META-HEURISTICS:
Theory & Applications

edited by

Ibrahim H. Osman
and
James P. Kelly

Kluwer Academic Publishers
Boston/London/Dordrecht

Distributors for North America:
Kluwer Academic Publishers
101 Philip Drive
Assinippi Park
Norwell, Massachusetts 02061 USA

Distributors for all other countries:
Kluwer Academic Publishers Group
Distribution Centre
Post Office Box 322
3300 AH Dordrecht, THE NETHERLANDS

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available from the Library of Congress.

Printed on acid-free paper.

Printed in the United States of America

26

Solving Dynamic Stochastic Control Problems in Finance Using Tabu Search with Variable Scaling

Fred Glover
School of Business
University of Colorado at Boulder
Boulder, CO 80309
E-mail: fred.glover@colorado.edu

John M. Mulvey*
Department of Civil Engineering and Operations Research
Princeton University
Princeton, NJ 08544
E-mail: mulvey@macbeth.princeton.edu

Kjetil Hoyland
Department of Managerial Economics and Operations Research
University of Trondheim
N-7034 Trondheim, Norway
E-mail: kth@iok.unit.no

Abstract

Numerous multistage planning problems in finance involve nonlinear and nonconvex decision controls. One of the simplest is the fixed-mix

* Research supported in part by NSF grant CCR-9102660 and Air Force grant AFOSR-91-0359. We acknowledge the assistance of Michael T. Tapia in preparing this paper for publication.

investment strategy. At each stage during the planning horizon, an investor rebalances her/his portfolio in order to achieve a target mix of asset proportions. The decision variables represent the target percentages for the asset categories. We show that a combination of Tabu Search and Variable Scaling generates global optimal solutions for real world test cases, despite the presence of nonconvexities. Computational results demonstrate that the approach can be applied in a practical fashion to investment problems with over 20 stages (20 years), 100 scenarios, and 8 asset categories. The method readily extends to more complex investment strategies with varying forms of nonconvexities.

Key Words: Dynamic stochastic control, financial modeling, tabu search, variable scaling.

1. Introduction

Efforts to explain financial markets often employ stochastic differential equations to model randomness over time. For example, interest rate modeling has been an active research area for the past decade, resulting in numerous approaches for modeling the yield curve. See Brennan and Schwartz (1982), Hull (1993), Ingersoll (1987) and Jarrow (1995) for details. These equations form the basis for various analytical studies such as estimating the "fair" market value of securities that display behavior conditional on the future path of interest rates. Often, for tractability (Hull 1993), these techniques discretize the planning horizon into a fixed number of time steps $t \in \{1,...,T\}$ and the random variables into a finite number of outcomes.

Our goal is to employ a similar discretization of time and randomness. But instead of estimating fair market value, we are interested in analyzing alternative investment strategies over extended planning periods -- 10 or even 20 years. The basic decision problem is called asset allocation. The universe of investments is divided into a relatively small number of generic asset categories (US stocks, bonds, cash, international stocks) for the purposes of managing portfolio risks. There is much evidence that the most critical investment decision involves selecting the proportion of assets placed into these categories, especially for investors who are well diversified and for large institutions such as insurance companies and pension plans.

In our model, we assume that future economic conditions are presented in the form of a modest number of scenarios. A scenario is described as a complete path of all economic factors and accompanying returns for the assets over the planning horizon. Asset allocation decision are made at the end of each of the T time periods. They cannot be changed during the period, but they can and do respond to the changing state of the world (including the condition of our portfolio) at the beginning of the next period. To keep the problem in a form that can be understood by financial planners and investment managers, we have assumed a decision rule that is intuitive and commonly used -- the fixed-mix rule. Under this rule, the portfolio has a predetermined mix at the beginning of each period -- for instance, 60% stock, 30% bonds, and 10% cash. The portfolio must be rebalanced by selling and buying assets until the proper proportions are attained. This strategy has been used successfully by Towers Perrin (Mulvey 1995) and other long term investment managers. See Perold and Sharp (1988) for a detailed description of the fixed-mix and other dynamic strategies for asset allocation.

In this paper, we show that Tabu Search can be combined with Variable Scaling to solve important cases of the asset allocation problem. These decisions are difficult due to nonconvexities in the objective function. Also, the inclusion of real world considerations, such as taxes and other transaction costs (Mulvey 1993), may cause difficulties for continuous optimization algorithms. In contrast, Tabu Search takes advantage of the discretization of the solution space.

