Optimal Rebalancing Strategy Using Dynamic Programming for Institutional Portfolios

Walter Sun[1,†], Ayres Fan[1], Li-Wei Chen[1], Tom Schouwenaars[1], Marius A. Albota[2]

Abstract

Institutional fund managers generally rebalance using *ad hoc* methods such as calendar basis or tolerance band triggers. We propose a different framework that quantifies the cost of a rebalancing strategy in terms of risk-adjusted returns net of transaction costs. We then develop an optimal rebalancing strategy that actively seeks to minimize that cost. We use certainty equivalents and the transaction costs associated with a policy to define a cost-to-go function, and we minimize this expected cost-to-go using dynamic programming. We apply Monte Carlo simulations to demonstrate that our method outperforms traditional rebalancing strategies like monthly, quarterly, annual, and 5% tolerance rebalancing. We also show the robustness of our method to model error by performing sensitivity analyses.

[1] Laboratory for Information and Decision Systems, Massachusetts Institute of Technology

[2] Research Laboratory of Electronics, Massachusetts Institute of Technology

[†] Corresponding author: waltsun@mit.edu

CONTENTS

I. Introduction

Institutional money managers develop risk models and optimal portfolios to match a desired risk/reward profile. Utility functions express risk preferences and implicitly reflect the views of the trustees or directors. Once a manager determines a target portfolio, the problem of maintaining this balance of assets is non-trivial. In particular, a manager must actively rebalance because different asset classes can exhibit different rates of return. Furthermore, managers also must rebalance if the weights in the target portfolio are altered. This occurs when the model for expected returns of asset classes change or the risk profile is altered.

Most academic theory ignores frictional costs and assumes that a portfolio manager can simply readjust their holdings dynamically without any problems. In practice, trading costs are non-zero and affect the decision to rebalance. The transaction costs involve commissions and market impact as well as the cost of manpower and technological resources. If the transaction costs exceed the expected benefit from rebalancing, then no adjustment should be made. However, without any quantitative measure for this benefit, we cannot accurately determine whether or not to trade.

Conventional approaches to portfolio rebalancing include periodic and tolerance band rebalancing [8], [13]. With periodic rebalancing, the portfolio manager adjusts to the target weights at a consistent time interval (*e.g.*, monthly or quarterly). The drawback with this method is that trading decisions are independent of market behavior. Thus, rebalancing may occur even if the portfolio is nearly optimal. Tolerance band rebalancing requires managers to rebalance whenever any asset class deviates beyond some predetermined tolerance band (*e.g.*, $\pm 5\%$). When this occurs, the manager fully rebalances to the target portfolio. While this method reacts to market movements, the threshold for rebalancing is fixed, and the process of rebalancing involves trading all the way back to the optimal portfolio.

Previous research on dynamic strategies for asset allocation [17] has established the existence of a no-trade region around the optimal target portfolio weights [9]. If the proportions allocated to each asset at any given time lie within this region, trading is not necessary. However, if current asset ratios lie outside the no-trade region, Leland has shown that it is optimal to trade but only to bring the weights back to the nearest edge of the no-trade region rather than to the target ratios. The optimal strategy has been shown to reduce transaction costs by approximately 50%. However, the full analytical solution involves a complicated system of partial differential equations in multiple dimensions.

Mulvey and Simsek [15] have modeled the problem of rebalancing in the face of transaction costs as a generalized network with side conditions and developed an algorithm for solving the resulting problem.

Meanwhile, Mitchell and Braun [14] have described a method for finding an optimal portfolio when proportional transactions costs have to be paid. More recently, Donohue and Yip [8] have confirmed the results of Leland [9] and have characterized the shape and size of the no-trade region and compared the performance of different rebalancing strategies.

In this paper, we present an approach that explicitly weighs transaction costs and portfolio tracking error. We assume that we are living in a CAPM world [12] which means that asset returns are stationary and that mean and variance are the primary portfolio statistics of interest. Utility functions coupled with asset return models then yield a target portfolio which is a set of optimal weights for different asset classes. We also assume that the portfolios are either tax-free or tax-deferred, which is the case for endowments, charities, pension funds, and most individual retirement funds.

The main difficulty with reconciling transaction costs and tracking error is that they are expressed in different units. Transaction costs are something tangible: dollars. Tracking error is a more abstract concept. Because the optimal portfolio is that which maximizes our given utility function, we can express tracking error as the shortfall in utils from our current portfolio to the optimal portfolio. Our first contribution is applying the concept of *certainty equivalents* [3] to create risk-adjusted returns that allow us to convert tracking error into a dollar-denominated cost. Note that we are not restricted to quadratic utility but can use arbitrary utility functions. Once we have a dollar cost, we can then directly compare the transaction costs for rebalancing with the suboptimality costs for not rebalancing. This is leaving out an essential piece of the puzzle: our actions this period also affect outcomes and decisions in future periods. Our second contribution is to then apply the method of *dynamic programming* to minimize a cost functional that explicitly models this point. Thus our optimal policy trades only when the expected cost of trading is less than the expected cost of doing nothing with costs evaluated over the next period and all future periods. In addition, we search over the rebalancing space from 0% (no rebalancing) to 100% rebalancing (full rebalancing) and the points in between. In most cases, partial rebalancing can provide nearly the same utility as full rebalancing while saving on transaction costs. Our third contribution is a framework to quantitatively evaluate rebalancing strategies. We show that our method performs better than traditional methods of rebalancing and is robust to model error.

The organization of the paper is as follows. In Section II, we introduce utility functions and the method of dynamic programming. We define certainty equivalents and discuss how we use them to determine our optimal rebalancing strategy in Section III. We then demonstrate our rebalancing results in Section IV. We begin with a simple two-asset example to illustrate our algorithm and provide some simple sensitivity analyses and then examine the more general case with multiple assets. We conclude

	Utility Function	Expected Utility
Quadratic	$f_{\mathrm{q}}(x) = x - \frac{\alpha}{2}(x - x_0)^2$	$U_{\mathrm{q}}(\mu, \sigma) = \mu - \frac{\alpha}{2}\sigma^2$
Log wealth	$f_{\mathrm{l}}(x) = \log(1 + x)$	$U_{\mathrm{l}}(\mu, \sigma) = \log(1 + \mu) - \frac{\sigma^2}{2(1+\mu)^2}$
Power	$f_{\mathrm{p}}(x) = 1 - 1/(1 + x)$	$U_{\mathrm{p}}(\mu, \sigma) = 1 - \frac{1}{(1+\mu)} - \frac{\sigma^2}{(1+\mu)^3}$

TABLE I

UTILITY FUNCTIONS AND THEIR CORRESPONDING APPROXIMATE EXPECTED UTILITIES USED. THE UTILITY FUNCTIONS f_i ARE EXPRESSED IN TERMS OF THE RETURN x. THE EXPECTED UTILITY FUNCTIONS U_i ARE SPECIFIED IN TERMS OF THE MEAN RETURN μ AND THE STANDARD DEVIATION σ. FOR QUADRATIC UTILITY, α IS THE RISK AVERSION PARAMETER.

the paper in Section V.

II. BACKGROUND

A. Utility Functions

Evaluating individual preferences to risk and return and making corresponding portfolio allocation decisions is a difficult task. Investment professionals need to analyze various factors to develop portfolios that allow clients to reach their investment goals while taking into account risks associated with bear markets and singular events such as crashes.

