数理解析研究所講究録
第 1773 巻 2012 年 47-56

# Towards Global Optimization of Constant Rebalanced Portfolio[1]

東京工業大学 大学院・社会理工学研究科　高野 祐一 (Yuichi Takano)
Graduate School of Decision Science and Technology,
Tokyo Institute of Technology

Renata Sotirov
Department of Econometrics and Operations Research,
Tilburg University

**Abstract**

We address the multi-period portfolio optimization problem with the constant rebalancing strategy. This problem is formulated as a polynomial optimization problem (POP) by using a mean-variance criterion. In order to solve the corresponding POPs of high degree, we develop a cutting-plane algorithm based on semidefinite programming. Our algorithm can solve problems that can not be handled directly by any of known polynomial optimization solvers.

**Keywords:** Multi-period portfolio optimization, Polynomial optimization problem, Constant rebalancing, Semidefinite programming, Mean-variance criterion

## 1 Introduction

We consider the constant rebalancing strategy in the multi-period portfolio selection. In this strategy, we rebalance the portfolio at the beginning of every period so that the investment proportion will be restored to the fixed constant one. It is known that the constant rebalancing achieves the optimal growth rate of wealth if the asset prices in each period are independent and identically distributed (i.i.d.) (see e.g., [1]). On the assumptions of i.i.d. and infinite horizon, the problem to be solved is a relatively easy convex program (see e.g., [6]). However, the constant rebalancing strategy generally leads to nonconvex optimization. Because of its difficulty, most studies (e.g., [3, 11]) have focused on approximately solving the constant rebalanced portfolio optimization problem. To the best of our knowledge, only Maranas et al. [8] approached it through global optimization by developing a specialized branch-and-bound algorithm.

In this paper, we use a mean-variance (M-V) criterion to formulate the constant rebalanced portfolio optimization problem (see also [8]) as a polynomial optimization problem (POP). In

[1]This article is a short version of [12].

order to solve the resulting POPs, we develop a cutting-plane algorithm based on semidefinite programming (SDP). Our algorithm is iterative and solves in each iteration a POP of reduced degree by applying Lasserre's approach [5]. In [5], Lasserre proved that small and medium size POPs can be efficiently solved using SDP. This approach is intractable for the polynomial reformulation of the constant rebalanced portfolio optimization problem when the number of planning periods is large. However, it is rather easy to handle in the framework of our cutting-plane algorithm in which we implement corresponding polynomials of reduced degree. Our numerical results verify the efficiency of our approach.

## 2 Mean-Variance Portfolio Optimization with Constant Rebalancing Strategy

We define the terminology and notation as follows:

| | | |
|---|---|---|
| $\mathbb{R}^N$ | : | set of $N$-dimensional real vectors |
| $\mathbb{Z}^N_+$ | : | set of $N$-dimensional nonnegative integer vectors |

**Index Sets**

| | | |
|---|---|---|
| $\mathcal{I} := \{1, 2, ..., I\}$ | : | index set of investable financial assets |
| $\mathcal{T} := \{1, 2, ..., T\}$ | : | index set of planning periods |
| $\mathcal{S} := \{1, 2, ..., S\}$ | : | index set of given scenarios |

**Decision Variables**

| | | | |
|---|---|---|---|
| $v^s_t$ | : | portfolio value at the end of period $t$ under scenario $s$ | $(t \in \mathcal{T},\ s \in \mathcal{S})$ |
| $w_i$ | : | investment proportion in asset $i$ $(i \in \mathcal{I})$ (where $\boldsymbol{w} := (w_1, w_2, \cdots, w_I) \in \mathbb{R}^I$) | |

