

Journal of Economic Dynamics and Control 21 (1997) 1405-1425

# Solving long-term financial planning problems via global optimization

C.D. Maranas, <sup>a</sup> I.P. Androulakis, <sup>a</sup> C.A. Floudas, <sup>a,\*</sup> A.J. Berger, <sup>b</sup> J.M. Mulvey <sup>b</sup>

<sup>a</sup> Department of Chemical Engineering, Princeton University, Princeton, NJ 08544-5263, USA <sup>b</sup> Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ 08544, USA

### Abstract

A significant multi-stage financial planning problem is posed as a stochastic program with decision rules. The decision rule – called dynamically balanced – requires the purchase and sale of assets at each time stage so as to keep constant asset proportions in the portfolio composition. It leads to a nonconvex objective function. We show that the rule performs well as compared with other dynamic investment strategies. We specialize a global optimization algorithm for this problem class – guaranteeing finite  $\varepsilon$ -optimal convergence. Computational results demonstrate the procedure's efficiency on a real-world financial planning problem. The tests confirm that local optimizers are prone to erroneously underestimate the efficient frontier. The concepts can be readily extended for other classes of long-term investment strategies.

Keywords: Global optimization algorithm; Financial planning problems; Fixed-Mix problem

### 1. Introduction

This paper addresses a significant problem in planning under uncertainty: the allocation of financial assets to broad investment categories over a long-run horizon (10 to 20 years). The long-term asset allocation problem plays a critical role in successful investing. It has been the subject of research over many years. For example, see Berger and Mulvey (1996), Dantzig and Infanger (1994), Davis and Norman (1990), Grauer and Hakansson (1985), and Merton (1971). In many of these studies, the problem is posed as a multi-stage stochastic program and

<sup>\*</sup> Corresponding author.

solved using efficient solution algorithms. Unfortunately, traditional multi-stage stochastic programs are difficult to solve. Not only does the problem size grow exponentially as a function of the number of stages and random variables, but the precision of the recommendations is difficult to measure since it is costly to perform out of sample experiments.

We investigate an alternative approach. The basic idea is to set up the investment problem as a stochastic program with a popular decision rule. Time is discretized into n-stages across the planning horizon. Investment decisions are made at the beginning of each time period. The optimal parameters for the decision rule are then determined by means of a global optimization algorithm. Although the model results in a nonconvex optimization problem, we have developed a specialized global solver to handle this problem class. The approach extends the ideas of Maranas and Floudas (1993, 1994), and Floudas and Visweswaran (1990, 1993).

Due to nonconvexities, multiple local solutions exist which render the location of the global one a very difficult task. Traditional local algorithms can only guarantee convergence to a local solution at best, thus failing sometimes to locate the optimal recommendation. Since the proposed algorithm finds the global solution, the best tradeoff between risk and expected profit can be established for the multi-period investment problem. As a consequence, stochastic programming with decision rules (SPDR) provides a viable alternative to multi-stage stochastic programming (MSP) (see, e.g. Garstka and Wets, 1974; Mulvey, 1996). Whereas the MSP may lead to outstanding decisions under uncertainty, there are several impediments to be overcome. First the size of the MSP must be considered. The number of decision variables grows exponentially as a function of the number of time stages and random coefficients. The stochastic program represents conditional decisions for each branch of the scenario tree. Efficient algorithms exist for solving these problems, for instance, see Dantzig and Infanger (1994), Dempster and Thompson (1995), Mulvey and Ruszczynski (1992), Rockafellar and Wets (1991), and Zenios and Yang (1996). Regardless, the resulting optimization problem can be long running for most multi-stage examples.

An important consideration for MSP and SPDB models involves the generation of scenarios. This area has received attention by researchers who have applied importance sampling and other variance reduction methods (e.g. Dantzig and Infanger, 1994). Others have developed probabilistic algorithms, e.g. Higle and Sen (1991) for stochastic programs with fixed recourse. Also see Mulvey (1996), who designed a scenario generation system for the consulting company Towers Perrin. Because of the need to perform sampling when generating scenarios, it is critical that the precision of the recommendations be ascertained. The SPRB approach has a distinct advantage in this regard since it can be easily tested with out a sample data – thereby providing confidence that the recommendations are robust. A similar test is much harder in the case of MSP. For an example, see Worzel et al. (1994), who evaluated multi–stage stochastic programs in the context of a rolling–horizon simulation. The decision rule-based model SPDR may appear more intuitive at least to some investors, perhaps since the concepts are closer to the way that individuals handle their own affairs. Unfortunately, the use of SPDR model hinges on discovering policies that are intuitive and that will produce superior results. The modelers must thoroughly understand the application in order to find an appropriate set of decision rules. This step can be hard to carry out. Also, decision rules may lead to nonconvexities and highly nonlinear functions. The nonconvexity barrier may become less constraining, however, as algorithms such as the one proposed in this report becomes available.

#### 2. The dynamically balanced strategy

In this section, we give a mathematical description of the decision rule we consider, namely the dynamically balanced (DB) investment policy. We also show through empirical test on historical data that the DB decision rule compares favorably to a stochastic optimal control approach and another decision rule. Also, the DB rule and its variants are based on considerable theoretical results (e.g. Davis and Norman, 1990; Merton 1971; Taksar et al., (1988).

#### 2.1. Model definition

First, a set of discrete time steps

 $t = \{1, 2, \dots, T\}$ 

in which the portfolio will be rebalanced is selected. We are interested in long planning horizons -10 to 20+ years. Second, a number of *assets* 

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

where I is in most cases between 3 and 15, is selected where the initial wealth will be distributed at each stage during the planning horizon. A given set of *scenarios* 

 $s = \{1, 2, \dots, S\}$ 

is generated based on the method presented in Mulvey (1996), depicting plausible outcomes for the uncertainties across the planning horizon. Five hundred to one thousand scenarios are typically required for adequately capturing the uncertain future trends in realistic portfolio problems.

The following parameters specify the DB model.

