INFORMATICA, 2005, Vol. 16, No. 1, 93–106

© 2005 *Institute of Mathematics and Informatics, Vilnius*

An Algorithm for Portfolio Optimization Problem [*]

Jong Soo KIM
Department of Information and Industrial Engineering, Hanyang University
Ansan Campus, Ansan, Korea
e-mail: jskim@mecors.hanyang.ac.kr

Yong Chan KIM, Ki Young SHIN
Department of Industrial Engineering, Hanyang University
Haengdang 1-dong, Seongdong-gu, Seoul 133-791, Korea
e-mail: yckim@mecors.hanyang.ac.kr, sky@mecors.hanyang.ac.kr

Received: November 2003

Abstract. Portfolio optimization is to find the stock portfolio minimizing the risk for a required return or maximizing the return for a given risk level. The seminal work in this field is the mean-variance model formulated as a quadratic programming problem. Since it is not computationally practical to solve the original model directly, a number of alternative models have been proposed.

In this paper, among the alternative models, we focus on the Mean Absolute Deviation (MAD) model. More specifically, we derive bounds on optimal objective function value. Using the bounds, we also develop an algorithm for the model. We prove mathematically that the algorithm can solve the problem to optimality. The algorithm is tested using the real data from the Korean Stock Market. The results come up to our expectations that the method can solve a variety of problems in a reasonable computational time.

Key words: portfolio, MAD, integer programming.

1. Introduction

The traditional portfolio optimization problem is to find an investment plan for securities with a reasonable trade-off between the rate of return and risk. The mean-variance model of Markowitz (1952) is a single period static model to obtain the portfolio which can achieve a specified average rate of return with the minimum risk. Following Markowitz's paper, a number of alternative models have been proposed for the same problem. The main objective of these alternative models is to overcome the computational complexity of the original quadratic programming problem. A piecewise linear approximation, the Mean Absolute Deviation (MAD), the Weighted Goal Programming (WGP) and the Minimax (MM) models are some examples of them. For more details on these models, please refer to Satchell and Scowcroft (2003).

The MAD model proposed by Konno and Yamazaki (1991) is a linear programming (LP) model where risk is measured by absolute deviation instead of variance. They

[*] This work was supported by the research fund of Hanyang University (HY-2004-1).

showed that it is equivalent to the Markowitz model if returns are multivariate normally distributed. Later, Zenios and Kang (1993) analyzed the model for other asymmetric distributions and found that the MAD model does not require any specific type of return distributions. Speranza (1993) generalized the MAD model using a weighted risk function. The author showed that a more compact equivalent model can be constructed when the coefficients in the linear combination are selected suitably. Three other papers followed to introduce more flexible models (Speranza, 1996; Mansini, 1997; Mansini and Speranza, 1997). Thereafter, Mansini and Speranza (1999) extended their previous work by developing three heuristics. More recent papers, Konno and Wijayanayake (2001) and Kellerer *et al.* (2001) examined the problems, which take into account more realistic features such as fixed transaction costs or minimum transaction lots. Mansini and Ogryczak (2003) introduced a systematic overview of the LP solvable models with a wide discussion of their theoretical properties.

In this paper, we derive objective bounds for the MAD model, which includes fixed transaction costs and minimum transaction lots. Finding such bounds is important because the bounds can be applicable to a wide variety of problems that have similar structure. By taking advantage of the bounds, we also developed an efficient algorithm for the model. Computational experiments using the real data from the Korean Stock Market show that the bounds we have found are tight and the algorithm is very efficient.

The rest of this paper is organized as follows. In Section 2, a brief review of the MAD model is introduced. In Section 3, we derive the bounds and a few properties followed by the presentation of our algorithm. In the following section the results of computational experiments are discussed. Finally, a conclusion follows in the last section.

2. Mathematical Model

In this section, we briefly discuss the MAD model. First, we define necessary notations:

Indices

j: index of security ($j = 1, \ldots, J$),

t: index of time period ($t = 1, \ldots, T$).

Parameters

r_{jt}: observed rate of return of security j at time period t,

r_j: average rate of return of security$j$, i.e., $r_j = \sum_{t=1}^{T} r_{jt}/T$,

C_L, C_U: lower and upper limits of the money to be invested,

p_j: unit purchase price of security j,

d_j: transaction cost rate of security j,

l_j: minimum units of security j that must be included in the portfolio,

u_j: maximum units of security j that can be included in the portfolio,

ρ: required rate of return.

Decision variables

x_j: units of security j to be included in the portfolio,

y_t: deviation below the average rate of return at time period t .

