2018 INFORMS O.R. & Analytics Student Team Competition – ENTRY FORM

Entry Number: [2018ORASTC252]

Executive Summary (not to exceed 2 pages)

Team Makeup & Process

Framing the Problem

Data

1. The Structure of the Data

   The data from Principal consists of 3 parts which are timeseries data, riskmodels data, and result template data.

   – Time Series Data :

   – Riskmodels Data :

   – Result Template Data :

2. Data Pre-processing and Rescailing

   There are some difficulties in applying the data given from Principal directly to the model. Therefore, we present the following data preprocessing process. To make the data composed of three parts more flexible, a process of data preprocessing formed dataframe using the Pandas Library(http://pandas.pydata.org) in Python.

   In particular, as rebalancing is carried out, it is configured to extract not all time series data, but only the data needed for the iteration. The parameters used in model were extracted from data frames formed for each period, when each parameter (Alpha Score(α), Beta(β),4 Weekly Returns (r), etc) was set with the 'SEDOL' index as the key value. This dictionary data type is proper to consider a list of assets that may change every period.

   The parameter Omega (Ω) is specified as the covariance matrix for a given data. In this case, the index in the columns and rows are 'SEDOL', the key value of the time series dictionary derived from the above, and are constructed in the form of full matrix.

3. The Analysis of the Data

Methodology Approach & Model Building

1. Formulation Approach

Model Building: Before buiding the model, we tried to solve the Quadratic Constraint Quadratic Programming (QCQP) problem by using the IBM CPLEX, which is well known Optmization tool, to determine the difficulty of the problem. Since CPLEX can not deal with non-linear constraints,the given formulation is required to be reformulated. First, some constraitns could be reformulated as follows:

$$
\begin{array}{lll}
\text{(MIP)} & (1) \sim (8) & \\
 & y_i \geq w_i, \quad \forall i \in N & (9^*) \\
 & y_i \leq w_i + 0.999 \quad \forall i \in N & \\
 & 50 \leq \sum_{i \in N} y_i \leq 70 & \\
 & y_i \in \{0,1\}, \quad \forall i \in N & \\
 & d^T \Omega d \ \leq 0.01 & (11*)
\end{array}
$$

The decision variables y_i is 1 if asset $i \in N$ is selected where w_i is bigger than 0. Because $w_i \leq 0.001$ is considered to $w_i = 0$, $y_i = 0$ if $w_i = 0$. The opposite is the same as well. Constraints (10) and (11) are also non-linear. However, constraints(10) and (11) can not be easily linearized unlike constraitns(9). However, if non-convex constraints are linearized, they significantly impact speed by increasing the number of variables to determine and the number of constraints to consider. In other words, it is very inefficient to reformulate constraints (10) and (11), which represent the limits of Active Share and Tracking Error, respectively. Therefore, the experiments are conducted to solve MIP which did not consider the constraints (10) and the left side of the constraints (11), and then we check whether the best feasible solution from MIP violate limit range of Tracking Error and Active Share. The first date was used for the experiements.

Table 1 shows the number of feasible solutions found by branch and bound for a given time and several indicators of the best feasible solution among feasible solutions of MIP. In particular, GAP represents the difference (%) from the best integer solution and it can be confirmed that no good solution is found within 3 minutes. In terms of finding a feasible solution, the following obeservations were derived.

Observation1. In most cases, the range of Active Share is satisfied.

Observation2. Tracking Error is less than lower bound (0.05).

Table 1: The results of solving MIP by Cplex

B&B time (sec)	# of feasible	GAP(%)	Tracking Error	Active Share	Violated Constraints
30	0	-	-	-	-
60	0	-	-	-	-
90	2	100.3%	0.0018	0.72	(11) TE
120	2	100.3%	0.0012	0.64	(11) TE
150	2	100.3%	0.0018	0.72	(11) TE
180	8	100.6%	0.0004	0.50	(10)AS, (11) TE