**Given Constants**

| | | | |
|---|---|---|---|
| $V$ | : | initial wealth for investment | |
| $R^s_{i,t}$ | : | total return of asset $i$ at period $t$ under scenario $s$ | $(i \in \mathcal{I},\ t \in \mathcal{T},\ s \in \mathcal{S})$ |
| $P_s$ | : | occurrence probability of scenario $s$ | $(s \in \mathcal{S})$ |
| $L_i\,(U_i)$ | : | lower (upper) bound of the investment proportion in asset $i$ | $(i \in I)$ |

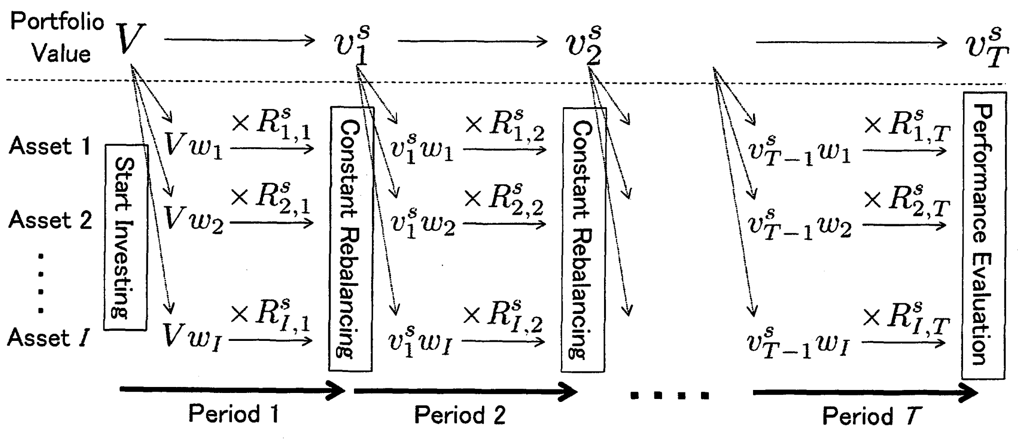
**User-Defined Parameters**

| | | |
|---|---|---|
| $\lambda$ | : | trade-off parameter between return and risk (where $\lambda \in (0,1)$) |

Figure 1 illustrates a portfolio dynamics under scenario $s$. Suppose that one starts investing $Vw_i$ in each asset $i$. Because of the return of each asset, the invested amount $Vw_i$ is changed to $R^s_{i,1}Vw_i$ over the first period. Accordingly, the portfolio value at the end of the first period under scenario $s$ is given by

$$v^s_1 = \sum_{i \in \mathcal{I}} R^s_{i,1} V w_i = V \sum_{i \in \mathcal{I}} R^s_{i,1} w_i. \tag{1}$$

The amount $R^s_{i,1}Vw_i$ is adjusted to $v^s_1 w_i$ according to the constant rebalancing strategy at the beginning of the second period. Because of the return of each asset, the invested amount



Portfolio Value
$V$
$v_1^s$
$v_2^s$
$v_T^s$
Asset 1
Asset 2
Asset $I$
Start Investing
$Vw_1$ $\times R_{1,1}^s$
$Vw_2$ $\times R_{2,1}^s$
$Vw_I$ $\times R_{I,1}^s$
Constant Rebalancing
$v_1^s w_1$ $\times R_{1,2}^s$
$v_1^s w_2$ $\times R_{2,2}^s$
$v_1^s w_I$ $\times R_{I,2}^s$
Constant Rebalancing
$v_{T-1}^s w_1$ $\times R_{1,T}^s$
$v_{T-1}^s w_2$ $\times R_{2,T}^s$
$v_{T-1}^s w_I$ $\times R_{I,T}^s$
Performance Evaluation
Period 1
Period 2
Period $T$

Figure 1: Portfolio Dynamics under Scenario $s$

$v_1^s w_i$ is changed to $R_{i,2}^s v_1^s w_i$ over the second period. Accordingly, the portfolio value at the end of the second period under scenario $s$ is given by

$$v_2^s = \sum_{i \in \mathcal{I}} R_{i,2}^s v_1^s w_i = v_1^s \sum_{i \in \mathcal{I}} R_{i,2}^s w_i. \tag{2}$$

