Optimal Rebalancing:
A Scalable Solution

Mark Kritzman
Windham Capital Management, LLC, 5 Revere Street, Cambridge, MA 02138,
mkritzman@windhamcapital.com
617-234-9410

Simon Myrgren
State Street Associates, 138 Mt. Auburn Street, Cambridge, MA 02138,
smyrgren@statestreet.com
617-234-9416

Sébastien Page
State Street Associates, 138 Mt. Auburn Street, Cambridge, MA 02138,
spage@statestreet.com
617-234-9462

Institutional investors usually employ mean-variance analysis to determine optimal portfolio weights. Almost immediately upon implementation, however, the portfolio's weights become sub-optimal as changes in asset prices cause the portfolio to drift away from the optimal targets. We apply a quadratic heuristic to address the optimal rebalancing problem, and we compare it to a dynamic programming solution as well as to standard industry heuristics. The quadratic heuristic provides solutions that are remarkably close to the dynamic programming solution. Moreover, unlike the dynamic programming solution, the quadratic heuristic is scalable to as many as several hundreds assets.

This version: June 9, 2008

We thank Harry Markowitz and Krishna Ramaswamy for helpful comments.

Portfolio Rebalancing: A Scalable Solution

Abstract: Institutional investors usually employ mean-variance analysis to determine optimal portfolio weights. Almost immediately upon implementation, however, the portfolio's weights become sub-optimal as changes in asset prices cause the portfolio to drift away from the optimal targets. We apply a quadratic heuristic to address the optimal rebalancing problem, and we compare it to a dynamic programming solution as well as to standard industry heuristics. The quadratic heuristic provides solutions that are remarkably close to the dynamic programming solution. Moreover, unlike the dynamic programming solution, the quadratic heuristic is scalable to as many as several hundreds assets.

I. Introduction

Institutional investors usually employ mean-variance analysis to determine optimal portfolio weights. Almost immediately upon implementation, however, the portfolio's weights become sub-optimal as changes in asset prices cause the portfolio to drift away from the optimal targets. In an idealized world without transaction costs investors would rebalance continually to the optimal weights. In the presence of transaction costs investors must trade off the cost of sub-optimality with the cost of restoring the optimal weights. Most investors employ heuristics that rebalance the portfolio as a function of the passage of time or the size of the misallocation. Sun et al. (2006) employ dynamic programming to determine optimal rebalancing rules, and they demonstrate that their approach is significantly superior to standard industry heuristics.

Their approach is seriously limited, however, because it does not scale beyond a few assets. It suffers from the curse of dimensionality.

Markowitz and van Dijk (2004) present a quadratic heuristic for rebalancing a portfolio to capture shifting views about the mean returns of portfolio assets. It has been shown previously that we can closely approximate a variety of utility functions with quadratic functions (see, for example: Levy and Markowtiz (1979), Kroll, Levy and Markowtiz (1984), Cremers, Kritzman and Page (2003), Cremers, Kritzman and Page (2005)).

We adapt the Markowitz-van Dijk heuristic to address the asset weight drift problem, and we compare its solution to the unscalable dynamic programming solution as well as to solutions based on standard industry heuristics. Our tests reveal that the Markowitz-van Dijk heuristic provides solutions that are remarkably close to the dynamic programming solutions for those cases in which dynamic programming is feasible and far superior to solutions based on standard industry heuristics. In the case of five or more assets, in fact, it performs better than dynamic programming due approximations required to implement the dynamic programming algorithm. Moreover, unlike the dynamic programming solution, the Markowitz-van Dijk heuristic is scalable to as many as several hundred assets.