2. The Fixed-Mix Dynamic Control Problem

In this section we provide the mathematical description of the fixed-mix control problem, a dynamic stochastic control model. Other reallocation rules could be employed, such as the life cycle concept; these lead to similar nonconvex optimization problems.

To state the model, we define the following sets:

a set of discrete time steps in which the portfolio will rebalanced:

$t = \{1,2,\ldots,T\}$;

a set of asset classes:

$i = \{1,2,\ldots,I\}$;

a set of scenarios, each of which describes the full economic situation for each asset category i, in each time period t:

$s = \{1,2,\ldots,S\}$.

Define the following decision variables:

$x_{i,t}^{s}$: amount of money (in dollars) invested in asset category i, in time period t, for scenario s;

w_t^s: wealth (in dollars) at the beginning of time period t under scenario s;

λ_i: fraction of the wealth invested in asset category i (constant across time), $0 \le \lambda_i \le 1,\ i = 1,2,\ldots,I$ *and* $\sum_{i=1}^{I} \lambda_i = 1$ (not allowing short sales).

Define the following parameters:

w_1: initial wealth in the beginning of time period 1 (in dollars);

$r_{i,t}^{s}$: percentage return for asset i, in time period t, under scenario s;

$\bar{r}_t^s$: average return in time period t under scenario s,

$$\bar{r}_t^s = \frac{\sum_{i=1}^{I} (x_{i,t}^s \cdot r_{i,t}^s)}{\sum_{i=1}^{I} x_{i,t}^s};$$

p^s: probability that scenario s occurs, $\sum_{s=1}^{S} p^s = 1$;

t_i: percentage transaction cost for asset class i -- to simplify the presentation, symmetric transaction costs are assumed (i.e. cost of selling equals cost of buying); the implemented model can also handle nonsymmetric transaction costs.

The initial asset allocation will be:

$$x_{i,1}^s = w_1 \lambda_i . \tag{1}$$

The fixed-mix control rule ensures that a fixed percentage of the wealth is invested in each asset category. Wealth at the beginning of the second period will be:

$$w_2^s = \sum_{i=1}^{I} (1 + r_{i,1}^s) x_{i,1}^s - \sum_{i=1}^{I} \left| r_{i,1}^s - \bar{r}_1^s \right| x_{i,1}^s t_i . \tag{2}$$

The second term of (2) takes the transaction cost into account assuming linear transaction costs. To compute these values, the average return, $\bar{r}_t^s$, is calculated for each time period t and for each scenario s. A portion of the asset classes with returns higher than the average return is sold, while the asset classes with below average returns are bought.

The equation for rebalancing at the beginning of period t is

$$x_{i,t}^s = w_t^s \lambda_i \quad \forall i \text{ and } s, \text{ and } t = \{2, 3, .., T\}, \tag{3}$$

while the equation for wealth at the beginning of period t+1 is

$$w_{t+1}^s = \sum_{i=1}^{I} (1 + r_{i,t}^s) x_{i,t}^s - \sum_{i=1}^{I} \left| r_{i,t}^s - \bar{r}_t^s \right| x_{i,t}^s t_i \quad \forall\ s \text{ and } t = \{2, 3, .. T\}. \tag{4}$$

Typically the model includes linear constraints on the asset mix which we for now assume constant over time:

$$A\lambda - b \leq 0. \tag{5}$$

These constraints can take any linear form, for instance upper and lower bounds on the proportions. A typical example will be for investors to limit the international exposure to some value (say 30%).

The objective function depends on the investor's risk attitude. A multiperiod extension of the Markowitz's mean-variance model is applied for this analysis. Two terms are needed -- the average total wealth, Mean(w_T), and the variance of the total wealth, Var(w_T), across the scenarios at the end of the planning horizon (at the end of time period T). Other objective functions can also be used. For example, we can maximize the von Neumann-Morgenstern expected utility of the wealth at time T (Keeney and Raiffa 1993). Alternatively, we can maximize a discounted utility function (Ziemba and Vickson 1975). Other objective function forms, such as multiattribute utility functions, can also be used to model multiple investor goals (Keeney and Raiffa 1993).