 $w_0$ : initial wealth (in dollars)  $r_{i,t}^s$ : return (in %) for asset *i*, in time period *t*, under scenario *s*  $p^s$ : probability of occurrence for scenario *s*, (note that  $\sum_{s=1}^{s} p^s = 1$ ). The decision variables are defined as follows:

 $w_t^s$ : wealth (in dollars) at time period t, and according to scenario s.

 $\lambda_i$ : fraction of wealth invested in asset category *i* (constant across time). Note that  $0 \leq \lambda_i \leq 1$ ,  $\forall i = 1, ..., I$  and  $\sum_{i=1}^{I} \lambda_i = 1$ . At the beginning of each period, the investor rebalances his portfolio. The rule ensures that a fixed percentage of one's wealth is invested in each asset category – at least at the beginning of each period.

Wealth  $w_1^s$  at the end of the first period will be

$$w_1^s = w_0 \sum_{i=1}^{I} (1 + r_{i,1}^s) \lambda_i, \quad s = 1, 2, \dots, S.$$
 (1)

Accordingly, the relation in wealth for scenario s between any two consecutive periods t - 1 and t is given by

$$w_t^s = w_{t-1}^s \sum_{i=1}^I (1+r_{i,t}^s) \lambda_i, \quad t = 1, 2, \dots, T, \ s = 1, 2, \dots, S.$$
 (2)

By utilizing (1) and (2), and assuming no transaction costs, we express the wealth  $w_T^s$  for scenario s at the end of the last period T as a function of only  $\lambda_i$ , i = 1, ..., I after eliminating all  $w_i^s$ .

$$w_T^s = w_0 \prod_{t=1}^T \left[ \sum_{i=1}^I \left( 1 + r_{i,t}^s \right) \lambda_i \right], \quad s = 1, \dots, S$$
(3)

By considering all scenarios, the expected wealth  $Mean(w_T)$  at the end of the last period T is

$$Mean(w_T) = \sum_{s=1}^{S} p^s w_T^s = w_0 \sum_{s=1}^{S} p^s \left\{ \prod_{i=1}^{T} \left[ \sum_{i=1}^{I} \left( 1 + r_{i,i}^s \right) \lambda_i \right] \right\}$$
(4)

The goal of the planning system is a *multi-period* extension of Markowitz's mean-variance model. Two competing terms in the objective function are needed: (1) the average total wealth  $Mean(w_T)$  and (2) the variance  $Var(w_T)$  across all scenarios s of the total expected wealth at the end of the planning horizon T. More specifically,

$$U(w_T) = \beta \operatorname{Mean}(w_T) - (1 - \beta) \operatorname{Var}(w_T),$$

where

$$Mean(w_T) = \sum_{s=1}^{S} p^s w_T^s,$$
(5)

$$\operatorname{Var}(w_T) = \sum_{s=1}^{S} p^s \left[ w_T^s - \operatorname{Mean}(w_T) \right]^2$$
(6)

and

 $0 \leq \beta \leq 1.$ 

Mean $(w_T)$  measures expected profitability of the particular investment strategy and Var $(w_T)$  represents the inherent risk. Clearly, Mean $(w_T)$  is desired to be as large as possible whereas Var $(w_T)$  to be as small as possible. Parameter  $\beta$ provides multiple weightings for the competing objectives. By varying  $\beta$  from zero to one, the multi-period *efficient frontier* of the investment problem can be obtained. The problem of selecting an optimal dynamically balanced decision rule can now be formulated as the following nonlinear optimization model (for each  $\beta$  with  $0 \le \beta \le 1$ ).

$$\max_{\lambda_{i}, w_{i}^{s}} \quad \beta \, Mean(w_{T}) \, - \, (1 - \beta) \, Var(w_{T})$$
s.t.
$$w_{T}^{s} = w_{0} \prod_{t=1}^{T} \left[ \sum_{i=1}^{I} \left( 1 + r_{i,t}^{s} \right) \lambda_{i} \right], \quad s = 1, \dots, S, \quad (7)$$

$$\sum_{i=1}^{i} \lambda_i = 1,$$

$$0 \le \lambda_i \le 1, \quad i = 1, \dots, I.$$
(8)

The set of nonlinear equality constraints (7) along with the variables  $w_T^s$ , s = 1, ..., S, Mean $(w_T)$ , Var $(w_T)$  can be eliminated by substituting relations (7), (5) and (6) into the objective function f. For convenience, the maximization of U is replaced by minimization of -U = f.

$$\begin{split} \min_{\lambda_{i}} \quad f &= -\beta \ w_{0} \sum_{s=1}^{S} p^{s} \left[ \prod_{t=1}^{T} \left( \sum_{i=1}^{I} \left( 1 + r_{i,t}^{s} \right) \lambda_{i} \right) \right] \\ &+ (1 - \beta) \ w_{0}^{2} \left\{ \left[ \sum_{s=1}^{S} p^{s} \left[ \prod_{t=1}^{T} \left( \sum_{i=1}^{I} (1 + r_{i,t}^{s}) \lambda_{i} \right) \right] \right]^{2} \\ &- \sum_{s=1}^{S} p^{s} \left[ \prod_{t=1}^{T} \left( \sum_{i=1}^{I} (1 + r_{i,t}^{s}) \lambda_{i} \right) \right]^{2} \right\} \end{split}$$
(DB)  
s.t. 
$$\begin{aligned} \sum_{i=1}^{I} \lambda_{i} &= 1, \end{split}$$

 $0 \leq \lambda_i \leq 1, \quad i = 1, \dots, I.$ 

This results in an optimization problem involving a single linear equality constraint (8) and a modest number of variables  $\lambda_i$ , i = 1, ..., I. The compensating cost is a more complicated objective function. The function f is a nonconvex multivariable polynomial function in  $\lambda_i$  involving multiple local minima.

