金融工学アプリケーションにおける数理計画法の実際

田辺隆人
（株）数理システム

東京都新宿区新宿 2-4-3 フォーシーズンビル 10F
tanabe@msi.co.jp

概要

ポートフォリオ最適化, イールドカーブフィッティング, あるいは倒産判別など, 金融工学アプリケーションにおいて数理計画法は広く応用されている．ここでは, 実務の現場で現れる様々な問題と数理計画法アルゴリズムとの適合性, パフォーマンス改善のためのテクニックを紹介し, 数理計画アルゴリズムの実装を行う立場からの解説を試みる.

Keywords: 二次計画法，線形計画法，整数計画法，半正定値計画法，ポートフォリオ最適化，非線形フィッティング

1　二次計画法の適用

1.1　ファクターモデル

x_j（$j \in Asset$）が各銘柄の投資比率を示すとしてポートフォリオの収益率は

$$R(x) \equiv \sum_{j \in Asset} r_j x_j \qquad (1)$$

と表わされる．ここで，$Asset$ は投資可能な銘柄の集合，r_j（$j \in Asset$）は各銘柄の収益率（確率変数）である．x_j（$j \in Asset$）が投資比率を表すことから

$$\sum_{j \in Asset} x_j = 1 \qquad (2)$$

という制約は前提となる．また，３章で議論するロングショートモデルのケースを除いて,

$$x_j \geq 0 \qquad j \in Asset \qquad (3)$$

であることを前提とする.

r_j（$j \in Asset$）が確率変数であることから，その線形関数である $R(x)$ も確率

変数となる. ポートフォリオの好ましさを定量化する尺度として, 例えば $R(x)$ の期待値：

$$E\left[R(x)\right] \equiv \sum_{j \in Asset} \overline{r}_j x_j \qquad (4)$$

と収益率の分散：

$$V\left[R(x)\right] \equiv \sum_{i,j \in Asset} Q_{ij} x_i x_j \qquad (5)$$

を採用する. ここで, $\overline{r}_j$（$j \in Asset$）は各銘柄の収益率の期待値, Q_{ij}（$i, j \in Asset$）は各銘柄の収益率の分散共分散行列である.

r_M（ポートフォリオの収益率期待値の下限値）を所与として $E\left[R(x)\right]$ を

$$E\left[R(x)\right] \geq r_M \qquad (6)$$

と制約し, $V\left[R(x)\right]$ を最小化するような x_j（$j \in Asset$）を求めるには二次計画問題を解けばよい. この定式化はマルコビッツモデルとして知られる.

実務においては, 各銘柄の収益率 r_j を経済指標などの説明変数 f_k（ファクター）によって

$$r_j = \alpha_j + \sum_k \beta_{jk} f_k + \varepsilon_j \qquad k \in Factor,\ j \in Asset,\ \varepsilon_j \in N(0, \sigma_j^2) \qquad (7)$$

と記述する.（7）を用いれば収益率の分散は次のように表現される.

$$V\left[R(x)\right] = \sum_{i,j \in Asset} \sum_{k,l \in Factor} Q^f_{k,l} \beta_{j,k} \beta_{j,l} x_i x_j + \sum_{j \in Asset} \sigma_j^2 x_j^2 \qquad (8)$$

（5）と（8）を比較すると

$$Q_{i,j} = \begin{cases} \sum_{k,l \in Factor} Q^f_{k,l} \beta_{j,k} \beta_{j,l} & (i \neq j) \\ \sum_{k,l \in Factor} Q^f_{k,l} \beta_{j,k} \beta_{j,l} + \sigma_j^2 & (i = j) \end{cases} \qquad i, j \in Asset \qquad (9)$$

であるが, ファクターモデルを実際に解く際には, 中間変数 s_k（$k \in Factor$）と, その定義に等価な

$$s_k = \sum_j \beta_{j,l} x_j \qquad (10)$$

なる線形制約式を導入し（8）を

$$V\left[R(x)\right]=\sum_{k,l}Q^{f}_{k,l}s_k s_l+\sum_{j}\sigma_j^2 x_j^2 \qquad (11)$$

