

実用的ポートフォリオ最適化

K V Fernando

NAG Ltd
Wilkinson House
Jordan Hill
Oxford OX2 8DR
United Kingdom
email:vin ce@nag.co.uk

概要

NAG ライブラリには大規模なポートフォリオ最適化や金融業界の選択問題の解決に使用できる強力で信頼性のある多くのオプティマイザーがあります。これらの万能なルーチンは学術研究や教育にも適しています。

キーワード

マーコウィッツ、平均分散解析、最適ポートフォリオ、最小分散ポートフォリオ、ポートフォリオ選択、ポートフォリオ配分、ポートフォリオ多様性、ポートフォリオ最適化、効率的フロンティア、平均分散フロンティア、MV効率性

目次

1	概要	1
2	NAG 最適化ルーチン	2
2.1	ライブラリルーチンの選択	2
2.2	線形制約つき二次計画法	2
2.3	非線形計画法	3
2.4	スパース行列問題用ルーチン	3
2.5	Forward 及び Reverse Communication	3
2.6	ハードウェア、オペレーティングシステム及び環境	4
3	ルーチンへのインターフェース	4
3.1	ポートフォリオウェイト	4
3.2	一次データ	4
3.3	一般線形制約	5
3.4	非線形制約	6
3.5	コールド及びウォームスタート	6
4	最適化問題	6
5	生データの処理	7
5.1	因数分解された共分散行列	7
5.2	コレスキー因子の特異値の決定	9
5.3	共分散行列が既に存在する場合	9
5.4	共分散行列の固有値	10
5.5	欠測値	10
6	数値例：株式の選択	10
7	数値例：資産分配	15
8	取引コスト	15
9	サンプルプログラム	17
10	謝辞	20

1 概要

資産や株式の選択は、単に投資対象を見つける問題ではありません。正しい資産のポートフォリオの設計は人間の直感だけで行うことはできません。オブティマイザーと呼ばれる近代的で強力な信頼性のある数学的プログラムが必要です。ニューメリカルアルゴリズム (NAG) は数値計算アルゴリズムの研究において世界的に有名であり、NAG 最適化ルーチンは産業界、商業界、学界に広く使用されています。

どんな投資家も投資から最高のリターンを得たいと思っています。しかしながら、これは投資家がリスクを負うことができる、もしくはリスクを負いたいと思うそのリスク量により相殺されなければなりません。予想されるリターンと分散により測定されるリスク (あるいは、分散の平方根である標準偏差) はポートフォリオの二つの主要な特徴です。残念ながら、ハイリターンの株式は通常ハイリスクと相関性があります。

ポートフォリオの動きはポートフォリオを構成する個々の要素の動きとはかなり異なる可能性があります。主要マーケットの株式から正しく構築されたポートフォリオのリスクは、ポートフォリオの個々の資産のリスクを足し合わせたものの半分になる可能性があります。これは、個々の資産あるいは株式の間の複雑な相関パターンによります。優れたオブティマイザーは相関、期待リターン、リスク (分散) やユーザ制約を有効に利用して最適化されたポートフォリオを得ることができます。NAG 最適化ルーチンは、投資家の期待に沿うよう、最適化され、多様化されたポートフォリオを提供することができます。

ポートフォリオの最適化の数学的問題は 50 年代に Harry Markowitz (ハリー・マーコウィッツ) 教授により始められました。彼は 1990 年に William Sharpe 教授と Merton Miller 教授と共にノーベル経済学賞を受賞しました。[8]

NAG オブティマイザーは、従来のマーコウィッツの最適化問題[7], [9], [10]と現代の多くの進展した問題[4], [11], [13], [14], [15]を処理できます。NAG はまた数学、数値計算、プログラミング、可視化の問題を解くためのコンサルティングサービスを金融部門に提供しています。ポートフォリオ最適化はしばしば平均分散(mean-variance: MV)最適化と呼ばれます。この平均(mean) は、投資の平均リターンもしくは期待リターンを意味し、分散(variance) はポートフォリオに関連するリスクの尺度を意味します。数理問題は様々な方法で公式化できますが、主な問題は下記のように要約することができます。

1. 特定の期待リターンのリスクを最小化する
2. 特定のリスクの期待リターンを最大化する
3. 特定のリスク回避因子を用いてリスクを最小化し、期待リターンを最大化する
4. 期待リターンにかかわらずリスクを最小化する
5. リスクにかかわらず期待リターンを最大化する

6. リスクにかかわらず期待リターンを最小化する

上記の問題には線形制約もしくは非線形制約、等式もしくは不等式制約がある可能性があります。最初の3つの問題は、基本的には数学的に等しく、その解は平均分散(mean-variance: MV)効率性と呼ばれます。リターンーリスクのグラフの有効点は効率的フロンティアと呼ばれます。

4番目の問題では、用心深い投資家のための最小分散の解が得られます。その解は他のポートフォリオの比較やベンチマークにも使用されます。5番目の問題では、期待リターンの到達可能な上限が得られます。これは比較にも役立ちます。最後の問題は最悪のシナリオを示しています。市場の状況や投資家のリスクの優先度が変わる場合、ポートフォリオのリバランスが望ましいです。上記問題のいずれも既存のポートフォリオやベンチマークと関連して解決することができます。

新規のポートフォリオの購入やポートフォリオのリバランスに関連する取引コストは投資家にとってかなりの金額になる可能性があります。NAG 最適化ルーチンは取引コストに対処し、ポートフォリオの構成に大きく作用します。

2 NAG 最適化ルーチン

2.1 ライブラリルーチンの選択

NAG ライブラリは一つのチャプターを関数最適化の記述に割いています。表1はNAGライブラリで利用可能なルーチンを要約したものです。

ルーチン名	制約	目的関数	密／スパース (疎)
e04nc	線形	二次	密
e04nf	線形	二次	密
e04nk	線形	二次	スパース
e04uc	非線形	非線形	密
e04uf	非線形	非線形	密
e04ug	非線形	非線形	スパース

表1:最適化ルーチン

2.2 線形制約つき二次計画法

e04nf ルーチンは主に線形プログラミング (Linear Programming: LP) と二次計画法 (Quadratic Programming: QP) 用で、以下の目的関数を最小化することができます。