To see why function f is nonconvex, let us give a small example. Say, we have two assets – stock and bond. We consider two time periods. In each time period, there is only one possible scenario for the return of each asset. Hence, in this example, I = T = 2, and S = 1. Let us assume that the possible returns of the assets are as follows:  $r_{1,1}^1 = r_{1,2}^1 = a$ , and  $r_{2,1}^1 = r_{2,2}^1 = b$ , for some a, b > 0. Since there is only one scenario, it follows that  $p^1 = 1$ . Suppose that  $\beta = 1$  and  $w_0 = 1$ . Then the objective function of (DB) is

$$f = -\prod_{t=1}^{2} \left[ \sum_{i=1}^{2} (1+r_{i,t}^{1})\lambda_{i} \right]$$
$$= -\left[ (1+a)\lambda_{1} + (1+b)\lambda_{2} \right]^{2}.$$

It is easy to verify that the above f is not a convex function of  $(\lambda_1, \lambda_2)$ . In fact, it is a concave function. This shows that the objective function f in (DB) is not convex even in a very special situation and hence it is not convex in general.

The (DB) model can be readily modified to address other measures of risk aversion across time. For example, we could maximize the expected utility of wealth at the end of the planning horizon; or we could conduct a multi-objective decision analysis. The proposed global optimization algorithm can be extended for these alternative models.

In Section 3, we introduce a global optimization algorithm which guarantees  $\varepsilon$ -converge to the global optimum solution in a finite number of iterations.

### 2.2. Empirical results

In this subsection, we test several popular versions of the dynamically balanced decision rule based on the 10-year historical data (January 1982 – December 1991) of monthly returns in three asset categories: cash, bond and stock. The investment problem involves a monthly decision on what portion to hold on each of the three asset categories. We examine seven variants of the DB rule, namely, mix 1, mix 2, ..., mix 7, as shown in Table 1. These mixes are often proposed in the literature and by investment advisers as benchmarks for comparative studies. We conducted a backtest of the seven specified dynamically balanced decision rules as follows: At the beginning of each month, we examined the portfolio and re-balanced it to the target mixes. Thus, we sold (bought) stock and bought (sold) bonds if the equity market outperformed (underperformed) the bond market. Geometric means of monthly returns ( $\mu$ ), standard deviations ( $\sigma$ ) and turnover ( $\rho$ ) of the seven rules as well as the 100% stock rule computed based on the given data are illustrated in Table 2.

	mixl	mix2	mix3	mix4	mix5	mix6	mix7
Cash	0%	0%	0%	0%	0%	0%	0%
Bond	20%	30%	40%	50%	60%	70%	80%
Stock	80%	70%	60%	50%	40%	30%	20%

Table 1 Composition of the seven mixes tested

Table 2

Means ( $\mu$ ), standard deviations ( $\sigma$ ) and turnovers ( $\rho$ ) of the seven DB rules and the 100% stock rule during 1982–1991 period

DB	100% stock	mixl	mix2	mix3	mix4	mix5	mix6	mix7
μ	1.5%	1.48%	1.46%	1.45%	1.42%	1.40%	1.35%	1.33%
σ	4.7%	4.0%	3.6%	3.3%	3.2%	3.1%	3.0%	2.8%
ρ	0%	1.10%	1.10%	1.08%	1.05%	1.00%	0.93%	0.86%

Table 3

Means ( $\mu$ ) and standard deviations ( $\sigma$ ) of sc1, sc2 and sc3 during 1982–1991 period (from Brennan et al., 1996)

	sc1	sc2	sc3
μ	1.3%	1.5%	1.4%
σ	2.8%	3.1%	3.1%

Using the same data, Brennan et al. (1996) computed monthly mean of return  $(\mu)$  and standard deviation  $(\sigma)$  by a stochastic optimal control approach. They employed three slightly different stochastic optimal control strategies, denoted here as sc1, sc2, and sc3, and obtained three pairs of  $\mu$  and  $\sigma$ , as exhibited in Table 3. Comparing our result (Table 2) to the result of the stochastic optimal control (Table 3), as shown in Fig. 1, we can see that (1) rule mix7 has the same standard deviation as strategy sc1 but with higher return; (2) rule mix5 has identical mean and standard deviation as strategy sc3; and (3) although rules mix2 and mix3 are dominated by sc2, they have much lower standard deviations than the 100% stock rule while they have quite close means to the 100% stock rule.

It is worth noting that the trading turnover involved in the BLS strategies are much higher than the dynamically balanced decision rules, as can be easily observed by the fact that their asset proportions vary greatly with time (see figures illustrated in their paper) while the turnover of our rules are very low – less than 1.2% on average, as exhibited in Table 2. This indicates that in the presence of transaction costs, their strategies will pay much higher amount of commission fee than ours. Thus, the geometric returns of strategy sc2 would be considerably reduced once the transaction costs are included in the results.



Fig. 1. Dynamically balanced vs. stochastic optimal control.

Table 4 Means  $(\mu)$  and standard deviations  $(\sigma)$  of the seven B&H rules

B&H	b&h1	b&h2	b&h3	b&h4	b&h5	b&h6	b&h7
μ	1.46%	1.44%	1.43%	1.39%	1.38%	1.33%	1.31%
σ	4.0%	3.65%	3.35%	3.24%	3.14%	3.01%	2.8%

We also compare the DB decision rule with the standard Buy and Hold (B&H) decision rule. Using the same initial mixes: mix1, mix2, ..., mix7, we can have seven corresponding B&H rules, called here b&h1, b&h2, ..., b&h7, respectively. Using the same 10-year data, we calculate the means and standard deviations for the seven B&H decision rules, as illustrated in Table 4. The comparison between the DB decision rules and the B&H decision rules is more clearly exhibited in Fig. 2, where 1 represents 'mix1' and 1' b&h1, and etc. These results clearly indicate that each DB decision rule dominates the corresponding B&H decision rule for the indicated time period.

#### 3. Global optimization algorithm

The solution of optimization problems that possess nonconvexities requires considerable care. A standard approach is to employ a nonlinear programming software system (e.g. CONOPT System, Drud, 1994) in conjunction with a method for restarting the algorithm at widely separated points. Then, the best solution found becomes the recommended investment strategy. Of course, there is no



Fig. 2. Dynamically balanced vs. buy and hold.

assurance that such a solution is the global optimum. Adaptive memory programming (AMP) (also called Tabu Search) provides an alternative approach to the solution of stochastic programs with decision rules. Berger and Mulvey (1996) applied AMP to financial planning for individual investors. Again, there is no guarantee that the optimal solution has been found.