と表現するのが計算の効率化に貢献する．以下は実務的な規模（3156 銘柄，ファクター数 48）において，目的関数として（9）から得られる（5）をそのまま使用した場合と，（10）を導入して（11）を使用した場合における二次計画問題の内点法，有効制約法による求解例である．なお，ここではリスク最小化問題とし，（6）の制約は除いている．

本稿における計算機実験は特に断りがないかぎり，

CPU: Core2 1.6GHz

メモリ：2G バイト

にて行っている．また，数理計画法パッケージとしては NUOPT（バージョン 9）を用いている．

表 1　定式化の異なるファクターモデルの計算結果

	内点法		*有効制約法*	
	計算時間	メモリ所要	計算時間	メモリ所要
（5）を使用	201 秒	550M	74 秒	550M
（11）を使用	1 秒	30M	0.75 秒	30M

問題全体を表現するのに必要なメモリ量の差を反映して，（11）を使用する定式化は（5）に比べて高速かつ省メモリである．

1.2　有効制約法と内点法

表 1 の結果では有効制約法が内点法よりも速い．これは，最適解において正の値を取っている x_j のコンポーネントが 196 個（全体の 6%程度）と少ないことに依る．言い換えると変数の非負制約のうち大部分が最適解において有効となっており，有効制約法が同時に考慮すべき非零の変数の数は少なく済むので，有効制約法の探索は高速である．一方，主・双対変数のすべてを正に拘束し，制約の有効性を意識しない内点法では，このような問題の性質を利用することができないので，効率上不利となる．

しかしながら，有効制約法はポートフォリオモデルにおいて常に内点法に対して優位ということではない．

インデックス運用の場合には（6）の下限制約の下でベンチマークポートフォリオ x_j^B との収益率の差の分散を最小化する問題を解く．具体的には

$$V\left[R(x)-R(x^B)\right]=\sum_{i,j\in Asset}\sum_{k,l\in Factor}Q^{f}_{k,l}\beta_{j,k}\beta_{j,l}(x_i-x_i^B)(x_j-x_j^B)+\sum_{j\in Asset}\sigma_j^2(x_j-x_j^B)^2$$

（１２）

が目的関数となる.

次は組み入れ銘柄数が 1329 であるようなx_j^Bを与え，（６）の下限制約の右辺を一定の刻みずつ増大させた三通りの設定に関する実験結果である.

表 2　（１２）を目的関数とした場合の結果

	組み入れ銘柄数	*内点法*	*有効制約法*
設定１	782	1.0 秒	2.6 秒
設定２	404	1.0 秒	1.2 秒
設定３	179	1.0 秒	0.8 秒

有効制約法の計算時間が最適解における組み入れ銘柄数に対して敏感に変動すること，一方で内点法は安定的なパフォーマンスを与えることがわかる.

2　線形計画法の適用

2.1　複雑なリスク構造を持つアセットの問題

１章に現れたQ_{ij}や$\overline{r}_j$（$i, j \in Asset$）は，各銘柄の収益率r_j（$j \in Asset$）の分布を多次元正規分布と仮定した場合のパラメータで，その分布およびパラメータの値を所与の入力としてポートフォリオを最適化している．一方で信用リスクや多期間モデルなどの複雑なリスク構造を持つアセットを含む問題の場合には収益率の分布を仮定せず収益率サンプル点r_j^k（$j \in Asset$，$k \in Sample$）をモンテカルロシミュレーションやヒストリカルデータ，あるいはシナリオデータから得て，各サンプル点におけるポートフォリオの収益率：

$$R^k(x) \equiv \sum_{j \in Asset} r_j^k x_j \qquad k \in Sample \qquad （１３）$$

を陽に定式に組み込む方法が提案されている.

（１３）を関数の列ととらえ，数理計画法のモデリング技法として知られた方法を適用すれば，複雑なリスク構造を持つアセットを含むポートフォリオ最適化問題を記述することができる.