II. The General Portfolio Rebalancing Problem

We begin by assuming an investor with log-wealth utility wishes to select a set of portfolio weights that maximize expected utility over a forthcoming period. The expected utility *E(U)* of the portfolio is written as the weighted sum of the *n* security expected returns under *m* scenarios, each with associated *p* probability

(1) $$E(U) = \sum_{i=1}^{m} p_i \ln\left(1 + \sum_{j=1}^{n} X_j \mu_{ij}\right) = p \ln(1 + \mu X')$$

where

$$\mu = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1n} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2n} \\ & & \vdots & \\ \mu_{m1} & \mu_{m2} & \cdots & \mu_{mn} \end{bmatrix}$$

is the matrix of expected returns, $X = [X_1, \ldots, X_n]$ are the current portfolio weights in percentage, and $p = [p_1, \ldots, p_m]$ are the probabilities associated with the *m* scenarios. Let X^{opt}, $X^{opt} = [X_1^{opt}, \ldots, X_n^{opt}]$, denote the optimal portfolio weights. *E(U)* is then maximized when $X = X^{opt}$ and denoted *E(U*)*. With the passage of time asset prices change, and *X* deviates from X^{opt} resulting in a loss of expected utility. For a given sub-optimal *E(U)*, we quantify the loss in expected utility as the certainty equivalent cost (CEC), which for the log wealth investor is given by:

(2) $$CEC = e^{E(U^*)} - e^{E(U)}$$

Doing so conveniently converts the portfolio's sub-optimality cost into units that are directly comparable to transaction costs.

The transaction costs (*TC*) at period *t* are written as

(3) $$TC_t = \sum_{i=1}^{n} C_j \left| X_{jt} - X_{jt-1} \right|$$

where C_j is the cost per unit of trading security *j* from the previous weights X_{jt-1} to the new weights X_{jt}.

The general portfolio rebalancing problem is therefore to minimize the combined costs associated with deviations from X^{opt} as defined in (2) while also minimizing transaction costs as defined in (3). Rebalancing decisions in the current period will influence future costs and decisions, which must be accounted for in the optimal solution.

III. The Dynamic Programming Solution

Bellman (1952) introduced dynamic programming in the same year that Markowitz published his landmark article on portfolio selection. Dynamic programming provides solutions to multistage decision processes and is used in a diverse set of applications including automatic sign language recognition, hydropower optimization, sequential bidding in auctions, ecological management, and robotics, to mention just a few.[1]

Following Sun et al. (2006), we apply dynamic programming to determine when and how to rebalance an institutional portfolio. We define the dynamic programming solution to portfolio rebalancing as the recursive minimization of the cost function

(4) $$J_t\left(X_t, X_{t-1}\right) = Min\left\{CEC_t + TC_t + J_{t+1}\left(X_{t+1}, X_t\right)\right\}$$

where the total cost for the current period $J_t(X_t, X_{t-1})$, is a function of the current CEC and TC, but also of future costs $J_{t+1}(X_{t+1}, X_t)$. In our experiments, we estimate the potential future cost of each decision as the discounted average cost across 50 potential allocations randomly generate by Monte Carlo sampling.

Unfortunately, this approach suffers from the curse of dimensionality. To rebalance a portfolio among three assets in increments of 1%, for example, we must consider 5,151 possible portfolios[2] and analyze 26,532,801 ($5{,}151^2$) rebalancing decisions for each period. Moreover, to solve this problem recursively we must generate at least 50 Monte Carlo paths for each possible decision at each time step. For a one-year horizon with monthly monitoring (12 time steps), we must therefore perform 14,619,573,351 ($5{,}151^2$ x 50 x 11 + $5{,}151^2$) calculations. Table 1 shows how the number of portfolios and the number of calculations grow as we add more assets.

[Insert Table 1 here]

In our experiment we use a 28-processor grid computing platform. Grid computing relies on parallel processing to allocate process execution efficiently, thus enabling faster processing of large-scale computation problems. Even with access to a grid computer, deriving the optimal decisions associated with a 10 asset portfolio and a choice of 1% granularity is computationally intractable. On a regular workstation, for example, the computing time required to solve this problem would be nearly 12,000 times of times the age of the universe.

IV. The Markowitz and van Dijk Heuristic

Table 1 underscores the limitations of dynamic programming when we wish to consider more than a few assets. Markowitz and van Dijk (2004) propose an alternative approach for determining optimal rebalancing rules. Although they apply their heuristic to account for changing means in asset returns, we adapt it to address the asset weight drift problem.