This section describes a third approach to the solution of stochastic program with decision rules. It is the first algorithm for solving nonconvex optimization problems for the dynamically balanced decision rule that provides global optimality guarantees. As such, the method generates bounds on the optimal solution.

The global optimization algorithm is based on a convex lower bounding of the original objective function f and the successive refinement of converging lower and upper bounds by means of a standard branch-and-bound procedure.

#### 3.1. Convex lower bounding and properties

A convex lower bounding function  $\mathscr{L}$  of f can be defined by augmenting f with the addition of a separable convex quadratic function of  $\lambda_i$  as proposed in Maranas and Floudas (1994).

$$\mathscr{L} = f + \alpha \sum_{i=1}^{I} (\lambda_i^{\mathrm{L}} - \lambda_i) (\lambda_i^{\mathrm{U}} - \lambda_i),$$

where

$$\alpha \geq \max\left\{\max_{\substack{i \\ i_i^L \leq \lambda_i \leq i_i^U}} 0, \left(-\frac{1}{2} \operatorname{eig}_i^f\right)\right\}.$$

Note that  $\lambda_i^L$ ,  $\lambda_i^U$  are the lower and upper bounds of  $\lambda_i$  initially set to  $\lambda_i^L = 0$  and  $\lambda_i^U = 1$ . Also,  $\alpha$  is a nonnegative parameter which ensures convexity and must be greater or equal to the negative one-half of the minimum eigenvalue of f over  $\lambda_i^L \leq \lambda_i \leq \lambda_i^U$ . The parameter  $\alpha$  can be estimated either through the solution of an optimization problem or by using the concept of the measure of a matrix (Maranas and Floudas, 1994). The effect of adding the extra term,

$$\alpha \sum_{i=1}^{I} (\lambda_i^{\mathsf{L}} - \lambda_i) (\lambda_i^{\mathsf{U}} - \lambda_i)$$

to f is to make  $\mathscr{L}$  convex by overpowering the nonconvexity characteristics of f with the addition of the term  $2\alpha$  to all of its eigenvalues,

$$\operatorname{eig}_{i}^{\mathscr{L}} = \operatorname{eig}_{i}^{f} + 2\alpha, \quad i = 1, \dots, I$$

Here  $\operatorname{eig}_{i}^{\mathscr{L}}, \operatorname{eig}_{i}^{f}$  are the *i*th eigenvalues of  $\mathscr{L}, f$  respectively. This function  $\mathscr{L}$  defined over the box constraints  $[\lambda_{i}^{L}, \lambda_{i}^{U}]$ ,  $i = 1, \ldots, I$  involves a number of properties which enable us to construct a global optimization algorithm for finding the global minimum of f. These properties, whose proof is given in Maranas and Floudas (1994), are as follows.

Property 1.  $\mathcal{L}$  is a valid underestimator of f.

 $\forall \lambda_i \in [\lambda_i^{\mathrm{L}}, \lambda_i^{\mathrm{U}}], \ \mathscr{L}(\lambda_i) \leq f(\lambda_i).$ 

Property 2.  $\mathcal{L}$  matches f at all corner points.

 $\forall \lambda_i \text{ such that } \lambda_i = \lambda_i^{\mathrm{L}} \text{ or } \lambda_i = \lambda_i^{\mathrm{U}}, \quad \mathscr{L}(\lambda_i) = f(\lambda_i).$ 

Property 3.  $\mathscr{L}$  is convex in  $[\lambda_i^{L}, \lambda_i^{U}]$ .

Property 4. The maximum separation between  $\mathcal{L}$  and f is bounded and proportional to  $\alpha$  and to the square of the diagonal of the current box constraints.

$$\max_{\lambda_i^{\mathsf{L}} \leq \lambda_i \leq \lambda_i^{\mathsf{U}}} (f - \mathscr{L}) = \frac{1}{4} \alpha \sum_{i=1}^{I} (\lambda_i^{\mathsf{U}} - \lambda_i^{\mathsf{L}})^2.$$

*Property 5.* The underestimators constructed over supersets of the current set are always *less tight* than the underestimator constructed over the current box constraints for every point within the current box constraints.

Property 6.  $\mathscr{L}$  corresponds to a relaxed dual bound of the original function f (see Floudas and Visweswaran, 1990, 1993).

### 3.2. Highlights of the algorithm

Based on the aforementioned properties, the global optimization algorithm locates the global minimum  $f^*$  by constructing converging lower and upper bounds on  $f^*$ . A lower bound on  $f^*$ , denoted as  $f^L$ , within some box constraints is derived by invoking Properties (1) and (3). Based on these properties  $\mathcal{L}$  is a convex lower bounding function of f. Therefore, its single global minimum within some box constraints is a valid lower bound  $f^L$  on the global minimum solution  $f^*$  and can be guaranteed to be found with available local optimization algorithms. An upper bound on  $f^*$ , denoted as  $f^U$ , is then simply the value of f at the global minimum point of  $\mathcal{L}$ .

The next step, after establishing an upper and a lower bound on the global minimum, is to refine them by using Property (5). This property implies that the value of  $\mathscr{L}$  at every point, and therefore at the global minimum, is increased by restricting the box constraints within which it has been defined. Tighter box constraints occur by partitioning the rectangle defined by the initial box constraints into a number of smaller rectangles. One way of partitioning is to successively divide the current rectangle in two subrectangles by halving on the middle point of the longest side of the initial rectangle (bisection). At each iteration the lower bound of  $f^*$  is simply the minimum over all the minima of  $\mathcal{L}$  in every subrectangle composing the initial rectangle. Therefore, a straightforward (bound improving) way of tightening the lower bound  $f^{L}$  is to halve at each iteration only the subrectangle responsible for the infimum of the minima of  $\mathcal L$  over all subrectangles, according to the rules discussed earlier. This procedure generates a nondecreasing sequence for the lower bound  $f^{L}$  of  $f^{*}$ . Furthermore, we construct a *nonincreasing* sequence for the upper bound  $f^{U}$  by selecting it to be the infimum over all the previously recorded upper bounds. Clearly, if the global minimum of  $\mathscr{L}$  in any subrectangle is greater than the current upper bound  $f^{U}$ we can ignore this subrectangle because the global minimum of f cannot be situated inside it (fanthoming step).