表 3 ポートフォリオのリスク尺度と定式化技術の対応

リスク尺度	*対応する定式化技術*
絶対偏差	$\left\|R^k(x) - E\left[R^k(x)\right]\right\|$ の和の最小化
下方リスク	$R^k(x)$ で負のものの絶対値の和の最小化
一次の下方積分積率	$R^k(x)$ が所与の定数値を下回るものの和の合計の最小化
CVaR 最小化	$-R^k(x)$ の最大から p（所与）個の平均値の最小化
Maximum drawdown	$R^k(x)$ の部分列の最大および最小の差

（１３）が線形であり，上記の定式化技法がいずれも線形計画法の範疇にて表現できるため，表 3 の一連のリスク尺度によるポートフォリオ最適化は線形計画法によって可能である．ただし，分布を詳細に記述するためにはサンプル点 $|Sample| >> |Asset|$ と設定されるので, 解くべき問題のサイズは通常大規模となる.

2.2 CVaR 最小化問題

CVaR 最小化の元になっている定式化技法は，関数列 $R^k(x)$（$k \in Sample$）の最大から p（定数）個の値の平均値を最小化するというものである．具体的には

非負制約を持つ，スラック変数 s_k（$k \in Sample$）と自由変数 α を導入,

$$\frac{1}{p}\sum_{k \in Sample} s_k + \alpha \qquad (14)$$

を目的関数とし,

$$s_k + \alpha \geq -R^k(x) \qquad k \in Sample \qquad (15)$$

を制約として追加する． p の値は CVaR の閾値 β から算出する．具体的には β-CVaR であれば， $p \leftarrow (1-\beta)\cdot|Sample|$ と設定する．
次は $|Asset|=3$ のデータに関する計算例である．ただし， $\beta = 0.99$ とした．

表 4　CVaR 最小化問題の計算時間

$\|Sample\|$	*内点法*	*単体法*
5000	0.6 秒	3.3 秒
10000	2.1 秒	24.3 秒
20000	4.7 秒	107.5 秒
50000	12.9 秒	——
100000	42.3 秒	——

単体法が低速な理由は問題の性質による．最適解においては（１５）の制約式の $(1-\beta)\cdot|Sample|$ 個のみが有効であり, 問題を標準形に変形する際に導入される（１５）の各制約式に対応するスラック変数の大部分が非零となるためである.

このようなケースにおいては，双対問題を解くようにするのが単体法の効率向上には効果的である[2]. 以下は CVaR 最小化問題の双対問題の計算例であるが，一部単体法が内点法よりも高速となっている.

表 5　CVaR 最小化問題の双対問題の計算時間

$\|Sample\|$	*内点法*	*単体法*
5000	0.4 秒	0.2 秒
10000	1.5 秒	0.7 秒
20000	4.0 秒	2.6 秒
50000	18.6 秒	15.0 秒
100000	41.7 秒	67.9 秒

2.3　効率的フロンティア曲線の描画

実際の投資判断においては，（6）の制約の右辺を変化させて得られる複数の最適ポートフォリオの情報が必要である．それら複数の最適ポートフォリオの収益率の平均値とリスクをプロットしたのが効率的フロンティアである．効率的フロンティアの描画のためには，収益率の制約の右辺を一定の刻みで変化させて繰り返し求解を行うことになるので，単体法（・有効制約法）を用いているのならば，二度目以降の最適化時には前回の最適解において有効であった制約の情報（基底情報）を参考にして探索を行う（リスタート）のが有利である．しかし，例えば CVaR 最小化問題は単体法の効率が悪いため，一連の計算の出発点となる基底情報を得るのに時間を所要してしまう．そのような場合に用いられるのは，最初の計算の際に，内点法の解から最適な基底解を得て（クロスオーバー），それを元にリスタートを行うという方法である．以下は，$|Sample|=10^5$ の問題について，16 点からなる効率的フロンティアを求めるのに所要した時間である.

表 6　CVaR 最小化問題の効率的フロンティア描画のための計算時間

クロスオーバー	0.1 秒
リスタート時間平均	23.7 秒
リスタート時間最小	12.8 秒
リスタート時間最大	24.9 秒
総合計	397.8 秒

クロスオーバーによる最適な基底解の構成は十分高速である．不利な定式化でも基底情報が存在すれば，単体法は比較的高速に動作することがわかる.

