Taylor & Francis
Taylor & Francis Group

The Engineering Economist
A Journal Devoted to the Problems of Capital Investment

ISSN: 0013-791X (Print) 1547-2701 (Online) Journal homepage: https://www.tandfonline.com/loi/utee20

An extension to the classical mean–variance portfolio optimization model

Çelen N. Ötken, Z. Batuhan Organ, E. Ceren Yıldırım, Mustafa Çamlıca, Volkan S. Cantürk, Ekrem Duman, Z. Melis Teksan & Enis Kayış

To cite this article: Çelen N. Ötken, Z. Batuhan Organ, E. Ceren Yıldırım, Mustafa Çamlıca, Volkan S. Cantürk, Ekrem Duman, Z. Melis Teksan & Enis Kayış (2019): An extension to the classical mean–variance portfolio optimization model, The Engineering Economist, DOI: 10.1080/0013791X.2019.1636440

To link to this article: https://doi.org/10.1080/0013791X.2019.1636440

Published online: 22 Jul 2019.

THE ENGINEERING ECONOMIST
https://doi.org/10.1080/0013791X.2019.1636440

TECHNICAL NOTE

An extension to the classical mean–variance portfolio optimization model

Çelen N. Ötken, Z. Batuhan Organ, E. Ceren Yıldırım, Mustafa Çamlıca, Volkan S. Cantürk, Ekrem Duman, Z. Melis Teksan, and Enis Kayış

Department of Industrial Engineering, Ozyegin University, Istanbul, Turkey

ABSTRACT

The purpose of this study is to find a portfolio that maximizes the risk-adjusted returns subject to constraints frequently faced during portfolio management by extending the classical Markowitz mean–-variance portfolio optimization model. We propose a new two-step heuristic approach, GRASP & SOLVER, that evaluates the desirability of an asset by combining several properties about it into a single parameter. Using a real-life data set, we conduct a simulation study to compare our solution to a benchmark (S&P 500 index). We find that our method generates solutions satisfying nearly all of the constraints within reasonable computational time (under an hour), at the expense of a 13% reduction in the annual return of the portfolio, highlighting the effect of introducing these practice-based constraints.

Introduction

Portfolio management is the science of selecting the appropriate investment opportunities that meet certain criteria. One of the well-known papers in the portfolio management field is Markowitz (1952), which aims to balance the trade-off between the risk and the expected return of a portfolio. There are only two constraints in the Markowitz mean–variance model: (1) the available capital has to be fully invested and (2) only long positions are allowed. In our specification of the problem there are three additional sets of constraints: (1) cardinality, (2) active share, and (3) tracking error, on top of the Markowitz model and other diversification constraints regarding the portfolio. These changes result in a mixed-integer nonlinear program, which is a hard problem to solve. Our main contribution is to solve this problem that includes the cardinality, sector capitalization, tracking error, and active share constraints using a two-step heuristic approach. In the first step, we select a candidate portfolio while ignoring the three additional constraints in the extended formulation. In the second step, we adjust the weights of the assets in the candidate portfolio to create a feasible solution. Our method finds a solution with lower realized returns compared to the benchmark, which violates only the tracking error constraint, within a reasonable amount of time. The reason for the lower realized returns with the methodology proposed might be the cardinality and

CONTACT Enis Kayış enis.kayis@ozyegin.edu.tr Department of Industrial Engineering, Ozyegin University, Nişantepe Mh., Orman Sk. 34794 Çekmeköy, Istanbul, Turkey.

time constraints posed, the nature of the alpha values given as input, and the randomization-based nature of some steps in the proposed solution methodology. We tested our solution methodology in a real-life data set provided by Principal and based on the stocks in the S&P 500 index between 2007 and 2016.

This article is organized as follows. The following section formulates the problem at hand. Our solution method, GRASP & SOLVER, is introduced next. The computational results are discussed the following section. The final section provides conclusions and future work.