No single portfolio can meet the needs of every investor. As mentioned in the introduction, one way to specify an investor's risk preference is through the use of utility functions [11]. A utility function indicates how much satisfaction (utils) we get for a given level of return x. Clearly, most people prefer higher levels of return to lower levels of return, so generally utility functions monotonically increase with x. If the marginal utility decreases with x (*i.e.*, utility grows sublinearly), then an individual is said to be *risk averse*. Numerous other risk characteristics can be imparted through the shape of the utility function. A manager then chooses portfolio weights w so that expected utility is maximized.

Because future returns are unknown, we need to use expected utility to create an optimal portfolio or to decide on a rebalancing policy. It has been shown by Levy and Markowitz [10] that for most relevant utility functions, this expected utility U can be approximated using truncated Taylor series expansions to be a function of mean and standard deviation, $\mathrm{U}(\mu, \sigma)$.

In Table I, we list three utility functions and the corresponding expected utilities that we use [6]. For each utility, $f_i(x)$ for $i = \{\mathrm{q}, \mathrm{l}, \mathrm{p}\}$ (where q indicates quadratic, l indicates logarithmic, and p indicates power) represents the utility in utils given a return x, which we also refer to as the empirical utility. $\mathrm{U}_i(\mu, \sigma)$ for $i = \{\mathrm{q}, \mathrm{l}, \mathrm{p}\}$ is the expected utility. Appendix I provides the derivation for the expected utility.

B. Dynamic Programming

Dynamic programming [1], [2], [4] is an optimization technique that finds the policy that minimizes expected cost given a cost functional and a dynamic model of state behavior. At time t, w_t is our state, u_t is our policy, and n_t is the state uncertainty. The state transition is defined by an arbitrary function h:

action

$$w_{t+1} = \mathrm{h}(w_t, u_t, n_t), \tag{1}$$

where w_{t+1} represents the new state which is influenced by the prior state w_t, the action taken u_t, and the uncertainty in the system dynamics n_t. We write the cost functional recursively as:

$$\mathrm{J}_t(w_t) = \mathrm{E}\left[\mathrm{G}(w_t, u_t, n_t) + \mathrm{J}_{t+1}(w_{t+1})\right], \tag{2}$$

where E is mathematical expectation, G is the cost for the current period, and J_t is the so-called *cost-to-go* function. J_t is the expected future cost from t onwards given all future decisions. So, the cost at any given period is the expected cost from t to $t+1$ along with the expected cost from $t+1$ onwards. At each time t, the optimal strategy is to choose u_t such that the cost is minimized:

Same Bellman in chess playing

$$\mathrm{J}_t^*(w_t) = \min_{u_t} \quad \mathrm{E}\left[\mathrm{G}(w_t, u_t, n_t) + \mathrm{J}_{t+1}(w_{t+1})\right]. \tag{3}$$

Equation (3) is the discrete-time Bellman Equation. Assuming convergence, this recursion approaches a fixed point such that $\mathrm{J}_t^*(w) = \mathrm{J}_{t+1}^*(w) = \mathrm{J}^*(w)$. The challenge is therefore to determine the cost-to-go values $\mathrm{J}^*(w)$. Once these values are known, the optimal rebalancing decision is to choose the policy u_t^* that minimizes (3).

We can determine the cost-to-go values using a technique called *value iteration*. The idea behind value iteration is to choose an arbitrary set of cost-to-go values $\mathrm{J}_t(w)$ for some time t that we imagine to be very far in the future. We then repeatedly apply (3) to obtain cost-to-go values successively closer to the present. After a sufficient number of iterations, we will approach a steady-state, and the cost-to-go values should converge to the optimal values $\mathrm{J}^*(w)$.

J* is a function. discretize and enumerate all W_t to get J*

The 5-year idea

III. Optimal Rebalancing Using Dynamic Programming

In this section, we investigate optimal rebalancing strategies for portfolios with transaction costs. In general, we consider a multi-asset problem where we are given an optimal portfolio consisting of a set of target portfolio weights $w^* = \{w_1^*, \ldots, w_N^*\}$, where N is the total number of assets. The optimal strategy should be to maintain a portfolio that tracks the optimal portfolio as closely as possible while minimizing the transaction costs.

We consider a model where we observe the contents of the portfolio w_t at the end of each month. At this point, we have the option of rebalancing the portfolio (i.e., apply our policy, or control, u_t). Thus, the portfolio at the beginning of the next month is $w_t + u_t$. Assuming normal returns in the process noise n_t, we use a simple multiplicative dynamic model so that $w_{t+1} = (1+n_t)(w_t+u_t)$, although in general, w_{t+1} can be an arbitrary function of w_t, u_t, and n_t.

In general, the decision to rebalance should be based on a consideration of three costs: the tracking error associated with any deviation in our portfolio from the optimal portfolio, the trading costs associated with buying or selling any assets during rebalancing, and the expected future cost from next month onwards given our actions in the current month. The optimal strategy dynamically minimizes the total cost, which is the sum of these three costs.

To apply dynamic programming, we must specify the cost function in the Bellman Equation. In our case, we write:

$$\mathrm{E}\left[\mathrm{G}(w_t, u_t, n_t)\right] = \tau(u_t) + \epsilon(w_t + u_t), \tag{4}$$

where $\tau(u_t)$ is the trading cost associated with applying our rebalancing decision u_t. This can include tangible costs such as commissions and market impact, but can also model indirect costs such as employee labor. $\epsilon(\cdot)$ represents the suboptimality cost, the cost of not having an optimal portfolio. $\epsilon(w_t + u_t) = 0$ whenever $w_t + u_t = w^*$ (i.e., choose u_t so that we rebalance to the target portfolio); otherwise, $\epsilon(\cdot) > 0$.

A. Modeling Tracking Error using Certainty Equivalents

μ: mean
Λ: var.

Note that the cost-to-go values, and hence the optimal strategy, will depend on the cost functions $\tau(\cdot)$ and $\epsilon(\cdot)$ chosen. In the certainty equivalence approach, we model the investor's preferences using a utility function (see Section II-A). For any portfolio weights w, we can express the expected utility as $\mathrm{U}(\mu^{\mathrm{T}}w, w^{\mathrm{T}}\Lambda w)$. We observe that there exists a risk free rate (which we will denote as $r_{CE}(w)$) that produces an identical expected utility. We therefore call $r_{CE}(w)$ the *certainty equivalent return* for the weights w. The condition for this is $\mathrm{U}(r_{CE}, 0) = \mathrm{U}(\mu^{\mathrm{T}}w, w^{\mathrm{T}}\Lambda w)$. The certainty equivalents for the three expected utility functions that we are using are:

1) Quadratic: $r_{CE}(w) = \mathrm{U_q}(\mu^{\mathrm{T}}w, w^{\mathrm{T}}\Lambda w)$
2) Log wealth: $r_{CE}(w) = \exp(\mathrm{U_l}(\mu^{\mathrm{T}}w, w^{\mathrm{T}}\Lambda w)) - 1$
3) Power: $r_{CE}(w) = 1/(1 - \mathrm{U_p}(\mu^{\mathrm{T}}w, w^{\mathrm{T}}\Lambda w)) - 1$.

One interpretation of the certainty equivalent then is as a risk-adjusted rate of return given the risk preferences embedded in the utility function.