Property (4) addresses the question of how small these subrectangles must become before the upper and lower bounds of f inside these subrectangles are within  $\varepsilon$ . If  $\delta$  is the diagonal of the subrectangle,

$$\delta = \sqrt{\sum_{i=1}^{I} \left(\lambda_i^{\mathrm{U}} - \lambda_i^{\mathrm{L}}\right)^2}$$

and  $\varepsilon$  is the convergence tolerance, from Property (4) we have the following condition for convergence,

$$\varepsilon \geq \frac{1}{4} \alpha \sum_{i=1}^{I} \left( \lambda_i^{U} - \lambda_i^{L} \right)^2 = \frac{1}{4} \alpha \delta^2 \geq f^{U} - f^{L}$$

which means that if the diagonal  $\delta$  of a subrectangle is

$$\delta < \sqrt{\frac{4\varepsilon}{\alpha}}$$

then  $\varepsilon$ -convergence to the global minimum of f has been achieved. In practice, however, not only  $\varepsilon$ -convergence to the global minimum is required, but convergences in a finite number of iterations. By analyzing the structure (sparsity) of the branch-and-bound tree resulting from the subdivision process, finite upper and lower bounds on the total number of required steps for  $\varepsilon$ -convergence can be derived (see Maranas and Floudas, 1994):

Iter<sub>max</sub> = 
$$\left[\frac{\sum_{i=1}^{I} (\lambda_i^{U} - \lambda_i^{L})^2}{\sqrt{4\epsilon/\alpha}}\right]^{I} - 1,$$

Iter<sub>min</sub> = 
$$I \log_2 \left[ \frac{\sum_{i=1}^{I} \left( \lambda_i^{U} - \lambda_i^{L} \right)^2}{\sqrt{4\epsilon/\alpha}} \right] - 1$$

The basic steps of the proposed global optimization algorithm are as follows:

# 3.3. Outline of algorithmic steps

Step 1: Initialization. A convergence tolerance  $\varepsilon$  is selected and the iteration counter Iter is set to one. Lower and upper bounds on the global minimum  $f^{\text{LBD}}$ ,  $f^{\text{UBD}}$  are initialized.

Step 2: Update of upper bound  $f^{\text{UBD}}$ . If the value of f at the current point is less than  $f^{\text{UBD}}$ , then  $f^{\text{UBD}}$  is set equal to the value of f at the current point.

Step 3: Partitioning of current rectangle. The current rectangle is partitioned into two rectangles by halving along the longest side of the initial rectangle.

Step 4: Solution of convex problems in two subrectangles. The following convex nonlinear optimization problem is solved in both subrectangles using a standard convex nonlinear solver.

$$\min_{\lambda_{i}} \quad \mathscr{L} = f + \alpha \sum_{i=1}^{I} \left(\lambda_{i}^{\mathsf{L}} - \lambda_{i}\right) \left(\lambda_{i}^{\mathsf{U}} - \lambda_{i}\right)$$
  
s.t. 
$$\sum_{i=1}^{I} \lambda_{i} = 1,$$
$$\lambda_{i}^{\mathsf{L}} \leq \lambda_{i} \leq \lambda_{i}^{\mathsf{U}}, \quad i = 1, \dots, I.$$

If a solution is less than the current upper bound, it is stored along with the value of the variable  $\lambda_i$  at the solution point.

# Step 5: Update iteration counter Iter and Lower Bound $f^{LBD}$ .

The iteration counter is increased by one, and the lower bound  $f^{\text{LBD}}$  is updated to be the incumbent solution. Further, the selected solution is erased from the solutions set. Stop if iteration limit is reached.

Step 6: Update current point and variable bounds. The current point is selected to be the solution point of the previously found minimum solution in Step 5, and the current rectangle is updated to be the one containing the previously found solution,

# Step 7: Check for convergence. IF $(f^{UBD} - f^{LBD}) > \varepsilon$ , then return to Step 2

Otherwise,  $\varepsilon$ -convergence has been reached.

A detailed description of the algorithmic steps as well as a mathematical proof of  $\varepsilon$ -convergence to the global minimum for the employed global optimization algorithm can be found in Maranas and Floudas (1994).

## 3.4. Geometric interpretation

A geometric interpretation of the proposed algorithm when applied to a onedimensional problem is shown in Fig. 3. The objective is find the global minimum  $F(X^*)$  of a nonconvex function F in a single variable X within the interval  $[X^L, X^U]$ . F involves two distinct local minima and thus traditional local optimization algorithms might miss the global minimum of F. Based on the proposed approach the initial interval  $[X^L, X^U]$  is partitioned into two subintervals  $[X^L, X^0]$ ,  $[X^0, X^U]$  and the convex lower bounding function

$$L(X) = F(X) + \alpha \left( X^{L} - X \right) \left( X^{U} - X \right)$$

is constructed in both subintervals. Since L(X) is convex, its single global minima  $L(X^1), L(X^2)$  in each subrectangle can be found with currently available convex solvers. Clearly, the value of the function F at  $X^0$  is an upper bound on the global minimum solution  $F(X^*)$  and the minimum over  $L(X^1), L(X^2)$  a lower bound on  $F(X^*)$ . Therefore, at the end of the first iteration we have  $UB = F(X^0)$  and LB  $= L(X^1)$ .