Continuing in the same vein, the portfolio value at the end of the planning horizon of $T$ periods under scenario $s$ is given by

$$\begin{aligned} v_T^s &= v_{T-1}^s \sum_{i \in \mathcal{I}} R_{i,T}^s w_i \\ &= v_{T-2}^s \left( \sum_{i \in \mathcal{I}} R_{i,T-1}^s w_i \right) \left( \sum_{i \in \mathcal{I}} R_{i,T}^s w_i \right) = \cdots = V \prod_{t \in \mathcal{T}} \left( \sum_{i \in \mathcal{I}} R_{i,t}^s w_i \right). \end{aligned} \tag{3}$$

In the sequel we formulate the constant rebalanced portfolio optimization problem. We consider here both, minimizing the variance of the portfolio value:

$$\begin{aligned} \mathrm{Var}(\boldsymbol{w}) &:= \sum_{s \in \mathcal{S}} P_s (v_T^s)^2 - \left( \sum_{s \in \mathcal{S}} P_s v_T^s \right)^2 \\ &\overset{(3)}{=} \sum_{s \in \mathcal{S}} P_s \left( V \prod_{t \in \mathcal{T}} \left( \sum_{i \in \mathcal{I}} R_{i,t}^s w_i \right) \right)^2 - \left( \sum_{s \in \mathcal{S}} P_s V \prod_{t \in \mathcal{T}} \left( \sum_{i \in \mathcal{I}} R_{i,t}^s w_i \right) \right)^2, \end{aligned} \tag{4}$$

and maximizing the mean of the portfolio value:

$$\sum_{s \in \mathcal{S}} P_s v_T^s \overset{(3)}{=} \sum_{s \in \mathcal{S}} P_s V \prod_{t \in \mathcal{T}} \left( \sum_{i \in \mathcal{I}} R_{i,t}^s w_i \right), \tag{5}$$

at the same time by taking the weighted sum of them (see also [8]). This leads to the following optimization problem:

$$\left|\begin{array}{ll}\underset{\boldsymbol{w}\in\mathbb{R}^I}{\text{minimize}} & (1-\lambda)\left(\sum_{s\in\mathcal{S}} P_s\left(V\prod_{t\in\mathcal{T}}\left(\sum_{i\in\mathcal{I}} R^s_{i,t}w_i\right)\right)^2-\left(\sum_{s\in\mathcal{S}} P_s V\prod_{t\in\mathcal{T}}\left(\sum_{i\in\mathcal{I}} R^s_{i,t}w_i\right)\right)^2\right) \\ & -\lambda\left(\sum_{s\in\mathcal{S}} P_s V\prod_{t\in\mathcal{T}}\left(\sum_{i\in\mathcal{I}} R^s_{i,t}w_i\right)\right) \\ \text{subject to} & \sum_{i\in\mathcal{I}} w_i = 1;\ \ L_i\le w_i\le U_i,\ \ i\in\mathcal{I}.\end{array}\right. \tag{6}$$

The above problem can be reformulated as the following POP:

$$\left|\begin{array}{ll}\underset{\boldsymbol{w}\in\mathbb{R}^I}{\text{minimize}} & \mathrm{OF}(\boldsymbol{w}) := (1-\lambda)\sum_{\boldsymbol{\alpha}:\ \sum_{i\in\mathcal{I}}\alpha_i=2T} \mathcal{C}_5(\boldsymbol{\alpha})\boldsymbol{w}^{\boldsymbol{\alpha}} - \lambda\sum_{\boldsymbol{\alpha}:\ \sum_{i\in\mathcal{I}}\alpha_i=T} \mathcal{C}_2(\boldsymbol{\alpha})\boldsymbol{w}^{\boldsymbol{\alpha}} \\ \text{subject to} & \sum_{i\in\mathcal{I}} w_i = 1;\ \ L_i\le w_i\le U_i,\ \ i\in\mathcal{I},\end{array}\right. \tag{7}$$