LP1: $c^T x$ 。この選択肢は分散にかかわらず期待リターンの最大もしくは最小を見つけるのに使用

することができます。ベクトル c は期待リターンベクトル a に設定されます。ポートフォリオのウェイトベクトルは x で表されます。

QP1: $\frac{1}{2}x^t H x$ 。この選択肢は分散を最小化するのに使用できます。行列 H は共分散行列 V に設定されます。

QP2: $\frac{1}{2}x^t H x + c^t x$ 。この選択肢は特定のリスク回避因子の分散を最小化するのに使用できます。行列 H は共分散行列 V に設定され、 c は λ がリスク回避パラメータで a が期待リターンベクトルである場合に、 $-\lambda a^t x$ に設定されます。目的関数は、 $\frac{1}{2}(x^t V x - \mu a^t x)$ として定義される場合があります。その場合、新しいリスク回避因子 μ は 2λ に等しくなります。

QP3: $\frac{1}{2}x^t H^t H x$ 。これは分散が共分散行列 V のコレスキー因子 H により提供されることを除いて QP1 に類似しています。詳細はセクション 5.1 を参照ください。

QP4: $\frac{1}{2}x^t H^t H x + c^t x$ 。これは共分散が共分散行列 V のコレスキー因子 H により提供されることを除いて QP2 に類似しています。

ルーチン e04nc もまた最小二乗の最小化問題や回帰問題を処理できます。多くのポートフォリオ最適化問題が最小二乗問題や回帰問題として定式化できます。そのため同一ルーチン e04nc は最小二乗問題や回帰問題として定式化されるポートフォリオ問題の他、一般のポートフォリオ最適化を解くのに使用できます。

2.3 非線形計画法

より一般的なルーチン e04uc と e04uf は非線形制約つき非線形目的関数に適合できます。

2.4 スパース行列問題用ルーチン

NAGライブラリの最適化ルーチンのほとんどは密行列問題用です。制約行列はほとんどゼロ以外の要素であると想定されています。しかしルーチン e04nk と e04ug はスパース制約行列で動作するように設計されています。ルーチン e04nk は線形計画法や二次計画法用で、ルーチン e04ug は非線形計画法用です。ルーチン e04nk は共分散行列 V を明示的には必要としていませんが、ユーザは行列ベクトル積 Vx を提供する必要があります。

2.5 Forward 及び Reverse Communication

最適化ルーチンのほとんどは forward communications に基づいています。

このようなプログラムでは、ルーチンは結果を得るのに 1 回だけ呼び出されます。そしてユーザは必要な全ての情報をサブルーチン経由で NAG ルーチンに提供します。

しかし、状況次第では、最適化を段階的に行い、最新の情報を得るために繰り返しユーザルーチンを呼び出す必要があります。

NAG ルーチン `e04uc` は forward communication ルーチンで、`e04uf` は reverse communication 同等ルーチンです。この reverse communication ルーチンは、プロシージャ引数のサポートを NAG ルーチンと互換性のある方法で十分には行っていない別の言語（例えば、Microsoft VBA）から呼び出される場合特に便利です。

2.6 ハードウェア、オペレーティングシステム及び環境

NAG ライブラリはすべての一般的ハードウェアシステムやオペレーティングシステムで利用可能です。PC、ワークステーション、SGI/Cray T3E や IBM SP3などの並列コンピュータ、ワークステーションネットワーク、強力な SMP マシンでの実装があります。Windows (2000, 98, NT)、DOS や PC版Linux から大規模システムのUNIX や VMSまで様々なオペレーティングシステムがサポートされています。

全ての最適化ルーチンは DLL としてご利用いただけます。

3 ルーチンへのインターフェース

3.1 ポートフォリオウェイト

資産 i の保有価値によるウェイトや比率は x_i で表されます。

3.2 一次データ

NAG 最適化ルーチンは、ポートフォリオについて以下の詳細を必要とします。

n ポートフォリオの資産の数

a 長さ n のベクトルの資産の期待リターン

s 長さ n のベクトルの資産のボラティリティ (標準偏差)

C $n \times n$ 行の対称行列の相関係数

l_i 資産 i のウェイトの下限

u_i 資産 i のウェイトの上限

また一方、ボラティリティベクトル s と相関行列 C の代わりに、以下が利用可能です：

$n \times n$ 行の対称行列の V 共分散行列

通常、NAG 最適化ルーチンでは共分散行列 V が必要です。もし得られない場合は、標準の偏差ベクトル s と相関行列 C から簡単に計算できます。しかし、セクション 5.1では、情報の消失

を避けるため、共分散行列 V は明示的に作成されるべきではないと推奨しています。必要なのは、共分散行列 V のコレスキー因子 R であり、行列 V 自身ではありません。

3.3 一般線形制約

一次等式制約とは、すべてのウェイトが一つの定数、通常 1 になるということです。それは以下で表されます。

$$\sum_{i=1}^n x_i = 1$$

そして 上記は予算制約や投資制約として知られています。行列の表記では以下で表されます。

$$e^t x = 1$$

カラムベクトル x が資産の割合をもつウェイトベクトルである場合、以下で表されます。

$$x = (x_1, x_2, \dots, x_n)^t$$

そして e は全ての要素が 1 であるカラムベクトルである場合、以下で表されます。

$$e = (1, 1, \dots, 1)^t$$

期待リターンは a_i が資産あるいは株 i の期待リターンを含む場合、以下で表されます。

$$a = (a_1, a_2, \dots, a_n)^t$$

NAG オプティマイザーは、多くの等式制約と不等式制約を処理できます。それらは以下の形式で表されます。

$$\begin{aligned} l &\leq x \leq u \\ L &\leq Ax \leq U \\ \text{where } l &= (l_1, \dots, l_n)^t \\ u &= (u_1, \dots, u_n)^t \\ L &= (L_1, \dots, L_m)^t \\ U &= (U_1, \dots, U_m)^t \end{aligned}$$

l_i と L_i は下限です。 u_i と U_j は上限を表します。 m は行列 A の行数を表します。

等式制約の場合、下限と上限は等しい値に設定されます。また上限は $+\infty$ に設定でき、下限は $-\infty$ に設定できます。

これらの制約は、セクター、産業、国、エクスポージャー(モメンタム、QRQ、P/B)やその他ユーザ定義の財政的制約を表すことができます。

3.4 非線形制約

NAG 最適化ルーチン、特にルーチン e04uc、e04uf と e04ug は線形制約を処理することができます。また非線形目的関数もサポートしています。

3.5 コールド及びウォームスタート

コールドスタートとは最初から問題を解くことを意味します。しかし、ルーチンが繰り返し呼び出されると近似解を前の解から得ることができます。この場合、次回の反復の初期条件が前の反復から与えられる可能性があります。このようなウォームスタートと呼ばれる機能が多くのNAG最適化ルーチンで利用可能です。