The MAD model is as follows:

Original Problem (OP):

$$Min\, Z = \sum_{t=1}^{T} y_t/T, \tag{1}$$

s.t.

$$-\sum_{j=1}^{J}(r_{jt} - r_j)p_j x_j \leqslant y_t, \quad t = 1, \dots, T, \tag{2}$$

$$\sum_{j=1}^{J}(1 + d_j)p_j x_j \geqslant C_L, \tag{3}$$

$$\sum_{j=1}^{J}(1 + d_j)p_j x_j \leqslant C_U, \tag{4}$$

$$\sum_{j=1}^{J}(r_j - d_j - \rho)p_j x_j \geqslant 0, \tag{5}$$

$$l_j \leqslant x_j \leqslant u_j, \quad x_j \text{ integer}, \ j = 1, \dots, J, \tag{6}$$

$$y_t \geqslant 0, \quad t = 1, \dots, T. \tag{7}$$

We will refer to the above MAD model as Original Problem (OP) throughout the paper. Constraints (3) and (4) specify the maximum and minimum of capital investment. Constraint (5) makes the selected portfolio attain the required rate of return. (5) shows that the fixed proportional transaction costs are assumed to incur and are deducted from the portfolio return. Constraint (6) is to specify the maximum and minimum units of each security type to be included in the portfolio. The objective function is to minimize the sum of below average deviation per period.

3. The Algorithms

3.1. *The Previous Heuristics*

One of the recent important developments for the MAD model is the work by Mansini and Speranza (1999). They introduced three heuristic algorithms and evaluated them using Milan Stock Exchange data. The three heuristics developed are called Procedures A, B, and C.

Procedure A solves an LP-relaxation of the original problem and, subsequently, solves a smaller MILP (mixed integer linear programming) problem, which is composed of the basic variables of the optimal solution of the LP. This smaller problem is called local MILP. Procedure B extended the range of variables in the local MILP by including non-basic variables too. Procedure C is a further extension of Procedure B utilizing an iterative scheme to change the security type included in the local MILP.

A serious flaw the three algorithms have in common is that they can not guarantee the optimality of their solution. Since the portfolio problem is a static planning problem, which rarely requires an approximate solution hurriedly, this kind of flaw can be a serious obstacle to real world applications. For this reason, we developed a practical method, which guarantees the optimal solution.

3.2. *The Proposed Algorithm*

We first define an LP as follows:

LP1:

$$Min\, f^M = \sum_{t=1}^{T} y_t/T, \tag{8}$$

s.t.

$$-\sum_{j=1}^{J}(r_{jt} - r_j)p_j x_j \leqslant y_t, \quad t = 1,\dots,T, \tag{9}$$

$$\sum_{j=1}^{J}(1 + d_j)p_j x_j = M, \tag{10}$$

$$\sum_{j=1}^{J}(r_j - \rho - \rho d_j)p_j x_j \geqslant 0, \tag{11}$$

$$l_j \leqslant x_j \leqslant u_j, \quad j = 1,\dots,J, \tag{12}$$

$$y_t \geqslant 0, \quad t = 1,\dots,T. \tag{13}$$

Note that LP1 is the same as the LP-relaxation of OP but with a new equation (10) and without the original equations (3) and (4). M in (10) denotes a variable for the investment amount, which is to be set to an appropriate value as we process the algorithm. We found some useful properties for LP1.

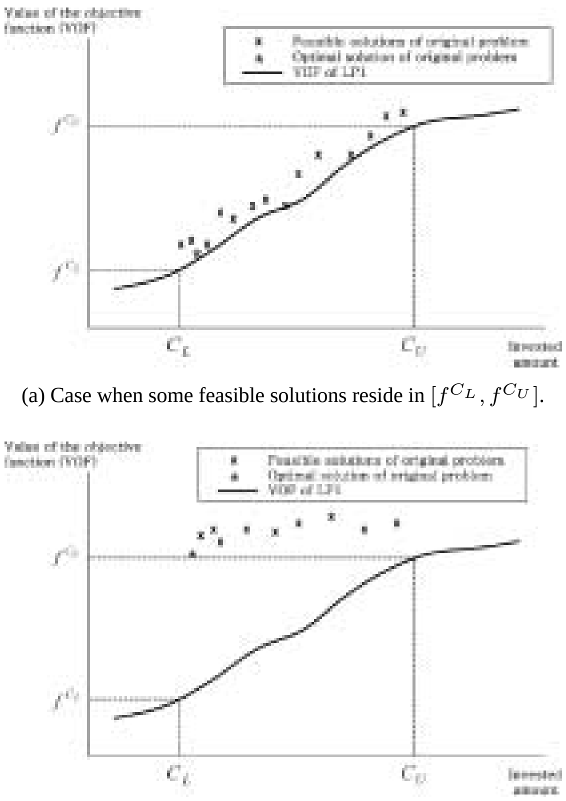
Property 1. *Let f^{C_L} be the value of the optimal objective function (VOF) of LP1 when $M = C_L$ and f^{C_U} be the VOF of LP1 when$M = C_U$. If optimals exist, then*
(i) $f^{C_L} \leqslant f^{C_U}$,
(ii) $f^{C_L} \leqslant Z^*$ *where Z^* is the VOF of OP.*