3　混合変数計画法の適用

3.1　ロングショートモデル

実務家がポートフォリオを継続的に保守する目的においては，現在の組み入れ比率からの変化量（リバランス）が数理計画法の出力であることが望ましい．さらに，買いポジション（long）の他に売りポジション（short）を取り得るような投資戦略が一般的になりつつあり，伴って（２），（３）は必須の制約式ではなくなっている．
このような場合には，１章で紹介したファクターモデルにおける組み入れ比率 x_j（$j \in Asset$）を，４つの非負変数

x_j^{long}（$j \in Asset$）：組み換え後の正の投資比率の大きさ

x_j^{short}（$j \in Asset$）：組み換え後の負の投資比率の大きさ

d_j^{buy}（$j \in Asset$）：購入量

d_j^{sell}（$j \in Asset$）：売却量

によって

$$x_j \equiv x_j^{long} - x_j^{short} = x_j^0 + d_j^{buy} - d_j^{sell} \quad （j \in Asset）　　（１６）$$

と記述した問題を解く．意味的に x_j^{long} と x_j^{short}，d_j^{buy} と d_j^{short}（$j \in Asset$）の組は同時に正にはなれない（相補性制約）とする．相補性制約の導入方法は数理計画モデルの作成において重要である．x_j^{long} と x_j^{short}，d_j^{buy} と d_j^{short} が，例えば $x_j^{long} + x_j^{short}$ あるいは $d_j^{buy} + d_j^{sell}$ のような形で組になって，最小化される目的関数や上限制約の左辺にのみ現れるのであれば，陽に相補性を制約として導入する必要はない．しかし実際には，ポジションの総量制約：

$$\sum_j x_j^{long} = LT, \quad \sum_j x_j^{short} = ST　　（１７）$$

さらに，取引要件を表現するうえで d_j^{buy} あるいは d_j^{short} のみを含む線形制約が存

在するため, 銘柄 j について０－１整数変数 $\delta_j^{long}, \delta_j^{buy} \in \{0,1\}$（$j \in Asset$）を導入,

U_j^{long}, U_j^{short} を最大の正・負の組み入れ比率，U_j^{buy}, U_j^{sell} を最大の購入・売却量とするとき

$$x_j^{long} \le U_j^{long}\delta_j^{long},\ x_j^{short} \le U_j^{short}(1-\delta_j^{long}) \qquad (j \in Asset) \qquad (18)$$

$$d_j^{buy} \le U_j^{buy}\delta_j^{buy},\ d_j^{sell} \le U_j^{sell}(1-\delta_j^{buy}) \qquad (j \in Asset) \qquad (19)$$

なる制約を通じて０－１整数変数と組み入れ比率を表現する変数を連動させ, 全体を混合整数二次計画問題として定式化する必要が生じる.

3.2　ロングショートモデルの補強

前述の問題は分枝限定法を用いて求解することになるが，最適解における組み入れ銘柄数が少ない状況，例えば収益率の制約が厳しい場合を除いて最適解を得るのが困難である．実務的な投資制約や線形な取引コストを考慮した，銘柄数 894 の問題を混合整数二次計画問題によって求解を試みたところ，収益率を最大化する設定の場合には 15.4 秒で最適解が得られる一方，リスクを最小化する設定の場合には複数の暫定解は得られるものの，最適解が得られないまま，探索木が発散してメモリーエラーを起こしてしまう.
分枝限定法のパフォーマンス向上のためには，制約式を追加することによって，緩和解を暫定解に近づける「補強」が有効である．このモデルの場合には，現在組み入れが正，負であるような銘柄のインデクスの集合をそれぞれ $Long$, $Short$

$$Long \equiv \{j \mid x_j^0 > 0\},\ Short \equiv \{j \mid x_j^0 < 0\} \qquad (20)$$

と定義し，インデクスが $Long$, $Short$ に属する銘柄毎に

$$d_j^{buy} \le x_j^{long} \le x_j^0 + d_j^{buy},\ x_j^{short} \le d_j^{sell} \qquad , j \in Long \qquad (21)$$

$$d_j^{sell} \le x_j^{short} \le -x_j^0 + d_j^{sell},\ x_j^{long} \le d_j^{long} \qquad , j \in Short \qquad (22)$$

なる制約を補強する．これらは購入量・売却量と組み換え後の保有比率の間に常に成り立つ関係を示している.

