非凸型取引コスト下でのポートフォリオ最適化における整数計画法と分枝限定法の比較

Comparison of Branch and Bound Method and Integer Programming in Protfolio Optimization under Nonconvex Transaction Cost

経営システム工学専攻　今野研究室　泉　一雄　Izumi Kazuo

1　はじめに

近年、最適化の世界では、計算機とアルゴリズムの進歩のおかげで、一昔前には考えられなかったような複雑な問題が実用的な時間で解けるようになってきている。しかし、計算機とアルゴリズムが進歩したことと同様に、OR の研究対象である社会システムの複雑さが増してきているのも確かである。また、システムの複雑さに反して、近年は社会全体がスピードを重視する傾向も持っている。そのため、社会の要求に応えるべく、複雑化したシステムを最適化する際には、解の正確さと供に計算速度を向上させる必要がある。

そこで、本論文では、一般的に大規模問題が容易に解けなかったり、計算に大きな時間がかかると言われている非凸型コスト関数を含んだ問題に焦点を当て研究を行っている。具体的には、非凸型取引コスト付きポートフォリオ問題を題材にして、正確でより速く解ける方法の提案を行う。

2　定式化

$$
\begin{array}{ll}
\text{最大化} & \displaystyle\sum_{j=1}^{n}\{r_j x_j - c_j(x_j)\} \\
\text{条件} & \displaystyle\sum_{t=1}^{T} f_t y_t \le wM \\
 & \displaystyle y_t - \sum_{j=1}^{n}(r_{jt} - r_j)x_j \ge 0,\ t = 1,\cdots,T \\
 & y_t \ge 0,\ t = 1,\cdots,T \\
 & \displaystyle\sum_{j=1}^{n} x_j = M \\
 & 0 \le x_j \le u_j,\ j = 1,\ldots,n
\end{array}
\qquad (1)
$$

ただし、

r_j:期待収益率、 $c_j(x_j)$:取引コスト、w:リスク

M:投資金額、u_j:投資上限

3　解法

式 (1) を解くには、整数計画法と分枝限定法の 2 つの方法が存在する。ここでは、それらの解法について解説する。

3.1　整数計画法による定式化

$$
\begin{array}{ll}
\text{最大化} & \displaystyle\sum_{j=1}^{n}\{r_j x_j - c_j(x_j)\} \\
\text{条件} & \displaystyle\sum_{t=1}^{T} f_t y_t \le wM \\
 & \displaystyle y_t - \sum_{j=1}^{n}(r_{jt} - r_j)x_j \ge 0,\ t = 1,\cdots,T \\
 & y_t \ge 0,\ t = 1,\cdots,T \\
 & \displaystyle\sum_{j=1}^{n} x_j = M \\
 & 0 \le x_j \le u_j,\ j = 1,\ldots,n \\
 & \displaystyle x_j = \sum_{l=1}^{k_j} \lambda_{jl} x_{jl} \\
 & \displaystyle c_j(x_j) = \sum_{l=1}^{k_j} \lambda_{jl} c_j(x_{jl}) \\
 & \displaystyle\sum_{l=1}^{k_j} \lambda_{jl} = 1 \\
 & \lambda_{jl} \ge 0,\ l = 1,\cdots,k_j \\
 & \lambda_{j1} \le \phi_{j1} \\
 & \lambda_{j2} \le \phi_{j1} + \phi_{j2} \\
 & \cdots\cdots\cdots\cdots\cdots \\
 & \lambda_{jk_j} \le \phi_{jk_{j-1}} + \phi_{jk_j} \\
 & \displaystyle\sum_{l=1}^{k_j} \phi_{jl} = 1,\ 0 \le \phi_{jl} \in Z,\ l = 1,\cdots,k_j
\end{array}
\qquad (2)
$$