In the second iteration, since  $L(X^1) < L(X^2)$  we further partition the first subinterval  $[X^L, X^0]$  into two new subintervals  $[X^L, X^1]$  and  $[X^1, X^0]$  and two new convex lower bounding functions L are constructed one for each subinterval. The global minima of L in these two new subintervals are, respectively,  $L(X^3)$ and  $L(X^4)$ . Since  $F(X^1) < F(X^0)$  the new upper bound on  $F(X^*)$  is UB =  $F(X^1)$ . Also because  $L(X^4) = \min(L(X_2), L(X_3), L(X_4))$ , the new lower bound



Fig. 3. Geometric interpretation.

on  $F(X^*)$  is  $LB = L(X^4)$ . Note how tightly the upper and lower bounds UB, LB are bracketing the global minimum solution  $F(X^*)$ .

In the next section, the proposed approach is applied to the DB financial planning problem.

#### 4. Computational results

The global optimization algorithm is applied to a DB investment problem involving I = 9 assets, T = 20 time periods and S = 100 scenarios. The following investment choices represent the I = 9 assets: (1) cash equivalent; (2) treasury bonds; (3) large capitalization stocks; (4) international bonds; (5) small capitalization stocks; (6) venture capital; (7) international stocks; (8) real estate; and (9) a government/corporate bond index. According to the mean-variance model the expected return Mean( $w_T$ ) quantifies the profitability of the particular investment and the variance  $Var(w_T)$  measures the associated risk. The initial wealth  $w_0$  is set equal to one so as the profit Mean( $w_T$ ) and the risk  $Var(w_T)$  are normalized. The multi-period efficient frontier (risk vs. profit) at the end of the planning horizon for the particular investment fixed-mix problem is then generated by successively varying  $\beta$  from zero to one and solving the global optimization problem for each value of  $\beta$ . As in Mulvey (1994), the probabilities of occurrence for scenarios s = 1, ..., S are equal. Thus, we have

$$p^s = \frac{1}{S} = 0.01, \quad \forall s = 1, \dots, S.$$

Next, by applying the scenario generation technique introduced in Mulvey (1996), the  $(I = 9) \times (T = 20) \times (S = 100) = 18000$  returns  $r_{i,t}^s$  for asset *i*, in time step *t*, and scenario *s* are estimated. To provide some insight on their numerical values, their respective minimum and maximum value as well as their mean and standard deviation are as follows:

In the next step, we solve the nonconvex problem (DB) with different values of the parameter  $\beta$ . Three different alternatives for the absolute convergence tolerance  $\varepsilon_1 = 0.05$ ,  $\varepsilon_2 = 0.03$  and  $\varepsilon_3 = 0.01$  are considered. Regardless of the initial point, convergence to the  $\varepsilon$ -global minimum solution is achieved. Exact convergence to the global minimum  $f^*$  is then achieved by slightly perturbing the obtained solution  $f^U$  with a gradient-based optimization algorithm until the KKT conditions are satisfied.

For different values of  $\beta$  the value of the objective function at the global minimum solution  $f^*$  as well as the associated profit

$$\operatorname{Profit} = \operatorname{Mean}(w_T^*) = w_0 \sum_{s=1}^{S} p^s \left\{ \prod_{t=1}^{T} \left[ \sum_{i=1}^{I} \left( 1 + r_{i,t}^s \right) \lambda_i^* \right] \right\}$$

and risk

$$\operatorname{Risk} = \operatorname{Var}(w_T^*) = w_0^2 \left\{ \sum_{s=1}^{S} p^s \left[ \prod_{t=1}^{T} \left( \sum_{i=1}^{I} (1 + r_{i,t}^s) \lambda_i^* \right) \right]^2 - \left[ \sum_{s=1}^{S} p^s \left[ \prod_{t=1}^{T} \left( \sum_{i=1}^{I} (1 + r_{i,t}^s) \lambda_i^* \right) \right] \right]^2 \right\}$$

along with the required number of iterations Iter and CPU (seconds) time on a HP-730 workstation at Princeton University are shown in Table 6. Three cases of absolute tolerances of  $\varepsilon_1 = 0.05$ ,  $\varepsilon_2 = 0.03$  and  $\varepsilon_3 = 0.01$  are considered. More specifically, the progress of the upper and lower bounds on the global optimum solution for the case  $\beta = 0.5$  is shown in Table 5. In Table 7 the global optimum

Iter	$f^{LBD}$	$f^{UBD}$	CPU (s)
1	-2.534569	-2.466787	3.340
2	-2.527700	-2.468392	4.960
3	-2.517993	-2.468392	6.560
			(Convergence for $\varepsilon_1 = 0.05$ )
4	-2.508569	-2.468392	8.210
5	-2.501949	-2.468392	9.410
6	-2.499233	2.468392	10.790
7	2.498880	-2.468392	12.470
8	-2.495425	-2.468597	14.410
			(Convergence for $\varepsilon_2 = 0.03$ )
9	-2.495425	-2.468597	16.350
10	-2.492574	-2.468915	18.540
11	-2.487626	-2.468915	20.260
12	-2.482876	-2.468915	22.350
13	-2.482548	-2.468915	24.220
14	-2.479623	-2.468915	25.640
			(Convergence for $\varepsilon_3 = 0.01$ )

Upper and lower bounds on global optimum for  $\beta = 0.5$ 

Table 5

values of the  $\lambda_i$ 's are illustrated for different values of  $\beta$ . These values show the evolution of the global optimum  $\lambda_i$ 's as the parameter  $\beta$  changes. By plotting risk vs. profit of global minimum solutions corresponding to different values of  $\beta$  the efficient frontier of the particular investment strategy is obtained (See Fig. 4).