4 最適化問題

ここにNAGオブティマイザーを使用して解くことができるいくつかの問題があります。しかし、このリストは全てを表してはおらず、もっと多くの最適化問題がNAGのソフトウェアを使用して解くことができます。数学的な考え方では、これらの問題は以下のように記載することができます。

問題 1 (リスクの最小化)

線形及び (または非線形) 制約を受ける特定の期待リターンが $r = a^t x$ となる、
分散 $x^t V x$ を最小化します。

問題 2 (期待リターンの最大化)

特定の分散が $v = x^t V x$ となる、
期待リターン $r = a^t x$ を最大化します。

これはまた $-r$ の最小化と同等です。

問題 3 (リスク回避して期待リターンを最大化)

λ がリスク回避パラメータである場合に線形及び (または非線形) 制約を受ける、
 $\lambda a^t x - x^t V x$ を最大化します。

これはまた以下と同じです。

$x^t V x - \lambda a^t x$ を最小化します。

問題 4 (リスクの最小化)

線形及び (または非線形) 制約を受ける、

分散 $x^t V x$ を最小化します。

問題 5 (期待リターンの最大化)

線形及び (または非線形) 制約を受ける、

期待リターン $r = a^t x$ を最大化します。

これはまた以下と同じです。

- $a^t x$ を最小化します。

問題 6 (期待リターンの最小化)

線形及び (または非線形) 制約をうける、

期待リターン $r = a^t x$ を最小化します。

上記の 6 つの問題はベンチマークや既存のポートフォリオについて解くことができます。以下は一つの例で、より多くの変形が可能です。

問題 7 (ベンチマークに関してリスクを最小化)

$(x - x_b)^t V (x - x_b)$ を最小化します。

以下の特定の期待相対リターン r を用います。

$$r = a^t (x - x_b)$$

これは、 x_b がベンチマークポートフォリオを表す場合に線形及び (または非線形) 制約をうけます。

5 生データの処理

5.1 因数分解された共分散行列

行列 Y の要素 $y_{i,j}$ について i 回での資産 j のリターンを表しましょう。 m 回の間で n 個の資産がある場合、行列 Y は $m \times n$ 行列になります。資産 j のサンプル平均は以下で表されます。

$$a_j = \frac{1}{m} \sum_{i=1}^m y_{i,j}$$

一旦サンプル平均がリターンから除外されると、サンプル平均から偏差が得られます。この偏差 \hat{Y} は次のように得られます。

$$\hat{Y} = Y - ea^t$$

e は各要素が 1 である長さ m のカラムベクトルの場合、以下になります。

$$e = (1, \dots, 1)^t.$$

サンプルの共分散行列 V は通常以下のように計算されます。

$$V = \frac{1}{m} \hat{Y}^t \hat{Y}$$

しかし、統計学者によっては下記の公式のほうを好むかもしれません。

$$V = \frac{1}{m-1} \hat{Y}^t \hat{Y}$$

しかし、このような共分散行列の計算が情報消失をもたらすことは数値アナリストによく知られています。[1], [5] または [6] を参照ください。

ここに、共分散の作成で情報が消失されたよく知られている例があります。

\hat{Y} が以下だとします。

$$\hat{Y} = \begin{pmatrix} 1 & 1 & 1 \\ \theta & & \\ & \theta & \\ & & \theta \end{pmatrix}$$

これは θ がゼロ以外であるフルランク行列です。共分散行列は以下で表されます。

$$\hat{Y}^t \hat{Y} = \frac{1}{3} \begin{pmatrix} 1 + \theta^2 & 1 & 1 \\ 1 & 1 + \theta^2 & 1 \\ 1 & 1 & 1 + \theta^2 \end{pmatrix}$$

倍精度浮動小数点演算のIEEE マシンではマシン精度 ϵ はおよそ 2^{-53} です。もし θ が $\sqrt{\epsilon}$ より小さい場合、浮動小数点演算ではコンピュータは $\hat{Y}^t \hat{Y}$ の対角要素を丸め誤差により以下のように決定します。

$$1 + \theta^2 \rightarrow 1$$

この場合、共分散行列は以下のように求められます。

$$\hat{Y}^t \hat{Y} = \frac{1}{3} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \frac{1}{3} \mathbf{e} \mathbf{e}^t$$

これはランク 1 の行列であり、従って特異です。そのため共分散行列の明示的な作成は避けるべきです。このような情報消失をさけるための方法の一つは、 $m \times n$ 行列 Q が直交カラム（例えば、 $Q^t Q = I$ ）を含み、 R が $n \times n$ 上三角行列である場合に下記に定義される \hat{Y} のQR 分解を計算することです。

$$QR = \hat{Y}$$

NAG ライブラリにはQR分解を計算するルーチン（例えば、f08ae または f01qc）があります。共分散行列は以下で表されます。

$$V = \frac{1}{m} R^t Q^t Q R = \frac{1}{m} R^t R$$

しかし、上記の積はポートフォリオ最適化で計算する必要はありません。コレスキー因子 R/\sqrt{m} は最適化ルーチンの入力として与えることができます。この場合、最適化オプション QP3 または QP4 (Section 2.2を参照) が使用されます。行列 Q を計算する必要がないことをご注意ください。

5.2 コレスキー因子の特異値の決定

共分散行列が正定値かどうか確認するのに、コレスキー因子 R の特異値が計算される場合があります。共分散行列 H の固有値はコレスキー因子の特異値の二乗です。このため、共分散行列の固有値は共分散行列を作成しなくても決定される可能性があります。