As with the dynamic programming approach, we wish to minimize the combined costs of sub-optimality and rebalancing, taking into account the current period's costs as well as the discounted expected costs of future choices. However, we replace $J_t\left(X_{t+1}, X_t\right)$ in (4) by a quadratic function of the current and optimal portfolio weights. In general, a quadratic approximation Q to $J_t\left(X_{t+1}, X_t\right)$ has the following form:

(5) $$Q = \sum a_i X_i + \sum b_i^2 X_i^2 + \sum_i \sum_{j>i} c_{ij} X_i X_j$$

To simplify our experiments, however, we conjecture that Q is separable and is minimized by the target portfolio, so that it is proportional to the squared deviations (the "drifts") multiplied by a coefficient *d*:

(6) $$Q = d \sum_{i=1}^{n} \left(X_i - X^{opt\prime} \right)^2$$

The cost function (4) then becomes

(7) $$J_t\left(X_t, X_{t-1}\right) = CEC_t + TC_t + Q_t$$

To determine the value of coefficient *d* in (6) we use Monte Carlo simulations. We generate 200 return paths. We minimize cost as defined in (7)[3] at each decision point during the simulation. We continue to run simulations and change *d* until we find its best performing value. Computational intensity, which is low to begin with, remains manageable as we add more assets[4].

V. Results

We test the relative efficacy of dynamic programming and the MvD heuristic with data on domestic equities, domestic fixed income, non-US equities, non-US fixed income, and emerging market equities. For these portfolios the expected portfolio return is

(8) $$E_p = \sum_{i=1}^{n} X_i \mu_i = X\mu'$$

and the expected portfolio variance is

(9) $$V_p = \sum_{i=1}^{n}\sum_{j=1}^{n} X_i X_j \sigma_{ij} = XCX'$$

where $X = [X_1, \ldots, X_n]$ is the set of asset weights, $\mu = [\mu_1, \ldots, \mu_n]$ is the set of expected returns on the n assets, σ_{ij} is the covariance between assets i and j, and C is the covariance matrix (σ_{ij}).

Table 2 shows our returns, standard deviations, and transaction cost assumptions.

[Insert Table 2 here]

Table 3 shows our correlation assumptions. We use monthly returns from October, 2001 through September, 2006 to measure standard deviations and correlations. To estimate expected returns we solve for the implied returns under the assumption that the allocations in Table 4 are optimal under mean-variance utility and a fully invested budget constraint:

(10)
$$E(U) = X\mu' - \frac{\lambda}{2} XCX' \quad \text{s.t. } X1_N' = 1$$

Here λ is the risk aversion parameter (7.5) and 1_N is a vector of ones. We thus calculate the implied returns as follows:

(11)
$$\mu_{impl} = \lambda CX' + \frac{-\lambda + 1_N C^{-1} \mu'}{1_N C^{-1} 1_N'} 1_N'$$

[Insert Table 3 here]

We use domestic stocks and domestic fixed income for the two-asset case. We add non-US equities for the three-asset case, non-US fixed income for the four-asset case, and emerging market equities for the five-asset case. Table 4 shows the assumed optimal portfolio weights, which as stated before are optimal under the standard mean-variance utility function. The choice of the initial portfolio weights is arbitrary. In our example, we use optimal portfolios based on a set of reasonable expectations and a mean-variance utility function[5]. We could just as well substitute optimal weights based on other descriptions of expected utility. Long-only investment managers, for example, would rely on a mean-tracking error utility function, while behavioral investors might use an s-shaped value function. Investors mostly concerned with large loses would use a kinked utility function (see Cremers, Kritzman, and Page, 2005). Also, the assumption that returns are normally distributed is convenient but not necessary for optimal rebalancing – as long as the distributions can be generated via Monte Carlo simulations.

[Insert Table 4 here]

We assume that we have a two-year investment horizon over which we wish to minimize the aggregate total cost; that is, the cumulative sum of trading costs and sub-optimality costs. For the calendar heuristics, we fully rebalance the portfolio at pre-determined time intervals. For the tolerance band heuristics, we fully rebalance the portfolio when asset weights breach pre-determined thresholds of 0.25, 0.5, 0.75, 1, 2, 3, 4, and 5 percent. Although we cannot extend the dynamic programming algorithm beyond five assets, we test the MvD heuristic and the other heuristics for portfolios of 10, 25, 50, and 100 assets using individual stocks, which are listed in the appendix.

