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 are required to be reformulated as follows:

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

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, 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, after solving the problem without reflecting constraints (10) and (11), we tried to identify the following observations.

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

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

According to observation2, Tracking Error should be raised. At this time, $d^T \Omega d$ can be expressed as $w^T \Omega w - w_{\text{bench}}^T \Omega w_{\text{bench}}$, which means that the difference between the benchmark weight and the portfolio weight is too small that Tracking Error does not satisfy the lower bound. In other words, the difference between the two weights must be increased to some extent to satisfy the constraints. The bisectional search algorithm reflects that if some portfolio weights are increased, the othe portfolio weights of the assets become smaller because of the constraints(4). Since the following $w_{\text{bench}}^T \Omega w_{\text{bench}}$ is fixed, it is required to satisfy the constraints by changing $w^T \Omega w$. Because the weight value of all of the unselected assets from approximately 500 candidates is zero, the weight value is less than the benchmark weight

value ($\because w_{bench} > 0$). On the other hand, the most of weight value of the 50 to 70 selected assets may be larger than the benchmark weights. Thus, we took a look for the average of portfolio weight and have noticed that about a third is smaller than average and two-thirds 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 larger than the average. Conversely, the assets whose portfolio weights are smaller than the average are decreased. As a result, the Tracking Error value becomes large, making the solution feasible.

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 greater 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 $\underline{Z}$ 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

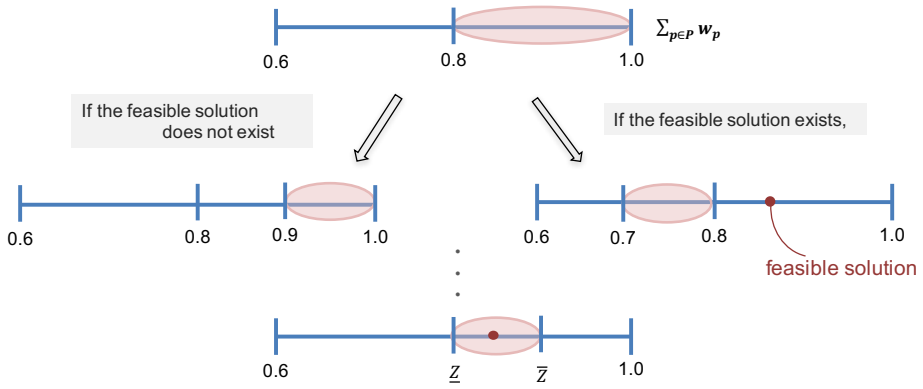

Figure 1: An Example of Bisectional Search

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.

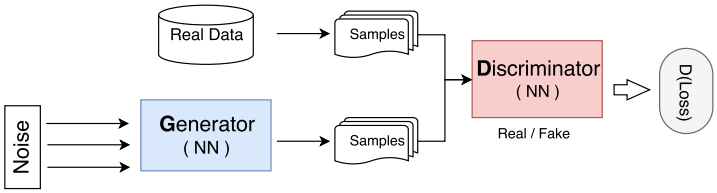

Figure 2: The strucutre of basic GAN

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 in G checks both the feasibility of the solution and the quality of objective value. 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 :

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

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

Figure 3: The strucutre of GAN for portfolio optimization

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.

(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.

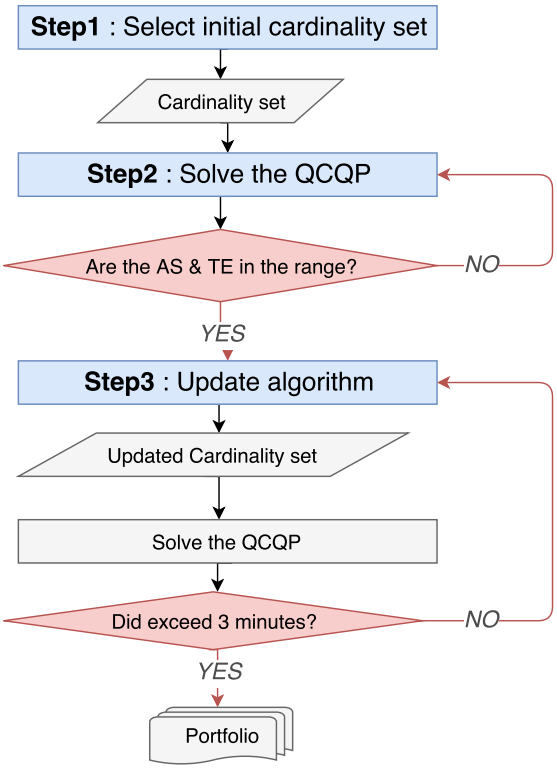

Figure 4: A Flowchart of Hybrid Approach

– Experiment Environment
  * Number of Experiments : 50 times
  * The B&B Timelimit of MIP and the Training Time for GAN : 60 seconds
  * Solver of MIP : Cplex
  * Deep Learning Library : Pytorch
  * The range of assets selected : $60 \sim 70$
  * Data of the first period (2007/01/03)

– Experiment Results :

  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 :