If we hold a suboptimal portfolio w, the utility of that portfolio $\mathrm{U}(w)$ will be lower than $\mathrm{U}(w^*)$, with a correspondingly lower certainty equivalent return. We can interpret this as losing a riskless return (equal to the difference between the two certainty equivalents) over one period, corresponding to the penalty paid for tracking error. Therefore, under the certainty equivalence approach, the tracking error has the cost function ☆

Discuss.

$$\epsilon(w) = r_{CE}(w^*) - r_{CE}(w). \tag{5}$$

This difference between the certainty equivalent of a non-optimal portfolio and that of the optimal portfolio is defined as the cost of not being optimal.

The reason why we use a certainty equivalent is because in our cost function, $\tau(\cdot)$ and $\epsilon(\cdot)$ must have commensurate values. We know that the cost will be in terms of dollars or basis points or some other absolute measure. It is more straightforward to then convert portfolio tracking error into a similar absolute measure using certainty equivalents rather than trying to express the trading costs in terms of diminished expected utility, though this can be done as well.

B. *Modeling Transaction Costs*

Assume that we have a portfolio w and we want to go to another portfolio w'. The simplest model for transaction costs is simply to assume a linear cost. Under this model, we assume that for asset i we pay a transaction cost of c_i per dollar to buy or sell the asset. Under this model,

$$\tau(w', w) = c^T|w' - w|, \tag{6}$$

same .

where $c^T = [c_1, \ldots, c_N]$ is the vector of transaction cost coefficients. This is just one specific choice of transaction cost. Alternative transaction cost functions can be chosen as appropriate. For example, price impact models can be used or an affine model can be used to discourage frequent trading.

IV. Experimental Results

In the results that we present in this section, we will use five asset classes: Hedge Funds, Developed Markets Equity, Emerging Markets Equity, US Equity, and Private Equity. Table II shows the mean and standard deviation of each asset class for historical monthly returns from the last decade. The correlation matrix used is shown in Table III. Of the different assets, Private Equity provides the most expected return but has the greatest amount of risk. On the other extreme, Hedge Funds have both the least expected

	Index as Proxy (Source)	Mean Return (%)	Std. Dev. (%)
Hedge Funds	HFR Mkt Neutral (Bloomberg)	5.28	10.16
Developed Markets	MSCI EAFE+Canada (Datastream)	6.65	16.76
Emerging Markets	MSCI EM (Datastream)	7.88	23.30
US Equity	Russell 3000 (Datastream)	6.84	14.99
Private Equity	Wilshire LBO (Bloomberg)	12.76	44.39

TABLE II

ANNUAL MEAN RETURNS AND ANNUAL STANDARD DEVIATIONS FOR THE ASSET CLASSES.

	Hedge Funds	Developed Markets	Emerging Markets	US Equity	Private Equity
Hedge Funds	1.00	0.09	0.21	0.29	0.36
Developed Markets	0.09	1.00	0.42	0.46	0.38
Emerging Markets	0.21	0.42	1.00	0.45	0.40
US Equity	0.29	0.46	0.45	1.00	0.64
Private Equity	0.36	0.38	0.40	0.64	1.00

TABLE III

CORRELATION COEFFICIENT MATRIX.

return and the least amount of variability. The mean returns were provided by State Street Associates and the variances and correlations were computed empirically from data acquired from Datastream and Bloomberg.

A. Two-Asset Model

To introduce the problem of portfolio rebalancing, we first consider an example involving only two of the asset classes: Developed Markets and Hedge Funds. The benefits of the two risky asset model are that the optimal portfolio can be computed in closed form (see [5] for a derivation), we can visually examine the changes in portfolio weights (since a single asset's weight represents the full description of our portfolio), and the parameters are few enough that we can easily perform sensitivity analyses. We follow this example with extensive simulations of a multi-asset model.

	(a)	(b)	(c)	(d)
	Trading Cost (bps)	Suboptimality Cost (bps)	Aggregate Cost (bps)	Utility Shortfall (utils x 10^4)
Ideal	0.00	0.00	0.00	0.00
Optimal DP	1.57	0.46	2.03	2.15
No Trading	0.00	4.74	4.74	4.56
5% Tolerance	3.68	0.13	3.81	3.65
Monthly	12.92	0.00	12.92	12.94
Quarterly	7.46	0.05	7.51	7.46
Annual	3.71	0.26	3.97	4.05

TABLE IV

Annualized trading cost, suboptimality cost, aggregate cost, and utility shortfall using six different rebalancing strategies on two risky assets over a twenty year period. Results obtained by averaging over $10{,}000$ realizations.

For brevity, in this section we only consider quadratic utility with risk aversion parameter $\alpha = 1.5$. Using this assumption, the optimal portfolio balance is 51% in Developed Markets and 49% in Hedge Funds. To illustrate the behavior of our rebalancing method, we simulate the returns of the two asset classes over a single twenty-year realization. Figure 1 shows how the portfolio weight of Developed Markets moves over one 240-month sample path. With no rebalancing (Figure 1(a)), the weight drifts from the optimal amount of 49% down to under 25%, resulting in large suboptimality costs (the exact costs are described below). Our optimal rebalancing strategy (Figure 1(b)) rebalances only when necessary. During months 110 to 120 and 170 to 215, the portfolio partially rebalances nearly every month to handle sharp changes in the portfolio, while for months 120 to 160, the lack of strong market movements in either direction allow us to avoid any transaction costs. The market movement during the times cited can be seen by examining the change in portfolio weights in Figure 1(a) where there is no rebalancing.

We evaluate different rebalancing algorithms using a Monte Carlo simulation process. Each month is sampled independently from the others, so we do not model effects such as trends, momentum, or mean reversion. For each sample path, we simulate the various rebalancing methods by generating a return value for each month that is net of transaction costs. Table IV shows the annualized costs of different rebalancing strategies. Trading costs were 40 bps for buying or selling Developed Markets and 60 bps