The generated efficient frontier for this example forms a bounded continuous concave curve. The lower bound of this curve (risk=0.22) corresponds to the minimum possible risk attainable for the particular investment model which cannot be further reduced not even by accepting lower than (expected profit=3.40) returns. The upper bound (expected profit=16.03) is the absolute maximum expected return for the investment model at hand for which it is impossible to increase not even by withstanding higher than 662.35 level of risk. The importance of this curve lies in the fact that by choosing a value for the risk within  $0.22 \le \text{Risk} \le 662.35$  the global maximum expected profit can be found and vice versa by selecting a desired expected profit within  $3.40 \le \text{Profit} \le 16.03$  the global minimum expected risk can be obtained. As expected, at the beginning of the efficient frontier substantial increases in the expected profit can be achieved by accepting higher than the minimum level of risk. However, after about (Risk  $\approx 50$ ) this trend seizes and only marginal improvements on the expected profits are derived even for substantially higher levels of risk.

A comparison of the efficient frontier generated with the global optimization approach with the one obtained using a local solver (Lasdon et al., 1978) is

Table 6 Expected <sub>f</sub>	profit and risk for	different values of $\not \mid$	3 and correspondir	ig required iterat	ions and CPU tin	ne (seconds).			
B	* 2	Risk	Profit	$Iter_1$ $\varepsilon_1 = 0.05$	$CPU_1$ $\varepsilon_2 = 0.03$	Iter <sub>2</sub> $\varepsilon_3 = 0.01$	CPU <sub>2</sub>	Iter <sub>3</sub>	CPU3
0.000	-0.216963	0.216963	3.399787	2	2.800	2	2.680	∞	9.540
0.100	0.175781	0.246934	3.980224	2	3.330	5	6.830	16	21.600
0.200	0.634769	0.359652	4.612451	ę	4.970	9	8.790	15	24.530
0.300	1.172335	0.566784	5.230278	r,	5.150	6	9.220	14	22.460
0.400	1.793346	0.807104	5.694021	e.	4.850	5	7.750	14	22.910
0.500	2.469084	1.058045	5.996214	3	4.830	8	12.610	15	25.660
0.600	3.221678	1.622239	6.450956	4	6.580	10	15.850	21	33.640
0.700	4.106418	2.732108	7.037215	5	6.730	10	12.940	20	29.180
0.800	5.213183	5.026511	7.773107	5	6.250	=	13.490	22	30.010
0.900	6.832546	12.373188	8.966517	5	5.330	7	7.440	16	19.050
0.910	7.054916	14.060312	9.143236	5	5.610	8	8.260	21	22.470
0.920	7.298454	16.267080	9.347631	5	5.120	×	8.340	22	23.090
0.930	7.569941	19.261830	9.589537	5	5.110	7	7.340	20	20.710
0.960	8.596387	30.574230	10.228497	5	4.260	7	6.230	20	16.010
0.970	9.056114	42.286782	10.644039	5	3.980	7	5.790	20	15.940
0.980	9.722067	74.981914	11.450719	5	3.660	9	4.580	14	10.990
0.985	10.245140	115.578747	12.161240	5	3.510	9	4.370	7	5.380
0.988	10.691630	161.149760	12.778772	2	1.440	5	3.790	Ξ	8.910
0660	11.080610	204.812652	13.261350	2	1.380	5	3.350	11	7.470
0.991	11.313331	235.018119	13.550448	2	1.380	5	3.530	Π	7.790
0.992	11.580623	273.855880	13.882530	2	1.400	5	3.550	П	7.080
0.993	11.893022	325.373905	14.270533	2	1.450	5	3.270	Ξ	6.970
0.994	12.266417	396.515497	14.733913	2	1.440	2	1.320	11	7.530
0.995	12.726252	500.151831	15.303529	2	1.370	2	1.320	9	4.090
1.000	16.030205	662.348760	16.030205	2	1.220	2	1.160	2	1.310

β	γ <mark>1</mark> *	Å2*	ئ*	д* Д4	2 <del>.</del> *	λ <del>"</del>	۶ <del>،</del> ۲ <del>,</del>	λ <del>*</del>	2,*
0.000	0.649741	0.028132	0.076547	0.235045	0.00000	0.010535	0.000000	0.00000	0.00000
0.100	0.468424	0.115252	0.147128	0.218042	0.00000	0.023116	0.00000	0.010013	0.018025
0.200	0.264850	0.093321	0.179713	0.207471	0.024087	0.033100	0.00000	0.022981	0.174477
0.300	0.091111	0.073204	0.192691	0.199119	0.056749	0.040721	0.000000	0.039425	0.306980
0.400	0.00000	0.070626	0.201995	0.194707	0.088711	0.047715	0.00000	0.051092	0.345154
0.500	0.00000	0.098580	0.207528	0.192832	0.126476	0.055201	0.001109	0.058613	0.259661
0.600	0.00000	0.143648	0.209584	0.170349	0.163405	0.062722	0.038795	0.071562	0.139935
0.700	0.000000	0.189269	0.212357	0.145747	0.210798	0.071624	0.082719	0.087486	0.00000
0.800	0.000000	0.069557	0.224844	0.125808	0.282628	0.087719	0.133575	0.075869	0.000000
0.900	0.000000	0.000000	0.152845	0.069575	0.435846	0.120036	0.220693	0.001005	0.000000
0.910	0.00000	0.00000	0.118728	0.056194	0.463112	0.125112	0.236854	0.00000	0.00000
0.920	0.000000	0.00000	0.079066	0.040746	0.493534	0.130984	0.255668	0.00000	0.000000
0.930	0.000000	0.00000	0.033623	0.022879	0.527640	0.137983	0.277875	0.00000	0.000000
0.940	0.000000	0.00000	0.000000	0.00000	0.552284	0.149331	0.298385	0.00000	0.00000
0.950	0.000000	0.00000	0.000000	0.00000	0.544832	0.180501	0.274667	0.000000	0.000000
0.960	0.000000	0.00000	0.000000	0.00000	0.530058	0.228150	0.241792	0.000000	0.000000
0.970	0.00000	0.00000	0.00000	0.00000	0.500960	0.305596	0.193444	0.00000	0.000000
0.980	0.00000	0.00000	0.000000	0.000000	0.448060	0.437499	0.114441	0.000000	0.00000
0.985	0.000000	0.00000	0.00000	0.00000	0.410410	0.538430	0.051160	0.000000	0.00000
0.988	0.00000	0.00000	0.00000	0.00000	0.381924	0.618076	0.000000	0.000000	0.000000
0.990	0.00000	0.00000	0.000000	0.00000	0.317034	0.682966	0.00000	0.000000	0.000000
166.0	0.00000	0.000000	0.000000	0.00000	0.279928	0.720072	0.00000	0.000000	0.000000
0.992	0.000000	0.00000	0.000000	0.00000	0.238736	0.761264	0.00000	0.000000	0.00000
0.993	0.000000	0.00000	0.000000	0.00000	0.192338	0.807662	0.00000	0.000000	0.00000
0.994	0.000000	0.00000	0.000000	0.00000	0.139073	0.860927	0.00000	0.000000	0.00000
0.995	0.00000	0.000000	0.000000	0.00000	0.076349	0.923651	0.00000	0.00000	0.00000
1.000	0.00000.0	0.00000	0.000000	0.00000	0.000000	1.000000	0.000000	0.000000	0.000000

