10.0%
Standard Deviation of the Log Return for the Low-Risk Asset	7.5%
Standard Deviation of the Log Return for the High-Risk Asset	20.0%
Correlation Coefficient Between the Log Returns for the Low-Risk and High-Risk Assets	0.50

Table 2b
Mean and Covariance Matrices for the Three-Asset Case

Mean of the Log Return for the Low-Risk Asset	3.0%
Mean of the Log Return for the Medium-Risk Asset	5.0%
Mean of the Log Return for the High-Risk Asset	10.0%
Standard Deviation of the Log Return for the Low-Risk Asset	5.0%
Standard Deviation of the Log Return for the Medium-Risk Asset	7.5%
Standard Deviation of the Log Return for the High-Risk Asset	20.0%
Correlation Coefficient Between the Log Returns for the Low-Risk and Medium-Risk Assets	0.40
Correlation Coefficient Between the Log Returns for the Low-Risk and High-Risk Assets	0.25
Correlation Coefficient Between the Log Returns for the Medium-Risk and High-Risk Assets	0.50

As mentioned before, one of the aims of this study is to measure the degree of suboptimality in failing to incorporate short selling restrictions in a standard analytical model. In order to achieve this, we compare the results from a standard analytical model with the results from a genetic algorithm. The portfolio weights from the standard analytical model are curtailed to the range [0, 1] if they violate the short selling restrictions. Hence, the restrictions are actually not incorporated in the analytical model, but they are, rather, imposed upon after the model produces the portfolio weights. On the other hand, the short selling restrictions are incorporated in the genetic algorithm. Hence, it follows that the difference between the performances of these two approaches in terms of the statistical properties of the net replacement ratio is expected to give a measure for the degree of suboptimality in failing to incorporate short selling restrictions in standard analytical models.

For the two asset-case, the above mentioned comparison is made between the outputs of the analytical model and the real value encoded genetic algorithm. For the three asset case, on the other hand, the comparison is made between the outputs of the analytical model, the real value encoded genetic algorithm, and the binary value added genetic algorithm. Hence, the second comparison will also provide information regarding the relative performances of the real value encoded and the binary value encoded versions of the genetic algorithm for the optimal investment allocation decision in defined contribution pension schemes.

In our simulations, we set the initial fund level at zero. For each case, 100 simulations are carried out to simulate the net replacement ratio, which may be considered as the single most

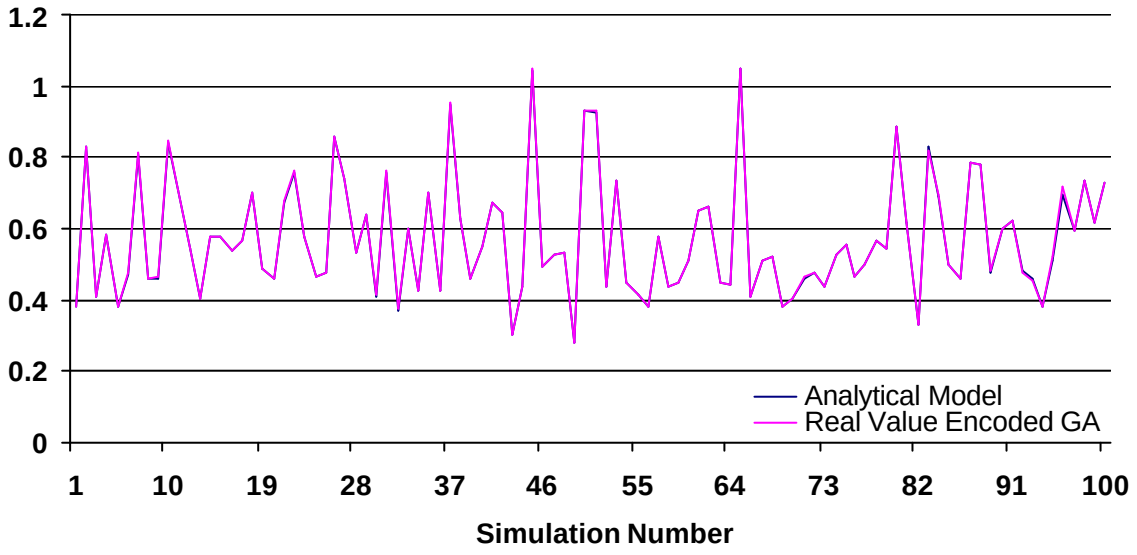
important indicator for the success of a retirement fund. For each sample of 100 simulations, we compute a number of risk measures, namely the probability of failing the target, mean shortfall, and the 5th percentile.[5]