Our financial planning model is thus

$$\text{Max} \quad Z = \beta\, Mean(w_T) - (1-\beta) Var(w_T) \tag{6}$$

$$\text{s.t.} \quad Mean(w_T) = \sum_{s=1}^{S} p^s w_T^s \,, \tag{7}$$

$$Var(w_T) = \frac{1}{S} \sum_{s=1}^{S} [w_T^s - Mean(w_T)]^2 \,, \tag{8}$$

equations (1) - (5), and $0 \leq \beta \leq 1$.

Mean(w_T) measures the expected profit from the investment strategy, while Var(w_T) measures the risk of the strategy. A tradeoff exists between the expected profit and the risk of an investment strategy. By solving the problem while allowing β to vary between 0 to 1 we obtain the multiperiod efficient frontier. The quadratic equality constraint (3) causes the fixed-mix model to be nonconvex.

3. Specializations to Tabu Search

An advantage in solving the fixed-mix control problem with Tabu Search is that each application of Variable Scaling effectively discretizes the solution space. We thus define the solution space as a discrete set of possible investment proportions in each asset category. Let us define the following vectors:

$\lambda(i)$: proportion of the total investment in asset category i;

$l(i)$: lower bound on the investment proportion in asset category i;

$u(i)$: upper bound on the investment proportion in asset category i.

Restricting the investment proportions in each asset category to be an integer percentage share of the total investment means that we allow $\lambda(i)$ to take the following percentage values:

$$l(i),\ l(i)+1,\ l(i)+2,\ \ldots,\ u(i)-1,\ u(i)\ ,$$

where $0 < l(i) < u(i) < 100$ (not allowing short sale) and $u(i)$ and $l(i)$ are integer.

Tabu Search operates under the assumption that a neighborhood can be constructed to identify adjacent solutions. The fixed-mix problem is constrained such that the sum of all the investment proportions must equal 100%. We define a neighbor solution by choosing two variables, $\lambda(up)$ and $\lambda(down)$ and assigning them new values:

$$\lambda(up) = \lambda(up) + delta, \text{ and}$$

$$\lambda(down) = \lambda\ (down) - delta,$$

such that $\lambda(up) < u(up)$ and $\lambda\ (down) > l(down)$, where *up* is the index of the variable increasing its value, *down* is the index of the variable decreasing its value, and $delta > 0$ is the stepsize for the neighborhood.

For a given discrete solution space, the following neighborhood search algorithm (Glover and Laguna 1993) is an attempt to solve the fixed-mix control problem. This approach gives the global optimal solution (in the discrete solution space) if the problem is convex. However, if the problem is nonconvex, as is the case with the fixed mix control problem, the neighborhood search algorithm will terminate at the first (discrete) local optimal solution. The area of the solution space that has been searched will be limited using this approach.

Algorithm: SIMPLE LOCAL SEARCH

Step 1: Initialization.
    Select a feasible starting solution, λ_*start*.
    Set *current_solution* = λ_*start*.
    Calculate objective value of this solution, $f(\lambda$_*start*$)$, and set
        best_objective = $f(\lambda$_*start*$)$.
Step 2: Stopping criterion.
    Calculate the objective value of the neighbors of *current_solution*.
    If no neighbors have a better objective value than *best_objective*, stop.
    Otherwise, go to Step 3.
Step 3: Update.
    Select a neighbor, λ_*next*, with the best objective value, $f(\lambda$_*next*$)$ (or select any neighbor that gives an improving objective value).
    Update: *current_solution* = λ_*next*, *best_objective* = $f(\lambda$_*next*$)$
    Go to Step 2.

Due to nonconvexities, we need a more intelligent method for finding a global solution. Tabu Search extracts useful information from previous moves to direct the search to productive areas of the solution space. It uses specialized memory structures to characterize moves with certain attributes as tabu, and assigns other moves values or penalties in order to guide the search through the solution space without becoming trapped at local optima.

We define the total neighborhood of the current solution, *N(current_solution)*, as all possible ways of increasing the amount invested in one asset category and decreasing the amount invested in another asset category, while satisfying the constraints. This neighborhood is modified to *N(H, current_solution)*, where *H* represents the history of the moves in the search process. Therefore, the search history determines which solutions might be reached from the current point. In short term Tabu Search strategies, *N*(*H*, *current_solution*) is typically a subset of *N*(*current_solution*). In intermediate and longer term strategies *N*(*H*, *current_solution*) may contain solutions not in *N*(*current_solution*). See Glover (1995) for further details.