According to Observation 2, Tracking Error should be raised. For the given data, benchmark weights are distributed over about 500 assets, while the portfolio weight is 0 ($w_i = 0$) for most of the remaining assets except for 50 to 70 selected assets. Thus, for non-selected assets i, d_i can be expressed as $-w_{\text{bench}}$. In other words, if the assets whose w_{bench} are large are not selected, the $\sum_{i\ inN} d_i$ becomes larger which means the Tracking Error is increased. On the other hand, for the selected 50 to 70 assets, most w_i are larger than w_{bench}. If the difference between w_i and w_{bench} is small, $\sum_{i\ inN} d_i$ is also small. The difference between w_{bench} and w_i are required to be large for all selected assets, so $\sum_{i\ inN} d_i$ becomes larger when the difference between w_{bench} and w_i are large for all assets i. That is, the tracking error also becomes large. First, We took a look for the average of portfolio weight and have noticed that about two-thirds is smaller than average and a third is larger than the average. Therefore, based on the average of portfolio weight, the portfolio weights are raised for the assets whose portfolio weight are smaller than the average. Conversely, the assets whose portfolio weights are larger than the average are decreased. As a result, the Tracking Error value becomes large, making the solution feasible. To sum up, there are the ways to increase the tracking error by not selecting the assets with a larger w_{bench} or by increasing the difference between w_{bench} and w_i for the selected assets. Since the value of the bench weight is practically insignificant, the method which increases the difference between w_{bench} and w_i for the selected assets is more efficient.

Bisectional Search Algorithm: We applied the idea to the bisectional search algorithm. The bisectional search algorithm is an iterative algorithm that narrows the range and finds a feasible solution. When narrowing the range, if the feasible solution exists in the range, search in the right range based on the median of the range, and search in the left range if the feasible solution does not exist. An example is shown in 1. Let $\sum_{p\in P} w_p = 0.6$, where P is a set of assets whose w_i is smaller than the avarage of the portfolio weights. First, the solution is searched by dividing the interval between 0.8 and 1.0. If the solution is a feasible solution that satisfies both the range of Active Share and Tracking Error, narrow the range and search again in the left section. The search is stopped when the width of the range($\underline{Z} \sim \overline{Z}$) is less than the end condition which is set to 0.001. One of the feasible solutions, obtained from the bisectional search with the lowest objective value, is returned. Tracking Error increases as

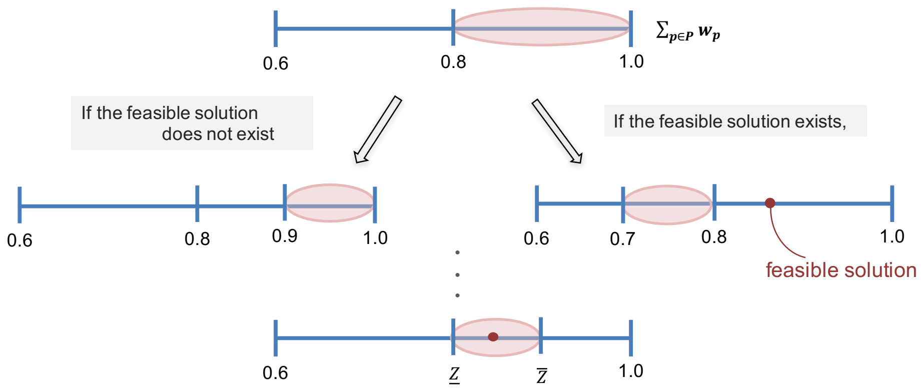

Figure 1: An Example of Bisectional Search

$\underline{Z}$ approaches to 1. Conversely, Tracking Error decreases as the $\overline{Z}$ approaches to $\sum_{p\in P} w_p$ (0.6 in the example). Therefore, the following constraints are derived through the bisectional algorithm : $\sum_{p\in P} w_i = (\underline{Z} + \overline{Z})/2$. Experimental results including the constraints show that Tracking Error is converged at the lower bound (0.05), except for the case where the difference between Tracking Error value and the lower bound is large before the start of the bisectional search.

2. Neural Network Approach

The application of GAN : **Generative adversarial networks (GANs)** are well known deep neural net(NN) architectures comprised of two networks. One neural network, called the **Generator(G)**, generates new data instances, while the other, the **Discriminator(D)**, decides whether each instance of data is real dataset or not. Both neural networks learn to alternate with each other. The G trains to generate data similar to real data, and the D trains to distinguish real data from fake data generated by G.