As indicated in section 3, for each portfolio we sample several hundred values for *d* until we find the *d* which yields the lowest average figure of merit (AFOM)[6], which we define as the average total cost over the 200 Monte Carlo runs:

$$AFOM = \frac{1}{200}\sum_{i=1}^{200}\sum_{p=1}^{24} CEC_{i,p}^{MV} + TC_{i,p} \tag{12}$$

where $CEC_{i,p}^{MV}$ is the certainty equivalent cost for the ith portfolio path in the pth period under mean-variance utility and $TC_{i,p}$ are the transaction costs (3) for the ith portfolio path in the pth period.

Table 5 summarizes the results. It shows that the MvD heuristic performs quite well compared to the dynamic programming solution for the two asset case and substantially better than other heuristics[7]. As we increase the number of assets we find that the advantage of dynamic programming over the MvD heuristic shrinks and is reversed at five assets. We are not able to apply dynamic programming beyond five assets, but we are able to extend the MvD heuristic up to 100 assets. We find that the MvD heuristic reduces total costs relative to all of the other heuristics by substantial amounts. In the appendix we present a more detailed cost analysis that partitions costs into trading and sub-optimality components.

[Insert Table 5 here]

Although the performance of the MvD heuristic improves relative to the dynamic programming solution as more assets are added, this improvement reflects a growing reliance on approximation for the dynamic programming approach. For the two-asset case the dynamic programming solution searches within an interval of plus or minus 5% around the optimal portfolio, and divides this range into 5,000 units. For greater than two assets, the search is confined to plus or minus 3% around the optimal portfolio, and this space is divided into increasingly coarser units, as shown in Table 6.

[Insert Table 6 here]

We have no way of knowing how well the MvD heuristic would track the ideal but unobtainable dynamic programming solution, but we are encouraged that its advantage over the next best heuristic increases as we add more assets. Moreover, we would not know ex ante which heuristic is the next best; hence a fairer assessment of the relative efficacy of the MvD heuristic might be to compare it to the average result of the other heuristics.

Part VI. Conclusion

Portfolio allocations drift from their optimal weights as prices shift. Most investors employ naïve heuristics to rebalance their portfolios. We describe how dynamic programming can be used to identify an optimal rebalancing schedule, which significantly reduces rebalancing and sub-optimality costs compared to naïve heuristics.

Unfortunately the curse of dimensionality prevents us from applying dynamic programming to more than a few assets. As an alternative we examine the efficacy of a more sophisticated heuristic called the MvD heuristic, which scales up to several hundred assets. Our tests show that the MvD heuristic performs almost as well as dynamic programming for up to four assets and better than dynamic programming for five assets.

In theory, of course, dynamic programming always yields the best result, but we cannot observe these results beyond a few assets. Therefore, we have no way of determining how the MvD heuristic would compare to the unobservable "correct" dynamic programming solution. To the extent of our knowledge, however, the MvD heuristic is the best alternative by far for rebalancing portfolios with more than just a few assets.

The scalability of the MvD heuristic opens to the door to several new applications of portfolio rebalancing. Passive managers could use the MvD heuristic to optimize the tradeoff between tracking error and transaction costs. Quantitative asset managers could use it to minimize alpha decay between rebalancing dates.

Plan sponsors in particular could benefit from the MvD heuristic, as they are continually confronted with asset mix rebalancing decisions.   Moreover, plan sponsors could customize the optimal rebalancing process to existing tolerance bands, tracking error targets, cash inflows, and benefit payments.