3.1 The Memory Structures

Tabu Search makes use of adaptive (recency based and frequency based) memory structures to guide the search. The recency and the frequency memory structures each consist of two parts. The first part of the recency based memory keeps track of the most recent iteration at which each variable received each of the values assigned to it in the past (*tabu_time*(*i*, *value*)). The second part of the recency based memory keeps track of the most recent iteration at which each variable changed its value (*tabu_last_time*(*i*)).

The first part of the frequency based memory keeps track of the number of times each variable has been assigned its specific values (*tabu_count*(*i*, *value*)). The second part measures the number of iterations the variables have resided at the assigned values (*tabu_res*(*i*, *value*)). Let *duration*(*i*) measure the number of iterations since the variable received its current value.

3.2 Move Attributes

To make use of these memory structures, Tabu Search uses the concept of move attributes. Generally a move $\lambda\_next(up) = \lambda(\text{up}) + delta$ and $\lambda\_next(down) = \lambda(down) - delta$ has four attributes:

(1) $\lambda(up)$: value of increasing variable, prior to increase;
(2) $\lambda\_next(up)$: value of increasing variable, after increase;
(3) $\lambda(down)$: value of decreasing variable, prior to decrease;
(4) $\lambda\_next(down)$: value of decreasing variable, after decrease.

If a move attribute is tabu, this attribute is said to be tabu-active. A move is classified tabu as a function of the tabu status of its attributes. In the next section we show what combinations of tabu-active move attributes classifies a move tabu.

3.3 Tabu Rules Based on Recency Memory Structure

An important parameter of this process is *tabu_tenure*, that is the number of iterations in which a move attribute is tabu-active. For our approach, *tabu_tenure* will have two values, *t_from* and *t_to*, where *t_from* is the number of iterations an assignment λ_i = *value* is tabu-active, to discourage moving λ_i "from" its present value (*now_value*), and *t_to* is the number of iterations an assignment λ_i = *value* is tabu-active, to discourage moving λ_i "to" a specific value (*next_value*) (Glover and Laguna 1993).

An attribute (*i*, *value*) is declared tabu-active when

tabu_time(*i*, *value*) ≠ 0, *tabu_time*(*i*, *value*) ≥ *current_iteration* - *t*,

where *t* = *t_to* if *value* = *next_value*. (This classification will discourage a variable from returning to a value it recently has moved away from.) Also, an attribute *i* (implicitly, an attribute (*i*, *value*) for *value* = *now_value*) is declared tabu-active when

tabu_last_time(*i*) ≠ 0, *tabu_last_time*(*i*) >= *current_iteration* - *t*,

where *t* = *t_from*. (This classification will discourage moving a variable from its present value if it recently received this value.)

Both dynamic and static rules can be applied to set *tabu_tenure*. For our purposes, we have used simple static rules, such as setting *t_from* to a constant value between 1 and 3 and *t_to* to a constant value between 4 and 6. Our choice of values for *t_to* and *t_from* is based on preliminary experimentation. We choose *t_to* $\cong 2\cdot\sqrt{n}$ as a rule of thumb in deciding the size of this parameter (in our test cases n = 8).

In the implementation we introduce one other kind of tabu-active classification: an attribute (*i, value*) is "strongly from tabu-active" if

$$tabu\_last\_time(i, value) >= current\_iteration - 1.$$

For this study, the basic rule for defining a move tabu is if one of the following is true:

i) either attribute (1) or attribute (3) is strongly tabu-active;

ii) two or more of the remaining attributes are tabu-active.

3.4 Tabu Rules Based on Frequency Memory Structure

The frequency memory structure is usually applied by assigning penalties and inducements to the choice of particular moves. Let *S* denote the sequence of solutions generated by the search process up to the present point and define two subsets, *S1* and *S2*, which contain high quality and low quality solutions (in terms of objective function values), respectively. A high residence frequency in the subsequence *S1* indicates that an attribute is frequently a participant in high quality solutions, while a high residence frequency in the subsequence *S2* indicates that an attribute is frequently a participant in low quality solutions. These relationships guide the search in the direction of high quality solutions, by increasing or decreasing the incentive to choose particular moves based on the quality of past solutions that contain attributes provided by these moves. This constitutes an instance of a Tabu Search intensification process. Define another set, *S3*, containing both high and low quality solutions. Assigning a penalty to high frequency attributes in this set pushes the search into new regions, creating a diversification process. A high transition frequency might indicate that an attribute is a "crack filler", which is an attribute that alternates in and out of the solution to "perform a fine tuning function" (Glover and Laguna 1993).