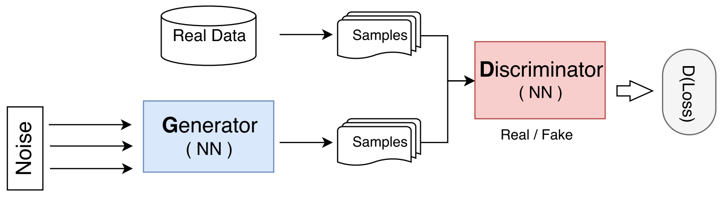

Figure 2: The strucutre of basic GAN

See Figure 2. G generates fake data when arbitrary noise z is given as a input through a NN composed of layers which are a input layer, multiple hidden layers and a output layer. G tries to make $D(x) = 1$ for sample x extracted from fake data, and D tries to make $D(G(z)) = 0$ for sample $G(z)$, where $z \sim p_z(z)$. Therefore, the loss function $V(D, G)$ of GAN can be expressed as follows :

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\ p_{data}(x)}[\log D(x)] + \mathbb{E}_{x\ p_x(x)}[\log(1-D(G(z)))]$$

We use the GAN 's basic idea of learning networks with adversarial relationships between two networks to solve the problem of portfolio optimization. G determines the decision variable w of the formulation, and D determines if the data generated from G is feasible. D is based on the formulation, and the sample data $G(z)$ generated checks both the feasibility of the solution and the quality of objective value. Specifically, when w_i is determined in G, the discriminator finds the sum of the objective value and the scaled value of the sum of values deviating from each constraint. Therefore, the value of loss in D is the sum of the degree of violation from the constraints and objective value. G is a deep NN structure consisting of one input layer, three hidden layers and one ouput layer. G tries to minimize the loss of D for the sample $G(z)$ generated from the random input noise $z \sim \mathcal{U}(-1,1)$. That is, the decision variable w generated in G is trained to be feasible for all constraints and the objective value to be minimized. Unlike the basic GAN structure, GAN for portfolio optimization has G for NN structure, but D is not, and only G is required to train. The loss function $V(G)$ of GAN can be expressed as follows :

$$\max_G \mathbb{E}_{x\ p_{data}(x)}[\log(D(G(z)))]$$

The training is performed with a total of 2000 random datasets so that the loss value is minimized. At this time, the batch size is 1000 datasets, and then sample data is extracted after training 1000 datasets. At this time, if the value of D (G (z)) is larger than the threshold value (0.995) for sample data G (z), the solution is considered as feasible. In order to understand whether GAN is being well trained, we let GAN train with 2000 random datasets and examined the results. As a result of the experiments, we realized some of constraints are violate. However, we also realized that only about 5 percent of the constraints of the whole are violated, and the degree of violation is very small. Because of threshold, the results of trained GAN allow in some degree of infeasible.

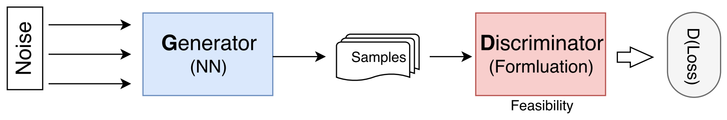

Figure 3: The strucutre of GAN for portfolio optimization

3. GAN-MP Hybrid Approach

We propose a new **GAN(Generative adversarial networks)-MP(Mathmatical Programming) Hybrid approach** that complements the disadvantages of formluation approach and neural network approach and enhances their advantages. Several limitations could be derived from the three approaches we present above. When solving the problem with the formulation appraoch, we could not find a reasonable solution because to consider about 500 assets per iteration the number of decision variables to decide and the number of constraints to be satisfied are too high . Especially, constraints (9) which select cardinalities considering the target range on number of stocks make the problem difficult. In the case of Neural Network Approach, there is a great advantage in that a solution that does not get significantly out from the whole constraints and that has a good objective value can be considered at the same time. However, it is difficult to find a soluton satisfying the feasibility for all the constraints and the fact that the training time takes too much time. The GAN-MP hybrid approach is a highly efficient approach that maximizes the advantages of GAN and formulation-based optimizations. This appraoch consists of three steps as shown in Figure 4. Step1 is the process of finding the cardinality set which is the combination of the selected assets. The cardinality set selected in Step1 solves the QCQP problem with the selected assets fixed and finds the local optimal solution in Step2. According to the result of Step2, the update algorithm finds cardinality set that can reduce the objective value and improves the solution.