Model

Our objective is to create a portfolio from available assets ($i \in \{1, 2, \ldots, n\}$) in order to balance the trade-off between risk and return. Mathematically, a portfolio is determined by a proportion, w_i, of the portfolio held in each asset i. In our case, we have a benchmark portfolio and are interested in the difference between our solution and the benchmark. For each asset w_i, we introduce $d_i = w_i - w_{bench,i}$, which is defined as the active weight, where $w_{bench,i}$ denotes the benchmark weight of asset i. The covariance matrix, Ω, gives the risk relation between each asset, and α is the vector of expected returns of each asset.

The mathematical model formulated to solve our problem is as follows:

$$\min\left(d^T \Omega d - \lambda d^T \alpha\right) \tag{1}$$

s.t.

$$w_i \geq 0, \quad \forall i \in \{1, 2, \ldots, n\} \tag{2}$$

$$\sum_{i=1}^{n} w_i = 1 \tag{3}$$

$$-0.01D \leq d_i \leq 0.01D, \quad \forall i \in \{1, 2, \ldots, n\} \tag{4}$$

$$-0.01S \leq \sum_{i \in Sector_j} d_i \leq 0.01S, \quad \forall j \in \{1, 2, \ldots, J\} \tag{5}$$

$$-0.01M \leq \sum_{i \in MCAPQ_k} d_i \leq 0.01M, \quad \forall k \in \{1, 2, \ldots, 5\} \tag{6}$$

$$-0.01B \leq \sum_{i=1}^{n} \beta_i d_i \leq 0.01B \tag{7}$$

$$50 \leq card(w_i \neq 0) \leq 70 \tag{8}$$

$${}_{AS} \leq 1 - \sum_{i=1}^{n} min\left(w_i, w_{bench,i}\right) \leq 1 \tag{9}$$

$${}_{TE} \leq \sqrt{d^T \Omega d} \leq \overline{TE}. \tag{10}$$

The objective function has two distinct parts; the first part is the risk of the portfolio and the second part is expected return. In the objective function formulation, λ denotes a tuning parameter that can vary from 0 to 10, and it can be used to assess different scenarios by modeling different weights for the two distinct parts of the objective function. A λ value of 0 is excessively focused on minimizing the risk, whereas a λ value closer to 10 is more focused on maximizing returns. The boundaries of λ are presented to us by the competition committee. Trying other values outside of these boundaries was outside the scope of the competition.

Constraint (2) explains that the weight given to an asset cannot be negative. Constraint (3) ensures that the portfolio must be fully invested. Constraint (4) limits the dispersion of the selected weights from the benchmark to at most D. Constraint (5) improves the diversification of the portfolio by concentrating on different assets from different sectors. Each sector is represented by the index j, and Sector_j denotes the list of assets belonging to sector j. Total sector active weight is expected within a range of $\pm S\%$. Constraint (6) further improves the diversification of the portfolio by considering market capitalization (company sizes). Companies are distributed into five subsets according to their sizes (referred to as Market Capitalization Quantile, MCAPQ_k). A value of $k=1$ denotes the largest 20% of companies and $k=5$ denotes the smallest 20% of companies in terms of benchmark weights assessed in the S&P 500 market. Constraint (7) controls the sensitivity level of the portfolio against the market volatility, which is assessed with the β_i parameter and must lie between $\pm B$. Constraint (8) limits the number of distinct assets between 50 and 70. Constraint (9) is about controlling how the portfolio is following a chosen benchmark index in terms of active weights. Active and passive management strategies are defined by this constraint. Constraint (10) is the tracking error constraint and controls how the portfolio follows a chosen benchmark in terms of a risk measure. We use the standard deviation of active weights as our risk measure.

Solution methodology

Our extension to the classical Markowitz mean–variance portfolio optimization problem is considered as a mixed-integer nonlinear programming problem and is known to be NP-hard (Kalayci, Ertenlice, Akyer, & Aygoren, 2017; Mutunge & Haugland, 2018). Exact solution methods for this extension are computationally expensive; thus, finding a solution in a reasonable amount of time (e.g., 3 min) is exceptionally difficult. Another challenge arises due to the nonconvex nature of the problem: we could be stuck in local optima. Hence, we compare different methodologies—(1) the exact solution methods for the original formulation without the extensions and (2) heuristics—and measure the quality of the generated solutions compared to a given benchmark.