4. Solution Strategy

Tabu Search methods often do not "turn on" their memory or restrictions until after reaching a first local optimum. In our solution strategy we take advantage of an efficient method to find the local optimum namely a Variable Scaling approach. Our Variable Scaling procedure is an instance of a candidate list method, which is the name given to a class of strategies used in Tabu Search (by extension of related procedures sometimes used in optimization) to generate subsets of good moves without the effort of examining all moves possible. In the present setting, where the number of values available to each variable is effectively infinite, we elect a candidate list strategy that "scales" the variables to receive only a relatively small number of discrete values. However, we allow the scaling to be variable, permitting the scaling interval to change each time a local optimum is reached in the restricted part of the neighborhood defined by the current scaling. When no change of scaling (from the options specified) discloses an improving move -- so that the current solution is truly a local optimum relative to the scalings considered -- we apply a recency based Tabu Search approach to drive the solution away from its current location, and then again seek a local optimum by our Variable Scaling approach. The Variable Scaling approach is outlined in section 4.1, while the complete strategy in outlined in section 4.2.

4.1 Variable Scaling Approach

We define a set of stepsizes (or scaling intervals), each of which gives rise to a restricted neighborhood. A local neighborhood search is done over a given restricted neighborhood until no improvement is possible. When local optimality is reached in that neighborhood (for one particular stepsize), we switch to another restricted neighborhood via a new stepsize. We choose a set of stepsizes in decreasing order. (For our purposes, typically the biggest stepsize is 5% and the smallest stepsize is 1%.) The smallest stepsize determines the accuracy of identifying a solution as a local optimum.

For a convex problem, applying such a candidate list approach based on Variable Scaling speeds up the search since we move in bigger steps toward the optimal solution in the beginning of the search, and only as we get "close to the top" (given we are maximizing) do we decrease the stepsize. For a nonconvex problem (like ours) with several local optima, the approach can reduce the number of necessary iterations to get to a local optimum, and it can also move the search away from a local optimum. If one stepsize is "stuck" in a local solution, a change in the stepsize can take us away from the local optimum. All of these ideas are incorporated in *Algorithm*: VARIABLE SCALING.

Algorithm: VARIABLE SCALING

Step 0: Initialize.
  Construct a stepsize list: *stepsize_1*, *stepsize_2*, .. , *stepsize_NS*, where *NS* is the number of stepsizes. Choose stepsizes in decreasing order.
  Set *stepsize* = *stepsize_1*.
  Construct an initial solution, λ*_start*. Let *current_solution* = λ*_start*.
  Set *current_iteration* = 0.
  Let *best_objective* = *f*(λ*_start*), the objective value of λ*_start*.
Step 1: Calculate neighbors.
Step 2: Evaluate neighbors.
  If objective value of at least 1 of the neighbors is better than *best_objective*:
    Pick the neighbor with the best objective value, λ*_next*;
    Update *current_solution* = λ*_next* and *best_objective* = *f*(λ*_next*);
    Set *current_iteration* = *current_iteration* + 1;
    Go to Step 1.
  If none of the neighbors improves the solution:
    If *stepsize* ≠ *stepsize_NS*, use next stepsize from list and go to Step 1;
    If *stepsize* = *stepsize_NS*, check if objective has improved for any of the last *NS* stepsizes: If the objective has improved, set *stepsize* to *stepsize_1*, and go to Step 1. If it has not improved, go to Step 3.
Step 3. STOP: Local optimal solution is obtained.