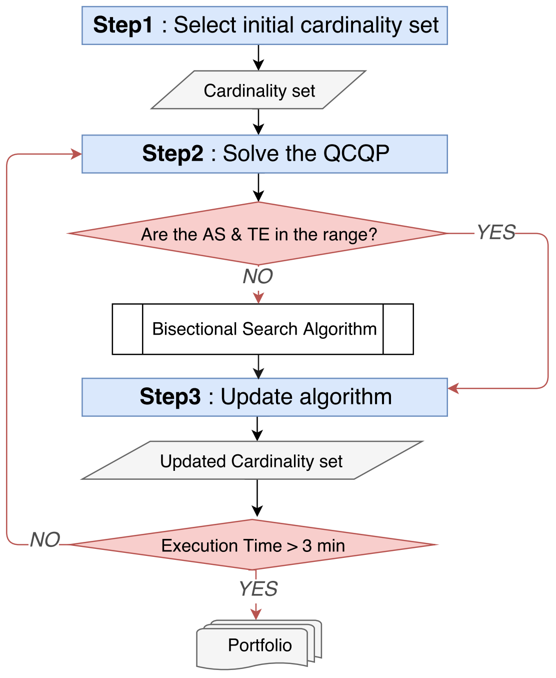

Figure 4: A Flowchart of Hybrid Approach

(Step1) Select Initial Cardinality Set & (Step2) Solve the QCQP : When the number of assets to be considered is large, the problem becomes large as the number of constraints increases as well as decision variables. Thus, the process of step1 filters out good assets that are crucial for minimizing the objective value. At this time, the set of good assets must meet the following criteria.

Criterion 1. The combination of assets should be satisfy all constraints

Criterion 2. The combination of assets is chosen to minimize the risk and maximize return.

In order to meet these criteria , the whole assets are required to be considered. We propose a formulation approach that finds the solution through Mixed Integer Programming (MIP) and a neural network-based GAN as a way to satisfy constraints and to take low values of objective value globally. Therefore, the experiments are carried out with two algorithms (MIP, GAN). For a given time, the MIP proceeds branch and bound(B&B) and the GAN proceeds to train the generator consisting of NN (Step1). After the given time, the assets, $w_i > 0.001$ for all asset $i \in N$,were extracted in the best solution found in each algorithm. The portfolio optimization problem that determines the weight of each assets as the cardinality set is fixed (Step2) . Accordingly, constraints (9) does not to be considered anymore, and the set N changes to S, which stands for set of selected assets for all of the constraints. Therefore, the number of constraints and decision varaibles are reduced.

- Experiment Environment
  - Number of Experiments : 50 times
  - The B&B Timelimit for MIP : 90 seconds (minimun time for feasible solution)
  - Training Time for GAN : 20 seconds
  - Solver of MIP : Cplex
  - Deep Learning Library : Pytorch
  - Data of the first period (2007/01/03)
- Experiment Results :