Proof. See the Appendix.

Usually $f^{C_L} \leqslant Z^* \leqslant f^{C_U}$ is satisfied. But, theoretically, the case of $f^{C_L} \leqslant f^{C_U} \leqslant Z^*$ exists.

Property 2. *There exists a case where $f^{C_L} \leqslant f^{C_U} \leqslant Z^*$.*

Proof. See the Appendix.



(a) Case when some feasible solutions reside in $[f^{C_L}, f^{C_U}]$.

(b). Case when all feasible solutions reside in (f^{C_U}, ∞).

Fig. 1. Graphical view of the feasible solutions.

Fig. 1 shows the two cases of $f^{C_L} \leqslant Z^* \leqslant f^{C_U}$ and $f^{C_L} \leqslant f^{C_U} \leqslant Z^*$. We will refer to the former as a usual case (Us-case) and the latter as an exceptional case (Ex-case). Note that the Ex-case can not happen unless there is a security whose unit price is more expensive than the gap between the limits, that is$C_U - C_L$. In either case, $f^{C_L} \leqslant Z^*$ is satisfied. Thus, f^{C_L} is a valid lower bound on the optimal objective value of OP, Z^* .

We realize that the search time of the OP's optimal can be shortened considerably if we know in advance which of the above two cases a given problem belongs to. The following LP called Feasibility Problem (FP) is developed to identify the type of a given problem. C in the objective function (14) denotes an arbitrary constant value.

FP:

$$Min\,C \tag{14}$$

s.t.

$$-\sum_{j=1}^{J}(r_{jt} - r_j)p_j x_j \leqslant y_t, \quad t = 1, \ldots, T, \tag{15}$$

$$\sum_{j=1}^{J}(1+d_j)p_jx_j \geqslant C_L, \tag{16}$$

$$\sum_{j=1}^{J}(1+d_j)p_jx_j \leqslant C_U, \tag{17}$$

$$\sum_{j=1}^{J}(r_j-\rho-\rho d_j)p_jx_j \geqslant 0, \tag{18}$$

$$\sum_{t=1}^{T}y_t/T \geqslant f^{C_L}, \tag{19}$$

$$\sum_{t=1}^{T}y_t/T \leqslant f^{C_U}, \tag{20}$$

$$l_j \leqslant x_j \leqslant u_j, \quad x_j \text{ integer}, \ j=1,\dots,J, \tag{21}$$

$$y_t \geqslant 0, \quad t=1,\dots,T. \tag{22}$$

We notice that either of two things can happen when we solve FP. FP may terminate with an optimal solution or with infeasibility. When OP belongs to the case of $f^{C_L} \leqslant Z^* \leqslant f^{C_U}$, it must have at least one feasible solution with its VOF in the interval $\left[f^{C_L}, f^{C_U}\right]$ (refer to Fig. 1(a)). Thus, in this case, feasible solutions to OP satisfy (19) and (20). Since FP has the same set of constraints as OP except (19) and (20), the feasible solutions in this case are also feasible to FP. Because the objective function of FP is set to a constant value, FP will generate one of the feasible solutions as optimal. We also realize that the value calculated by plugging this optimal solution into (1) is a valid upper bound on the VOF of the original problem. Let this upper bound be denoted by υ(Note that the optimal to FP is a feasible to OP, but not necessarily optimal. Thus it is an upper bound). So, in this case $f^{C_L} \leqslant Z^* \leqslant \upsilon$ is satisfied.

On the other hand, if $f^{C_L} \leqslant f^{C_U} \leqslant Z^*$ (Fig. 1(b)) or if OP is infeasible, then OP can not have a feasible solution with its VOF in $\left[f^{C_L}, f^{C_U}\right]$ and, consequently, FP becomes infeasible. When FP is infeasible, we may proceed further to check whether it is $f^{C_L} \leqslant f^{C_U} \leqslant Z^*$ or OP is infeasible via Property 3.