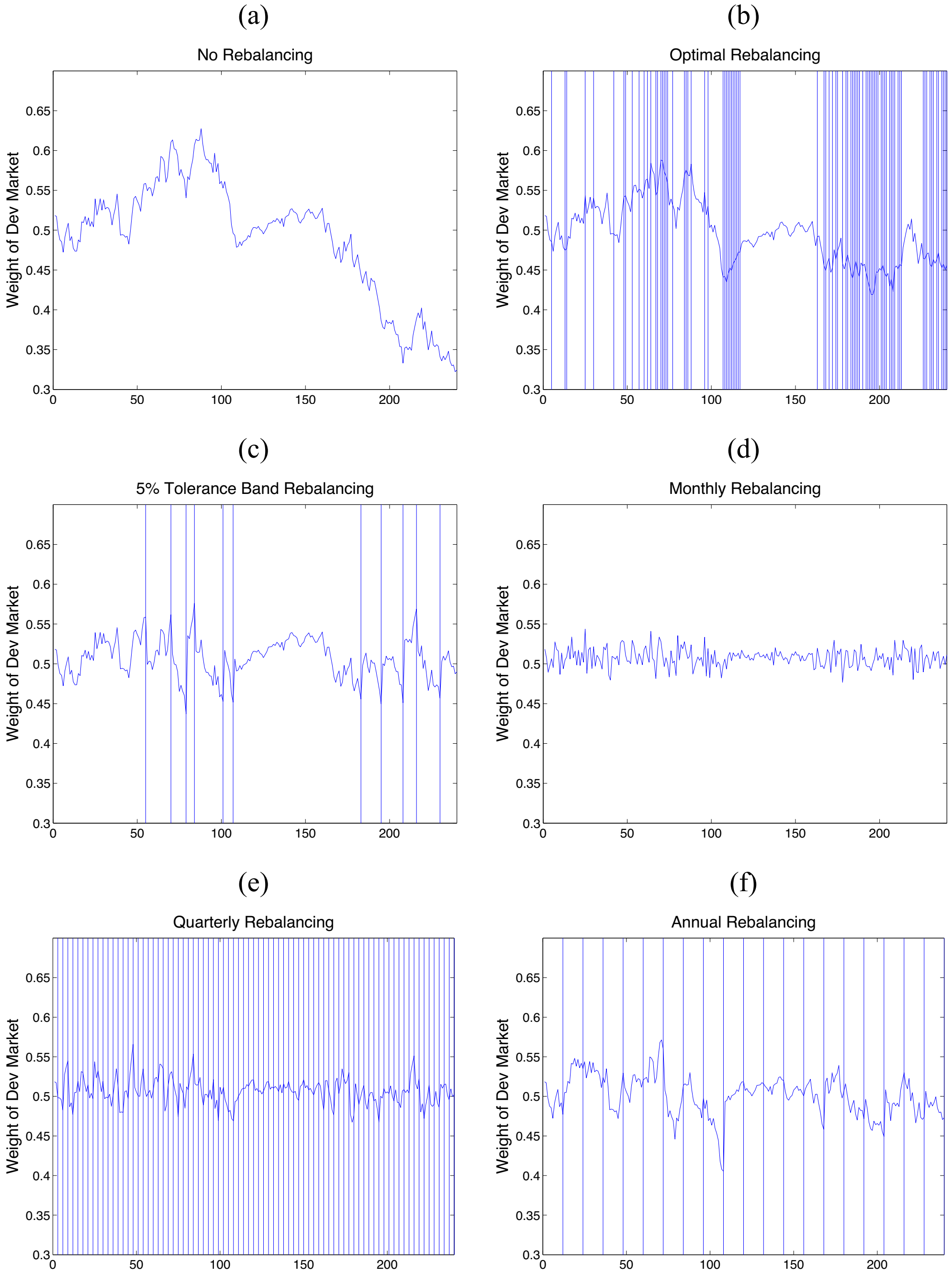

Fig. 1. Plots of Developed Markets weighting in the two asset example using different rebalancing models. The vertical lines indicate months where rebalancing was done (for monthly rebalancing, this is omitted since trading occurs in every month).

for buying or selling Hedge Funds.

We show two metrics to evaluate performance. In both cases, the numbers are in terms of shortfall from an idealized rebalancing strategy that is allowed to rebalance to the optimal portfolio for free every month. The first metric is aggregate expected cost shown in column (c). This is simply the sum of the trading costs in column (a) and the suboptimality costs in column (b). The suboptimality cost was computed at the end of each month as the difference between the expected risk-adjusted returns for the optimal portfolio and the current portfolio after the rebalancing policy has been applied. Note that this is precisely the metric that our dynamic programming approach is designed to minimize. The second metric is what we call empirical utility shortfall. For each month, we compute the return net of transaction costs and then compute the utils associated with that return using our empirical utility function f_i (see Table I). This metric is similar to aggregate cost but works with actual utility rather than the expected utility used to compute certainty equivalents. In the case of an ergodic process, we would expect the sample average and the expected value to converge which they do in our case[1] [16].

From the table, we observe that the aggregate cost is minimized by our method. Assuming a portfolio of $1 billion, the aggregate annual cost of our algorithm is $203,000. The cost for rebalancing using 5% tolerance, the next least expensive method, is $381,000 annually. The results for each rebalancing method make intuitive sense. Monthly rebalancing leads to no deviation from optimality, but at the cost of high trading fees. Infrequent trading yields smaller trading costs, but higher suboptimality costs. Our method of rebalancing whenever the cost of non-optimality exceeds the trading costs allows us to adequately trade-off the cost of non-optimality with that of trading.

B. Sensitivity Analysis

So far, we have assumed that the model for each asset is known. In practice, the mean and variance of each asset's returns as well as the correlation between the assets must be estimated (e.g., using historical observations). Errors in the parameter estimate (i.e., deviation from the true unknown value) will cause inaccuracies in the costs-to-go obtained from the dynamic program resulting in suboptimal rebalancing. In this section, we investigate the impact of errors in each of these parameters on the rebalancing strategy.

In particular, we consider the mean, standard deviation, and correlation parameters. For each simulation, only one of the three parameters is varied to isolate the effects of changes to each of these factors. We

[1] The units on utils multiplied by 10^4 (in column (d)) are similar to the basis points in columns (a)-(c). This is clear for the quadratic case where the certainty equivalent is equal to the empirical utility. For the other two cases, taking a linear approximation around $x = 0$ shows that the utilities are proportional to x. So utils times 10^4 is reasonably commensurate with basis points and explains why the numbers in columns (c) and (d) are similar.

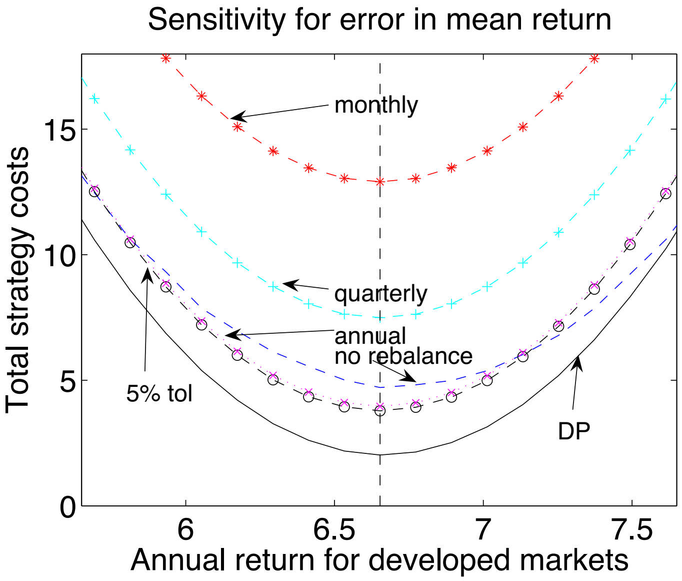

Fig. 2. Sensitivity analysis on mean return for the two asset example. The vertical line indicates the actual rate used by the rebalancing strategies.

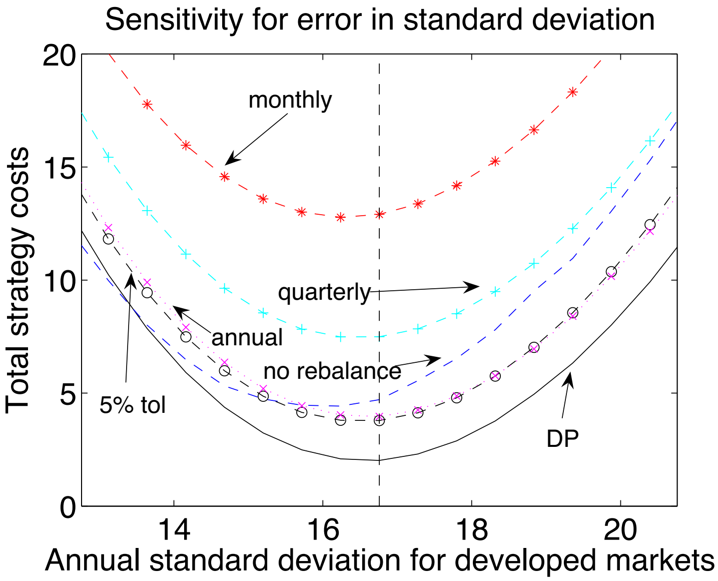

Fig. 3. Sensitivity analysis on standard deviation for the two asset example. The vertical line indicates the actual rate used by the rebalancing strategies.