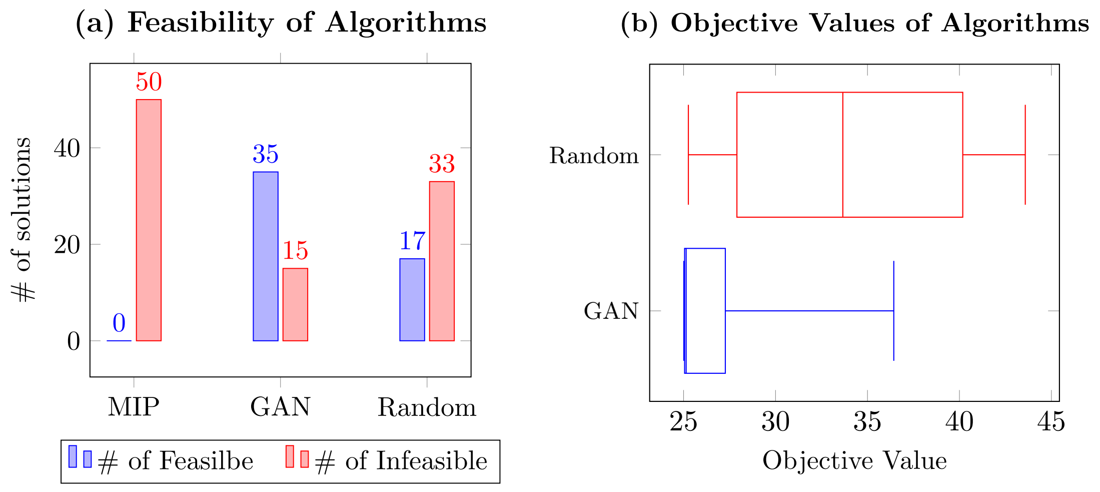

Figure 5: Results of Algorithms

The cardinality sets are derived by solving MIP problem with the Cplex, and all of the results were infeasible when solving the portfolio optimization problem with the fixed assets. One reason that cannot derive the feasible solutions is that Tracking Error is too low because the constraints of Tracking Error are not reflected in MIP. Thus, Cplex does not suit for selecting initial cardinality set. If random samples are specified as an initial cardinality set, the solutions are often infeasible, and objective values are lower than GAN. Therefore, the initial cardinality set is assigned to the GAN because the feasible solution is derived more reliably than others, and the objective value is relatively low.

– Threshold and Suggestion :

  GAN is composed of NNs, and the result quality and training time vary depending on computing environment of NN. Therefore, we recommend that Principal could improve the training speed and quality by using GPU-accelerated computing or through some advanced technologies such as Amazon Cloud services. By improving the environment, more accurate and better solutions are derived by adjusting the hidden layer addition and the learning rate that we could not reflect now because of computing power.

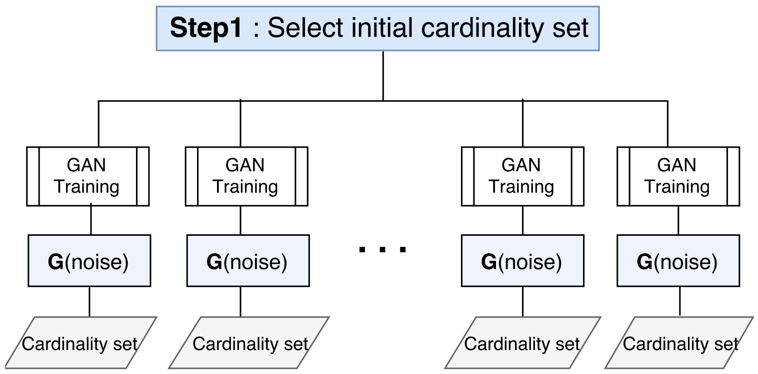

Figure 6: A Flowchart of Hybrid Approach

We present a model that can maximize the benefits of GAN in a given situation. The major advantage of the GAN is that it can extract solutions whenever noises are given to G by learning the weight and bias of the generator. The server that currently runs the code has eight cores, so we figure out the server can run up to 8 GANs simultaneously on 8 cores. Therefore, we use the parallel process to be 8 GANs training at the same time and then obtain 8 initial cardinality sets each as a result. The parallel process uses the Python Library of Ipython Parallel (https://ipyparallel.readthedocs.io), and Figure 6 shows the process of obtaining 8 sets of cardinality sets by running 8 GANs using Ipython Parallel.

(Step3) Update Algorithm : The assets to be included in the portfolio are selected by GAN, and the MIP is solved by Cplex for determining weights to each assets. As a result of solving QCQP problems, the solutions are derived. At this time, the local optimal solution may be improved depending on which asset is selected. Therefore, based on good initial cardinality set, it is necessary to perform a local search by changing the items of the asset, the weight of the asset, and the number of selected assets. However, if the initial cardinality set obtained from the GAN is not good in Step 1, changing the number of assets does not improve the result significantly. To determine the bad cardinality set, we check the quality of the solution when we solve the MIP with the initial cardinality set before updating the cardinality set. The following process is performed according to the result of 8 solutions. For 4 cardinality sets with bad results, it is required to find a new initial cardinality set (→ Step1. Select Initial Cardinality Set). For 4 the other cardinality sets with good results, it adjusts the assets to improve results (→ Step3. Update Algorithm).