Property 3. *If FP without* (20) *is infeasible, then there exists no feasible solution to OP. If FP without Eq.* (20) *is feasible, the value calculated by plugging its optimal into* (1) *is a valid upper bound on OP.*

Proof. See the Appendix.

In either case, because the objective function of FP is a constant value, FP can be solved very quickly. Note that equations (19) and (20) can be implemented as the upper and lower bounds rather than usual constraints for faster program execution. The algorithm presented below first determines which of the two cases (Us-case and Ex-case) a given problem belongs. Afterwards, based on the evaluation result, it develops the lower and upper bounds as explained above. Using the constructed bounds, we can find the optimal solution of the problem in a reduced computational time.

Algorithm

Step 1. Solve LP1 to obtain f^{C_L} and f^{C_U}.

Step 2. Perform the Steps 2.1–2.3 to check the feasibility and to construct the bounds.

Step 2.1. Solve FP to see if $f^{C_L} \leqslant Z^* \leqslant f^{C_U}$. If FP is infeasible, then go to Step 2–3.

Step 2.2. Calculate $\sum_{t=1}^{T} y_t/T$ using the solution of FP and let $\upsilon \leftarrow \sum_{t=1}^{T} y_t/T$ and $\lambda \leftarrow f^{C_L}$. Go to Step 3.

Step 2.3. Solve FP without constraint (20).

i) If a feasible solution is obtained, then calculate $\sum_{t=1}^{T} y_t/T$ using the solution and let $\upsilon \leftarrow \sum_{t=1}^{T} y_t/T$ and $\lambda \leftarrow f^{C_U}$.

ii) If it is infeasible, terminate the algorithm.

Step 3. Use λ and υ to create a modified problem (MP).

MP:

$$Min\, Z = \sum_{t=1}^{T} y_t/T, \tag{23}$$

s.t.

$$-\sum_{j=1}^{J} (r_{jt} - r_j) p_j x_j \leqslant y_t, \quad t = 1, \ldots, T, \tag{24}$$

$$\sum_{j=1}^{J} (1 + d_j) p_j x_j \geqslant C_L, \tag{25}$$

$$\sum_{j=1}^{J} (1 + d_j) p_j x_j \leqslant C_U, \tag{26}$$

$$\sum_{j=1}^{J} (r_j - \rho - \rho d_j) p_j x_j \geqslant 0, \tag{27}$$

$$\sum_{t=1}^{T} y_t/T \geqslant \lambda, \tag{28}$$

$$\sum_{t=1}^{T} y_t/T \leqslant \upsilon, \tag{29}$$

$$l_j \leqslant x_j \leqslant u_j, \quad x_j \text{ integer}, \ j = 1, \ldots, J, \tag{30}$$

$$y_t \geqslant 0, \quad t = 1, \ldots, T. \tag{31}$$

Step 4. Solve MP. The solution obtained is the optimal solution to the original problem.

In Steps 2.1 and 2.2, we construct the upper and lower bounds when it is Us-case. If a given problem is classified as Ex-case, we calculate the bounds in Step 2.3. In either case, we proceed to Step 3 and utilize the bounds to formulate MP. When we solve MP in Step 4, we can find the OP's optimal solution.

To complete the algorithm, we should solve the LP problem (LP1) once and the MILP problems (FP and MP) up to two times. FP can be solved routinely because it has a constant number as its objective function. Furthermore, the bounds specified in MP are tight enough to cut the computational time considerably. More specifics will be discussed in the next section. All in all, fast computational time enabled this way makes it possible to find the optimal solutions of complex problems, which could not be solved to optimality otherwise.

4. Computational Experiences

In this section we explain the results of computational experiments. The objective of the analysis is two-fold. The main purpose is to estimate how tight the bounds are. The secondary purpose is to check the performance of the algorithm in a real investment environment. For this reason, the experiments are performed using real data sets from the Korean Stock Market.

We have coded the algorithm in object-oriented Pascal on a personal computer with Intel Pentium processor (1.2 GHz), 256 Mb memory, and a Windows 2000 operating system. CPLEX 7.0 is used to solve the LP and MILP problems. Inside CPLEX, the tolerance gap of the best integer objective and the objective of the best node remaining was set to 10^{-6}. A program run is terminated when the computational time exceeds a practical limit, say, ten hours during each test case.