3.2 分枝限定法アルゴリズム

Step1. $\Gamma = (P_0)$, $\hat{f} = -\infty$, $k = 0$, $\varepsilon > 0$ とする。

Step2. $\Gamma = \phi$ なら *step*9 へ、そうでなければ *step*3 へ。

Step3. Γ の中から1つの問題を選び出し、それを (P_k) とする 。

$$(P_k) \left| \begin{array}{ll} \text{最大化} & f(\boldsymbol{x}) = \displaystyle\sum_{j=1}^{n} \{r_j x_j - c_j(x_j)\} \\ \text{条件} & (\boldsymbol{x}, \boldsymbol{y}) \in F \\ & \boldsymbol{\beta}^k \le \boldsymbol{x} \le \boldsymbol{\alpha}^k \end{array} \right.$$

$\Gamma = \Gamma \backslash (P_k)$

Step4. $c_j(x_j)$ を領域 $\beta_j^k \le x_j \le \alpha_j^k$ 上で下方線形近似した関数 $c_j^k(x_j)$ とし、線形計画問題

$$(Q_k) \left| \begin{array}{ll} \text{最大化} & g_k(\boldsymbol{x}) = \displaystyle\sum_{j=1}^{n} \left\{r_j x_j - c_j^k(x_j)\right\} \\ \text{条件} & (\boldsymbol{x}, \boldsymbol{y}) \in F \\ & \boldsymbol{\beta}^k \le \boldsymbol{x} \le \boldsymbol{\alpha}^k \end{array} \right.$$

を解く。(Q_k) が実行不能であれば、*step*2 に戻る。そうでないときは (Q_k) の最適解を $\boldsymbol{x}^k$ とする。

$\mid g_k(\boldsymbol{x}^k) - f(\boldsymbol{x}^k) \mid > \varepsilon$ なら *step*7 へ。そうでないときは $f_k = f(\boldsymbol{x}^k)$ とする。

Step5. $f_k < \hat{f}$ なら *step*7 へ。そうでないときは *step*6 へ。

Step6. $\hat{f} = f_k$, $\hat{\boldsymbol{x}} = \boldsymbol{x}^k$ とする。
$g_t(\boldsymbol{x}^t) \le \hat{f} + \varepsilon$ であるような子問題 (P_T) を Γ から削除する。

Step7. $g_k(\boldsymbol{x}^k) \le \hat{f} + \varepsilon$ なら *step*2 へにいく。そうでなければ *step*8 へいく。

Step8. $c_s(x_s^k) - c_s^k(x_s^k) = max \left\{c_j(x_j^k) - c_j^k(x_j^k) \,|\, j = 1, \cdots, n\right\}$

$$X_{l+1} = X_k \cap \left\{\beta_s^k \le x_s \le x_s^k\right\}$$
$$X_{l+2} = X_k \cap \left\{x_s^k \le x_s \le \alpha_s^k\right\}$$

とおいて２つの子問題

$$(P_{l+1}) \left| \begin{array}{ll} \text{最大化} & f(\boldsymbol{x}) \\ \text{条件} & (\boldsymbol{x}, \boldsymbol{y}) \in F \\ & \boldsymbol{x} \in X_{l+1} \end{array} \right.$$

$$(P_{l+1}) \left| \begin{array}{ll} \text{最大化} & f(\boldsymbol{x}) \\ \text{条件} & (\boldsymbol{x}, \boldsymbol{y}) \in F \\ & \boldsymbol{x} \in X_{l+2} \end{array} \right.$$