5.3 共分散行列が既に存在する場合

前のセクションで共分散行列は明示的に作成されるべきではないということを示しました。しかし、もし既に作成されていたら、オプション QP1 と QP2 (詳細は Section 2.2 を参照) を使用して直接 NAG 最適化ルーチンへの入力として与えることができます。この場合、共分散行列のコレスキー因子

$$V = R^t R$$

は最適化ルーチンを呼び出す前に決定される必要があります。もし共分散行列が正定値の場合、これはf07fd または f03ae を使用して行うことができます。もし因数分解が存在しない場合は、これらのルーチンは適当なエラー処理フラグにより失敗を示します。

共分散行列は通常正定値、あるいは少なくとも半正定値です。しかし、丸め誤差により共分散行列は正定値でない可能性があります。その場合、因数分解ルーチンは因子の計算ができないかもしれません。しかし、コレスキー因子がたとえあったとしても共分散行列が正定値である保証はありません。下記の例を考えてみてください。

$$H = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} = R^t R.$$

共分散行列 H にはコレスキー因子がありますが、正定値ではありません。

5.4 共分散行列の固有値

情報消失を避けるために、共分散行列は明示的には作成されるべきではないと述べました。しかし、もし既に作成されている場合は、共分散行列が正定値かどうかを見つける最善の方法は共分散行列の固有値を計算することです。もし固有値が正数の場合、行列は正定値です。しかし、たとえすべての固有値が整数であっても、(最小の固有値で割られた最大の固有値として定義される) 行列の条件数を計算することをお勧めします。もし条件数が高い場合は、行列はほとんど半正定値で、これは信ぴょう性のない結果をもたらします。

固有値は `f02aa` を使用して計算される場合がありますが、ほかにも使用できる多くのルーチンがあります。

5.5 欠測値

セクション 5.1では、行列 Y が定義されました。そしてそれには共分散行列と共分散行列コレスキー因子を計算するためのデータが含まれています。

実際に、この行列には欠測(未記録)データが含まれています。もしこれが本当ならば、欠測エントリへ値を割り当てるために適切なデータ補間ルーチンが使用される必要があります。

より教養のあるユーザは金融のBrownian プロセス用に `classical EM` アルゴリズム [3]を取り入れた `Brownian bridge EM` アルゴリズム [12]を使用して欠測データを見つけるかもしれません。

6 数値例：株式の選択

図で表すため、NAGのオプティマイザーから予測される結果についていくつかの例を示します。データは、Hang Seng (香港、31 銘柄)、DAX 100(ドイツ、85 銘柄)、FTSE 100 (イギリス、89 銘柄)、S & P 100 (アメリカ、98銘柄)そして日経(日本、225 銘柄)の株式のポートフォリオがNAG `e04`ルーチンを使用して研究がなされた際の [2]から得られています。私たちの効率的フロンティアはルーチン `e04nf`に基づいています。

データは週次のリターンと週次の分散を指しています。図1 から 5 まではこのデータの効率的フロンティアを示しています。優位な(赤の)曲線はウェイトがゼロから1の間に制約される場合(空売りは禁止)のフロンティアを表しています。より厳しい制約がウェイト($0.0 \leq x_i \leq 0.1$)に強いられる場合、異なるフロンティアが現れます。一般に、多くの制約があると、期待リターンは特定のリスク値に対して小さくなります。個々の点は個別の株式を表しています。最適化によるリスクの削減はこれらの図の効率的フロンティアをもつ株式を比較することで評価することができます。