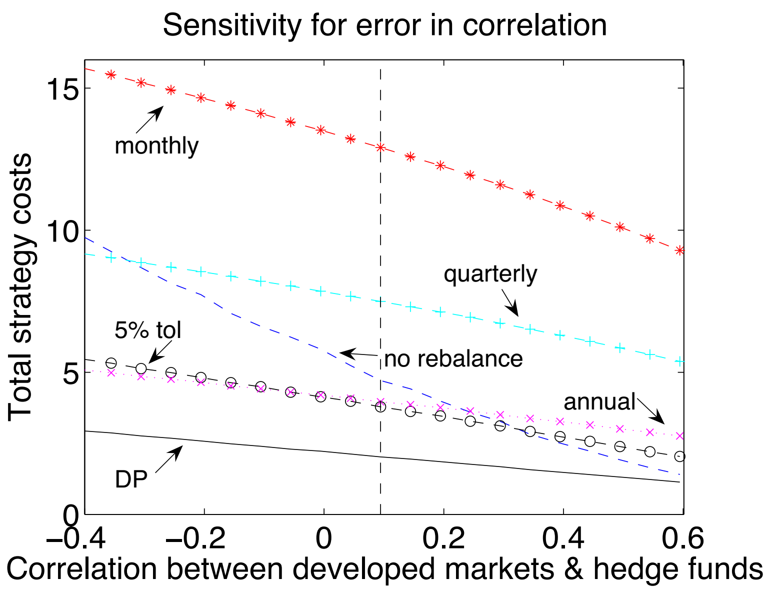

Fig. 4. Sensitivity analysis on correlation for the two asset example. The vertical line indicates the actual rate used by the rebalancing strategies.

error in μ, Σ not built in DP. but analyzed as sensitivity analysis.
DP rebalance less, and least amount, more robust to error.

then compute annualized aggregate cost while the true value of the third parameter varies around the assumed value. This cost is computed relative to an idealized portfolio which knows the correct value and rebalances for free to the true optimal portfolio every month.

Most rebalancing strategies are inherently model-based because the target portfolio is determined by the model. When the target portfolio is determined with incorrect parameters, then the target portfolio is itself suboptimal. Therefore we would expect all of the rebalancing methods to be sensitive to model error. The exception to this among the rebalancing methods we examine is the strategy of no rebalancing. For our testing methodology, the assumed model plays no role except in determining the initial portfolio. Therefore we would expect no rebalancing to be least sensitive to model error. Our algorithm should be even more strongly influenced by the model because incorrect assumptions can also result in suboptimal rebalancing decisions from the dynamic program. Other methods such as calendar and tolerance-band strategies are heuristic and do not directly rely on the model when forming the rebalancing policy.

The results for mean sensitivity are shown in Figure 2. For each point, 10,000 sequences of twenty years of monthly returns were generated, and the performance of each rebalancing strategy was averaged over each sequence. Our dynamic rebalancing strategy was as robust as tolerance-band and calendar rebalancing. We performed better than no rebalancing until the annual return exceeded 7.6%. This is because the annual return for Developed Markets is higher than that of Hedge Funds. So without rebalancing, as the annual return for Developed Markets increases, the average weight for Developed Markets tends to increase as well. At the same time, it becomes more advantageous to hold Developed Markets due to the higher return, so the true optimal portfolio will contain a higher percentage of Developed Markets. Thus in this case, a passive strategy will typically have the portfolio inadvertently move towards the optimal portfolio.

From Figure 3, we see that the dynamic programming approach again outperforms the other approaches even if there are large errors in estimating the standard deviation – it still outperforms the other methods even for inaccuracies in the standard deviation of several percentage points per year. In this instance, the strategy of no rebalancing performs better for lower standard deviations. Similarly to the mean sensitivity case, because Developed Markets has a higher average return than Hedge Funds, its weight tends to increase with time. So as the standard deviation for Developed Markets falls, the true optimal portfolio will again have a higher weight in Developed Markets. So the portfolio will again tend to drift towards the optimal portfolio. Finally, in Figure 4, we observe that the dynamic programming approach is relatively insensitive to errors in estimation of the correlations between the assets. Since the cost does not change much for different true correlations, we conclude that correlations need not be accurately estimated for the purposes of our approach.

C. *Multi-Asset Model*

Now that we have described and analyzed the simple two-asset model, we proceed to examine the general case of N risky assets. In this section, we consider the case of five risky assets and assert that another choice of $N > 2$ would proceed similarly with the main difference being computation time. big time! It is known [6] that standard mean-variance portfolio optimization produces optimal portfolios only if returns are normally distributed or if quadratic utility is assumed. Otherwise, full-scale optimization must be performed to compute optimal portfolios when using more advanced utility functions such as log wealth or power utility. However, recent work by Cremers *et al.* [7] indicates that except when returns are highly non-normal, it is sufficient to perform mean-variance optimization on a Markowitz-style approximate expected utility function (see Section II-A) in terms of just the mean and standard deviation. They show that the performance of the resulting portfolios and the performance of those generated from full-scale optimization do not differ significantly. When performing this approximate mean-variance optimization,

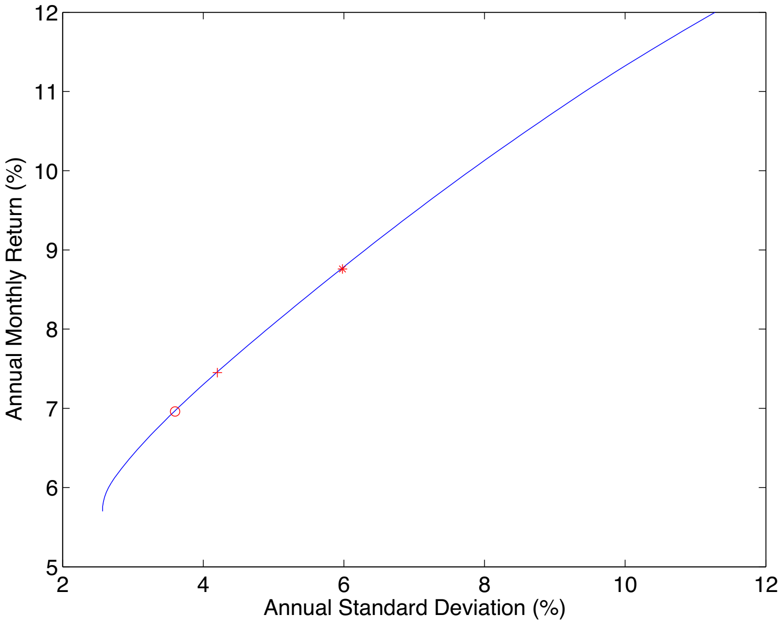

Fig. 5. Efficient frontier and optimal portfolios for the different utility functions discussed. (o) indicates the optimal portfolio for power utility. (+) indicates that for quadratic utility with $\alpha = 1.5$. (*) indicates the optimal portfolio for log wealth utility.

the optimal portfolio lies on the efficient frontier[2] [12]. Therefore, to construct optimal portfolios for different utility functions, we first compute the efficient frontier by solving a quadratic programming problem and then search over those portfolios to find the one with the highest expected utility.