Fast GRASP & SOLVER

Our heuristic solution is based on Baykasoglu, Yunusoglu, and Ozsoydan (2015) and Cura (2009). The method studied in these papers hybridizes a heuristic algorithm with an exact solution approach. They propose a new desirability criterion for each asset to include in the portfolio. One advantage of this method is that the new criterion evaluates the risk and return probabilities of one asset in one single parameter. Our solution, GRASP & SOLVER, consists of two steps. In the first step, we select a candidate portfolio while ignoring three additional constraints in the extended formulation. In the second step, we adjust the weights of the assets in the candidate portfolio to create a feasible solution.

The first step consists of three parts. In the first part, we find an exact solution to the original problem with the first six constraints only (excluding active share, tracking error, and cardinality constraints). This solution is used as a starting point because it provides a distribution of weights that satisfies the first six constraints and can be computed quite fast (i.e., less than a second).

In the second part, we create a desirability ranking of all of the assets. We denote a desirability value for each asset with the f_i parameter, following Baykasoglu et al. (2015). This parameter was proposed by Cura (2009), used by Sadigh, Mokhtari, Iranpoor, and Ghomi (2012), and slightly modified in our case by rescaling λ to fall in [0, 1].

$$f_i = \frac{1 + \frac{\lambda}{10}\left(\frac{\sum_{j=1}^{n} \sigma_{ij}}{n}\right)}{1 + \left(1 - \frac{\lambda}{10}\right)\alpha_i} \quad i = 1, 2, \ldots, n.. \tag{11}$$

In the third part, we sort the assets in ascending order according to this parameter to form a desirability ranking. Then we create a restricted assets list (RCL), which consists of the top 250 assets (out of 492 to 505 total number of assets, which varies across periods) in this desirability ranking. The reason why the RCL size is equal to 250 in this solution methodology, which is nearly half of the assets across all periods, is explained in Baykasoglu et al. (2015). In the article, it is stated that smaller sizes of RCL favor positive f_i values, which may cause the solution to be stuck in a local optima, whereas large RCL sets (close to the size of the asset set that includes all of the assets) have little difference from pure random selection of assets, in which creating an RCL set has little effect on the solution. We select one of the assets from RCL randomly and put it in a desired asset list (D). Once an asset is selected, the desirability parameter is chosen and ranking is performed again for the remaining $n-1$ assets. Again, another asset is randomly selected from the top RCL assets. The assets are randomly selected in this manner; which means that all of the assets in the RCL have the same probability of being chosen. In order not to anchor too much on the desirability parameter, which may lead to missing some good solutions, we did not alter weights based on the desirability parameter at this stage. This process continues until K assets are selected into the desired assets list D, which is taken to be 200 for this study. The selection of K assets is to reduce the size of selected assets in the portfolio while keeping the number of assets in the desired asset list D close to the RCL subset (for details, see Baykasoglu et al., 2015). This selection of an asset list also increases the randomization in the process and lowers the probability of being stuck in local optima.

Once we have a candidate portfolio, the second step in our solution is to adjust the weights, given by the starting solution, in order to have a feasible solution. To achieve this, we "dually allocate" the asset weights. The dual weight allocation took place between one randomly selected asset from the desired assets list and another random asset from the universal assets list, which consists of all of the assets evaluated for the considered period. A random amount of weight, U, is exchanged between these two assets, where U is a uniformly distributed random variable between 0 and 0.002. Because the expected value of mean for a uniformly distributed random variable between the interval [0, 1] is 0.5, we used $rand \times 0.002$, which leads to an expected average dual weight allocation of 0.001 between the assets. The random number generator of type Mersenne Twister with seed 0 in Matlab is used in this methodology.