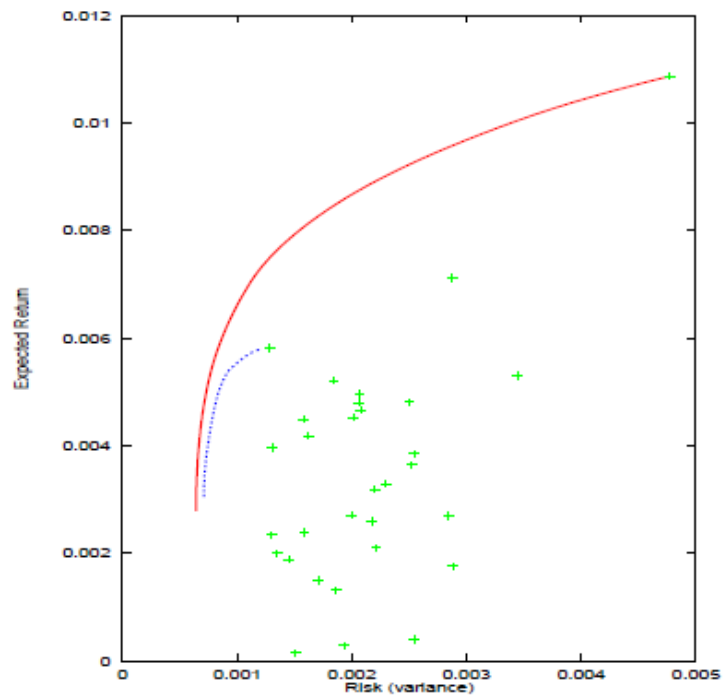


図1: Hang Seng の効率的フロンティアの例

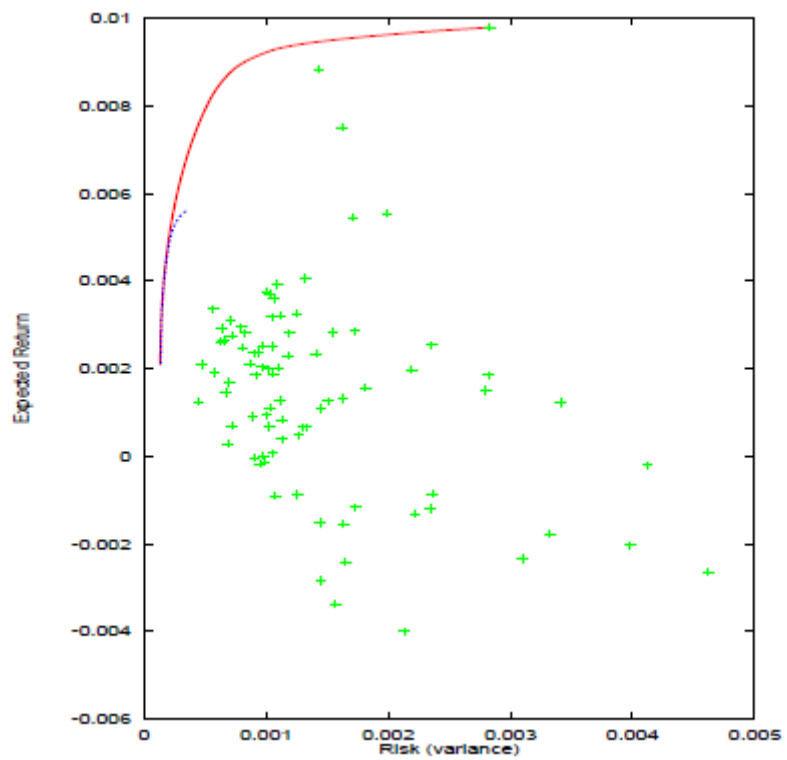


図2: DAX 100 の効率的フロンティアの例

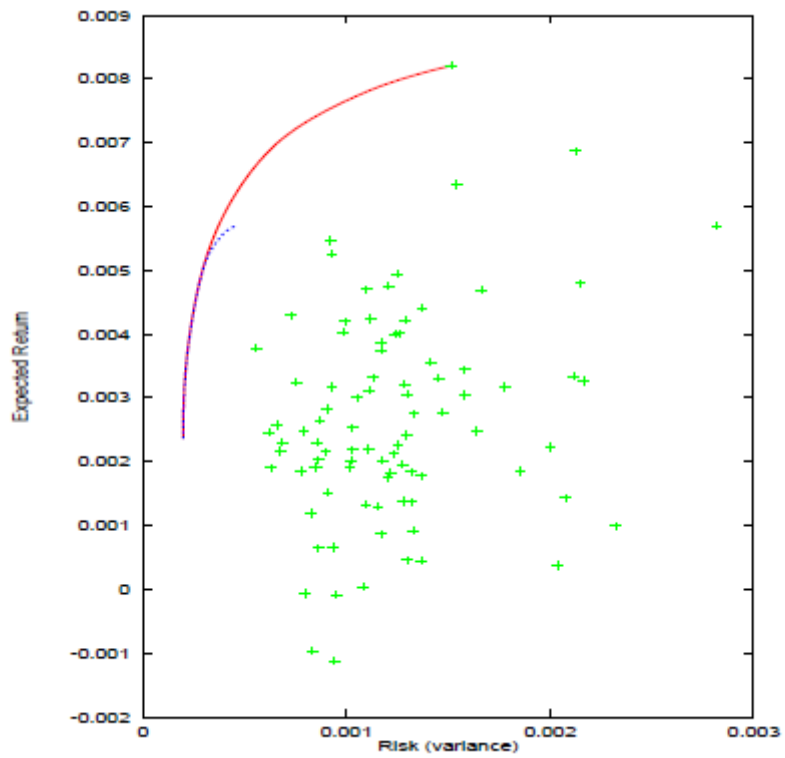


図3: FTSE 100 の効率的フロンティアの例

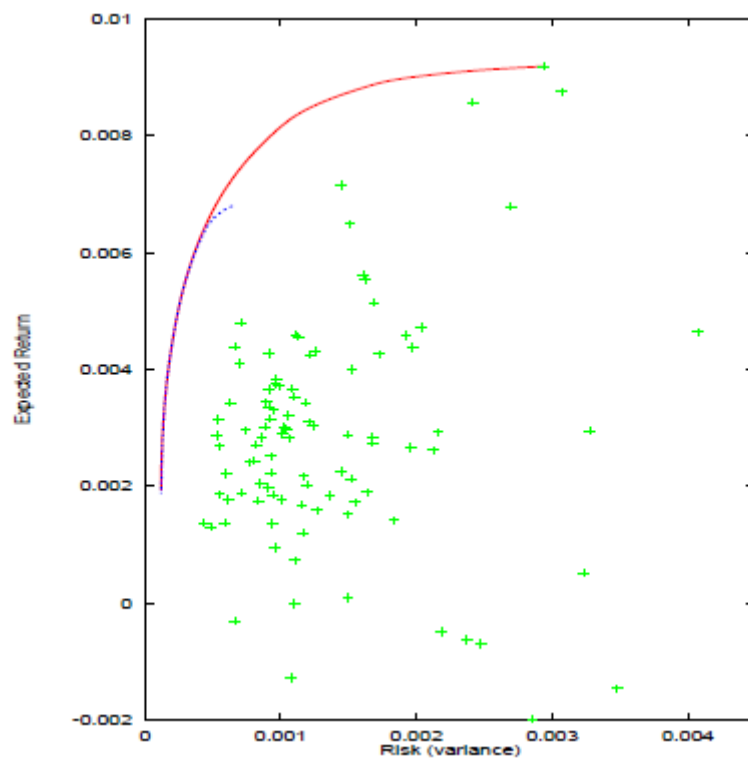


図4: S & P 100 の効率的フロンティアの例

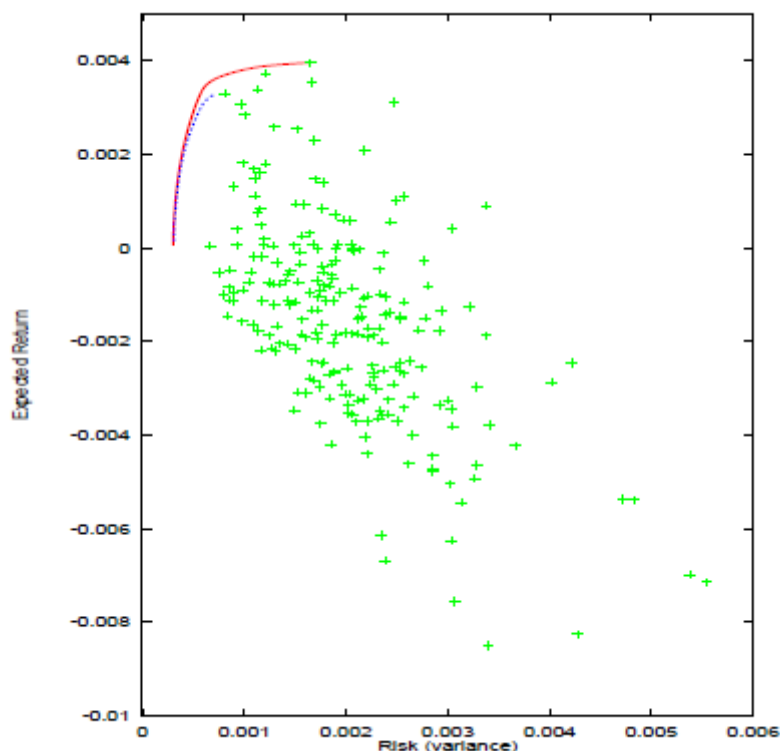


図5:日経の効率的フロンティアの例

実際問題、財政的制約が多いと、効率的フロンティアは小さな曲線もしくはほとんど点になるまで減少します。

日経の例の場合、私たちはリスク回避パラメータ λ の負の値を使用して得られる非効率的フロンティアも計算します。(セクション4の問題3を参照) 図6では、曲線の上半分(赤)は、効率的フロンティアを示し、曲線の下半分(青)は非効率的フロンティアを示しています。この例では空売りは禁止されています。

マーコウィッツ型最適化は資産分配に頻繁に使用されます。資産管理者は資産のポートフォリオを選択するため、株、公社債、国際株式や国際債券、不動産、ベンチャーキャピタルや金を所有できます。それぞれの資産は(最適化、または最適化なしの)ポートフォリオになる可能性があります。