を生成し、
$\Gamma = \Gamma \cup \{(P_{l+1}), P(_{l+2})\}$
として *step*3 に戻る。

Step9. 終了 ($\hat{\boldsymbol{x}}$ が (P_0) の ε 最適解となる)。

3.3 幅優先と深さ優先分枝限定法

ここでは、本論文で用いている幅優先分枝限定法と深さ優先分枝限定法について触れておく。

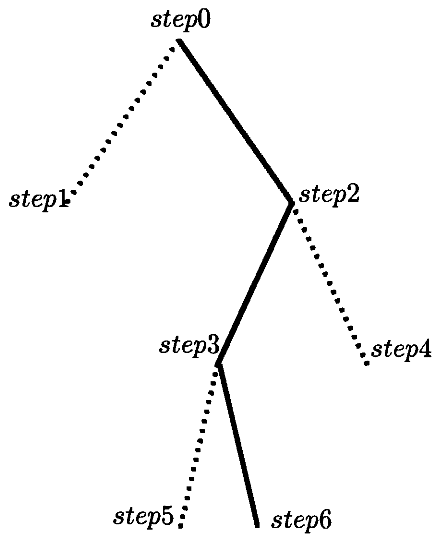

図１：深さ優先分枝限定法の経路

２つの手法のアルゴリズムは前節で紹介したものと本質的には同じものである。異なっている点はノードの選択方法である。幅優先分枝限定法の場合は、選択できるノードの中で $g_k(\boldsymbol{x})$ が最も大きいノードを選択していく。これに対して、深さ優先分枝限定法は、現在のノードの直下にある２つのノードの中から $g_k(\boldsymbol{x})$ の値が大きい方を選択して進んでいくという手法である（図２を参照）。

3.4 深さ優先分枝限定法における限定操作緩和法

この方法は限定操作の緩和の度合いを変えることでより速い時間で解を求めようとする方法である。まずは、限定操作の構造に触れることにする。

●限定操作の構造
大まかに述べると、以下の２つの限定操作を行っている。１つ目は、コストに関する許容誤差が満たされた $(g_k - f_k \leq \varepsilon)$ 後に行う限定操作（以後、"限定操作その１"と呼ぶ)、２つ目はコストに関する許容誤差を満たさなかった $(g_k - f_k \geq \varepsilon)$ 後に行う限定操作（以後、"限定操作その２"と呼ぶ）である。以下にそれぞれの限定操作に対応する式を書く。
＜限定操作その１＞

$$f_{max} + \eta \times \varepsilon \geq g_k \tag{3}$$

を満たすなら、子問題 k を削除する。ただし、$\eta > 0, \varepsilon > 0$ である。

＜限定操作その２＞

$$f_{max} + \gamma \times \varepsilon \geq g_k \tag{4}$$

を満たすなら、子問題 k を削除する。ただし、$\gamma > 0, \varepsilon > 0$ である。
式 (3.14),(3.15) 中の ε は限定操作を緩和するための基準となる誤差で、η, γ の値を変えることによって、緩和の度合いを変更できる（通常、η, γ)。

幅優先分枝限定法がほぼ"限定操作その１"しか行われていないのに対して、深さ優先分枝限定法が"限定操作その１"、"限定操作その２"共に行われていることが経験的に分かっている。このことを利用して、深さ優先分枝限定法の"限定操作その２"の γ を大きくしてやることで、計算速度を向上させる方法を本論文では扱っている。

4 考察

本研究は、取引コストを考慮した平均・絶対偏差モデルを分枝限定法・整数計画法を用いて解き、その計算面での振舞いを考察したものである。特に分枝限定法の分枝操作には、幅優先、深さ優先の両方を用いており、最適化解をはじめに見つかった暫定解の比較も行っている。　今回の実験で分析したデータは、1991 年 2 月から 2002 年 11 月までの間上場していた東証 1 部銘柄 900 社の月次収益率データであり、この期間から 10 セットのデータを作成した。また、変化させたパラメータは以下の通りである。

表 1 : パラメータ (%)

パラメータ	選択肢
期間T	24、36、48、60
ユニバースn	300、600、900、
ターゲットリスクw	2.5%、3.0%、3.5%
投資上限u	2.0%、2.5%
収束誤差 ε	10^{-3}、10^{-4}、10^{-5}

ここで、投資金額Mは20億円であり、投資上限 u=2.0 %の場合には個別銘柄の投資上限金額が 4000 万円、投資上限 u=2.5 %の場合には個別銘柄の投資上限金額が 5000 万円となる。また、整数計画法の収束誤差は、最適化ソフト CPLEX のデフォルト値である 10^{-4} としている。計算した組合わせは 4 × 3 × 3 × 2 × 3 × 10=2160 通りである。

各手法がどのように振舞うかをみるために、全てのパラメータについての計算結果を足し合せたものを表 2 に示す。

表 2 : 全体を考慮した各手法の結果 (%)