Table 1. The Curse of Dimensionality

Number of Assets	Number of Portfolios	Number of Calculations to Perform
2	101	5,620,751
3	5,151	14,619,573,351
4	176,851	17,233,228,186,751
5	4,598,126	11,649,662,254,243,700
6	96,560,646	5,137,501,054,121,460,000
7	1,705,904,746	1,603,471,162,336,350,000,000
8	26,075,972,546	374,655,945,665,079,000,000,000
9	352,025,629,371	68,281,046,097,460,800,000,000,000
10	4,263,421,511,271	10,015,396,403,505,300,000,000,000,000

Notes. This table shows the number of portfolios as a function of the number of assets, assuming 1% granularity. It also shows the number of calculations one would need to perform in order to solve the dynamic programming problem for a one-year horizon with12 time steps.

Table 2. Volatilities and Transaction Costs

Rebalancing Asset Class	Index	Standard Deviation	Transaction Cost
Domestic Equities	S&P 500	12.74%	0.40%
Domestic Fixed Income	Lehman US Agg	3.96%	0.45%
Foreign Developed Equity	MSCI EAFE + Canada	13.41%	0.50%
Foreign Bonds	CGBI World ex US	8.20%	0.75%
Foreign Emerging Equity	MSCI EM	18.51%	0.75%

Table 3. Correlations

	Domestic Equities	Domestic Fixed income	Foreign Dev. Equities	Foreign Fixed income
Domestic Fixed Income	-0.31			
Foreign Developed Equity	0.84	-0.19		
Foreign Bonds	-0.14	0.53	0.16	
Foreign Emerging Equity	0.77	-0.17	0.83	-0.05

Table 4. Optimal Portfolios

	Two Assets	Three Assets	Four Assets	Five Assets
Domestic Equities	60.00%	40.00%	40.00%	40.00%
Domestic Fixed Income	40.00%	40.00%	25.00%	25.00%
Foreign Developed Equity		20.00%	20.00%	15.00%
Foreign Bonds			15.00%	15.00%
Foreign Emerging Equity				5.00%

Table 5. Performance Comparison – Total Costs (bps)

Rebalancing Strategy	Two Assets	Three Assets	Four Assets	Five Assets	Ten Assets	Twenty Five Assets	Fifty Assets	Hundred Assets
Dynamic Programming	6.31	6.66	7.33	8.76	NA	NA	NA	NA
MvD Heuristic	6.90	7.03	7.58	8.61	25.57	20.38	17.92	12.46
0.25% Bands	15.19	17.01	19.81	21.37	41.93	42.96	41.53	26.88
0.50% Bands	14.11	15.75	17.81	18.92	41.73	38.42	31.15	21.82
0.75% Bands	12.80	14.09	15.32	16.27	40.05	32.95	31.46	25.02
1% Bands	11.54	12.52	13.15	14.13	37.71	31.95	36.74	29.47
2% Bands	8.73	9.20	9.79	10.73	41.94	48.59	66.96	39.33
3% Bands	8.51	8.66	10.14	11.43	61.29	73.78	89.03	41.54
4% Bands	9.46	9.52	12.08	13.78	88.49	93.23	98.55	41.96
5% Bands	11.20	11.21	14.80	16.77	120.19	106.38	102.38	42.03
Monthly	15.65	17.25	20.07	21.85	41.92	42.92	43.34	39.75
Quarterly	11.05	11.86	13.51	14.76	45.17	34.32	33.12	26.54
Semi-annually	11.13	11.53	12.67	13.95	69.97	40.75	37.33	24.41

Notes. This table shows results for 5,000 Monte Carlo simulations. For the 10 through 100 asset cases, which employ equally weighted portfolios of stocks drawn from the S&P 500, a dynamic programming solution is unachievable.

Table 6. Dynamic Programming Discretization Scheme

Number of Assets	Number of Discreetization Points	Number of Portfolios
2	5,000	5,001
3	60	3,323
4	14	2,174
5	7	1,508

Notes. Given that large fluctuations from the optimal allocation are improbable we chose to sample with higher density around the optimal allocation.

Appendix

Table A1 shows the securities used to create the stock portfolios for the 10, 25, 50, and 100 asset cases.