The data is a monthly record of 600 stocks covering from July 1998 to June 2000. The required rate of return, ρ, is varied from 0.25% to 1.5% per month, with a step size of 0.25%. These values are suitable to reflecting the possible interest rates of the Korean financial market. We selected eight different values (10, 20, 30, 40, 50, 60, 70, and 80 million Korean wons) for the lower limit of the capital,C_L. The gap between C_L and C_U is varied from 1.25% to 10.00% of the total investment. With the experimental design, we generated 48 problem instances by randomly selecting 400 stocks from the data.

As proven theoretically, the algorithm successfully found an optimal solution for all the problem sets. The computational time required ranges from 148.94 to 1434.03 with the average of 672.84 CPU seconds. Thus it is safe to say that the speed of the algorithm is pretty fast.

In contrast, the straightforward application of the branch and bound method using CPLEX alone generates an optimal solution in 35 out of 48 instances, which is about a 72.9% success rate. All the unsuccessful cases are due to memory overflow rather than the time limit. It requires a huge amount of main memory, e.g., 5 Gb, to handle the trees and the memory requirement. When it falls short of the physical memory, it resorts to memory paging. Eventually, even with the memory paging, the operation is terminated with a memory overflow error. Thus whether or not the branch and bound method can solve a given problem is mainly determined by the memory size rather than the actual computing power of a system.

To measure the efficiency of the derived bounds, we have compared them with the VOF. As summarized in Table 1, it is shown that the lower bound is very close to the VOF. It deviates no more than 1.31%. The upper bound, however, shows a little more deviation ranging 1.14% to 9.52% of difference. It is also noted that the gap between the upper and lower bounds nearly equals to the capital gap when the capital gap is relatively small. For example, we see that the bounds are (-0.03, 1.40) for the capital gap of 1.43%. This can be explained by the shadow price, i.e., the percent change in the objective function equals to that of the right hand side as long as the basis remains unchanged. For a larger gap, say 5.00%, two values show more disparity because the basis is changed in this case.

As mentioned in (Kellerer *et al.*, 2001), the amount we are allowed to invest is usually an exact amount rather than the amount in a range. Previous research, however, substitutes the exact amount with a capital range. This approximation is necessary and has been justified because it is more practical to handle the amount in a range. However, it is not recommended to allow the gap to be unrealistically large. 2.00% or less of the gap size would be appropriate for most cases. Returning to the results, we have observed that the bounds are very tight deviating no more than 2.00% for this realistic gap range.

Another concern that we have in mind is how much improvement in solution quality can be made possible by the algorithm. For this purpose, we selected two heuristics, A and C, by Mansini and Speranza (1999), and calculated the percent error from the optimal solution:

$$\frac{(Z^H - Z^*)}{Z^*} \times 100(\%), \tag{32}$$

where Z^H and Z^* are the VOF's of the heuristic and the optimal solutions respectively.

Table 1

Percent deviations of the lower and upper bounds from the VOF

Gap of capital range (%)	Required rate of return $\rho(\%)$					
	0.25	0.50	0.75	1.00	1.25	1.50
1.25	(−0.04, 1.21)*	(−0.11, 1.14)	(−0.04, 1.21)	(−0.04, 1.21)	(−0.03, 1.21)	(−0.02, 1.23)
1.43	(−0.03, 1.40)	(−0.06, 1.37)	(−0.02, 1.41)	(−0.03, 1.40)	(−0.02, 1.41)	(−0.04, 1.39)
1.67	(−0.10, 1.56)	(−0.07, 1.60)	(−0.03, 1.63)	(−0.06, 1.61)	(−0.05, 1.61)	(−0.04, 1.62)
2.00	(−0.06, 1.94)	(−0.09, 1.91)	(−0.07, 1.93)	(−0.10, 1.90)	(−0.10, 1.90)	(−0.04, 1.96)
2.50	(−0.08, 2.42)	(−0.11, 2.39)	(−0.06, 2.44)	(−0.08, 2.42)	(−0.06, 2.42)	(−0.09, 2.41)
3.33	(−0.11, 3.22)	(−0.12, 3.21)	(−0.61, 2.47)	(−0.11, 3.22)	(−0.07, 3.22)	(−0.08, 3.26)
5.00	(−0.13, 4.87)	(−0.24, 5.44)	(−0.71, 4.22)	(−0.18, 4.81)	(−0.24, 4.81)	(−0.11, 4.89)
10.00	(−0.65, 9.28)	(−0.44, 9.52)	(−1.31, 8.56)	(-1.27, 8.60)	(−0.77, 8.60)	(−0.54, 9.41)

* Negative values in the parentheses are the percent deviations below VOF and positive values are those of above.