さらに x_j^{long} と x_j^{short} の相補性から,

$$(x_j)^2 = (x_j^{long})^2 + (x_j^{short})^2 \qquad (23)$$

となる関係を用いて，$(x_j)^2$ の項を記述する．

これらの処置を行うと，リスク最小化，収益率最大化など問題の設定にかかわらずおおむね 30 秒程度で最適解を求めることができる．補強後の問題に対する 16 点から成る効率的フロンティア曲線（図 1）を描くのに必要な計算時間を以下にまとめる．

表 6　補強されたロングショートモデルの計算時間
（16 点の効率的フロンティア描画）

計算時間平均	23.7 秒
計算時間最小	11.2 秒
計算時間最大	34.0 秒
総時間合計	392.3 秒

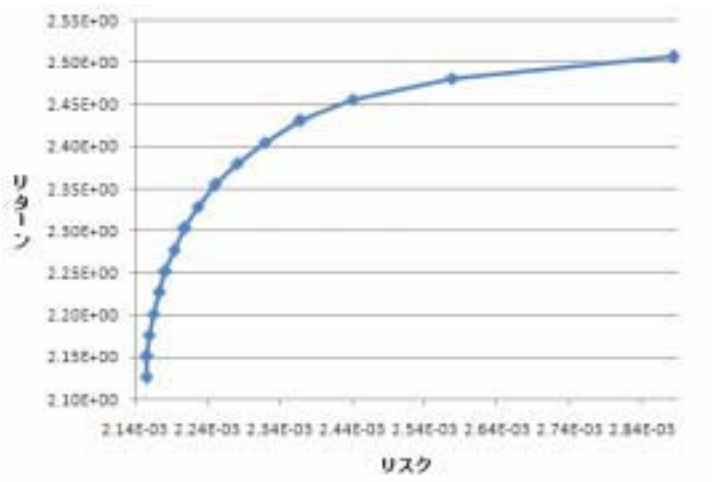

図 1　ロングショートモデルの効率的フロンティア

4　非線形計画法の適用

4.1　フィッティング問題

m 個の観測点 x_i（$i \in Sample$）に対する観測値 y_i（$i \in Sample$），観測値を説明とする n 個のパラメータを持つ一般に非線形な数式モデル $f(x_i; a_1, \cdots, a_n)$ が与えられているとき，パラメータ $a_1, \cdots, a_n$ を推定する問題は金融工学のさまざまなアプリケーションに現れる．

変数：　　$a_1, \cdots, a_n$

最小化：　$\sum_i e_i^{\,2}$

制約：　$e_i = f(x_i;a_1,\cdots,a_n) - y_i,\qquad i \in Sample$　　（２４）

以下は$n=391$，$m=180$で$f(x_i;a_1,\cdots,a_n)$が多項式，分数および指数関数によって表わされる具体例の計算結果である．一階・二階微係数の情報を得て信頼領域法に基づく主双対内点法[1]を適用した場合と，一階微係数の情報のみから準ニュートン法を用いて直線探索法を適用した場合を比較している．

表７　実務的なフィッティングモデルにおける計算結果

方法	*反復回数*	*停留条件残差*	*誤差二乗和*	*計算時間*
二階微係数	112 回	5.0e-7	1257.6	67.8 秒
一階微係数のみ	300 回	1.8e-2	2452.8	105.5 秒