where

$$\boldsymbol{\alpha} := (\alpha_1,\alpha_2,\cdots,\alpha_I)\in\mathbb{Z}^I_+ \ \text{ and } \ \boldsymbol{w}^{\boldsymbol{\alpha}} := \prod_{i\in\mathcal{I}} w_i^{\alpha_i},$$

and see [12] for details of $\mathcal{C}_2(\boldsymbol{\alpha})$ and $\mathcal{C}_5(\boldsymbol{\alpha})$.

## 3 Cutting-Plane Algorithm

If a POP contains a polynomial of high degree, then the relaxation order, $\omega$ (for details see [5]), is also high. Accordingly, the corresponding SDP relaxations are large-scale and it is hard to solve them. In this section, we present a cutting-plane algorithm for solving POPs of high degree.

The fundamental principle of our algorithm, which is regarded as a natural extension of Kelley's convex cutting-plane algorithm (see e.g., Section 14.8 of [7]), is to solve a sequence of relaxed POPs and to approximate the feasible region of the original problem by cutting off the current infeasible solution of the relaxed problem.

The algorithm for the M-V portfolio optimization problem (7) is described as follows (see [12] for details):

Algorithm CPMV: Cutting-Plane Algorithm for the M-V Portfolio Optimization Problem (7)

**Step 0.** (Initialization) Let $\varepsilon \ge 0$ be a tolerance for optimality, $K$ be the maximum number of iterations, and $\omega \ge \lceil T/2 \rceil$ be the relaxation order. Set

$$\mathcal{Z} \leftarrow \left\{(\boldsymbol{w},z)\,\middle|\,\sum_{i\in\mathcal{I}} w_i = 1;\ \ L_i\le w_i\le U_i,\ \ i\in\mathcal{I};\ \ z\ge 0\right\}.$$

Set the initial upper bound as $\mathrm{UB}_0 := \infty$. Set $k \leftarrow 1$.

**Step 1.** (Lower-Bound Estimation) Solve the the following POP by using the SDP approach [5] with the relaxation order $\omega$:

$$\left| \begin{array}{l} \underset{(\boldsymbol{w},z)\in\mathbb{R}^I\times\mathbb{R}}{\text{minimize}} \quad (1-\lambda)z - \lambda \sum_{\boldsymbol{\alpha}:\ \sum_{i\in\mathcal{I}}\alpha_i=T} \mathcal{C}_2(\boldsymbol{\alpha})\boldsymbol{w}^{\boldsymbol{\alpha}} \quad \text{subject to } (\boldsymbol{w},z)\in\mathcal{Z}. \end{array} \right. \tag{8}$$

Let $\mathrm{LB}_k$ be the objective function value, and $(\bar{\boldsymbol{w}}^k, \bar{z}^k)$ be the solution of (8).

**Step 2.** (Upper-Bound Update) If $\mathrm{OF}(\bar{\boldsymbol{w}}^k) < \mathrm{UB}_{k-1}$, then $\mathrm{UB}_k := \mathrm{OF}(\bar{\boldsymbol{w}}^k)$ and $\hat{\boldsymbol{w}} \leftarrow \bar{\boldsymbol{w}}^k$. Otherwise, $\mathrm{UB}_k := \mathrm{UB}_{k-1}$.

**Step 3.** (Termination Conditions) If one of the following conditions is satisfied, then terminate the algorithm with the solution $\hat{\boldsymbol{w}}$:

⟨a⟩ $\mathrm{UB}_k - \mathrm{LB}_k \le \varepsilon$, ⟨b⟩ $\bar{z}^k \ge \mathrm{Var}(\bar{\boldsymbol{w}}^k)$, ⟨c⟩ $k = K$,

⟨d⟩ no improvement in the gap, $\mathrm{UB}_k - \mathrm{LB}_k$.