Table 7 Optimal  $\lambda_i^*$ 's for different values of  $\beta$ 



Fig. 4. Efficient frontier.

shown in Fig. 5. Clearly, the local solver fails to correctly generate the efficient frontier instead yielding a suboptimal curve.

The proposed global approach has been implemented in C providing an efficient portable investment allocation tool GLOFP, (Global Optimization in Financial Planning). Computational results show that GLOFP can be used in an on-line fashion to construct the efficient frontier of even large-scale multi-stage investment models. Work is under way to accommodate several alternative financial planning models and other nonlinear dynamic policies.

#### 5. Conclusions

The deterministic global optimization tool GLOFP obtains the efficient frontier of dynamically balanced investment problems with no transaction costs over a multi-stage planning horizon. Computational results suggest that the approach can be applied in an online fashion even for large-scale investment problems – guaranteeing always convergence to the global optimum point. When applied to real-life dynamically balanced investment problems an improved efficient frontier over the one generated with a widely used local solver (Lasdon et al., 1978)



Fig. 5. Comparison with local solver.

was found. The presented global optimization algorithm readily generalizes for von Neumann Morgenstern expected utility functions, other financial planning models and decision rule-based strategies such as constant proportional portfolio insurance (e.g. Perold and Sharpe, 1988). Work in this direction is currently under way. In addition, we are exploring the intergration of stochastic programs with decision rules and traditional stochastic programs for long-term financial planning problems.

# References

- Berger, A., Mulvey, J., 1996. Integrative risk management for individuals. In: W. Ziemba and J. Mulvey (Eds.), Worldwide Asset and Liability Management. Cambridge University Press, Cambridge.
- Berger, A., Mulvey, J., Ruszczynski, R., 1995. A new scenario method for large-scale stochastic optimization. Operations Research 43, 477-490.
- Brennan, M, Schwartz, E., Lagnado, R., 1996. Stategic asset allocation. Journal of Economic Dynamics and Control, this issue.
- Dantzig, G., Infanger, G., 1994. Planning Under Uncertainty. Boyd and Fraser, Danvers, MA.
- Davis, M., Norman, A., 1990. Portfolio selection with transaction costs. Math of Operations Research 15, 676-713.

- Dempster, M., Thompson, R., 1995. Parallelization and aggregation of nested Benders decomposition. presented at APMOD 95, Brunel University.
- Drud, A., 1994. CONOPT A large-scale GRG Code. ORSA Journal on Computing, 6, 207-216.
- Floudas, C., Visweswaran, V., 1990. A global optimization algorithm for certain classes of nonconvex NLPs, part 1: theory. Computers and Chemical Engineering. 14, 1397–1417.
- Floudas, C., Visweswaran, V., 1993. Primal-relaxed dual global optimization approach, Journal of Optimization Theory and its Application. 78, 187-225.
- Garstka, S., R J-B Wets, 1974. On decision rules in stochastic programming. Mathematical Programming, 7, 117-143.
- Grauer, R., Hakansson, N., 1985. Returns on levered actively managed long-run portfolios of stocks, bonds and bills. Financial Analysts Journal 24-43.
- Higle, J., Sen, S., 1991. Stochastic decomposition: an algorithm for two-stage linear programming with recourse, Mathematics of Operations Research 16, 650-669.
- Lasdon, L., Warren, A., Ratner, M., 1978. Design and testing of a GRG code for nonlinear optimization. ACM Transaction on Mathematical Software 4, 34-50.
- Maranas, C., Floudas, C., 1993. Global optimization for molecular conformation problems, Annals of Operations Research 42, 85-117.
- Maranas, C., Floudas, C., 1994. Global minimum potential energy conformations of small molecules. Journal of Global Optimization 4, 155–170.
- Merton, R., 1971. Optimum consumption and portfolio rules in continuous time model. Journal of Economic Theory 3, 373-413.
- Mulvey, J., 1994. An Asset-liability investment system. Interfaces 24, 22-33.
- Mulvey, J., 1996. Generating scenarios for the Towers Perrin investment system. Interfaces 26, 1-15.
- Mulvey, J., Vladimirou, H., 1992. Stochastic Network programming for financial planning problems. Management Science 38, 1642–1664.
- Perold, A., Sharpe, W., 1988. Dynamic strategies for asset allocation. Financial Analysts Journal 1, 16-27.
- Rockafellar, R., R J-B. Wets, 1991. Scenario and policy aggregation in optimization under uncertainty. Mathematics of Operations Research 16, 119–147.
- Taksar, M., Klass M., Assaf, D., 1988. A diffusion model optimal portfolio selection in the presence of brokerage fees. Math of Operations Research 13, 277-294.
- Worzel, K., Vassiadou-Zeniou, C., Zenios, S., 1994. Integrated simulation and optimization models for tracking indices of fixed income securities. Operations Research 42, 223–233.
- Zenios, S., 1993. Financial Optimization, Cambridge University Press, Cambridge.
- Zenios, S., Yang, D., 1996. A scaleable parallel interior algorithm for stochastic linear programming and robust optimization. Computational Optimization and Applications. In print.