4.2 The Complete Algorithm

The implemented algorithm is:

Algorithm: COMPLETE ALGORITHM

Step 0, Step 1 and Step 2 from the Variable Scaling approach.
Step 3: Incorporate Recency Based Tabu Search (diversifying search).
  3.1: Set *stepsize* = *stepsize_div* (*stepsize* used for the diversifying search).
    Set *div_it_counter* = 0.
  3.2: Set *div_it_counter* = *div_it_counter* + 1.
    For all neighborhood moves do
      check if move is tabu.
      if tabu, go to next neighbor;
      if not tabu, calculate the objective value of the move.
  3.3: Pick the (nontabu) neighbor with the best objective value. Denote this neighbor λ*_best_div*, and its objective value $f(\lambda$*_best_div*$)$.
    If $f(\lambda$*_best_div*$)$ > *best_objective* set *stepsize* = *stepsize_1* and go to Step 1 in the Variable Scaling approach.
    If $f(\lambda$*_best_div*$)$ < *best_objective* go to Step 3.4.
  3.4: If *div_it_counter* < *max_div_it*, update tabu status (as done in Steps 1-3 in memory updating in section 3.1) and go to Step 3.2.
    If *div_it_counter* = *max_it_counter*, go to Step 4.
Step 4: STOP. The solution is equal or sufficiently close to the true (discrete) global optimal solution.

The procedure imposes the Tabu Search approach in conjunction with the candidate list strategy of Variable Scaling. This is just one of many ways that Tabu Search can be coordinated with a candidate list strategy. (More typical is to subject such a strategy directly to the guidance of Tabu Search memory, rather than invoking this guidance only at particular stages and for particular neighborhood instances.) Nevertheless, we have found this approach convenient for our present purposes. Specifically, in this alternating approach, we switch from the Variable Scaling method (over its chosen set of scalings) to the Tabu Search method (over another set of scalings) whenever Variable Scaling no longer improves the current solution. If Tabu Search generates a solution better than the one obtained from the Variable Scaling, we return to Variable Scaling, with the solution from Tabu Search as an initial solution. If Tabu Search fails to generate an improved solution, the algorithm will stop when the maximum number of iterations is reached. In this procedure the choice of the stepsizes is crucial for the algorithm's success. We typically choose small stepsizes in the range of 0.5% to 5% for the Variable Scaling, while a bigger one, typically 5% to 15% is applied for the Tabu Search.

5. Computational Results

The algorithm described in section 4.2 is applied to an investment problem with I = 8 asset categories, T = 20 time periods and S = 100 scenarios. The 8 different asset categories are cash equivalents, Treasury bonds, large capitalization US stocks, international bonds, international stocks, real estate, government/corporate bond index, and small capitalization US stocks. One hundred scenarios were generated by the technique introduced in Mulvey (1995). Each scenario is given equal probability $p_s = 1/S = 1\%$. Each scenario consists of returns for each asset, in each time period. Hence the total number of returns generated is equal to 16000. The initial wealth is set equal to unity.

In all the experiments, each point on the multiperiod efficient frontier is obtained as explained in section 4.2. The entire efficient frontier is obtained by solving the problem for 22 values of β. Through all the experiments, we have a set of basic test parameters[1].

5.1 Comparison with Global Method

Our solutions are compared with the solutions obtained from the global method described in Androulakis *et al.* (1994). This deterministic global optimization algorithm guarantees finite ε-convergence to the global optimum. In this case we assume no transaction cost or tax. The solutions obtained by our approach are very close to the solutions obtained by the global method and are often slightly better. The opti-

---

[1] Number of stepsizes for the Variable Scaling approach (*NS*): 3; Variable Scaling stepsizes *stepsize_1* = 5%, *stepsize_2* = 3%, *stepsize_3* = 1%; stepsize for the Tabu Search part of the algorithm (*stepsize_div*): 10%; maximum number of diversifying steps (*max_div_it*): 25; tabu tenures: *t_from*= 2, *t_to* = 4.

mal mixes generated by the two methods are indeed practically identical. These results are obtained despite the presence of nonconvexities.

5.2 Including Transaction Costs and Tax

We assumed percentage transaction costs for the asset categories[2]. The tax rate is assumed to be 28%. Figure 1 shows the efficient frontiers obtained by solving the model that takes tax and transaction cost in to account (tax-model) and the one that does not (no-tax model). As expected, the efficient frontier obtained by the tax-model is tilted down. More surprisingly, perhaps, the efficient frontiers cross in the low end of the variance area (for low β's). Transaction costs and taxes dampen the variance of the expected value and therefore a lower variance can be achieved for the tax-model. From Figure 2 we see that for $\beta > 0.94$ in the model where transaction costs and tax is included, it is optimal to have all the funds in the asset category with the highest expected value. This is in contrast to the case without taxes or transaction costs, where a more diversified optimal solution was found as soon as the variance was assigned a weight in the objective function.