4.2. Simulation Results

4.2.1. The Two Asset Case – Analytical Model vs. Real Value Encoded GA

Figure 1 gives the net replacement ratios simulated by the standard analytical model and the real value encoded genetic algorithm for the two asset case.

Figure 1
Net Replacement Ratios for the Two-Asset Case



As Figure 1 suggests, there is very little difference between the outputs of the analytical model and the real value encoded genetic algorithm for the two asset case. This can be verified by Table 3 which summarizes the statistical properties of the net replacement ratios and their differences together with the result of a t-test testing for the difference in means of two samples.

[5] See Haberman and Vigna (2002) for a description of these risk measures.

Table 3
Net Replacement Ratios for the Two-Asset Case*

Parameter	Mean	Standard Deviation
$NRR_{Analytical\ Model}$	0.5719	0.1645
$NRR_{Real\ Value\ Encoded\ GA}$	0.5728	0.1650
$NRR_{Analytical\ Model} - NRR_{Real\ Value\ Encoded\ GA}$	-0.0009	0.0027
$\lvert NRR_{Analytical\ Model} - NRR_{Real\ Value\ Encoded\ GA} \rvert$	0.0017	0.0023

* Performing a t-test assuming that the two samples come from normal distributions with unknown and possibly unequal variances, we cannot reject the null hypothesis that the net replacement ratios simulated by the analytical model and the real value encoded genetic algorithm come from distributions with equal means at the 5% significance level. The t-statistic is -0.0387 and the corresponding p-value is 0.4846.

Based on the evidence from this experiment, we cannot claim that failing to incorporate short selling restrictions in a standard analytical model is suboptimal for the two-asset case. Of course, this doesn't mean that analytical models without short selling restrictions *always* generate optimal results. Ours is just one experiment with a single set of assumptions for parameters. Other experiments with different assumptions may prove the opposite.

4.2.2. The Three Asset Case

4.2.2.1. Analytical Model vs. Real Value Encoded GA

Figure 2a gives the net replacement ratios simulated by the standard analytical model and the real value encoded genetic algorithm for the three-asset case.