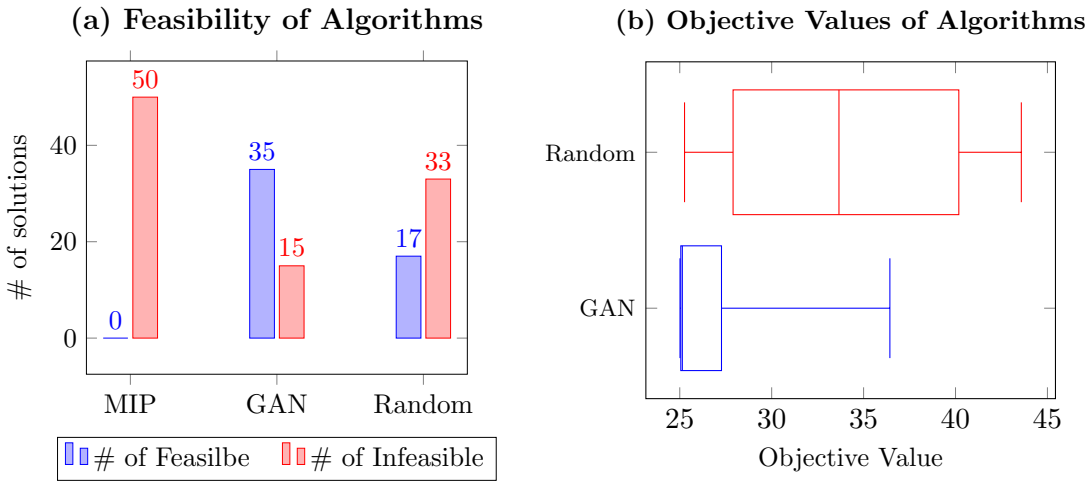

Figure 5: Results of Algorithms

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.

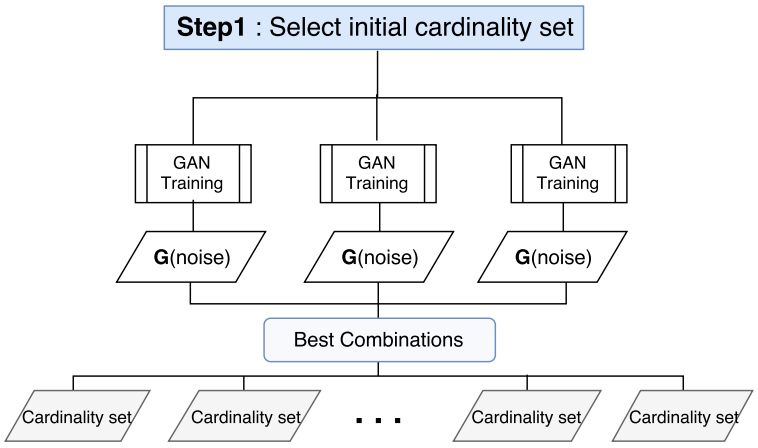

Figure 6: A Flowchart of Hybrid Approach

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, and we figure out the server can run up to three GANs simultaneously on 8 cores. Therefore, we use the parallel process to be 3 GANs training at the same time and then obtain several initial cardinality sets 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 3 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. At this time, the local optimal solution may be improved depending on which asset is selected. Therefore, based on 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. Basically, the update algorithm finds one or multiple assets whose objective value is lowered when one or multiple assets is/are added or removed. Also, if the result of current cardinality set is infeasible, it also adjusts the cardinality set so that it becomes feasible.

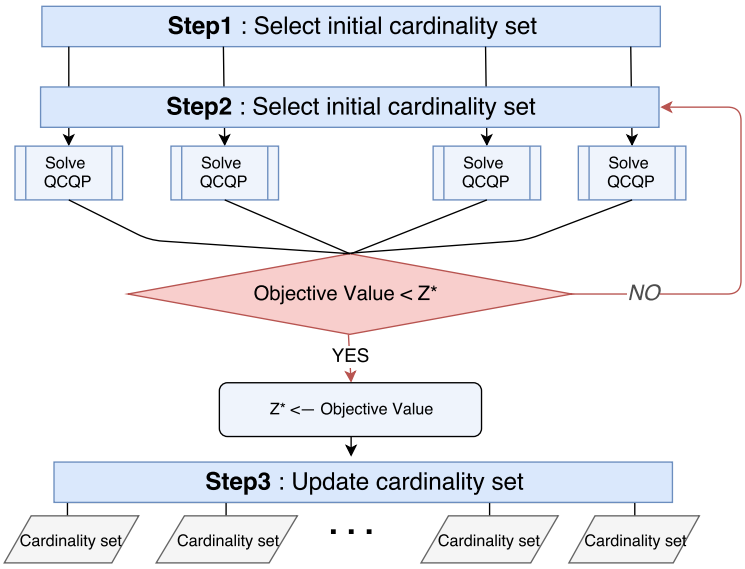

Figure 7: A Flowchart of Hybrid Approach

Analytics Solution and Results

- This Entry Form: Present your solution and results in this section, including completion of the Portfolio Performance Statistics chart below. You may supplement your analysis with additional charts, diagrams and/or other visualization; these supplements must be incorporated into this section of the Entry From.

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

Portfolio Performance Statistics

- Results Template: Populate and submit your numerical results using the Results Template. This template is provided as a separate file on the Competition download site. You can use either the Excel or CSV version.

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.