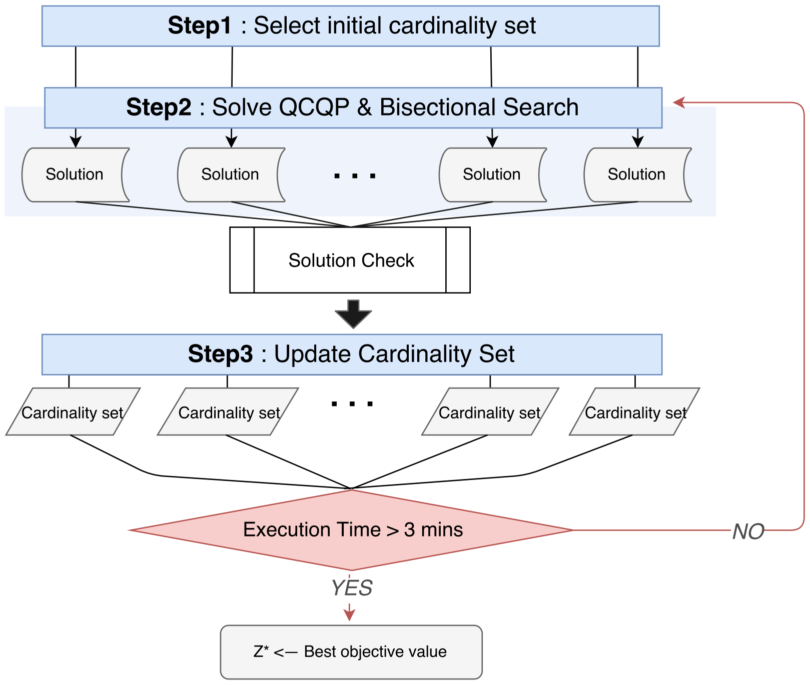

Figure 7: A Flowchart of Hybrid Approach

Basically, update algorithm has 2 goals.

(1) One is that the update algorithm finds one or multiple assets whose objective value is lower when one or multiple assets is/are added or removed.

(2) The other is if the result of current cardinality set is infeasible, it also adjusts the cardinality set so that it becomes feasible.

To accomplish these two goals, the update aglrithm is supposed to adjust the cardinality set solving the problem and improving the solution until it is reachead to the end condition. First, the update algorithm finds out which asset gets better to lower the objective value when it gets out of the portfolio, and which asset should be added to improve the outcome. From the viewpoint of determining which asset is included in the portfolio, there is a great difficulty in deriving an objective value that can be obtained only when the weight of each asset is determined. We therefore take into account the maximum value that an asset has on the determination of the objective value. In the objective function, the risk is minimized while the return is maximized. Let C is a set of assets whose asset i is in cardinality set, and K is a set of assets whose asset is not in cardinality set, that is candidate set for update algorithm. In order to find an good asset with higher return and lower risk, return and risk for the candidate asset $k \in K$ are calculated as follows:

- Return = The maximum return that occurs as the asset k comes in $\Rightarrow \alpha_k$
- Risk = The maximum risk that occurs as the asset k comes in $\Rightarrow \sum_{c \in C} \Omega_{ck} + \Omega_{kk}$

On the contrary, the asset $c^{\prime}$ that increases the objective value because the risk is large and the return is small among the assets included in the current cardinality set is also determined by the following criteria :

- Return = The maximum return that decreases as the asset $c^{\prime}$ leaves the cardinality set $\Rightarrow \alpha_{c'}$
- Risk = The maximum risk that decreases as the asset $c^{\prime}$ leaves the cardinality set $\Rightarrow \sum_{c \in \{c | c \in C \text{ and } c \neq c'\}} \Omega_{cc'} + \Omega_{c'c'}$

Therefore, in order to improve the objective function, the update algorithm minimizes the Risk − Return for the new candidate asset k and eliminates the asset $c^{\prime}$ which have a bad effect on the existing cardinality set. Repeating this process can result in a local optimal solution as a result of obtaining a good cardinality set. However, infeasible cardinality sets can be obtained in many cases when considering the risk and return to obtain a good cardinality set. The cardinality set is infeasible, which means that when a QCQP problem is solved after fixing an asset with a cardinality set, a feasible solution can not be found in violation of some constraints. We have noticed that most infeasilbe cardinality sets occur in Constraints 7 and 8 indicating the limits of Sector Active Share and Market Cap Quintile (MCAPQ) Active Share. To see why, we took a look at data for the first date (date of January 3rd, 2007) of a given dataset. Figure 6 shows the sum of the weights for each sector and MCAPQ to satisfy Constraints 6 and 7. Because w_{bench} is different for each asset, the scope varies depending on what asset is included in the sector and the MCPQA. In other words, when the sum of each sector and MCAPQ is sensitive, it becomes a difficult constraint to satisfy. Therefore, if the constraints 6 and 7 are violated in the process of improving the cardinality set, adding deficient sector and mcapq assets is prevent infeasible.