Only the feasible exchanges of weights are performed, according to the prespecified cardinality number C. Until this point, we ignored the cardinality constraint, which force the asset weights in the solution to be either 0 or greater than 0.001. After all of the weight exchanges, the assets with weights less than or equal to 0.001 are reduced to 0, and the total reduced weight (defined as *Excess* in the pseudocode below) is distributed equally (defined as *amount* in the pseudocode below) among the selected assets that satisfy the cardinality limit (assets that have weight greater than 0.001). Note that because the excess amounts of reduced weights are very small, distributing this amount equally among the selected assets almost never violates any of the constraints. The final part of the solution is to check whether the created solution is feasible with respect to all of the constraints. Moreover, if the solution takes more than 6 s, we stop further evaluation of this solution and move on to the next one due to the prespecified time limit. The solution is stored if no constraints are violated.

The time limit given for one period was between 3 and 5 min. In our computations, the procedure took at most 300 s per rebalance date, which was the prespecified upper limit for time per period. In order to fit into the time constraint, we set our solution methodology to run five trials for different cardinality values from 50 to 60 (inclusive) in each period. This resulted in approximately 6 s reserved for each trial. Each feasible solution resulting from these trials was stored. Out of these feasible solutions for the considered period, the one with the best objective value (as given in Equation (1) in the Model section) was selected to be the final solution for the period.

The detailed pseudocode for dual weight allocation is provided below:

```
Solve for objective function (1) to satisfy Constraints (2)–(7). Let this solution be starting asset list S.
C ← Cardinality                              ▷ Count the assets with weights larger than 0.001 as cardinality
RCL ← 250
K ← 200
for Cardinality ← 50, …, 60 do
   for iteration ← 1, …, 5 do
      f_i Creation method to create a desired assets list, D
      while Constraints are not satisfied do
         while Cardinality ≥ C do
            Choose random asset from D and S as d and s, respectively.
            if d > 0.001 then
               if s−rand × 0.002 ≥ 0 then
                  if s−rand × 0.002 ≤ 0.001 AND s > 0.001 then
                     C = C−1
                  end if
                  s = s−rand × 0.002
                  d = d + rand × 0.002
               end if
            end if
         end while
         if C ≥ Cardinality then
            Choose random asset from D and S as d and s, respectively.
            if s−rand × 0.002 ≥ 0 then
               if s−rand × 0.002 ≤ 0.001 AND s > 0.001 then
                  C = C−1
               end if
               s = s−rand × 0.002
               if d + rand × 0.002>0.001 AND d < 0.001 then
                  C = C + 1
```

(*continued*)

```
            end if
            d = d + rand × 0.002
            if Constraints (2)–(10) (except (7)) are satisfied then
              for i ← 1, ..., length(S) do
                if S_i ≤ 0.001 then
                  Excess = Excess + S_i
                  S_i = 0
                end if
              end for
                amount = Excess/C
                for i ← 1, ..., length(S) do
                  if S_i ≠ 0 then
                    S_i = S_i + amount
                    Excess = Excess−amount
                  end if
                end for
                  if Time > 6 then
                    Break
                  end if
              end if
            end if
          end if
          Check for all the constraints
        end while
      end for
end for
```

GRASP & SOLVER with turnover cost

Portfolio turnover is a measure of how frequently the assets are bought or sold in rebalance periods, because buying an asset or selling is not free. Every time a new allocation on assets is made, the portfolio manager needs to pay attention to keep the rebalancing (buying and selling assets) under a certain level, because rebalancing activities result in a turnover (or transaction) cost. This cost has a considerable impact on portfolio return.

In order to incorporate this cost into our solution methodology, first we extended the classical mean–variance model with a turnover limiting constraint for each rebalance period, which was introduced by Fabozzi (see, for example, Fabozzi, Kolm, Pachmanova, & Focardi 2007), as follows:

$$\sum_{i=1}^{N} |w_{i,t} - w_{i,t}^{Pre}| \leq L. \tag{12}$$

In addition to the classical model, we modified our GRASP & SOLVER heuristic as well. After the dual weight allocation step, one of the stored feasible solutions was selected based on a modified objective function that also considers the transaction cost. First, we calculated the adjusted weight of every asset in the previous period, $w_{i,t}^{Pre}$, as follows:

$$w_{i,t}^{Pre} = \frac{w_{i,t-1} * (1 + r_{i,t-1})}{\sum_{i=1}^{n} w_{i,t-1} * (1 + r_{i,t-1})}. \tag{13}$$