**Step 4.** (Cut Generation) Set

$$\mathcal{Z} \leftarrow \mathcal{Z} \cap \left\{(\boldsymbol{w},z) \,\middle|\, z \ge \sum_{\boldsymbol{\alpha}:\ \sum_{i\in\mathcal{I}}\alpha_i=T} \left(\sum_{s\in\mathcal{S}} G_s(\bar{\boldsymbol{w}}^k)\mathcal{C}_1(\boldsymbol{\alpha},s)\right)\boldsymbol{w}^{\boldsymbol{\alpha}} - \mathrm{Var}(\bar{\boldsymbol{w}}^k)\right\},$$

where $G_s(\bar{\boldsymbol{w}}^k)$ and $\mathcal{C}_1(\boldsymbol{\alpha},s)$ are defined in [12]. Set $k \leftarrow k+1$ and return to Step 1.

---

Note that the maximal degree of monomials in the problem (8) is $T$ while in the POP (7) is $2T$. This reduction of the degree is crucial for the success of our algorithm, i.e., the corresponding SDP relaxations are easier to solve. A general form of the above algorithm and its convergence theorem are shown in [12].

## 4 Computational Experiments

In our computations, we use the following numbers for assets $I \in \{4, 7, 10\}$, periods $T \in \{2, 4, 6\}$, and scenarios $S \in \{100, 1{,}000\}$. The initial wealth is $V = 1$ and the occurrence probability is $P_s = 1/|S|$ for all $s \in S$. The lower bound, $L_i$, and the upper bound, $U_i$, of the investment proportion are set to 0 and 0.5, respectively for each $i \in \mathcal{I}$. In the cutting-plane algorithm, we set the tolerance for optimality $\varepsilon = 10^{-5}$, and the maximum number of iterations $K = 30$. All computations were performed on a PC with a Core2 Duo CPU (1.40 GHz) and 2GB memory. We used MATLAB 7.10.0 (R2010a) to program our algorithm and the global optimization solver over polynomials GloptiPoly 3.6.1 [4], which uses SeDuMi 1.3 [10] to solve SDP problems. The results obtained by our algorithm are compared with the global optimization solver BARON [9] and the NLP solver CONOPT [2], that are available via NEOS Server[2]. In BARON, a tolerance for optimality is set to the same value as in the cutting-plane algorithm, i.e., to $10^{-5}$.

[2]`http://www-neos.mcs.anl.gov`

Table 1: Numerical Results for Solving (7) by GloptiPoly

| $S$ = 100 | | | $I$ | | | $S$ = 1,000 | | | $I$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 4 | 7 | 10 | | | | 4 | 7 | 10 |
| $T$ | 2 | Rel.Order | 2 | 2 | 2 | $T$ | 2 | Rel.Order | 2 | 2 | 2 |
| | | TotalCPU | 0.6 | 3.0 | 26.3 | | | TotalCPU | 0.6 | 2.8 | 26.0 |
| | 4 | Rel.Order | 4 | 4 | 4 | | 4 | Rel.Order | 4 | 4 | 4 |
| | | TotalCPU | 8.5 | OMS | OMG | | | TotalCPU | 8.6 | OMS | OMG |
| | 6 | Rel.Order | 7 | 6 | 6 | | 6 | Rel.Order | 7 | 6 | 6 |
| | | TotalCPU | 1,223.2 | OMG | OMG | | | TotalCPU | 1,180.2 | OMG | OMG |