Exhibit A1: Securities used for stock portfolios			
MICROSOFT	SLM	SIGMA ALDRICH	MORGAN STANLEY
IBM	GOLDEN WEST FINANCIAL	GENERAL DYNAMICS	GOLDMAN SACHS
CISCO SYSTEMS	PFIZER	DANAHER	FANNIE MAE
DELL	JOHNSON & JOHNSON	CENDANT	US BANCORP
ORACLE	AMGEN	GENERAL ELECTRIC	WASHINGTON MUTUAL
EBAY	UNITEDHEALTH GROUP	UNITED TECHNOLOGIES	PRUDENTIAL FINL.
YAHOO	MEDTRONIC	BOEING	LEHMAN BROTHERS
FIRST DATA	ELI LILLY	3M	METLIFE
ADOBE SYSTEMS	WYETH	TYCO INTL.	ALLSTATE
HOME DEPOT	CARDINAL HEALTH	UNITED PARCEL SER.	SAINT PAUL TRAVELERS
LOWE'S COMPANIES	GILEAD SCIENCES	CATERPILLAR	SUNTRUST BANKS
TARGET	SCHERING-PLOUGH	HONEYWELL INTERNATIONAL	BANK OF NEW YORK
STARBUCKS	GUIDANT	EMERSON ELECTRIC	FRANK.RES.
BEST BUY	CAREMARK RX	LOCKHEED MARTIN	HARTFORD FINANCIAL SERVICES
SEARS HOLDINGS	STRYKER	FEDEX	INTEL
NIKE	VALERO ENERGY	BURLINGTON NORTHERN SANTA FE CORPORATION	HEWLETT-PACKARD
AMAZON.COM	BURLINGTON RES	ILLINOIS TOOL WORKS	QUALCOMM
KOHLS	DEVON ENERGY	UNION PACIFIC	APPLE COMPUTERS
CLEAR CHANNEL COMMUNICATIONS	ANADARKO PETROLEUM	CITIGROUP	MOTOROLA
OMNICOM GROUP	PROCTER & GAMBLE	BANK OF AMERICA	TEXAS INSTRUMENTS
HARLEY-DAVIDSON	WAL MART STORES	AMERICAN INTERNATIONAL GROUP	CORNING
YUM! BRANDS	PEPSICO	JP MORGAN CHASE & COMPANY	EMC
AMERICAN EXPRESS	WALGREEN	WELLS FARGO & COMPANY	APPLIED MATERIALS
FREDDIE MAC	ANHEUSER-BUSCH	WACHOVIA	AUTOMATIC DATA PROCESSING
CAPITAL ONE FINANCIAL	ECOLAB	MERRILL LYNCH & COMPANY	ADVANCED MICRO DEVICES

For example, the first 10 securities in column one constitute the 10 asset portfolio, and the securities in the first column constitute the 25 asset portfolio.

We determine the risks and correlations of the securities in Table A1 based on daily historical returns from January, 2005 through January, 2006 and estimate the expected returns as the implied returns under the assumption that the equally weighted portfolio is optimal under mean-variance optimization.

Tables A2 through A9 show the trading cost and sub-optimality cost components for the various rebalancing algorithms.

Exhibit A2: Performance Comparison - Two Assets
(5,000 Monte Carlo Simulations)

Rebalancing Strategy	Trading	Costs (bps) Sub-optimality	Total
Dynamic Programming	4.87	1.44	6.31
MvD Heuristic	4.86	2.04	6.90
0.25% Bands	15.18	0.01	15.19
0.50% Bands	14.06	0.05	14.11
0.75% Bands	12.63	0.17	12.80
1% Bands	11.19	0.34	11.54
2% Bands	7.18	1.55	8.73
3% Bands	5.17	3.34	8.51
4% Bands	3.88	5.58	9.46
5% Bands	3.00	8.20	11.20
Monthly	15.65	0.00	15.65
Quarterly	9.31	1.74	11.05
Semi-annually	6.70	4.43	11.13

Exhibit A3: Performance Comparison - Three Assets
(5,000 Monte Carlo Simulations)