次の例では、8種類の資産(米政府短期証券3カ月物、米長期国債、SP 500、Wilshire 500、Nasdaq総合社債指数、EAFE及び金)を使用しました。年次のリターンは1973年から1994年までのものです。これらは下記で参照いただけます。

図7は二つの効率的フロンティアを示しています。優位の曲線(赤)は資産の上限が1に設定され下限がゼロに設定されたフロンティアを示しています。下限が0.05に上げられ、上限が0.5に下げられる場合、ドット曲線(青)で示される効率的フロンティアが得られます。

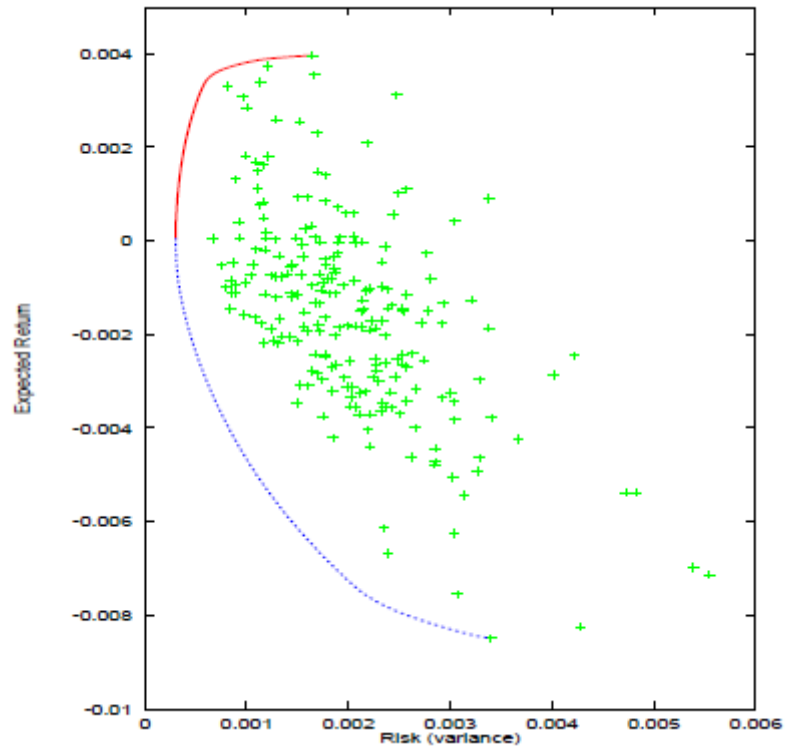


図6:日経の効率的及び非効率的フロンティアの例

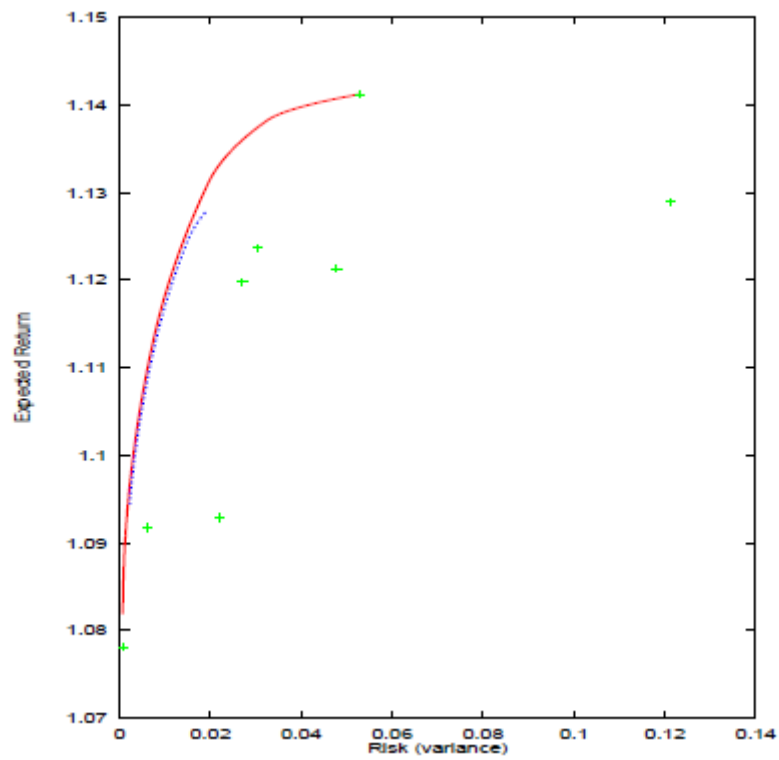


図7:資産分配の効率的フロンティアの例

7 数値例：資産分配

この数値例では、私たちは生のデータ（年次リターン）から始めました。そして共分散行列を明示的には作成しませんでした。そのかわり、セクション5.1で述べたように、共分散のコレスキー因子 R のみ計算しました。この場合、最適化オプションQP3 や QP4（セクション 2.2を参照）が使用されるべきです。なぜならコレスキー因子 R のみ入力として与えられる必要があり、共分散行列 V 全体は入力として与えられなくてもいいからです。図 7 はこの問題の効率的フロントティアを示しています。