Figure 2a
Net Replacement Ratios for the Three-Asset Case

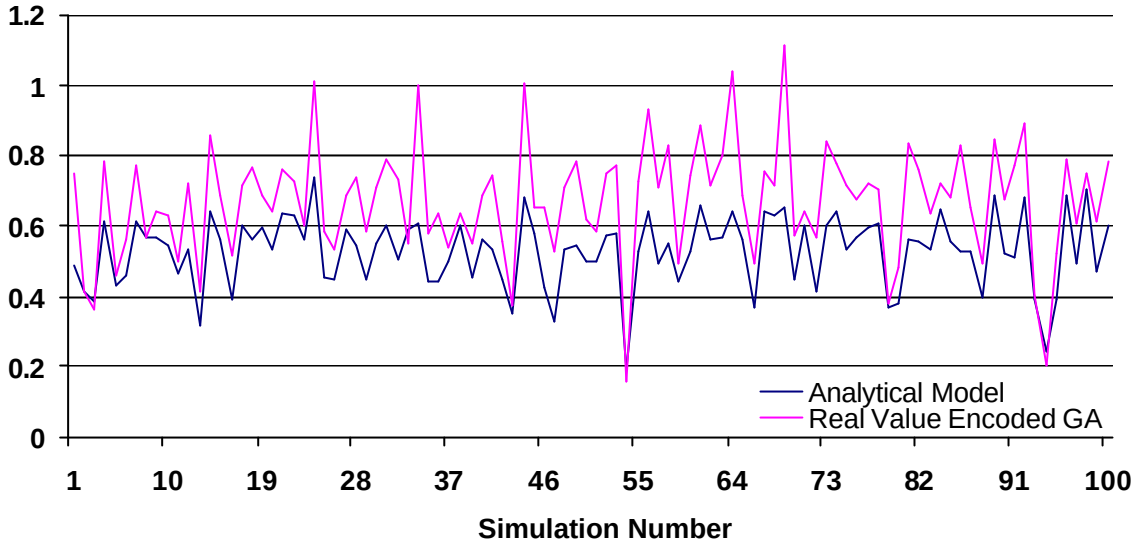


Figure 2a suggests that there is remarkable difference between the outputs of the analytical model and the real value encoded genetic algorithm for the three-asset case. This can be verified by Table 4a which summarizes the statistical properties of the net replacement ratios and their differences together with the result of a t-test testing for the difference in means of two samples.

Table 4a
Net Replacement Ratios for the Three-Asset Case*

Parameter	Mean	Standard Deviation
$NRR_{Analytical\ Model}$	0.5290	0.1017
$NRR_{Real\ Value\ Encoded\ GA}$	0.6727	0.1635
$NRR_{Analytical\ Model} - NRR_{Real\ Value\ Encoded\ GA}$	-0.1437	0.0936
$\lvert NRR_{Analytical\ Model} - NRR_{Real\ Value\ Encoded\ GA} \rvert$	0.1464	0.0892

* Performing a t-test assuming that the two samples come from normal distributions with unknown and possibly unequal variances, we reject the null hypothesis that the net replacement ratios simulated by the analytical model and the real value encoded genetic algorithm come from distributions with equal means at the 5% significance level. The t-statistic is -7.4619 and the corresponding p-value is 2.2728×10^{-12}.

Based on the evidence from this experiment, we conclude that failing to incorporate short selling restrictions in a standard analytical model is indeed suboptimal for the three-asset case. Actually, in 96 out of the 100 simulations, the real value encoded genetic algorithm dominates

the analytical model in terms of net replacement ratio. As can be seen in Table 4a, the mean net replacement ratio for the real value encoded genetic algorithm is 127% of the mean net replacement ratio for the analytical model. Again as indicated in Table 4a, the dominance in favor of the real value encoded genetic algorithm is statistically significant with a p-value which is virtually equal to 0.

The astonishing difference between the results obtained for the two-asset and three-asset cases shows how misleading it can be to make generalizations from a simple two-asset case to a case of n assets.

4.2.2.2. Analytical Model vs. Binary Value Encoded GA

Figure 2b gives the net replacement ratios simulated by the standard analytical model and the binary value encoded genetic algorithm for the three-asset case.