There is an explanation for this phenomenon. When all of the funds are concentrated in one asset category, no buying or selling is necessary to update the portfolio at the end of each time period. When the funds are split between asset categories, however, trading must be done at the end of each time period to reconstruct the portfolio to the predetermined weights of the fixed-mix rule. This implies cash outflows because of the transaction costs and taxes on assets sold for profit. So for $\beta > 0.94$ the gain from reduced variance by diversifying is offset by a larger loss in the expected value.

---

[2] Cash equivalents: 0%; Treasury bonds: 0.25%; large capitalization stocks: 0.25%; international bonds: 0.5%; international stocks: 1.0%; real estate: 6.0%; government/corporate bond index: 1.0%; small capitalization stocks: 1.0%.

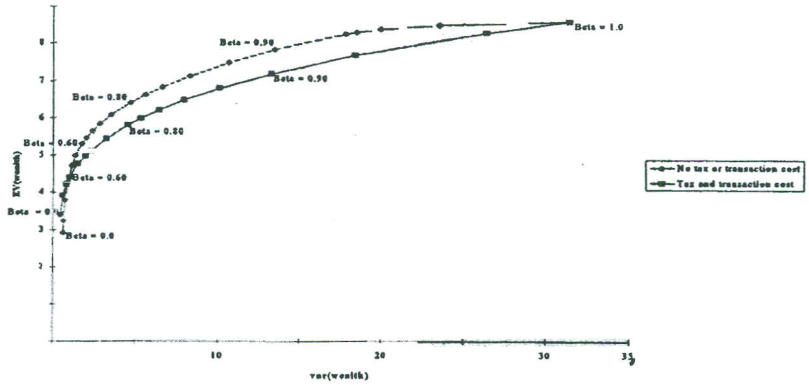

Figure 1: Efficient frontiers for model with and without tax and transaction costs. Notice how the efficient frontier for the tax-model is tilted down (as one would expect). Also see how the efficient frontiers crosses in the low variance end.

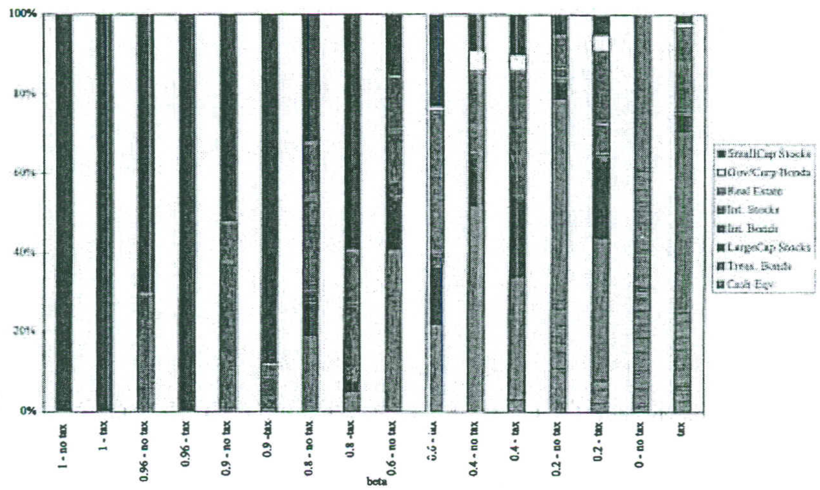

Figure 2: Optimal solutions for model with and without tax and transaction costs. Notice how the solutions obtained from the tax-model are less diversified for high β's (i.e. in the high variance/high expected value end of the efficient frontier), and more diversified for low β's.

At the other end of the efficient frontier (low β, i.e. low variance), we see that tax-model recommends solutions that are more diversified than the solutions from the no-tax model. Tax-model recommends a portfolios consisting of all 8 asset categories for $\beta \leq 0.4$, with no dominating asset category (except for $\beta = 0$). The no-tax model, however, recommends a portfolio concentrated in fewer asset categories There is a simple explanation for this result. With many asset classes, the return in some of asset classes will be close to the average return. Little trading is required in these asset categories to rebalance the portfolio to the prefixed weights. More trading is done in the asset categories with returns far from average, but these asset categories are a fraction of our total portfolio. This is not the case for a portfolio concentrated in fewer asset categories. Consider, for instance, the portfolio recommended by the no-tax model for $\beta = 0.2$ -- approximately 80% in cash equivalents and Treasury bonds. Since it is unlikely that both dominating asset categories have close to average returns, more trading is needed to rebalance the portfolio. This portfolio is likely to have more cash outflows from transaction costs and taxes than the more diversified one -- the gain in reduced variance by concentrating the portfolio in cash equivalents and Treasury bonds is offset by a loss in expected value caused by the increase in cash outflows.