This adjustment to $w_{i,t-1}$ is needed because the value of the previous period's portfolio changes between two periods. Given the adjusted weights from the previous period, we formulated the modified objective function as follows:

$$\min\left(d^T\Omega d - \lambda d^T\alpha + 0.005\sum_{i=1}^{n}|w_{i,t} - w_{i,t}^{Pre}|\right). \tag{14}$$

We sum up the absolute changes in asset weights and multiply this summation by 0.005, assuming that every 100% in portfolio turnover will reduce the subsequent overall portfolio return by 0.5%.

At the expense of computation time, we also increased our search space compared to the previous implementation of the GRASP & SOLVER method. For each rebalance date, we searched for 16 (instead of 11) different cardinality values between 50 and 65. For each cardinality value, we searched for a maximum of 20 different feasible solutions while increasing the time spent to search for each solution from 6 to 8 s.

Computational results

Given the problem formulation and the proposed solution method, we conducted a simulation study to assess the performance of our solution using the data provided by Principal. In this section, we provide computational results of our solution methodology and compare our solution against the benchmark using several key performance indicators (KPIs).

What would have happened if Principal had created a portfolio in each rebalance date using our solution methodology? In this section, we present the results of the simulation runs from the three methodologies described in the previous section. The portfolio optimization problem was solved for each of the 131 rebalance periods using each of the three methods. We compare the results of these methodologies using several KPIs. We used the following values for the parameters used in the model: $D = 5, S = 10, M = 10, B = 10, {}_{AS} = 0.6, {}_{TE} = 0.05, \overline{TE} = 0.1$.

For the Fast GRASP & SOLVER method, our solution time for a single period took about 2.97 min on average. Because the dual allocation step involved randomization, solution generation times varied between periods. The lower limit of the tracking error constraint was not satisfied for the last 29 periods. Due to the time limit of 6 s per feasible solution, some of these trials did not give any feasible solution. Even though the lower limit for this constraint was relaxed to 0.04, it took almost twice as long as average for these periods. On average, we had about 15 feasible solutions (out of 55 maximum) for every rebalance date.

For the GRASP & SOLVER with turnover cost method, computations for one period took significantly longer: 42.67 min on average. In order to reduce the risk of not finding any feasible solutions in some periods as discussed above, we lowered the tracking error to 0.04. This enabled us to get solutions in a shorter amount of time in the dual weight allocation trials and resulted in a smoother simulation run. Figure 1 shows that with the relaxation of the lower limit of the tracking error, the average tracking error value over 131 periods becomes 0.0415.

Finally, to compute the classical mean–variance model with the turnover limiting constraint, we took the value of L equal to 0.9. We should note, however, that the average turnover value over 131 periods after the simulation run was around 0.079. We could not run the code with lower L values due to prohibitively excessive computation times beyond 24 h.

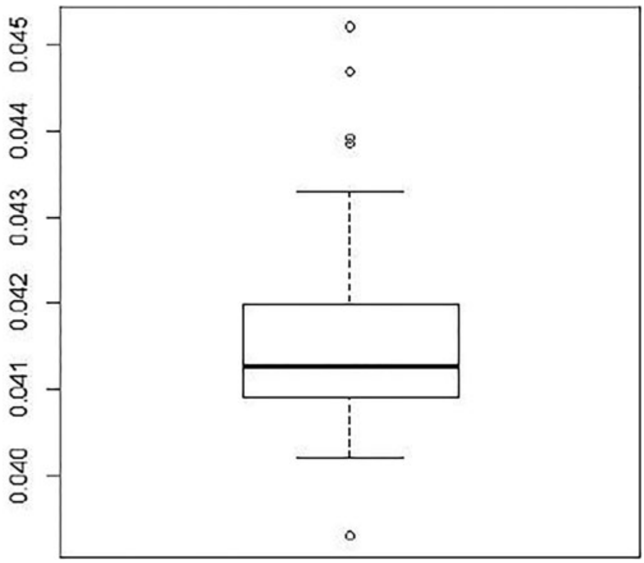

Figure 1. Box plot for tracking error values, GRASP & SOLVER with turnover cost.