8 取引コスト

マーコウィッツの古典的研究では、株式の売買に関する取引コストは認められていませんでした。しかし、今は新規のポートフォリオの構築や既存のポートフォリオのリバランスに取引コストを組み入れることの重要性が十分認識されています。一般に取引コストは 無視できるほど小さいものではありません。最適なポートフォリオは総取引コストに依存します。

株式 i の購入に関する取引コストを p_i とすると、下記のようにモデル化されます。

$$p_i = \begin{cases} (x_i - \bar{x}_i)g_i & \text{for } x_i > \bar{x}_i \\ 0 & \text{for } x_i \leq \bar{x}_i \end{cases}$$

ここで x_i は株式 i の新規ポートフォリオのウェイトとし、 \bar{x}_i は株式 i の元のウェイトとし、 g_i は株式 i の購入に関する定数とします。

同様に、 q_i が株式 i の売却のコストとしますと以下のように表されます。

$$q_i = \begin{cases} (\bar{x}_i - x_i)i_i & \text{for } x_i < \bar{x}_i \\ 0 & \text{for } x_i \geq \bar{x}_i \end{cases}$$

ここで定数 h_i は売却と関係するとします。なお、 p_i と q_i 両者とも同時にゼロ以外にはなることはありませんのでご注意ください。なぜなら売却が生じる場合購入コストがゼロだからであり、逆の場合も同じだからです。

$\phi(x)$ が取引コストなしの最小化の目的関数とします。取引コストのある新規の目的関数は以下で表されます。

$$\phi(x) + \sum_{i=1}^n p_i + q_i = \phi(x) + \sum_{i=1}^n \max\{p_i, q_i\}.$$

上記の目的関数は平滑ではありません（微分不可能）。しかし、これはそれぞれの株式 i に対して新規変数 y_i を含むことにより平滑問題に変換できます。新しい目的関数は以下ようになります。

$$\varphi(x) + \sum_{i=1}^n y_i$$

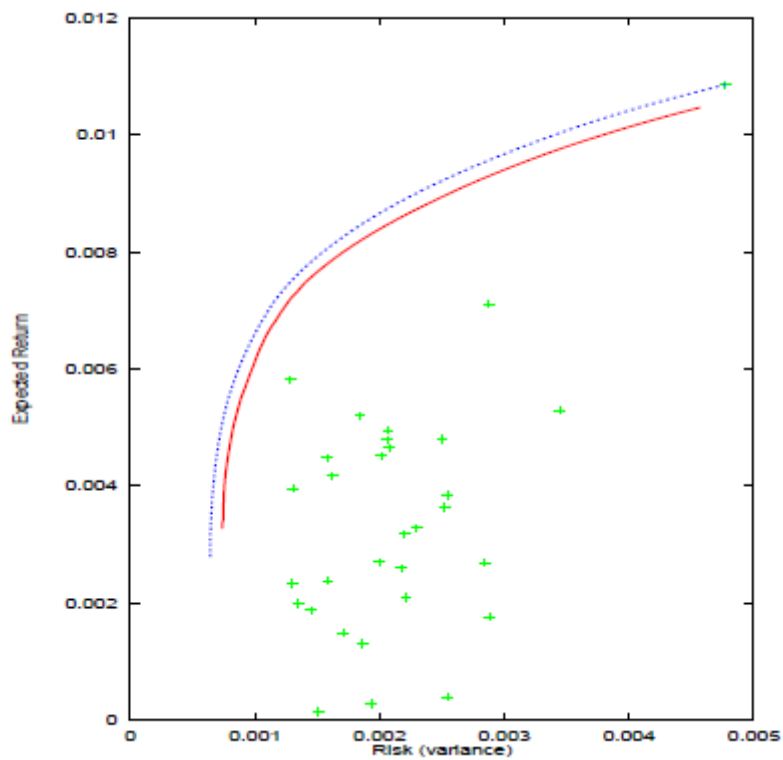


図8:取引コストがある場合／ない場合の効率的フロンティア

これは下記の制約を受けます。

$$p_i \leq y_i$$

及び

$$q_i \leq y_i$$

ここで、 y_i は株式 i の取引コストとし、 $i = 1$ から n まで繰り返します。上記2つの不等式は下記の形式で記述することができます。

$$-\infty \leq g_i x_i - y_i \leq g_i \bar{x}_i$$

$$h_i \bar{x}_i \leq y_i + h_i x_i \leq +\infty$$

残念ながら、このアプローチは最適化の変数の数を2倍にします。私たちはHang Sengの例の効率的フロンティアを、取引コストを含めるかたちで再計算しました。私たちはそれぞれの株式 i に対し p_i (購入コストを決定) が 0.0001 に等しく q_i (売却コストを決定) が 0.0002 に等しいと仮定しました。さらに、初期のポートフォリオがそれぞれの株式 i について等しいウェイト、 $x_i = 1/38$ 、であると仮定しました。

図 8 は取引コストがある場合/ない場合の効率的フロンティアを示しています。任意の分散では、期待リターン (任意の分散) における違いは取引コストに反映されます。

9 サンプルプログラム

実用的ポートフォリオ最適化を行うのに NAG ルーチンを使用するのは極めて簡単です。以下のサンプルルーチンは期待リターンが最小化される場合の (例えば、セクション 1 の問題 1)、最適なポートフォリオの計算のしかたを示しています。