Rebalancing Strategy	Trading	Costs (bps) Sub-optimality	Total
Dynamic Programming	4.68	1.98	6.66
MvD Heuristic	4.73	2.30	7.03
0.25% Bands	17.00	0.00	17.01
0.50% Bands	15.71	0.04	15.75
0.75% Bands	13.94	0.15	14.09
1% Bands	12.20	0.32	12.52
2% Bands	7.69	1.50	9.20
3% Bands	5.40	3.26	8.66
4% Bands	4.03	5.49	9.52
5% Bands	3.16	8.05	11.21
Monthly	17.25	0.00	17.25
Quarterly	10.24	1.61	11.86
Semi-annually	7.38	4.15	11.53

Exhibit A4: Performance Comparison - Four Assets (5,000 Monte Carlo Simulations)			
Rebalancing Strategy	Costs Trading (bps)	Costs Sub-optimality (bps)	Costs Total (bps)
Dynamic Programming	5.10	2.23	7.33
MvD Heuristic	4.94	2.64	7.58
0.25% Bands	19.80	0.00	19.81
0.50% Bands	17.73	0.08	17.81
0.75% Bands	15.05	0.27	15.32
1% Bands	12.57	0.58	13.15
2% Bands	7.29	2.50	9.79
3% Bands	4.82	5.32	10.14
4% Bands	3.33	8.75	12.08
5% Bands	2.29	12.51	14.80
Monthly	20.07	0.00	20.07
Quarterly	11.87	1.64	13.51
Semi-annually	8.50	4.17	12.67

Exhibit A5: Performance Comparison - Five Assets (5,000 Monte Carlo Simulations)			
Rebalancing Strategy	Costs Trading (bps)	Costs Sub-optimality (bps)	Costs Total (bps)
Dynamic Programming	6.21	2.55	8.76
MvD Heuristic	5.30	3.31	8.61
0.25% Bands	21.36	0.01	21.37
0.50% Bands	18.81	0.11	18.92
0.75% Bands	15.92	0.35	16.27
1% Bands	13.41	0.72	14.13
2% Bands	7.70	3.02	10.73
3% Bands	5.09	6.33	11.43
4% Bands	3.55	10.23	13.78
5% Bands	2.46	14.31	16.77
Monthly	21.85	0.00	21.85
Quarterly	12.95	1.82	14.76
Semi-annually	9.29	4.66	13.95

Exhibit A6: Performance Comparison - Ten Assets
(5,000 Monte Carlo Simulations)

Rebalancing Strategy	Costs Trading (bps)	Costs Sub-optimality (bps)	Costs Total (bps)
MvD Heuristic	19.59	5.98	25.57
0.25% Bands	41.93	0.00	41.93
0.50% Bands	41.68	0.05	41.73
0.75% Bands	39.21	0.83	40.05
1% Bands	34.47	3.24	37.71
2% Bands	20.76	21.18	41.94
3% Bands	14.11	47.19	61.29
4% Bands	10.14	78.35	88.49
5% Bands	7.42	112.76	120.19
Monthly	41.92	0.00	41.92
Quarterly	24.83	20.34	45.17
Semi-annually	17.69	52.28	69.97

Exhibit A7: Performance Comparison - Twenty-Five Assets
(5,000 Monte Carlo Simulations)

Rebalancing Strategy	Costs Trading (bps)	Costs Sub-optimality (bps)	Costs Total (bps)
MvD Heuristic	14.16	6.22	20.38
0.25% Bands	42.96	0.00	42.96
0.50% Bands	37.07	1.34	38.42
0.75% Bands	27.60	5.35	32.95
1% Bands	21.63	10.32	31.95
2% Bands	10.56	38.02	48.59
3% Bands	5.91	67.87	73.78
4% Bands	3.35	89.88	93.23
5% Bands	1.78	104.59	106.38
Monthly	42.92	0.00	42.92
Quarterly	25.32	9.01	34.32
Semi-annually	17.97	22.78	40.75