	Quadratic (α=1.5)	Logarithmic	Power
US Equity	0.194	0.160	0.210
Developed Market Equity	0.222	0.240	0.213
Emerging Market Equity	0.185	0.275	0.143
Private Equity	0.156	0.292	0.093
Hedge Funds	0.243	0.033	0.341

TABLE V

OPTIMAL PORTFOLIO WEIGHTS FOR DIFFERENT UTILITY FUNCTIONS $U(\mu, \sigma)$.

[2] Risk-averse expected utility functions are monotonically increasing in terms of return and monotonically decreasing in terms of risk. Hence if a portfolio is not on the efficient frontier, there exists a portfolio with equivalent return and less risk or more return and the same risk. Therefore this portfolio cannot be optimal.

Figure 5 displays the efficient frontier for the five asset classes when short sales are not allowed. Searching over this frontier for each of the utility functions results in the optimal portfolios as indicated in the figure with the weights shown in Table V. These weights are the optimal weights we use throughout our analysis. Note that power utility is the most risk-averse utility and log wealth is the risk-seeking utility.

1) Monte Carlo Simulations: Table VI shows the results of various rebalancing algorithms for the five-asset case. The results were generated in an analogous manner to the two-asset case with Monte Carlo simulations over 20 years for 10,000 sample paths. Trading costs were 60 bps for Hedge Funds, Emerging Markets, and Private Equity; 40 bps for Developed Markets; and 30 bps for US Equity.

For the quadratic utility case, we see that our optimal DP method performs approximately 30% better than the next-best method, 5% tolerance bands. If we examine the costs, we see, as expected, that monthly trading incurs no suboptimality at the expense of high trading costs. The other extreme of no trading incurs an extremely large suboptimality cost because over a twenty year period, assets can become quite unbalanced if unadjusted.

For power utility, our expected loss is 24% less than the runner-up, 5% tolerance band rebalancing. The benefits for this method are reduced from the quadratic utility case primarily because power utility is more risk averse, so the optimal portfolio has a lower variance. Therefore less rebalancing is needed overall because the portfolio simply does not move as much. This can be seen in the quarterly and annual rebalancing methods which trade less and suffer smaller suboptimality costs while doing so.

For the log wealth utility case, our algorithm results in expected loss 35% less than the best alternative, annual rebalancing in this case. The 5% tolerance method falls short in this simply because the log wealth portfolio is a high variance portfolio. In the quadratic case, the trading costs are only marginally higher than the annual rebalance method. But in the log wealth case, they are 48% higher because the tolerance bands are breached more often. As a general rule, we see it is more important to get the rebalancing right when dealing with higher-variance portfolios simply because many more rebalancing opportunities arise.

Note that even though tolerance bands do better than annual rebalancing for quadratic utility and power utility and worse for log wealth utility, it is not a valid conclusion that tolerance band methods perform better for low variance portfolios and calendar rebalancing methods perform better for high variance portfolios. There is a degree of freedom in each algorithm. For tolerance bands, the bands can be loosened or tightened depending on the variance of the optimal portfolio. For example, in the log wealth case, it is clear that the tolerance bands are too tight because of the sheer imbalance between trading costs and suboptimality costs. For calendar algorithms, the period between rebalancings can be

		(a)	(b)	(c)	(d)
		Trading Cost (bps)	Suboptimality Cost (bps)	Aggregate Cost (bps)	Utility Shortfall (utils x 10^4)
Quadratic $\alpha = 1.5$	Ideal	0.00	0.00	0.00	0.00
	Optimal DP	3.97	1.49	5.47	5.23
	No Trading	0.00	30.18	30.18	32.89
	5% Tolerance	7.29	0.70	7.99	7.88
	Monthly	23.67	0.00	23.67	23.73
	Quarterly	13.69	0.28	13.96	14.21
	Annual	6.84	1.55	8.39	8.43
Power	Ideal	0.00	0.00	0.00	0.00
	Optimal DP	3.39	1.15	4.54	4.32
	No Trading	0.00	25.99	25.99	26.04
	5% Tolerance	5.19	0.82	6.01	5.95
	Monthly	20.05	0.00	20.05	19.96
	Quarterly	11.59	0.18	11.78	11.87
	Annual	5.81	1.02	6.84	6.80
Log Wealth	Ideal	0.00	0.00	0.00	0.00
	Optimal DP	4.80	1.93	6.72	6.57
	No Trading	0.00	30.52	30.52	32.01
	5% Tolerance	11.95	0.43	12.38	12.72
	Monthly	28.15	0.00	28.15	28.19
	Quarterly	16.27	0.40	16.67	16.80
	Annual	8.05	2.17	10.22	10.65

TABLE VI

QUADRATIC ($\alpha = 1.5$), POWER, AND LOG WEALTH UTILITY: ANNUALIZED TRADING COST, NON-OPTIMAL UTILITY COST, AND AGGREGATE COST USING SIX DIFFERENT TRADING STRATEGIES ON FIVE RISKY ASSETS SIMULATED OVER A TWENTY YEAR PERIOD 10,000 TIMES.

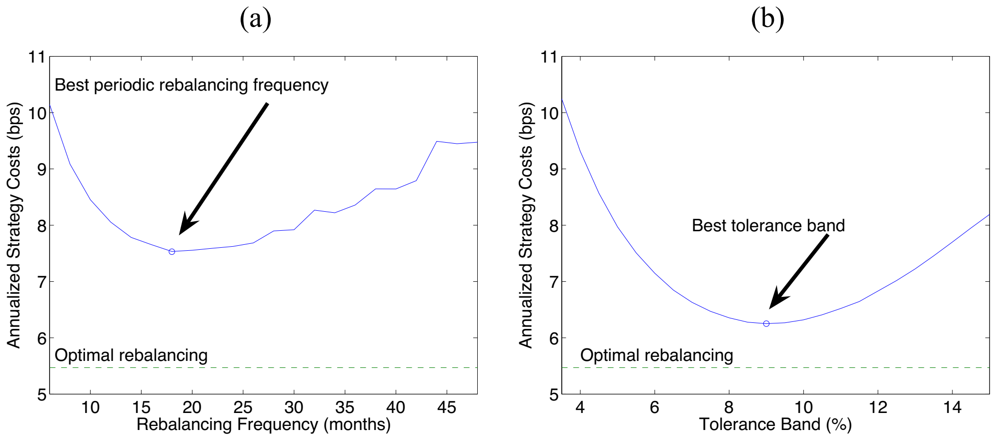

Fig. 6. Annualized strategy costs for different calendar-based and tolerance band settings using the quadratic utility with $\alpha = 1.5$. (a) Plot of costs as a function of different periodic rebalancing periods. (b) Plot of costs as a function of different tolerance band windows.

adjusted. For instance, setting the rebalancing time to two years for the power utility case results in an expected loss of 6.32 bps per annum, a savings of 0.52 bps over the annual strategy. This is achieved by accruing more than twice as much expected tracking error (2.21 bps versus 1.02 bps) but also reducing trading costs by 29% (4.11 bps versus 5.81 bps). A more exhaustive search of possible fixed-interval rebalancing strategies could presumably yield an even better result.