このプログラムは、[2]で使われるデータを用いて実行されます。さらに、ユーザはポートフォリオの期待リターンを指定する必要があります。プログラムはポートフォリオの最適なウェイトと分散を出力します。

```

/* Example Program for Portfolio Optimization
*
* Copyright 2000 Numerical Algorithms Group Ltd.
*
* This example program computes the optimal weights
* of the equities for a specified expected return
* of the portfolio */
#include <Nag/nag.h>
#include <stdio.h>
#include <Nag/nag_stdlib.h>
#include <Nag/nag_string.h>
#include <Nag/nage04.h>
static void ex1(void);
static void ex1();
#define MAXN 50

```

```

#define MAXLIN 2
#define MAXBND MAXN+MAXLIN

main()
{
Vprintf("Portfolio optimization using e04nfc.¥n");
ex1();
exit(EXIT_SUCCESS);
}

static voidex1()
{
double x[MAXN], cvec[MAXN], s[MAXN];
double a[MAXLIN][MAXN], h[MAXN][MAXN];
double bl[MAXBND], bu[MAXBND];
double objf, up, low, ret, tem, tex, hh;
Integer tda, tdh;
Integer i, j, n, nclin, nbnd, ii, jj;
Boolean print;
Nag_E04_Opt options;
static NagError fail, fail2;
Vscanf("%^[¥n"); /* Skip heading in data file */
Vscanf("%^[¥n"); /* Skip heading in data file */
fail.print = TRUE;
fail2.print = TRUE;
/* Readthe actual problem dimension.
* n = the number of equities. */
Vscanf("%ld", &n);
/* nclin = the number of general linear constraints . */
nclin = 2;
nbnd= n + nclin;
tda = MAXN;
tdh = MAXN;
/* a = the linear constraint matrix A.
* bl = the lower bounds on x and A*x.
* bu = the upper bounds on x and A*x.
* x = the initial estimate of the solution.
* s = the standard deviation of return

```

```

For simplicity we assume that
lower bound on x is low
upper bound for x is up
for all x
Read low and up */
Vscanf("%*[\n]"); /* Skip heading in data file */
Vscanf("%lf%lf",&low,&up);
/* Set the initial weights to 1/n
set bl and bu
set the second row of A to unity
set the linear constraints bl[n+1] and bu[n+1] */
for (i = 0; i < n; ++i) {
x[i] = 1.0/n;
bl[i] = low;
bu[i] = up;
a[1][i] = 1.0;
}
bl[n+1] = 1.0;
bu[n+1] = 1.0;
/* Read the expected return and sd
The expected return is in first column of A */
Vscanf("%*[\n]"); /* Skip heading in data file */
for (i = 0; i < n; ++i) {
Vscanf("%lf%lf",&a[0][i],&s[i]);
}
/* Read the correlations
and form the covariance matrix */
Vscanf("%*[\n]"); /* Skip heading in data file */
for (i = 0; i < n; ++i)
for (j = i; j < n; ++j) {
Vscanf("%ld%ld%lf",&ii,&jj,&hh);
h[ii-1][jj-1] = hh*s[i]*s[j];
h[jj-1][ii-1] = hh*s[i]*s[j];
}
/* Read the required expected return of the portfolio
and set that in bl[n+1] and bu[n+1] */

```

```

Vscanf(" %*[^%n]"); /* Skip heading in data file */
Vscanf("%lf",&ret);
bl[n] = ret;
bu[n] = ret;
e04xxc(&options); /* Initialise options structure */
/* Set one option directly
* Bounds >= inf_bound will be treated as plus infinity.
* Bounds <= -inf_bound will be treated as minus infinity. */
options.inf_bound= 1.0e21;
/* Obtain remaining settings */
fail.print = TRUE;
20
print = TRUE;
e04xyc("e04nfc", "stdin", &options, print, "stdout", &fail);
/* Set the problem type and other optional settings*/
options.prob = Nag_QP1;
/* Solve the problem from a coldstart. */
if (fail.code == NE_NOERROR)
e04nfc(n, nclin, (double *)a, tda, bl, bu, cvec, (double *)h, tdh,
NULLFN, x, &objf, &options, NAGCOMM_NULL, &fail);
Vprintf("ExpectedReturn = %lf %n",ret);
Vprintf("ExpectedVariance = %lf %n",2.0*objf);
/* Free memory allocated by e04nfc to pointers in options */
e04xzc(&options, "all", &fail2);
if (fail.code != NE_NOERROR || fail2.code != NE_NOERROR) exit(EXIT_FAILURE);
} /* ex1 */
.

```

10 謝辞

Anne Trefethen, David Sayers, George Levy そして Neil Swindellsの皆様には論議いただき、心より感謝申し上げます。

参考文献

- [1] A. Björk. *Numerical Methods for Least Squares*. SIAM, Philadelphia, 1996.
- [2] T. J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha. Heuristics

for cardinality constrained portfolio optimization. *Computers and Operations Research*, 2000, to appear.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B (Methodological)*, 39:1-38, 1977.

[4] E. J. Elton and M. J. Gruber. *Modern Portfolio Theory and Investment Analysis*. Wiley, New York, 1995.

21

[5] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1996.

[6] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problem*. SIAM, Philadelphia, 1995.

[7] H. Markowitz. Portfolio selection. *Journal of Finance*, 7:77-91, 1952.

[8] H. Markowitz, W. F. Sharpe, and M. Miller. *Founders of Modern Finance: Their Prize Winning Concepts and 1990 Nobel Lectures*. AIMR, Charlottesville VA, 1991.

[9] H. M. Markowitz. *Mean-Variance Analysis in Portfolio Choice and Capital Markets*. Blackwell, Oxford, 1987.

[10] H. M. Markowitz. *Portfolio Selection: Efficient Diversification of Investments*. Blackwell, Oxford, 1991.

[11] R. O. Michaud. *Efficient Asset Management*. Harvard Business School Press, Boston, 1998.

[12] W. Morokoff. The Brownian bridge e-m algorithm for covariance estimation with missing data. *Journal of Computational Finance*, 2:75-100, 1998.

[13] W. F. Sharpe. *Portfolio Theory and Capital Markets*. McGraw-Hill, New York, 1999.

[14] W. F. Sharpe, G. J. Alexander, and J. V. Bailey. *Investments*. Prentice Hall, Upper Saddle River, NJ, 1998.

[15] S. A. Zenios. *Financial Optimization*. Cambridge University Press, Cambridge, 1993.