Exhibit A8: Performance Comparison - Fifty Assets (5,000 Monte Carlo Simulations)			
Rebalancing Strategy	Costs Trading (bps)	Costs Sub-optimality (bps)	Costs Total (bps)
MvD Heuristic	12.05	5.86	17.91
0.25% Bands	41.22	0.31	41.53
0.50% Bands	25.23	5.92	31.15
0.75% Bands	17.46	14.00	31.46
1% Bands	12.93	23.73	36.66
2% Bands	5.15	61.82	66.96
3% Bands	1.80	87.23	89.03
4% Bands	0.59	97.95	98.55
5% Bands	0.23	102.16	102.38
Monthly	43.34	0.00	43.34
Quarterly	25.57	7.55	33.12
Semi-annually	18.14	19.19	37.33

Exhibit A9: Performance Comparison - Hundred Assets (5,000 Monte Carlo Simulations)			
Rebalancing Strategy	Costs Trading (bps)	Costs Sub-optimality (bps)	Costs Total (bps)
MvD Heuristic	7.55	4.91	12.46
0.25% Bands	24.75	2.13	26.88
0.50% Bands	12.95	8.88	21.82
0.75% Bands	8.13	16.89	25.02
1% Bands	5.39	24.08	29.47
2% Bands	0.71	38.61	39.33
3% Bands	0.10	41.44	41.54
4% Bands	0.02	41.94	41.96
5% Bands	0.01	42.02	42.03
Monthly	39.75	0.00	39.75
Quarterly	23.46	3.08	26.54
Semi-annually	16.63	7.78	24.41

References

Bellman, R.E. “On the theory of dynamic programming.” *Proceedings of the National Academy of Sciences*, 38 (1952), pp.716-719

Cremers, Jan-Hein, Kritzman, and Page. “Portfolio Formation with Higher Moments and Plausible Utility.” *Revere Street Working Paper Series. Financial Economics* 272-12 (2003).

Cremers, Jan-Hein, Kritzman, and Page. *“Optimal Hedge Fund Allocations.” Journal of Portfolio Management*, Vol. 31 (Spring 2005). No 3.

Kroll, Yoram, Levy, and Markowitz. “Mean Variance Versus direct Utility Maximization.” *Journal of Finance*, Vol. 39 (1984), No. 1. pp 47-61.

Levy, Haim and Harry M. Markowitz. “Approximating Expected Utility by a Function of Mean and Variance.” *American Economic Review*, Vol. 69 (1979), No. 3.

Markowitz, Harry M. “Portfolio Selection.” *Journal of Finance*, (1952), pp.77-91

Markowitz, Harry M and Erik L. van Dijk. Single-Period Mean–Variance Analysis in a Changing World (corrected). *Financial Analysts Journal*, Vol. 59 (March/April 2003.), No. 2, pp. 30-44

Smith, David K., “Dynamic programming: an introduction”, PASS Maths, http://plus.maths.org/issue3/dynamic/, September 1997

Sun, W., A. Fan, L. W. Chen, T. Schouwenaars, and M. Albota. “Optimal Rebalancing for Institutional Portfolios.” *Journal of Portfolio Management*, (Winter 2006) pp. 33-43

---

[1] A particularly intuitive illustration of dynamic programming is provided by Smith (1997). He demonstrates how dynamic programming can be used to find a soul mate.

[2] The number of portfolios is given by the formula, $N = (1/g + n-1)! \div ((n-1)! \cdot (1/g)!)$, where g equals granularity and n equals number of assets. In our experiments we exactly don’t use g = 1%. We use efficient sampling, which means that g is small around the optimal weights and gets larger as we move further from them.

[3] There are a variety of optimization algorithms to minimize this cost function. We use the fmincon function which is available in the optimization toolbox of MatLab.

[4] For example, finding the best coefficient d for a 100 asset case would take slightly more than 10 days without grid computing.

[5] Others have shown that results of comparisons between optimal rebalancing and industry heuristics are not sensitive to changes in utility function or changes in risk and return assumptions (see Sun et al. 2006).

[6] The term "Figure of Merit" is from in Markowitz and van-Dijk, 2003.

[7] Some investors might use more sophisticated heuristics. For example they might use different bands for each asset, or rebalance partially, for example to the edge of the band, rather than back to the optimal weights. Our approach will be useful to these investors, as it will help them optimize these decision rules.