Figure 2b
Net Replacement Ratios for the Three-Asset Case

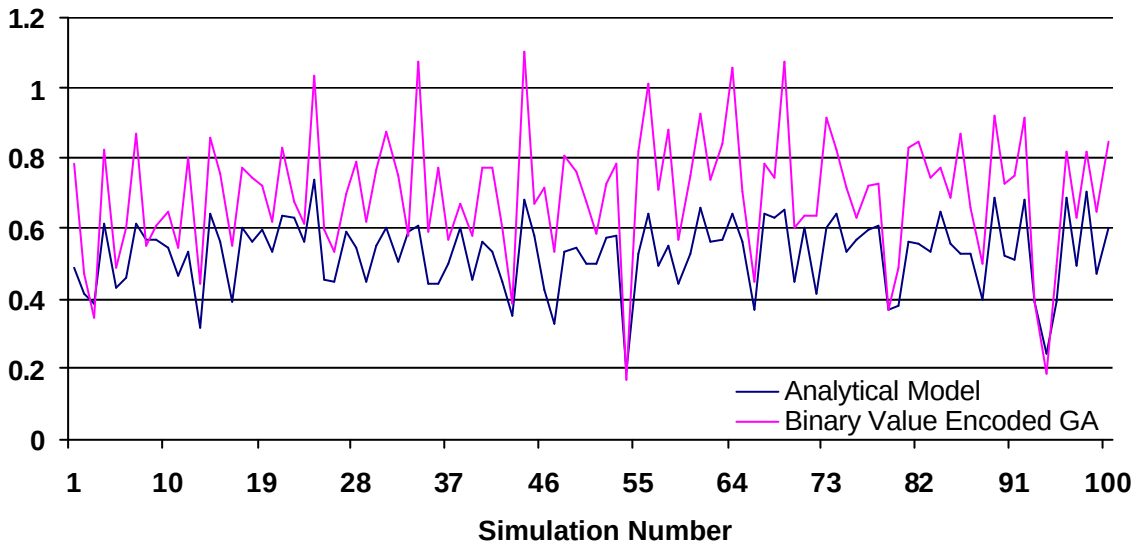


Figure 2b suggests that there is remarkable difference between the outputs of the analytical model and the binary value encoded genetic algorithm for the three-asset case. This can be verified by Table 4b which summarizes the statistical properties of the net replacement ratios and their differences together with the result of a t-test testing for the difference in means of two samples.

Table 4b
Net Replacement Ratios for the Three-Asset Case*

Parameter	Mean	Standard Deviation
$NRR_{Analytical\ Model}$	0.5290	0.1017
$NRR_{Binary\ Value\ Encoded\ GA}$	0.7006	0.1756
$NRR_{Analytical\ Model} - NRR_{Binary\ Value\ Encoded\ GA}$	-0.1716	0.1068
$\lvert NRR_{Analytical\ Model} - NRR_{Binary\ Value\ Encoded\ GA} \rvert$	0.1747	0.1016

* Performing a t-test assuming that the two samples come from normal distributions with unknown and possibly unequal variances, we reject the null hypothesis that the net replacement ratios simulated by the analytical model and the binary value encoded genetic algorithm come from distributions with equal means at the 5% significance level. The t-statistic is -8.4597 and the corresponding p-value is 8.2500×10^{-15}.

Similar to the previous case, the evidence from this experiment repeatedly manifests that failing to incorporate short selling restrictions in a standard analytical model is suboptimal. Actually, in 94 out of the 100 simulations, the binary value encoded genetic algorithm dominates the analytical model in terms of net replacement ratio. As can be seen in Table 4b, the mean net replacement ratio for the binary value encoded genetic algorithm is 132% of the mean net replacement ratio for the analytical model. Again, as indicated in Table 4b, the dominance in favor of the binary value encoded genetic algorithm is statistically significant with a p-value which is virtually equal to 0.

4.2.2.3. Real Value Encoded GA vs. Binary Value Encoded GA

Figure 2c gives the net replacement ratios simulated by the real value encoded genetic algorithm and the binary value encoded genetic algorithm for the three-asset case.

Figure 2c
Net Replacement Ratios for the Three-Asset Case

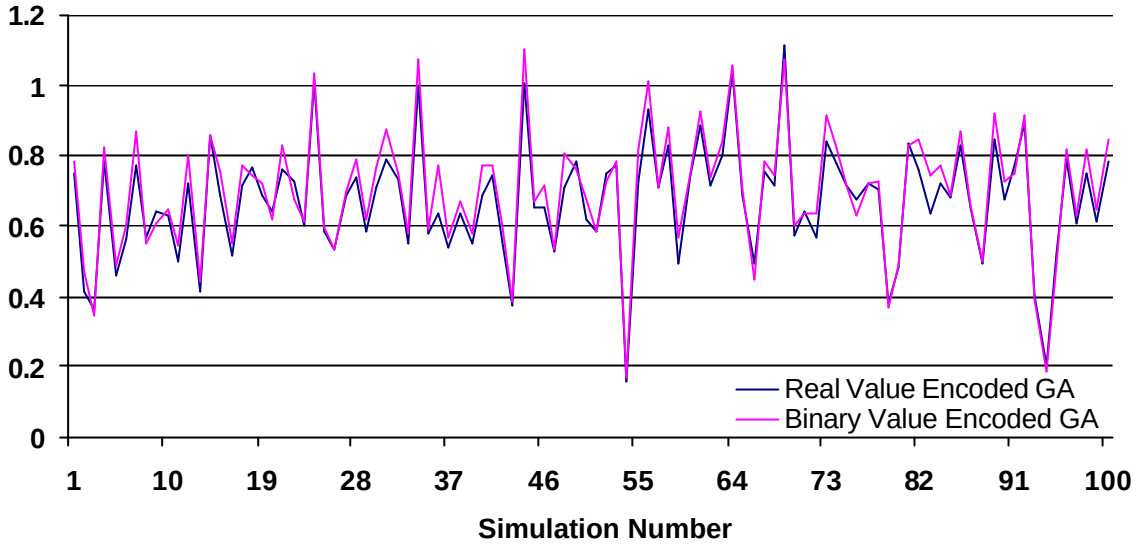


Figure 2c suggests that there is some difference between the outputs of the real value encoded genetic algorithm and the binary value encoded genetic algorithm for the three-asset case. This can be verified by Table 4c which summarizes the statistical properties of the net replacement ratios and their differences together with the result of a t-test testing for the difference in means of two samples.