2) Optimal Heuristic Algorithms: Previously, we simply chose a 5% tolerance and periodic rebalancing periods of monthly, quarterly, and annually. These specific choices are a subset of general tolerance band and calendar-based rebalancing algorithms. For our comparison, we made no claims that the parameters chosen were optimal but rather chose settings that seemed to be common in the literature. The work here can be thought of in two ways. First, we show that our method is superior to any tolerance band or calendar-based rebalancing method. It should be noted that our method can be thought of as a dynamic tolerance band approach. Fixed-tolerance methods are a subset of the controls available to our algorithm, so they can never do better. Second, the full dynamic programming algorithm is computationally intensive, especially with large numbers of assets. If we instead restrict the search over a smaller subset of acceptable policies, we have a strategy that is not necessarily optimal, but perhaps good enough.

In Figure 6, we present results for the quadratic utility case. Each point was obtained by performing Monte Carlo simulations over 20 years for 10,000 realizations. In Figure 6(a), we plot the total costs as a function of the number of months between rebalancings. The lack of smoothness in the curve, particularly between less frequent rebalancings, arises because the number of rebalancings in a 240 month period

	(a)	(b)	(c)	(d)
	Trading Cost (bps)	Suboptimality Cost (bps)	Aggregate Cost (bps)	Utility Shortfall (utils x 10^4)
Ideal	0.00	0.00	0.00	0.00
Optimal DP	2.60	0.85	3.45	3.75
No Trading	0.00	30.18	30.18	32.89
5% Tolerance	3.64	0.70	4.34	4.53
Monthly	11.83	0.00	11.83	11.86
Quarterly	6.84	0.28	7.12	7.18
Annual	3.42	1.55	4.97	5.15

TABLE VII

QUADRATIC UTILITY ($\alpha = 1.5$): TRADING COSTS, NON-OPTIMAL UTILITY COSTS, AND AGGREGATE COST USING SIX DIFFERENT TRADING STRATEGIES ON FIVE RISKY ASSETS SIMULATED OVER A TWENTY YEAR PERIOD 10,000 TIMES. TRANSACTION COSTS ARE HALVED FROM THE PREVIOUS EXPERIMENTS.

must necessarily change in discrete steps. We see that the best performance occurs for 18 month periods at a cost of 7.53 bps. Note, this is still over 2 bps worse than our algorithm's performance (shown as a horizontal line) but is nearly 1 basis point better than annual rebalancing. In Figure 6(b), we vary the tolerance setting that triggers rebalancing. We find that the best setting occurs for a tolerance band of 9% and results in a cost of 6.25 bps, again worse than the 5.47 bps obtained from our algorithm.

3) Computational Complexity: To provide some information regarding the computational complexity of our approach, we first state that we allow on the order of 15 possible weights for each asset. For five assets, we have an observation space of approximately 750,000 points (we must develop the optimal policy for each point). Our current implementation processes around 600,000 points per hour on a single PC[3]. Thus, the run-time estimate for five assets is 75 minutes. If we assume the possibility of M different weights for an additional asset, the addition of this asset into our N-asset model would increase computation by a factor of M. Memory requirements increase by a similar amount. Note that this is detailing the computation for learning the optimal policy. Once that is done, actually applying the policy is very fast.

Exponential Computing Time.

[3] Value iteration can be easily parallelized, so the total processing time also depends on the number of machines available.

4) Alternate Cost Functions: Before we complete this section, we address the possibility of a different trading cost function. In particular, while the numbers used are consistent with trading costs cited in other research papers [9], some may wonder if the results would be different for alternate trading costs. Table VII shows the results when we reduce the proportional trading costs in half and apply it to the quadratic utility strategy. 5% tolerance bands remains the next best strategy, but our advantage has narrowed from 30% down to 20%. The reason for this is that the other algorithms trade too much, and now they're being penalized less for it.

Transaction costs for the other methods are cut in half, while suboptimality remains the same. This produces reductions in aggregate cost ranging from 41% for annual rebalancing to 50% for monthly rebalancing (ignoring no rebalancing which obviously does not benefit). In contrast, our algorithm actually reduces trading costs by less than half. This means that we are sensibly trading more now that transaction costs have been lowered. So our algorithm automatically trades off a little extra trading to reduce the suboptimality costs by a greater amount.

V. Conclusion

The *ad hoc* methods of periodic and tolerance band rebalancing provide simple but suboptimal ways to rebalance portfolios. Calendar-based approaches rely on the fact that, on average, we expect the portfolio to become less and less optimal as time goes on but they do not use any knowledge about the actual state of the portfolio. The tolerance band approach uses the current portfolio to make a decision, but this method has no sense of the proper tolerance band setting, or even how large this band should be. In this work, we have shown that by formulating the rebalancing problem as an optimization problem and solving it using dynamic programming, we reduce the overall costs of portfolio rebalancing. We have demonstrated that the reduced costs hold for different investor risk preferences. Namely, we have compared the performance of our technique with others for quadratic, log wealth, and power utility functions.

The costs of transactions are much more tangible than the cost of being suboptimal. However, through the use of certainty equivalents, we have provided a method that quantifies the cost of being suboptimal. Our simulations have confirmed that this optimal method provides gains over the best of the traditional techniques of rebalancing.

It is worth noting that in our analysis we assume that returns are independent across different intervals. It has been discussed in the literature that mean reversion may exist. Under such circumstances, we expect our method to perform even better when compared with periodic rebalancing because our algorithm would likely rebalance with even less frequency.

Several extensions exist from our work. First, we may want to consider affine transaction costs. This model is appropriate if we believe that there is a fixed cost to making each and every transaction. Such an adjustment would likely favor dynamic trading methods over periodic rebalancing. Next, we may want to examine rebalancing over taxable portfolios. Asset managers of such funds have the additional consideration of tax consequences when a decision to transact needs to be made. The relaxation of the short sales constraint is another possible extension to the work. Although many tax-deferred funds do not allow short sales, several either explicitly allow short selling or implicitly participate in short sales through investments into hedge funds.

In our work, we assume an instantaneous rebalancing at the end of each month. We may want to incorporate more general trading models which consider the effects of price impact. Finally, for the multi-asset case, we search a one-dimensional policy space representing portfolios which are a linear combination of the current portfolio and the target portfolio. We ideally want to search over the entire space of possible portfolios around the optimal portfolio. This would be particularly useful when trading costs have a fixed component. In these situations, it may be better to trade on only a subset of assets rather than a portion of all asset classes.

Acknowledgments

The authors would like to thank Sebastien Page (State Street Associates) and Mark Kritzman (Windham Capital Management) for introducing us to this problem and for their valuable guidance and advice. In addition, we note that this paper originated from a project performed for a course at the Massachusetts Institute of Technology's Sloan School of Management entitled *Proseminar in Financial Engineering*. We also want to thank Ed Freyfogle and Josh Grover for their input.

Appendix I
Derivation of Expected Utility Functions

Figure 7 plots the three empirical utility functions as a function of return. The relative difference, not the absolute value, in utility for different x is what is important (we could scale the utility functions without affecting the corresponding optimal portfolio).