Analytics Solution and Results

In this section, we report results of computational experiments for solving portfolio optimization with the methodology we present. First, we show how to set the parameter λ to be considered in the problem in order to obtain the optimal solution before proceeding with the experiment. Also, we present how rebalancing reflects in our proposed model. Finally, the results of the computational experiments with the determined parameters are present.

1. Parameter Setting

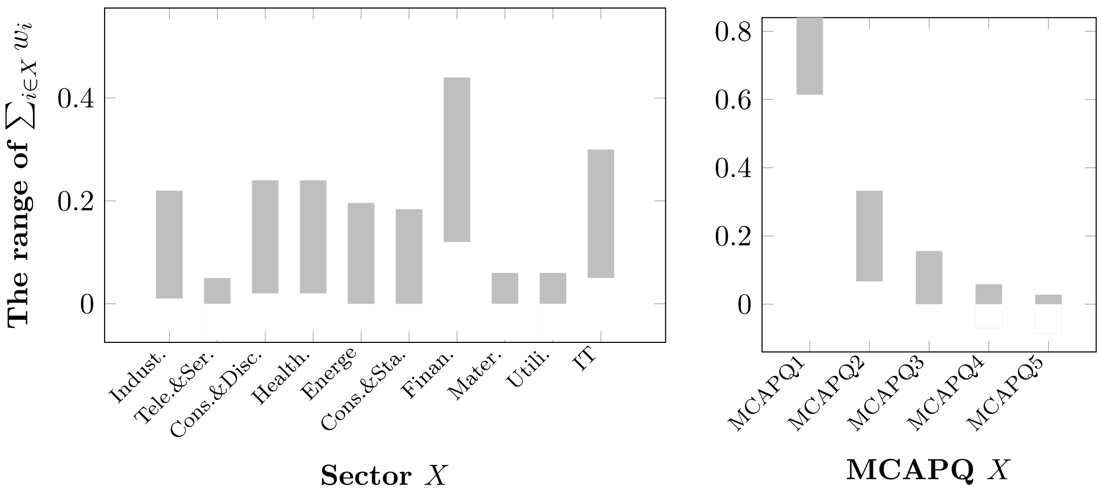

Figure 8: The sum of the weights for each Sector and MCAPQ

To set parameters, various experimental analyzes are required with real data. However, there is a limit to obtaining the real data such as α, Ω, β, and bench weight from the historical data. we assume that the year 2007 is a parameter adjustment period, and analyzed the data in 2007.

The Parameter λ: Parameter λ reflects how much return will be considered relative to risk in objective function. When the value of λ is large, the return is considered a lot, and when the λ value is low, the risk is considered more. We used 2007 data to analyze whether the $d^T\alpha$, which represents a return on objective, has an influence on the overall return index (r_{opt}) as we adjusted the λ. In other words, as the lambda value increases, more revenue will be taken into account, so that it can be inferred that the r_{opt} is more affected by return or risk.

2. Rabalancing

Portfolio turnover depends on how many assets in the portfolio chance from rebalance to rebalance, and the value of turnover has effects on Information Ratio(IR). Therefore, it is required to consider turnover in the portfolio optimization. We apply turnover to the both GAN for finding initial cardinality set and QCQP problem for determining the weight of the selected asset. For applying it to GAN, the calculated turnover$_t$($\sum_{i\ inN} |w_{i,t} - w^{pre}_{i,t-1}|$)was directly taken into account in the objective function. On the other hand,for applying it to MIP, it should be linearlized. We reflect turnover into the objective function in MIP as follows:

$$\begin{aligned} \text{(MIP)} \quad \min \quad & d^T\Omega d - \lambda d^T\alpha - \omega \sum_{i\in N} o_i \qquad (2^*) \\ \text{s.t} \quad & (3) - (11) \\ & o_i \geq w_{i,t} - w^{pre}_{i,t-1}, \quad \forall i \in N \\ & o_i \geq -w_{i,t} + w^{pre}_{i,t-1}, \quad \forall i \in N \end{aligned}$$

The decision variables o_i is the difference between w_i of previous period(t-1) and w_i of current(t) period for all asset $i \in N$. Because of constraints added, $w_{i,t} - w^{pre}_{i,t-1}$ is positive for all asset $i \in N$. The parameter ω represents the weight of turnover for objective value considering risk and return.