Table 2: Numerical Results of Algorithm CPMV

| $S$ = 100 | | | $I$ | | | $S$ = 1,000 | | | $I$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 4 | 7 | 10 | | | | 4 | 7 | 10 |
| $T$ | 2 | Rel.Order | 2 | 2 | 2 | $T$ | 2 | Rel.Order | 2 | 2 | 2 |
| | | TotalCPU | 13.0 | 65.6 | 960.9 | | | TotalCPU | 12.9 | 114.0 | 733.1 |
| | | #Iteration | 3.4 | 5.1 | 7.9 | | | #Iteration | 3.3 | 6.8 | 6.4 |
| | | #Ter.Con. | (8,0,0,0) | (8,0,0,0) | (8,0,0,0) | | | #Ter.Con. | (8,0,0,0) | (8,0,0,0) | (8,0,0,0) |
| | | Opt.Gap | 8.8E-06 | 9.9E-06 | 9.9E-06 | | | Opt.Gap | 9.4E-06 | 1.0E-05 | 9.5E-06 |
| | 4 | Rel.Order | 2 | 2 | 2 | | 4 | Rel.Order | 2 | 2 | 2 |
| | | TotalCPU | 5.7 | 60.6 | 498.7 | | | TotalCPU | 11.3 | 68.8 | 616.7 |
| | | #Iteration | 2.9 | 7.4 | 6.4 | | | #Iteration | 5.0 | 8.6 | 7.8 |
| | | #Ter.Con. | (6,0,0,2) | (1,0,0,7) | (5,0,0,3) | | | #Ter.Con. | (7,1,0,0) | (2,4,0,2) | (6,0,0,2) |
| | | Opt.Gap | 1.4E-04 | 9.0E-05 | 1.2E-04 | | | Opt.Gap | 1.7E-05 | 1.1E-04 | 3.2E-05 |
| | 6 | Rel.Order | 3 | 3 | 3 | | 6 | Rel.Order | 3 | 3 | 3 |
| | | TotalCPU | 29.9 | 6,743.2 | OMG | | | TotalCPU | 46.3 | 5,215.4 | OMG |
| | | #Iteration | 4.0 | 10.4 | --- | | | #Iteration | 6.0 | 7.6 | --- |
| | | #Ter.Con. | (8,0,0,0) | (1,6,0,1) | --- | | | #Ter.Con. | (8,0,0,0) | (2,4,0,2) | --- |
| | | Opt.Gap | 9.4E-06 | 5.9E-05 | --- | | | Opt.Gap | 9.6E-06 | 1.7E-04 | --- |

Numerical data and notations in Tables 1, 2 and 3 are as follows:
Rel.Order : the relaxation order $\omega$,
TotalCPU : the total CPU time (in seconds),
#Iteration : the number of iterations (i.e., $k$) in the cutting-plane algorithm,
#Ter.Con. : the number of times each termination condition was satisfied, ($\langle$a$\rangle$, $\langle$b$\rangle$, $\langle$c$\rangle$, $\langle$d$\rangle$), see CPMV algorithm and explanations therein,
Opt.Gap : the optimality gap, i.e., (the best upper bound) $-$ (the best lower bound),
#Mem.Sho. : the occurrence number of memory shortage,
OMS : out of memory in SeDuMi, and
OMG : out of memory in GloptiPoly.

For each pair of $(I, T, S)$ we solve eight problems corresponding to different values of the trade-off parameter, $\lambda \in \{0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.99\}$. In the tables we show the average value of the eight problems in TotalCPU and #Iteration, and the largest value of those in Opt.Gap.

Table 3: Numerical Results of BARON and CONOPT

| BARON | | | | | | CONOPT | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $S$ = 1,000 | | | $I$ | | | $S$ = 1,000 | | | $I$ | | |
| | | | 4 | 7 | 10 | | | | 4 | 7 | 10 |
| $T$ | 4 | TotalCPU | 804.2 | 5,031.7 | 7,537.8 | $T$ | 4 | TotalCPU | 0.5 | 0.8 | 1.7 |
| | | Opt.Gap | 3.4E-03 | 1.0E-05 | 1.2E-02 | | | Opt.Gap | --- | --- | --- |
| | | #Mem.Sho. | 4 | 0 | 5 | | | #Mem.Sho. | 0 | 0 | 0 |