Quadratic utility is a commonly used function, and using it in portfolio construction is akin to doing standard mean-variance optimization. Regardless of whether or not the assets are Gaussian-distributed, the expected utility only involves the first two moments, so any higher order moments are ignored. The α parameter can be adjusted to indicate risk tolerance. A larger number indicates that an investor is

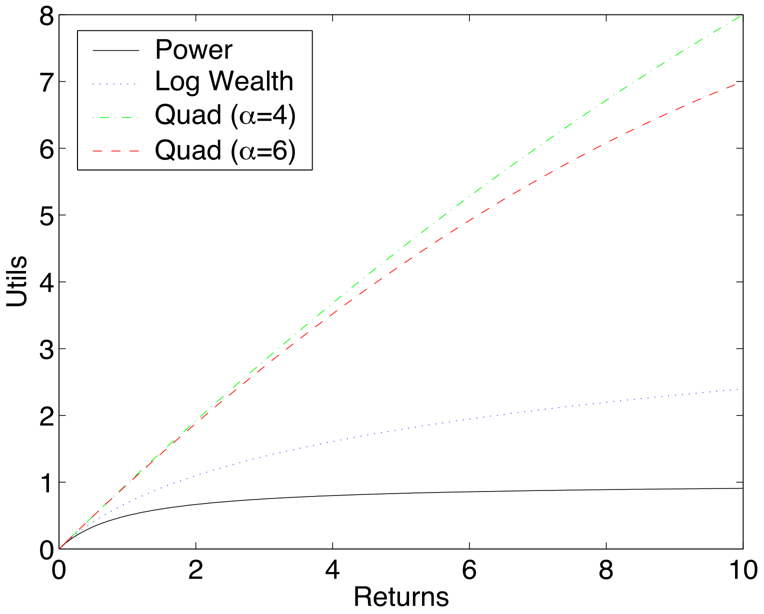

Fig. 7. Plots of the quadratic utility (for two different α's), log wealth utility, and power utility as a function of returns.

more risk averse. One main difficulty with quadratic utility is that it has the odd behavior that for a large enough return, it is *too* risk averse and the utility function actually prefers a smaller return because $\lim_{x \to \infty} f_q(x) = -\infty$. This behavior begins at $x = x_0 + \frac{1}{\alpha}$, the maximum of the quadratic function. The log wealth and power utility functions do not exhibit this behavior.

Even though it is true that the expected utility can be written just in terms of the mean and variance, the expression shown for $U_q(\mu, \sigma)$ is only an approximation. The true expression is:

$$\mathrm{U}_q(\mu, \sigma) = \mu - \frac{\alpha}{2}(\sigma^2 + (\mu - x_0)^2). \tag{7}$$

Note that if we had *a priori* knowledge of the portfolio return μ, we would just choose $x_0 = \mu$. Unfortunately, μ is a function of the portfolio weights w, so we cannot fix it ahead of time. In the typical operating regime, $\mu(w) \approx \mu(w^*)$ because we would typically rebalance before the portfolios become too unbalanced. If we choose $x_0 = \mu(w^*)$, then the $(\mu - x_0)^2$ term in (7) is small, and U_q as a reasonable approximation to the true expected utility. This leaves a much simpler expected utility function (especially in terms of μ).

The derivation of the expected utility functions for log wealth and power is non-obvious. Let's examine log wealth utility. We can expand the utility function around the point $x = \mu$ using a Taylor series:

$$\begin{aligned} f_l(x) = \log(1+x) &= \log(1+\mu) + \frac{1}{1!} f_l'(1+\mu)(x-\mu) + \frac{1}{2!} f_l''(1+\mu)(x-\mu)^2 + \cdots \\ &\approx \log(1+\mu) + \frac{x-\mu}{1+\mu} - \frac{(x-\mu)^2}{2(1+\mu)^2}. \end{aligned} \tag{8}$$

Thus we see that

$$\begin{aligned} U_l(\mu,\sigma) &= E[\log(1+x)] \\ &\approx E\left[\log(1+\mu) + \frac{x-\mu}{1+\mu} - \frac{(x-\mu)^2}{2(1+\mu)^2}\right] \\ &= \log(1+\mu) - \frac{\sigma^2}{2(1+\mu)^2}. \end{aligned}$$

Additional terms of the Taylor expansion may be used to improve the approximation. These will then involve the skewness and the kurtosis and other higher-order moments. A similar method is applied to derive the approximation for power utility.

APPENDIX II
EFFICIENT FRONTIER USING MEAN-VARIANCE OPTIMIZATION

Computing mean-variance efficient frontiers is a relatively straightforward process. This is an essential part of computing optimal portfolios for the non-normal returns or non-quadratic utility cases in order to avoid using full-scale optimization. We solve a series of quadratic programs [18], each minimizing the variance for a given expected portfolio return μ_p. Because we did not allow short sales, the optimization problem has the following form:

$$\begin{array}{ll} \min_w & w'\Lambda w \\ \text{s.t.} & w'\mu = \mu_p, \quad \sum_i w_i = 1, \quad w \geq 0, \end{array} \tag{9}$$

where w are the unknown portfolio weights, Λ is the covariance matrix of the available assets and μ is the vector of expected asset returns. This optimization can be efficiently performed using Matlab's `Quadprog.m` function. For the quadratic utility function, it is not necessary to compute the entire efficient frontier. The optimal weights can directly be determined by solving a different quadratic program:

$$\begin{array}{ll} \max_w & w'\mu - \frac{\alpha}{2} w'\Lambda w \\ \text{s.t.} & \sum_i w_i = 1, w \geq 0. \end{array} \tag{10}$$

References

[1] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[2] R. Bellman and S. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, Princeton, NJ, 1962.

[3] D. Bernoulli. Exposition of a new theory on the measurement of risk. *Econometrica*, 22(1):23–36, 1954.

[4] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA, 2000.

[5] R. Brealey and S. Myers. *Principles of Corporate Finance, 5th edition*. McGraw-Hill Companies, New York, 1996.

[6] J. Cremers, M. Kritzman, and S. Page. Portfolio formation with higher moments and plausible utility. *Revere Street Working Paper Series, Financial Economics*, 272(12), 2003.

[7] J. Cremers, M. Kritzman, and S. Page. Optimal hedge fund allocations: Do higher moments matter? *Revere Street Working Paper Series, Financial Economics*, 272(13), 2004.

[8] C. Donahue and K. Yip. Optimal portfolio rebalancing with transaction costs. *Journal of Portfolio Management*, Summer:49–63, 2003.

[9] H. E. Leland. Optimal portfolio management with transaction costs and capital gains taxes. *Haas School of Business Technical Report, UC Berkeley*, Dec, 1999.

[10] H. Levy and H. M. Markowitz. Approximating expected utility by a function of mean and variance. *American Economic Review*, 69(3):308–317, 1979.

[11] R. Duncan Luce. *Utility of Gains and Losses: Measurement-Theoretical and Experimental Approaches*. Lawrence Erlbaum Associates, Inc., 2000.

[12] H. M. Markowitz. Portfolio selection. *Journal of Finance*, 1952.

[13] S. J. Masters. Rebalancing. *Journal of Portfolio Management*, Spring:52–57, 2003.

[14] J. E. Mitchell and S. Braun. Rebalancing an investment portfolio in the presence of transaction costs. *Working Paper, Department of Rensselaer Polytechnic Institute*, Nov, 2002.

[15] J. M. Mulvey and K. D. Simsek. *Rebalancing Strategies for Long-term Investors*, pages 15–33. Kluwer Academic Publishers, New York, 2002.

[16] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Inc., New York, 1991.

[17] A. F. Perold and W. F. Sharpe. Dynamic strategies for asset allocation. *Financial Analysts Journal*, Jan-Feb:149–160, 1995.

[18] D. P. Bertsekas with A. Nedic and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, Belmont, MA, 2003.