Table 1. KPI values for the three alternative methods along with the benchmark.

	Benchmark (S&P 500)	Fast GRASP & SOLVER	GRASP & SOLVER with turnover cost	Classical mean–variance model
10-Year cumulative return	158.99%	21.11%	40.66%	324.03%
Annual return	9.90%	1.92%	3.44%	15.41%
Excess annual return	—	−14.60%	−13.07%	5.51%
Annualized tracking error	—	4.00%	3.18%	7.03%
Sharpe ratio	35.24	4.50	9.04	66.94
Information ratio	—	−311.28	−370.13	84.63

The resulting KPIs across all three methods as well as the benchmark are given in Table 1. For the benchmark KPIs, we used the values of the S&P 500, provided by Principal, and calculated the index returns based on these values. Though the S&P 500 index is not a feasible solution to our extended problem, it represents the performance of a benchmark portfolio that ignores the additional three constraints of our problem. Specifically, the benchmark portfolio almost always violates the cardinality and tracking error constraints: All of the assets in the benchmark in a particular period (on average 500 assets) have weights larger than 0 and thus the number of assets in the benchmark greatly exceeds the cardinality upper limit. Trivially, the tracking error will be zero for the benchmark portfolio violating the lower limit of 0.05. Finally, the benchmark portfolio also violates the active share constraint (i.e., Constraint (9)), because taking $w_i = w_{bench,i}$ results in a value of active share equal to zero, which is much smaller than the lower limit of 0.6.

For both versions of the GRASP & SOLVER method, the cumulative return KPI indicates that the generated portfolio has gained value over time and the excess annualized return KPI shows that selected portfolios are performing lower than the benchmark. However, the GRASP & SOLVER method with turnover cost performs better than the

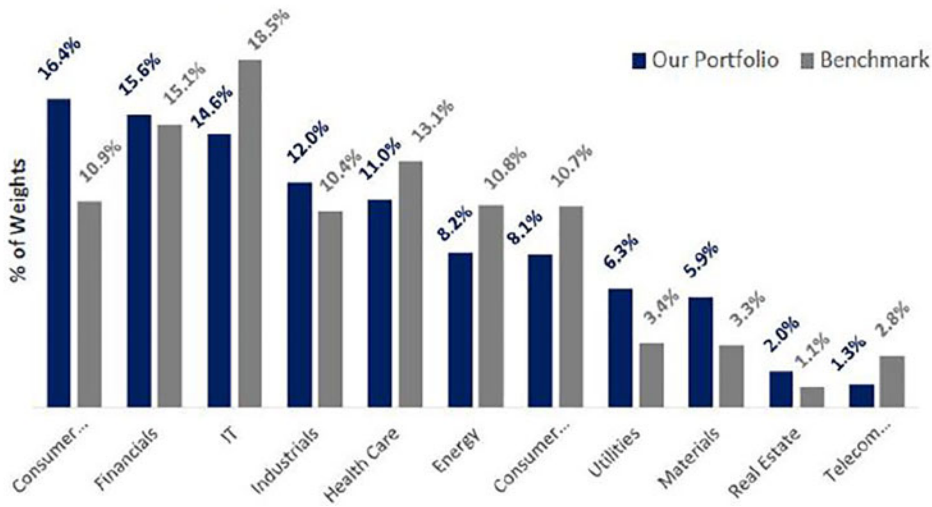

Figure 2. Average percentage of weights allocated per sector between fast GRASP & SOLVER method and the benchmark.

fast version. The improvement over the Sharpe ratio KPI is significant as well, an increase from 4.50 to 9.04, which means that the portfolio construction is made with a much smarter investment, in terms of risk-adjusted returns. Unfortunately, the information ratio decreased from −311.28 to −370.13. The reason behind this decrease is the lowered tracking error, because the standard error of the evaluated portfolio is in the denominator of the information ratio.