The Parameter ω:

3. Computational Experiments

Experiment environment : The experiments were performed on an Intel Core 3.5 GHz PC with 32GB memory and ILOG CPLEX 12.6 were used as an MIP solver, and Pytorch for Python 3.6 was used as a deep learning framework.

The Portfolio Performance Statistics :

Portfolio Performance Statistics

2007-01-01 to 2016-12-31	**Portfolio**	**Benchmark**
Cumulative Return	%	%
Annualized Return	%	%
Annualized Excess Return	%	–
Annualized Tracking Error	%	–
Sharpe Ratio		
Information Ratio		–

4. Robust Optimization

The results obtained earlier are derived by deterministic values which are α and Ω. However, this is a parameter value with uncertainty, so we figure out the change of α and Ω in the historical data. We first tried to secure the value of risk and alpha score for 2007 by analyzing the historical data. The analyzed data is obtained from Yahoo Finance and is historical data from 1970 to 2017. For the first time, we are looking at the risk of an asset in 2006 with extensive historical data. We obtained data on close stock prices for approximately 400 assets from historical data among 492 assets included in January 3, 2007 in S & P 500 and derived a cross covariance matrix. And then, the correlation between the covariance matrix and the full matrix of Ω was derived. We expected a large correlation between the two matrices, but the result was not. It is judged that it is difficult to deduce Ω value from the covariance matrix derived from historical data. Therefore, we obtain the range of parameter change for each asset with the first year data received from the Principal and reflect it in the following robust optimization. Let set $U := \{\tilde{\alpha} \mid \tilde{\alpha}_i = \hat{\alpha}_i + \bar{\alpha}_i\gamma_i, \quad -1 \leq \gamma_i \leq 1, \quad \sum_i |\gamma_i| = \Gamma\}$. We may a simplified formulation because the worst-case only occurs when $\tilde{d}_i = \hat{d}_i - \bar{\alpha}_i, \forall i \in N$(i.e$\gamma_i =$

-1). Let set $U := \{\tilde{\alpha} \mid \tilde{\alpha}_i = \hat{\alpha}_i + \bar{\alpha}_i\gamma_i, \quad 0 \leq \gamma_i \leq 1, \quad \sum_i \gamma_i = \Gamma\}$. For a given α, the objective function considering robustness can be expressed as follows :

$$\begin{aligned}
\text{(Robust)} \quad & \min_{\tilde{\alpha}\in U} d^T\tilde{\alpha} = \min \sum_i d^T(\hat{\alpha}_i - \bar{\alpha}_i\gamma_i) \\
& \sum_i \gamma_i \leq \bar{\alpha}_i d_i, \quad \forall i \in N \\
& 0 \leq \gamma_i \leq 1, \quad \forall i \in N
\end{aligned}$$

However, Thus,

$$\begin{aligned}
\text{(Robust-Dual)} \Rightarrow \quad \max \quad & \Gamma\pi + \sum_i \theta_i \\
\text{s.t} \quad & \pi + \theta_i \leq \bar{\alpha} d_i, \quad \forall i \in N \\
& \pi \geq 0 \\
& \theta_i \geq 0, \quad \forall i \in N
\end{aligned}$$

$$\begin{aligned}
\text{(ALL)} \quad \min \quad & d^T\Omega d - \lambda(\Gamma\pi + \sum_i \theta_i) \\
\text{s.t} \quad & (1) - (11) \\
& \pi + \theta_i \leq \bar{\alpha} d_i, \quad \forall i \in N \\
& \pi \geq 0 \\
& \theta_i \geq 0, \quad \forall i \in N
\end{aligned}$$

References

Please follow guidelines in the *Chicago Manual of Style,* 16th Edition. Here are examples:

- Journal article: Flynn J, Gartska SK (1990) A dynamic inventory model with periodic auditing. *Oper. Res.* 38(6):1089–1103.
- Book: Makridakis S, Wheelwright SC, McGee VE (1983) *Forecasting: Methods and Applications*, 2nd ed. (John Wiley & Sons, New York).
- Edited Book: Martello S, Toth P (1979) The 0-1 knapsack problem. Christofides N, Mingozzi A, Sandi C, eds. *Combinatorial Optimization* (John Wiley & Sons, New York), 237–279.
- Online reference, fictional example: American Mathematical Institute (2005) Better predictors of geospatial variability. Retrieved June 14, 2005, www.mathematicsinstitute.