Table 2

Percent errors of the Procedures A and C compared to the optimal

Gap of the capital range (%)	Required rate of return $\rho(\%)$						Average
	0.25	0.50	0.75	1.00	1.25	1.50	
1.25	(0.0102, 0.0024)*	(0.0185, 0.0153)	(0.0016, **0.0000**)⊗	(0.0125, 0.0050)	(0.0132, 0.0106)	(0.0326, 0.0141)	(0.0148, 0.0079)
1.43	(0.0078, 0.0044)	(0.0433, 0.0175)	(0.0059, 0.0034)	(0.0809, 0.0711)	(0.0363, 0.0003)	(0.0231, 0.0177)	(0.0329, 0.0191)
1.67	(0.0205, 0.0022)	(0.0820, 0.0321)	(0.0246, 0.0087)	(0.0860, 0.0828)	(0.0105, 0.0074)	(0.0643, 0.0222)	(0.0480, 0.0259)
2.00	(0.0315, 0.0090)	(0.0796, 0.0227)	(0.0314, 0.0056)	(0.0175, 0.0042)	(0.0353, 0.0109)	(0.0237, 0.0066)	(0.0365, 0.0098)
2.50	(0.0192, 0.0118)	(0.0648, 0.0111)	(0.0119, 0.0108)	(0.0016, **0.0000**)	(0.0712, 0.0081)	(0.0675, 0.0547)	(0.0394, 0.0161)
3.33	(0.0805, 0.0110)	(0.0742, 0.0099)	(0.0194, 0.0049)	(0.0198, 0.0165)	(0.0649, 0.0629)	(0.0419, 0.0076)	(0.0501, 0.0188)
5.00	(0.0253, 0.0137)	(0.1063, 0.0269)	(0.0548, 0.0157)	(0.1195, 0.0592)	(0.1260, 0.1022)	(0.0653, 0.0021)	(0.0829, 0.0366)
10.00	(0.5617, 0.0861)	(0.2150, 0.0871)	(0.4055, 0.2726)	(0.3175, 0.1128)	(0.2038, 0.0736)	(0.5160, 0.1268)	(0.3699, 0.1265)
Average	(0.0946, 0.0176)	(0.0855, 0.0278)	(0.0694, 0.0402)	(0.0819, 0.0439)	(0.0701, 0.0345)	(0.1043, 0.0315)	(0.0843, 0.0326)

* Elements in the parentheses are the percent errors of the Procedures A and C.

⊗ Bold case represents the instance where an optimal was found by the corresponding heuristic.

Table 2 shows the percent errors by Procedures A and C. Summarizing the results, we may reduce up to 0.5617% and 0.2726% errors by implementing our algorithm instead of Procedures A or C respectively. We also discovered that Procedures A and C can rarely find the optimal solution. Procedure C found twice and Procedure A did none out of 48 problem cases. From a practical point of view, less than 1% of improvement achieved by the algorithm may not be significant. Nevertheless, from a theoretical point of view, it is an important groundwork for more complex MILP models such as m-MAD.

In terms of computational time, our algorithm takes around 3.25 times longer than Procedure A, but around 2.81 times shorter than Procedure C on average. The details can be found in Table 3. We think the computational time is satisfactory because we would rather solve the portfolio problem to optimality, even though it takes some more time than to obtain an approximate solution.

Lastly, we have examined the relation between the gap of the investment, $C_U - C_L$, and the performance of the algorithms. More specifically, we investigated the influences of the gap size on the solution quality and the computational time. For the test, we assumed eight different values ranging from 1.25% to 10.00% for the gap. For each setting, we run 10 different test problems. The results are in Figs. 2 and 3. We first note that the accuracy of the proposed algorithm is not affected at all because it is an algorithm to find an optimal solution. But the performance of the other two algorithms is noticeably influenced by the gap. For the gap range of 5% or less, the percent error of the solution remains less than 0.10%. However, as the gap increases beyond 5.00%, the accuracy of

Table 3
Minimum, average and maximum of the computational times in CPU seconds

	Minimum	Average	Maximum
The proposed algorithm	148.94	672.84	1434.03
Procedure A	82.23	206.84	688.69
Procedure C	732.28	1892.95	6284.09

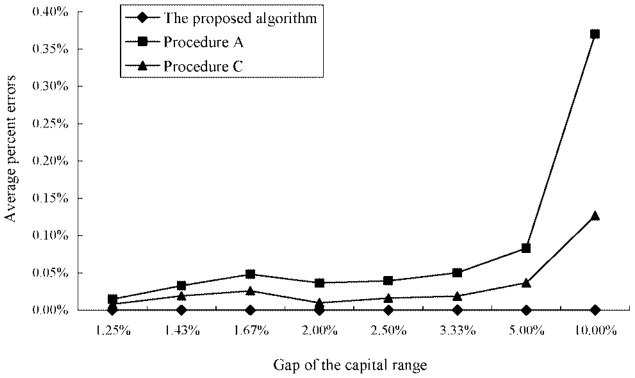


Fig. 2. Percent errors of the algorithms for various capital gaps.