Table 4c
Net Replacement Ratios for the Three-Asset Case*

Parameter	Mean	Standard Deviation
$NRR_{\text{Real Value Encoded GA}}$	0.6727	0.1635
$NRR_{\text{Binary Value Encoded GA}}$	0.7006	0.1756
$NRR_{\text{Real Value Encoded GA}} - NRR_{\text{Binary Value Encoded GA}}$	-0.0279	0.0377
$\lvert NRR_{\text{Real Value Encoded GA}} - NRR_{\text{Binary Value Encoded GA}} \rvert$	0.0371	0.0286

* Performing a t-test assuming that the two samples come from normal distributions with unknown and possibly unequal variances, we cannot reject the null hypothesis that the net replacement ratios simulated by the real value encoded genetic algorithm and the binary value encoded genetic algorithm come from distributions with equal means at the 5% significance level. The t-statistic is -1.1645 and the corresponding p-value is 0.1228.

In 78 out of the 100 simulations, the binary value encoded genetic algorithm dominates the real value encoded genetic algorithm in terms of net replacement ratio. As can be seen in Table 4c, the mean net replacement ratio for the binary value encoded genetic algorithm is

104% of the mean net replacement ratio for the real value encoded genetic algorithm. However, as indicated in Table 4c, the dominance in favor of the binary value encoded genetic algorithm is not statistically significant.

4.2.3. Risk Measures

Tables 5a and 5b summarize some risk measures, namely the probability of failing the target, mean shortfall and 5th percentile, for the two-asset and three-asset cases, respectively.

Table 5a
Risk Measures for the Two-Asset Case

Risk Measure	Analytical Model	Real Value Encoded GA
Probability of Failing the Target	0.09	0.09
Mean Shortfall	0.0458	0.0453
5th Percentile	0.3784	0.3804

Table 5b
Risk Measures for the Three-Asset Case

Risk Measure	Analytical Model	Real Value Encoded GA	Binary Value Encoded GA
Probability of Failing the Target	0.66	0.25	0.20
Mean Shortfall	0.0969	0.1089	0.1231
5th Percentile	0.3554	0.3774	0.3814

For the two asset case, the probability of failing the target is the same for the analytical model and the real value encoded genetic algorithm, whereas the other risk measures are very close. This is an expected outcome since the difference in means of two samples is not statistically significant, as we have demonstrated in Section 4.2.1.

For the three asset case, there is a substantial difference between the probabilities of failing the target in favor of the genetic algorithm. Also, the 5th percentiles for the genetic algorithm are slightly better than the 5th percentile for the analytical model. Together with the evidence presented in Section 4.2.2.1 and 4.2.2.2, the differences in these risk measures, particularly the difference in the probability of failing the target, saliently manifest the suboptimality due to the failure to incorporate short selling restrictions in the analytical model. Interestingly, the

mean shortfall for the analytical model is less than the mean shortfall for the genetic algorithm. Considering that the analytical model fails 66% of the time whereas the genetic algorithm fails only 20 to 25% of the time, the dominance in terms of shortfall is not a strong argument in favor of the analytical model. Actually, if we define "expected mean shortfall" as another risk measure to incorporate both the probability of failure and the mean shortfall, it is clearly evident that the analytical model without short selling restrictions generate suboptimal results.[6]

5. Conclusions

In this paper, we present genetic algorithms as an alternative solution for optimal investment allocation decision in defined contribution pension schemes. Most of the previous research papers attempt to solve the problem analytically. The problem with analytical solutions is that they make numerous restricting assumptions such as lognormal distributions, time-invariant covariance matrices, or short selling restrictions that are not (or, rather, that cannot be) incorporated into the model for the sake of mathematical tractability. Genetic algorithms provide numerical solutions that are not bound by such restricting assumptions. For instance, short selling restrictions can easily be incorporated in the genetic algorithm. This study focuses on the relative performance of genetic algorithms in solving the asset allocation problem for defined contribution pension schemes. In particular, we compare the simulation results from a standard analytical model with results from a genetic algorithm for analyzing the effect of short selling restrictions.