## 4.1 Numerical Results of POP Approaches

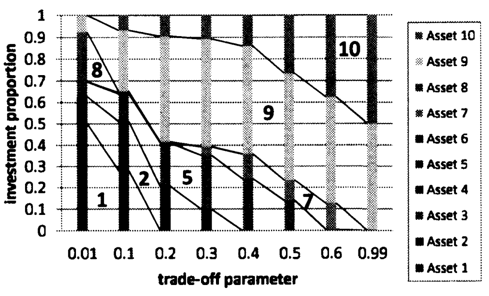
Numerical results for solving POP (7) by GloptiPoly and the cutting-plane algorithm are shown in Tables 1 and 2, respectively. Note that all solutions reported in Table 1 are globally optimal. However, only the problem involving four assets was solved when the number of periods was four. To the contrary, all problems were solved by using the cutting-plane algorithm except when $(I, T) = (10, 6)$ (see Table 2). The algorithm has terminated several times due to the numerical instability. Although it is possible that the attained solution is not very good in such cases, the obtained optimality gap was sufficiently small (see worst-case optimality gap, Opt.Gap in Table 2).

## 4.2 Comparison with BARON and CONOPT

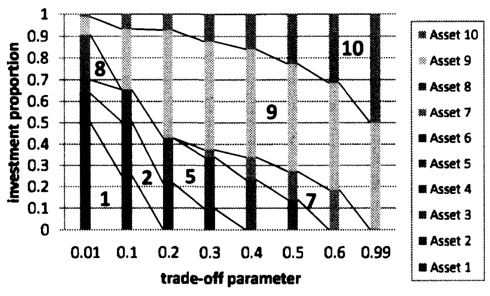
Numerical results of BARON and CONOPT are presented in Table 3, where four periods and 1,000 scenarios are considered. CPU times for solving problems by BARON were very long in comparison to the cutting-plane algorithm. In some cases, BARON stopped due to the memory shortage and returned a locally optimal solution (see the last row in Table 3). Although solutions obtained by CONOPT do not have a guarantee of global optimality, CONOPT attained locally optimal solutions in very short time without leading to memory shortage.

Figure 2 shows the optimal investment proportions obtained by different approaches. The solutions of the cutting-plane algorithm were slightly different from others. For instance, the proportion in Asset 7 for $\lambda$ =0.5 and 0.6 differs from other approaches (see Figure 2).

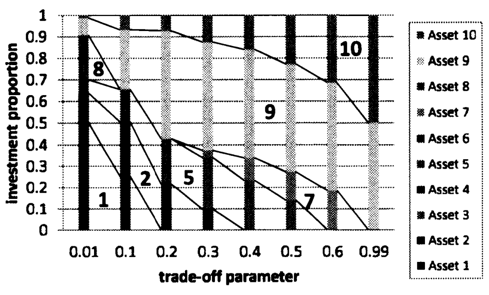
In Figure 3, we show the efficient frontiers of the solutions provided by different approaches with the value of the trade-off parameter. The horizontal and the vertical axis are mean and variance of the portfolio value, respectively. Although some solutions of the cutting-plane algorithm were slightly different from others, it is clear that solutions of the cutting-plane algorithm are not far from the frontiers of other approaches.