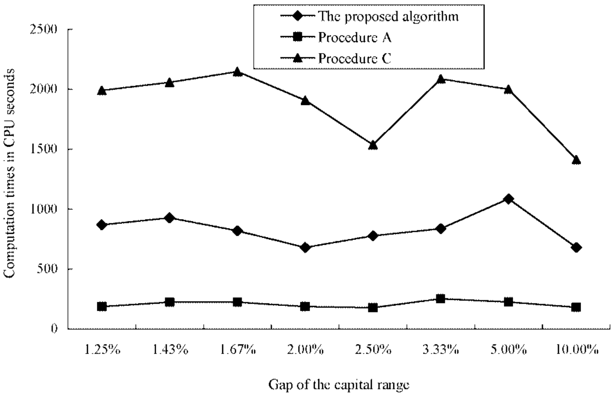


Fig. 3. Average computational times of the algorithms for various capital gaps.

the algorithms degrades very rapidly.

The running time of the three algorithms is depicted in Fig. 3. It shows that the proposed algorithm exhibits little change over the whole gap range except for a little hump at 5%. The other two algorithms reveal a similar feature. Overall, the computational time of all three algorithms does not seem to be influenced much by the gap of the capital range.

5. Conclusions

In this paper, we have investigated the MAD model to construct the upper and lower bounds on the optimal objective function value. By taking advantage of the bounds and properties that we found, we have developed an algorithm to solve the MILP problem. It is proved mathematically that the algorithm can solve the problem to optimality. Using the data from the Korean Stock Market, we performed extensive experiments to check the efficiency of the bounds as well as the performance of the algorithm. As a result, we have observed that the bounds are very close to the optimal objective function value, and the algorithm can solve a variety of problems in a reasonable computational time. For all problem sets, the method successfully generates optimal solutions in less than 1.434 CPU seconds.

To sum up, the method we have developed is accurate and efficient enough to solve realistic problems in a reasonable amount of time. The bounds we have derived will be applicable to a wide range of problems that have similar structure. We hope that the work in this paper would contribute to expanding the real world application of portfolio optimization and become a good addition to the existing solution methodologies.

Appendix

Proof for Property 1

(i) Suppose that we change M in (10) to $M + \varepsilon$ where ε is very small positive number. Then, to satisfy (10), some x_j should be increased by $\frac{\varepsilon}{(1+d_j)p_j}$. In terms of the objective function value, x_j can not have less below average deviation than the securities already in the optimal solution. Thus the increase in x_j can not improve the objective function value. Reasoning this way leads to the fact that the objective function is a nondecreasing function in M(More specifically, it is a nondecreasing piecewise linear convex function. See pp. 302-306 of Murty (1983)). Thus $f^{C_L} \leqslant f^{C_U}$.

(ii) At the optimal of the LP-relaxation of OP (LROP), equation (3) becomes binding whereas (4) is nonbinding. The reason is that the VOF of LROP is a piecewise linear and nondecreasing function in the money invested. Since other equations except (3) and (4) are common to both problems, LP1 with $M = C_L$ and LROP are equivalent problems. Since the VOF of LORP is a lower bound for the original problem, the VOF of LP1 with $M = C_L$ is also a lower bound. Thus $f^{C_L} \leqslant Z^*$.

Proof for Property 2

Consider LP1 when$M = C_U$. We assume that there are two kinds of securities shown in Table 4. Without loss of generality, we further assume that $t = 1$ and either security can satisfy all the constraints of LP1. Then the optimal solution to LP1 is $X^{LP1} = (\frac{C_U}{(1+d_1)p_1}, 0)$ with $f^{C_U} = 0$ while the optimal solution to OP is $X^{OP} = (0, 1)$ with its $Z^* = 1$. Thus, for this problem, $f^{C_U} = 0 < Z^* = 1$.