Although our solutions perform worse than the benchmark in terms of annual return, both of our portfolios satisfy the additional three constraints (with a slight relaxation of the lower limit for the tracking error constraint in some instances) and generate solutions in a comparatively short amount of time for all of the rebalance dates. These results indicate that the addition of the transaction cost calculation in the evaluation of feasible solutions, as well as increased search space, improves the results while keeping the computation time below 45 min per period. Adjustments in the cardinality, trial numbers, trial duration, and tracking error lower limit change could improve the performance of our method even more.

Our method generates portfolios as intended: They closely follow the benchmark's investments in assets from the same sector and MCAP group. Figures 2 and 3 show that the general trend of the fast GRASP & SOLVER portfolio closely follows the benchmark trend in terms of the percentage of the portfolio allocated for each sector (MCAPQ group). Our portfolio includes more assets from consumer discretionary, industrial, and financial sectors and fewer assets from IT, health, energy, and consumer staples sectors.

Finally, note that the classical mean–variance model significantly outperforms the benchmark model in all KPIs. However, it does not create a feasible portfolio for our model, because it does not consider the additional three constraints. This shows that additional constraints in the extended formulation significantly narrow down the

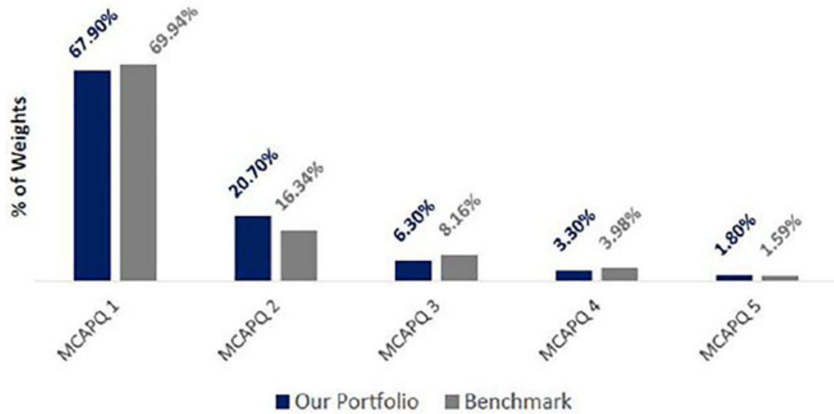

Figure 3. Average percentage of weights allocated per MCAPQ between fast GRASP & SOLVER method and the benchmark.

feasible set of portfolios; thus, a significant decrease in the annual return is not surprising. Instead, our method creates a portfolio under specific practice-based constraints with limited reduction in annual return that is computed within a reasonable amount of time.

Conclusion

In this article, we develop a heuristic-based methodology, called GRASP & SOLVER, to solve the extended Markowitz model with three additional constraints: upper and lower limits on the cardinality, active share, and tracking error. The classical Markowitz model can be solved effectively via quadratic programming. However our extended version is nonconvex; thus, exact solutions cannot be generated for real-life instances in a reasonable amount of time. Hence, we designed a heuristic solution, GRASP & SOLVER. We tested our method on a real-life data set between 2007 and 2016 and compared the results to the S&P index as a benchmark. This solution yields annual returns comparably lower than the benchmark and does not satisfy the tracking error constraint in several periods, between 103 and 131. The results show that the proposed algorithms follow the benchmark trend while satisfying all of the practice-based constraints, excluding the tracking error constraint.

The fast GRASP & SOLVER method ignores turnover costs, thus yielding comparatively lower adjusted returns. However, the GRASP & SOLVER with turnover cost method includes the turnover (transaction) cost in determining the best portfolio among a set of candidate feasible solutions. This modification yields higher annual returns than the fast version of this method.