(i) Algorithm CPMV



(ii) BARON



(iii) CONOPT

Figure 2: Optimal Investment Proportion ($I$ = 10, $T$ = 4, $S$ = 1,000)

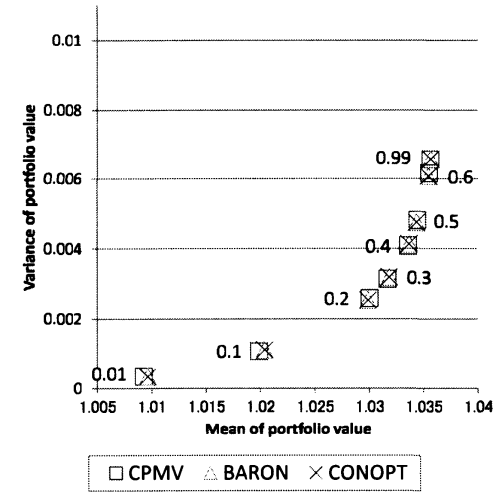


Figure 3: Efficient Frontier ($I$ = 10, $T$ = 4, $S$ = 1,000)

## 5 Conclusion

We have formulated the constant rebalanced portfolio optimization problem as a POP and developed a cutting-plane algorithm for solving it. The computational experiments show that our algorithm can solve large-size problems that can not be directly solved by the global optimization solver over polynomials GloptiPoly [4]. This success is due to implementation of the reduced degree polynomials in the iterative algorithm. Our numerical results show that our algorithm provides solutions with adequate accuracy for practical purposes. Moreover, our algorithm is comparable to state-of-the-art global optimization solver BARON.

Furthermore, if there is an effective warm-starting approach for SDP, then our cutting-plane algorithm might be even more efficient by starting a SDP solver from the solution attained in the previous iteration.

## Acknowledgments

The first author was supported by the Grant-in-Aid for JSPS Fellows and the Excellent Young Researcher Overseas Visit Program of the JSPS.

## References

[1] P.H. Algoet and T.M. Cover, "Asymptotic Optimality and Asymptotic Equipartition Properties of Log-Optimum Investment," *The Annals of Probability*, Vol.16, No.2, pp.876–898 (1988).

[2] A.S. Drud, "CONOPT — A Large Scale GRG Code," *ORSA Journal on Computing*, Vol.6, No.2, pp.207–216 (1992).

[3] S.-E. Fleten, K. Høyland and S.W. Wallace, "The Performance of Stochastic Dynamic and Fixed Mix Portfolio Models," *European Journal of Operational Research*, Vol.140, No.1, pp.37–49 (2002).

[4] D. Henrion, J. B. Lasserre and J. Löfberg, "GloptiPoly 3: Moments, Optimization and Semidefinite Programming," *Optimization Methods and Software*, Vol.24, No.4–5, pp.761–779 (2009).

[5] J.B. Lasserre, "Global Optimization with Polynomials and the Problem of Moments," *SIAM Journal on Optimization*, Vol.11, No.3, pp.796–817 (2001).

[6] D.G. Luenberger, *Investment Science*, Oxford University Press, USA (1997).

[7] D.G. Luenberger and Y. Ye, *Linear and Nonlinear Programming (Third edition)*, Springer, USA (2008).

[8] C.D. Maranas, I.P. Androulakis, C.A. Floudas, A.J. Berger and J.M. Mulvey, "Solving Long-Term Financial Planning Problems via Global Optimization," *Journal of Economic Dynamics & Control*, Vol.21, No.8–9, pp.1405–1425 (1997).

[9] N.V. Sahinidis and M. Tawarmalani, "A Polyhedral Branch-and-Cut Approach to Global Optimization," *Mathematical Programming*, Vol.103, No.2, pp.225–249 (2005).

[10] J.F. Sturm, "Using SeDuMi 1.02, A MATLAB Toolbox for Optimization over Symmetric Cones," *Optimization Methods and Software*, Vol.11–12, pp.625–653 (1999).

[11] Y. Takano and J. Gotoh, "Constant Rebalanced Portfolio Optimization under Nonlinear Transaction Costs," *Asia-Pacific Financial Markets*, Vol.18, No.2, pp.191–211 (2011).

[12] Y. Takano and R. Sotirov, "A Polynomial Optimization Approach to Constant Rebalanced Portfolio Selection," *Computational Optimization and Applications* (to appear).