一階微係数のみを利用した場合では反復回数がここで設定した上限（３００回）に達しても停留条件の残差が十分に下落していないことがわかる．

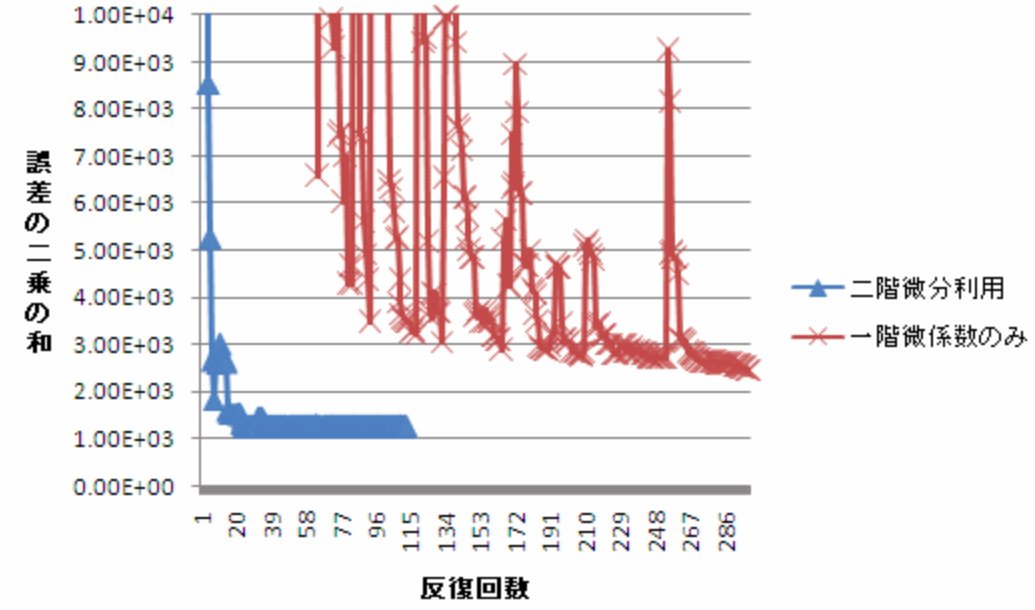

図 2　表 7 のケースにおける反復と誤差の二乗和のプロット

非線形関数の二階微係数の情報が計算の精度と安定性に大きく寄与する．このことから，非線形関数の二階微係数の計算は，非線形最適化のパフォーマンスに大きく貢献することがわかる．

4.2　誤差の尺度の変更

前述のフィッティング問題で，外れ値データに対する結果の安定性を向上させ

るため，誤差を二乗ではなく絶対値に変更した場合の影響について述べる．最も簡便には目的関数を

$$\text{最小化} \quad \sum_i |e_i| \qquad (25)$$

に変更することが考えられるが，絶対値関数は原点付近で微分不可能となるため，非線形計画法アルゴリズムの効率は一般に下落する．
この方法と絶対値関数を用いない次のような定式化：

$$\text{最小化} \quad \sum_i s_i$$

$$\text{制約} \quad s_i \geq f(x_i; a_1, \cdots, a_n) \geq -s_i, \quad s_i \geq 0 \qquad (26)$$

とのパフォーマンスを比較したのが表 8 である．

表 8　絶対値関数と中間変数を用いたケースの比較

方法	*反復回数*	*停留条件残差*	*誤差二乗和*	*計算時間*
絶対値関数導入	300 回	1.0e-1	5285.1	236.2 秒
中間変数導入	150 回	1.8e-7	1889.0	155.9 秒