Table 4

Data for Property 2

Security number(j)	Purchase price	$r_{j1} - r_j$
1	p_1	0
2	$\frac{C_U - \varepsilon}{1+d_2}$	$-\frac{(1+d_2)}{C_U - \varepsilon}$

p_1 satisfies $p_1 > C_U$,
ε: minimal monetary unit.

Proof for Property 3

FP without equation (20) has exactly the same set of constraints as OP. Thus, when it is infeasible, so is OP. On the other hand, if FP without (20) is feasible, then its optimal is also a feasible but not necessarily optimal to OP. Thus, the value calculated by plugging its optimal into (1) is a valid upper bound[1].

[1] Actually FP without (20) has one more equation, (20), than OP. However, it is concerned with the valid lower bound and, therefore, should be satisfied by any feasible solution to OP. Thus, (20) can be ignored when we consider the feasibility of OP.

References

Kellerer, H., R. Mansini and M.G. Speranza (2000). On selecting a portfolio with fixed costs and minimum transaction lots. *Annals of Operations Research*, **99**, 287–304.

Konno, H., and H. Yamazaki (1991). Mean-absolute deviation portfolio optimization model and its application to Tokyo stock market. *Management Science*, **37**, 519–531.

Konno, H., and A. Wijayanayake (2001). Portfolio optimization problem under concave transaction costs and minimal transaction unit constraints. *Mathematical Programming*, **99**, 287–304.

Mansini, R. (1997). *Mixed Integer Linear Programming Models for Financial Problems: Analysis, Algorithms and Computational Results.* PhD Thesis, Bergamo University (in Italy).

Mansini, R., W. Ogryczak and M.G. Speranza (2003). LP solvable models for portfolio optimization. *IMA Journal of Management Mathematics*, **14**, 187–220.

Mansini, R., and M.G. Speranza (1999). Heuristic algorithms for the portfolio selection problem with minimum transaction lots. *European Journal of Operational Research*, **114**, 219–233.

Markowitz, H. (1952). Portfolio selection. *Journal of Finance*, **7**, 77–91.

Murty, K.G. (1983). *Linear Programming.* John Wiley & Sons, New York.

Satchell, S.E., and A.E. Scowcroft (Eds.) (2003). *Advances in Portfolio Construction and Implementation*. Butterworth-Heinemann, Oxford.

Speranza, M.G. (1993). Linear programming models for portfolio optimization. *Finance*, **14**, 107–123.

Speranza, M.G. (1996). A heuristic algorithm for a portfolio optimization model applied to the Milan stock market. *Computers and Operations Research*, **23**, 433–441.

Zenios, S.A., and P. Kang (1993). Mean-absolute deviation portfolio optimization for mortgage-backed securities. *Annals of Operations Research*, **45**, 433–450.

J.S. Kim is a professor of information and industrial engineering at Hanyang University in Korea. He received his doctor of engineering and MS degrees from the University of California at Berkeley. His areas of research include mathematical programming, multi-echelon inventory system, game theory applications to logistics problems, and investment analysis.

Y.C. Kim received his MS degree in industrial engineering from Hanyang University. He is currently pursuing his PhD degree in industrial engineering from Hanyang University. His research interests are portfolio management and integer programming.

K.Y. Shin received his MS degree in industrial engineering from Hanyang University. He is currently pursuing his PhD degree in industrial engineering from Hanyang University. His research interests are mathematical programming and forecasting.

Portfelio optimizavimo algoritmas

Jong Soo KIM, Yong Chan KIM, Ki Young SHIN

Optimizuojant vertybinių popierių portfelį paprastai, arba minimizuojama rizika duotai grąžai, arba maksimizuojama grąža leistinam rizikos lygiui. Šios srities darbai remiasi vidurkio-dispersijos modeliu, formuluojamu kaip kvadratinio programavimo uždavinys. Kadangi pastarasis pasirodė praktiškai mažai taikytinas dėl skaičiuojamųjų problemų, buvo sukurta daug alternatyviųjų modelių. Šiame darbe mes nagrinėjame alternatyviuosius modelius, labiausiai skirdami dėmesio vidurkio-absoliutinio nuokrypio modeliui. Mes įvertiname optimalios tikslo funkcijos viršutinę bei apatinę ribą ir pateikiame uždavinio sprendimo algoritmą, kurio tinkamumas yra įrodomas matematiškai. Algoritmas yra testuojamas su Korėjos vertybinių popierių rinkos duomenimis. Testavimo rezultatai patvirtina teorines išvadas dėl algoritmo tinkamumo spręsti portfolio problemoms per priimtiną kompiuterio laiką.