As expected, the solution time increases considerably when transaction costs and taxes are included to the model. The average solution time per β-problem for the model with basic test parameters (see beginning of this section) is 46.7 CPU seconds (just over 17 CPU minutes to obtain the entire efficient frontier) on a Silicon Graphics Iris workstation. The results are highly attractive, particularly given that we are developing a long term projecting system.

5.3 The Effect of Including Tabu Search Restrictions.

To see the effect of the Tabu Search restrictions we consider the tax problem and compare the solutions of our approach (COMPLETE ALGORITHM) with the solutions of a pure Variable Scaling approach (the candidate list strategy used in our method). The largest stepsize in the pure Variable Scaling approach is set equal to the stepsize of the Tabu Search part of our algorithm (*stepsize_div* = 10%). The other stepsizes are also set equal for both approaches (5%, 3% and 1%). Hence, the sole difference between the two approaches is the recency memory restrictions on the 10% search applied in our algorithm. For $\beta > 0.5$ the solutions are identical. The pure Variable Scaling approach "gets stuck" in a local solution for $\beta < 0.4$, in contrast to the approach that includes Tabu Search memory guidance.

6. Conclusions

Tabu Search is an efficient method for obtaining the efficient frontier for the fixed-mix investment problem. The computational results show that the solutions obtained by the Tabu Search are very close to (often slightly better than) the ε-tolerance global optimal solutions obtained by the method of Androulakis *et al.* (1994) for the case with no taxes or transaction costs. In an expanded model, which addresses transaction costs and taxes for greater realism, our approach continues to obtain optimal solutions efficiently. The expanded model is beyond the capability of the global optimization approach, and its complicating features in general pose significant difficulties to global optimization solvers based on currently standard designs.

Some areas for future research are: (1) develop and test related dynamic stochastic control strategies (with nonconvexities); (2) design an approximation scheme for updating information between iterations in order to improve computational efficiencies; and (3) incorporate additional strategic elements of Tabu Search. Since the discretization of the solution space depends upon an investor's circumstances, research in this area is critical for successful use of this methodology.

7. References

I.P. Androulakis *et al.* Solving stochastic control problems in finance via global optimization, Working paper, SOR-94-01, Princeton University (1994).

M.J. Brennan and E.S. Schwartz, An equilibrium model of bond pricing and a test of market efficiency", *Journal of Financial and Quantitative Analysis*, 17 (1982) 75.

F. Glover, Tabu search: fundamentals and usage, Working paper, University of Colorado, Boulder (1995).

F. Glover and M. Laguna, Tabu search, in: *Heuristic Techniques for Combinatorial Problems*, ed. C. Reeves (1992).

J.C. Hull, *Options, Futures and other Derivative Securities*, (Prentice Hall, 1993).

J.E. Ingersoll, Jr. *Theory of Financial Decision Making*, (Rowman & Littlefield, 1987).

R. Jarrow, Pricing interest rate options, in: *Finance*, eds: R. Jarrow, V. Madsimovic, and W.T. Ziemba, (North Holland, Amsterdam, 1995).

R. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, (John Wiley, New York, 1976; reprinted by Cambridge University Press, 1993).

J.M. Mulvey, Incorporating transaction costs in models for asset allocation, in: *Financial Optimization*, ed: S. Zenios, (Cambridge University Press, 1993).

J.M. Mulvey, Generating scenarios for the Towers Perrin investment system, Working paper, SOR 95-04, Princeton University (to appear *Interfaces* 1995).

A.F. Perold and W.F. Sharpe, Dynamic strategies for asset allocation, *Financial Analysts Journal*, (1988) 16.

W.T. Ziemba and R.G. Vickson (eds.), *Stochastic Optimization Models in Finance*, (Academic Press, New York, 1975).