We believe that there may be several reasons for the comparatively lower returns of our solution: the cardinality constraint, alpha values, and the nature of the GRASP & SOLVER methodology applied in this article. According to Fabozzi et al. (2007), portfolios that are evaluated in the literature have cardinality values around 40, which led us to think that future research should investigate the impact of the cardinality constraint on the performance of our methodology. Other possible reasons stem from our implementation of the GRASP & SOLVER methodology: For example, in the dual weight allocation step of the methodology, pure randomization may lead to suboptimal exchanges. Instead, the process could leverage the f_i values. Finally, our implementation requires many computations and thus requires significant computational resources. A smarter search of the solution space could increase the efficiency of our methodology.

Another critical avenue of research relates to the parameters, especially alpha values, used in this study. Though both the returns and correlation among assets are assumed to be deterministic, it is known that these parameters cannot be estimated with 100% accuracy. Hence, stochastic optimization methods, such as sample-based average approximation, could be employed to take uncertainty in these estimates into consideration.

Notes on contributors

Çelen N. Ötken received a B.Sc. in industrial engineering from Ozyegin University in 2019. Currently, she is working as a data science project specialist at Ozyegin University.

Zeynel Batuhan Organ received a B.Sc. in industrial engineering with a minor in computer science from Ozyegin University in 2018. Since graduation, he has been working as a data scientist in Invent Analytics.

Elif Ceren Yıldırım received a B.Sc. in industrial engineering from Ozyegin University in 2018. Currently, she is working as a business intelligence lead in Vodafone Turkey.

Mustafa Çamlıca graduated from Ozyegin University in 2018 with a B.Sc. in industrial engineering. Currently, he is a master's student at Istanbul Technical University in the Department of Industrial Engineering and is working as a data analytics assistant specialist in Borusan Makina–Caterpillar.

Volkan S. Cantürk graduated from Ozyegin University in 2019 with a B.Sc. in industrial engineering and a minor in hotel management. He is currently working as an operator marketing specialist at Unilever Food Solutions.

Ekrem Duman obtained his B.Sc. in electrical and electronics engineering from Bogazici University and M.Sc. and Ph.D. in industrial engineering from the same university. Currently he works as a professor in the Department of Industrial Engineering at Ozyegin University. In addition to theoretical studies, he is very interested in applications and has been involved in many industrial (mostly banking) projects as a coordinator or consultant.

Zehra Melis Teksan received her B.Sc. and M.Sc. in industrial engineering from Bogazici University and her Ph.D. in industrial and systems engineering from the University of Florida. Currently she works as an assistant professor in the Department of Industrial Engineering at Ozyegin University. Her main research focus lies in the field of production planning and inventory theory.

Enis Kayış received B.Sc. degrees in mathematics and industrial engineering from Bogazici University, an M.Sc. in statistics, and a Ph.D. in management science and engineering from Stanford University. Currently he works as an assistant professor in the Department of Industrial Engineering at Ozyegin University. His current research interests include data-driven decision making and business analytics with applications in supply chain management and health care operations.

References

Baykasoglu, A., Yunusoglu, M.G. and Ozsoydan, F.B. (2015) A grasp based solution approach to solve cardinality constrained portfolio optimization problems. *Computers & Industrial Engineering*, 90, 339–351.

Cura, T. (2009) Particle swarm optimization approach to portfolio optimization. *Nonlinear Analysis: Real World Applications*, 10, 2396–2406.

Fabozzi, F.J., Kolm, P.N., Pachmanova, D.A. and Focardi, S.M. (2007) *Robust portfolio optimization and management*. Hoboken, New Jersey: John Wiley & Sons.

Kalayci, C.B., Ertenlice, O., Akyer, H. and Aygoren, H. (2017) An artificial bee colony algorithm with feasibility enforcement and infeasibility toleration procedures for cardinality constrained portfolio optimization. *Expert Systems with Applications*, 85, 61–75.

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

Mutunge, P. and Haugland, D. (2018) Minimizing the tracking error of cardinality constrained portfolios. *Computers and Operations Research*, 90, 33–41.

Sadigh, A.N., Mokhtari, H., Iranpoor, M. and Ghomi, S.M.T.F. (2012) Cardinality constrained portfolio optimization using a hybrid approach based on particle swarm optimization and Hopfield neural network. *Advanced Science Letters*, 17, 11–20.