For the case of two assets, the net replacement ratios and the risk measures for the standard analytical model and the real value encoded genetic algorithm are similar. Hence, we cannot claim that failing to incorporate short selling restrictions in a standard analytical model is suboptimal for the two-asset case. Of course, this doesn't mean that analytical models without short selling restrictions *always* generate optimal results. Ours is just one experiment with a single set of assumptions for parameters. Other experiments with different assumptions may prove the opposite.

[6] Define Expected Mean Shortfall = Probability of Failing the Target × Mean Shortfall. Then the expected mean shortfall is 0.0640, 0.0272, and 0.0246 for the analytical model, real value encoded genetic algorithm, and binary value encoded genetic algorithm, respectively.

For the case of three assets, on the other hand, there is a remarkable difference between the net replacement ratios of the analytical model and the real value encoded genetic algorithm. Furthermore, the dominance in favor of the real value encoded genetic algorithm is statistically significant with a p-value which is virtually equal to 0. Therefore, we conclude that failing to incorporate short selling restrictions in a standard analytical model is indeed suboptimal for the three-asset case.

We observe a similar, if not stronger, case in favor of the genetic algorithm when binary value encoding is used instead of real value encoding. The binary value encoded genetic algorithm slightly dominates the real value encoded genetic algorithm; yet, the difference is not statistically significant.

Risk measures, with the possible exception of mean shortfall, also manifest the suboptimality due to the failure to incorporate short selling restrictions in the analytical model. In particular, there is a substantial difference between the probabilities of failing the target in favor of the genetic algorithm.

The astonishing difference between the results obtained for the two-asset and three-asset cases shows how misleading it can be to make generalizations from a simple two-asset case to a case of n assets.

References

Arifovic J., 1996. The Behavior of the Exchange Rate in the Genetic Algorithm and Experimental Economies. Journal of Political Economy 104, 510-541.

Frick A., Herrmann R., Kreidler M., Narr A., Seese D., 1996. Genetic-Based Trading Rules - A New Tool to Beat the Market with - First Empirical Results. AFIR 2, 997-1017.

Haberman S., Vigna E., 2002. Optimal Investment Strategies and Risk Measures in Defined Contribution Pension Schemes. Insurance: Mathematics and Economics 31, 35-69.

Holland J. H., 1975. Adaptation in Natural and Artificial Systems. MIT Press.

Jackson A., 1997. Genetic Algorithms for Use in Financial Problems. AFIR 2, 481-503.

Lee B., Kim M., 1999. Application of Genetic Algorithm to Automobile Insurance for Selection of Classification Variables: The Case of Korea. Paper Presented at the 1999 Annual Meeting of the American Risk and Insurance Association.

McNelis P. D., 2005. Neural Networks in Finance. Elsevier.

Michalewicz Z., 1996. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag.

Senel K., Tuncer R., 2003. Simulation of Alternative Investment Strategies for Defined Contribution Pension Schemes when Asset Returns and Salary Growth Rates are not Independent. Paper presented at the 7th Congress on Insurance: Mathematics and Economics, Lyon, France.

Shapiro A. F., 2002. The Merging of Neural Networks, Fuzzy Logic, and Genetic Algorithms. Insurance: Mathematics and Economics 31, 115-131.

Tuncer R., Senel K., 2004. Optimal Investment Allocation Decision in Defined Contribution Pension Schemes with Time-Varying Risk. Paper presented at the 8th Congress on Insurance: Mathematics and Economics, Rome, Italy.

Vigna E., Haberman S., 2001. Optimal Investment Strategy for Defined Contribution Pension Schemes. Insurance: Mathematics and Economics 28, 233-262.

Wendt R. Q., 1995. Build Your Own GA Efficient Frontier. Risks and Rewards, December.