	深さ	深さの暫定解	cut緩和法	幅	整数計画法
目的関数の相対誤差(%)	0.01	0.03	0.01	0.01	0.00
計算時間(sec)	4130.62	521.88	1723.97	3140.61	6652.57

また、表 2 をグラフにしたものを図 3 に示す。

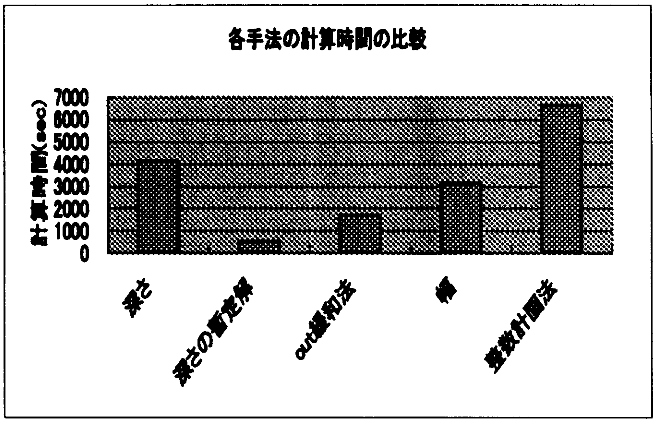

図 2：全体を考慮した各手法の結果 (%)

全体的な傾向をまとめると、計算速度は、
深さの暫定解＜ cut 緩和法＜幅優先＜深さ優先＜整数計画法
となっている。 各分枝限定法は整数計画法に比べ計算速度、計算精度において十分満足できる結果となっている。特に、深さの暫定解は整数計画法より 10 倍程度速く解けており、また、相対誤差も 3 ベーシス程度しかない。問題が解きにくかったり、解の精度に厳密さを求めないときであれば、深さの暫定解は有利な手法であるといえる。
また、深さ優先分枝限定法の cut 条件を緩和することにより、目的関数値の精度と計算速度を調節できることも分かった。

5　結論と今後の課題

本論文で行った計算結果からは、計算速度において、概ね
深さの暫定解＜ cut 緩和法＜幅優先＜深さ優先＜整数計画法
という順序になっていることが分かった。
特に、深さの暫定解は整数計画法に比べ、圧倒的に速く解けることが実証できた。さらに、深さ優先分枝限定法の cut 条件を変更することにより、計算速度と相対誤差を調節できることを示し、深さ優先分枝限定法の有効性を立証することができた。 今後の課題としては、超大型問題による計算機実験、計算時間だけではなく反復回数での比較、分枝限定法のメモリ管理を強化しての計算機実験などが考えられる。

参考文献

[1] 今野浩　著：「整数計画法」，産業図書，1981

[2] 今野浩　著：「線形計画法」，日科技連出版社，1987

[3] 今野浩　著：「理財工学 I 」，日科技連出版社，1995

[4] 今野浩　著：「理財工学II」，日科技連出版社，1998

[5] Konno,H. ,"Piecewise Linear Risk Functions and Portfolio Optimization", *J. of the Operations Research Society of Japan,33(1990) 139-156.*

[6] Konno,H. and Wijayanayake,A.,"Mean-Absolute Deviation Portfolio Optimization Model Under Transaction Costs", . *J. of the Operations Research Society of Japan,42(1990) 422-435.*

[7] Konno,H. and Yamamoto,R. ,"Global Optimization vs Integer Programming in Portfolio Optimization under Nonconvex Transaction Cost", *Technical Report 2003*,Department of Industrial and Systems Engneering,Chuo University .

[8] Konno,H. and Yamamoto,R.,"Integer Programming Approaches in Mean-Risk Model", *Technical Report 2003*,Department of Industrial and Systems Engneering,Chuo University .

[9] Phong,T.Q.,An,L.T.H. and Tao,P.D.,"Decomposition Branch and Bound Method for Globally Solving Linearly Constrained Indefinite Quadratic Minimization Problems" *Operations Research Letters,17(1995)215-220.*