絶対値関数を利用した場合では反復回数がここで設定した上限（３００回）に達しても停留条件の残差が十分に下落していないことがわかる．

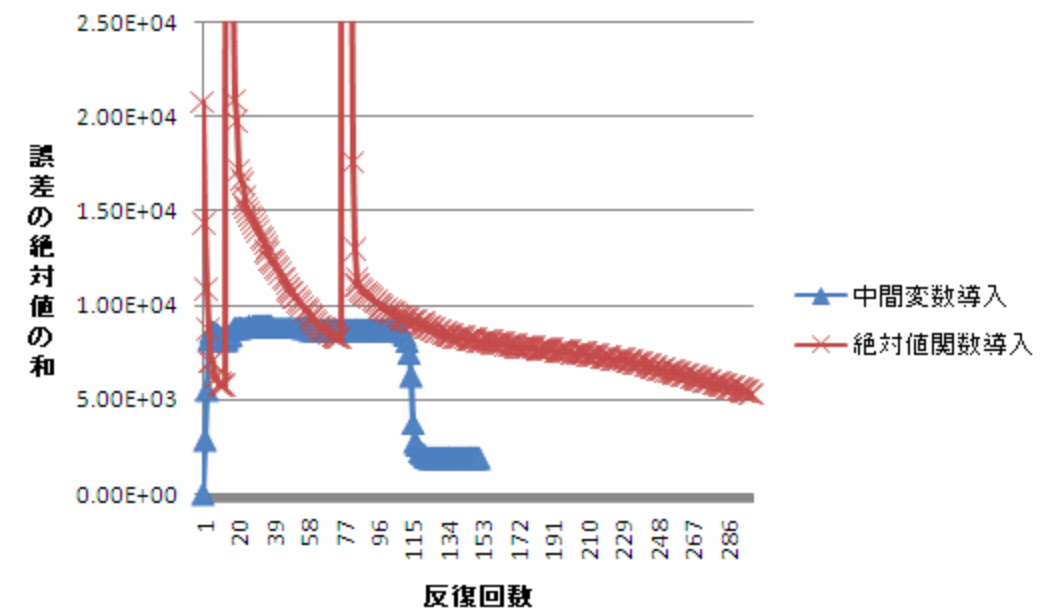

図 3　表 8 のケースにおける反復と誤差の二乗和のプロット

中間変数を導入し，制約式が増えても絶対値関数を排除する効果が顕著に認められる．

5　非線形半正定値計画法の適用

5.1　倒産判別における二次ロジットモデル

倒産判別における二次ロジットモデルの推定を行う際に，二次の係数に対して半正定値制約を課すことによって，モデルの予測力を向上させることができる[3]．[3]では二次計画問題の列を生成する切除平面法によって，半正定値制約を考慮しているが，非線形半正定値計画法[4]を適用すると，二次計画問題を繰り返し解く必要がないため，切除平面法に比べて大幅に高速化される．以下は実務におけるインスタンス（パラメータ数 8，データ件数 6084）についての結果である．

なお，以降の実験結果は

CPU:Pentium4（クロックスピード 3.2GHz）

メモリ：1GBytes

で行ったものである．NUOPT は次期リリース予定のバージョン１０のプロトタイプコードを用いている．

表 9　切除平面法と半正定値計画法との比較

方法	*尤度*	*計算時間*
切除平面法	143.745	30.3 秒
一階微係数のみ	143.747	1.5 秒

5.2　相関行列取得問題

与えられた対称行列Gに，最もフロベニウスノルムの意味で近い半正定値対称行列Xを求めるという問題：

$$\text{最小化} \qquad \|G-X\|_F$$

$$\text{制約} \qquad X \succ 0 \qquad (27)$$

は，正規乱数発生のための相関行列の取得に利用可能な問題であり，非線形半正定値計画法をそのまま応用することが可能である．計算結果を以下に示す．

表１０　相関行列取得問題の計算結果

G のサイズ	*反復回数*	*計算時間*
10	16 回	0.1 秒
20	18 回	0.4 秒
30	15 回	2.3 秒
40	17 回	10.4 秒
50	17 回	33.1 秒
60	16 回	72.2 秒

70	18 回	139 秒
80	15 回	259 秒
90	17 回	453 秒
100	17 回	763 秒

決定変数の数がGのサイズについて二乗のオーダーで増大するため，計算時間は急激に増大するが，GとXの要素が異なってもよいものを限定するなどの制約を加えた場合には計算時間の増大は上記よりも緩和される．

参考文献

[1] H. Yamashita, H. Yabe,T. Tanabe: A globally and superlinearly convergent primal-dual interior point trust region method for large scale constrained optimization, Mathematical Programming Ser.B, Vol.102, No.1, pp.111-151 (2005)

[2] 枇々木規雄，田辺隆人，ポートフォリオ最適化と数理計画法，朝倉書店 (2005)

[3] Konno, H., Kawadai, N.,Wu, D.: Estimation of failure probability using semi-definite Logit model. Computational Management Science 1, 59-73 (2004)

[4] Hiroshi Yamashita, Hiroshi Yabe, and Kouhei Harada, A primal-dual interior point method for nonlinear semidefinite programming, Technical Report Mathematical Systems Inc. (2006, Revised 2007)