Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Factor Model Based Clustering Approach for Cardinality Constrained Portfolio Selection [⋆]

Kening Jiang [*] Duan Li [**] Jianjun Gao [***] Jeffrey Xu YU [****]

[*] *Valuation Advisory Services, Deloitte & Touche Financial Advisory Services Limited, Hong Kong (e-mail: victoria957@gmail.com)*
[**] *Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong (e-mail: dli@se.cuhk.edu.hk)*
[***] *Department of Automation, Shanghai Jiao Tong University, Shanghai, China. (jianjun.gao@sjtu.edu.cn)*
[****] *Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong (e-mail: yu@se.cuhk.edu.hk)*

Abstract: Portfolio selection concerns identifying an optimal composition of various risky assets and their corresponding holding amounts such that the corresponding investment strategy strikes a balance between maximizing the expected investment return and minimizing investment risk. While market frictions make full diversification impractical, cardinality constrained mean-variance (CCMV) portfolio selection problem emerges as a natural remedy: Given an asset pool with total n assets and a given cardinality $s < n$, optimally choose s assets from the entire asset pool such as to achieve a mean-variance efficiency. Unfortunately, CCMV has been proved to be NP hard and has been posted in front of optimization society as a long-standing challenge. By invoking structural market information and utilizing fast clustering algorithm for classification, we develop in this paper an effective heuristic scheme to identify approximate solutions for large-scale CCMV problems. More specifically, by constructing grouping constraints generated from factor-model based clustering algorithm and attaching them to the mixed integer programming formulation associated with the CCMV problem, we are able to significantly reduce the computational complexity, thus offering a fast algorithm with relatively high quality solution.

Keywords: Financial optimization, portfolio selection, mean-variance formulation, clustering algorithm, factor model, mixed integer programming

1. INTRODUCTION

Modern portfolio selection starts with the ground breaking work of mean-variance formulation proposed by Markowitz (1952), which provides a quantitative framework as an analytical engine to achieve good investment performance. Most fundamentally, the mean-variance formulation captures an essential multi-objective nature between the two conflicting goals in portfolio management, maximizing the investment return and minimizing the investment risk, and offers a systematic approach to strike a balance between them. As a natural generalization of the mean-variance analysis, the framework of mean-risk trade-off analysis has become a standard in portfolio management, in accompanying with emergence of numerous alternative risk measures.

One of the most important guiding principles derived from the mean-variance framework is that investors should always diversify their investment, thus diversifying investment risk. If we faithfully follow Markowitz's doctrine, investors who primely care about the mean and variance of the portfolio return should construct their optimal portfolios from among *all* risky assets available in the market. Such an ideal solution in a frictionless world, however, becomes unrealistic in most real-life applications. Due to various forms of market frictions, such as management fees and transaction costs, almost all investors only invest in a limited number of risky assets. Such a significant gap between the ideal and real markets motivates the research community to investigate in the last 20 years the cardinality constrained mean-variance (CCMV) portfolio selection problem. More specifically, CCMV portfolio selection problem can be stated as follows: Given an asset pool with total n assets and a given cardinality $s < n$, optimally choose s assets from the asset pool such as to achieve a mean-variance efficiency. It is evident that the challenge is how to identify a small number of risky assets

[⋆] This work was partially supported by Research Grants Council of Hong Kong under grants 414610 and 414513 and by Natural Science Foundation of China under grant 71201102. The second author is also grateful to the support from the Patrick Huen Wing Ming Professorship of Systems Engineering & Engineering Management.

to achieve a performance as close as possible to the market portfolio.

Unfortunately, the cardinality constrained mean-variance portfolio selection problem has been known to be NP-hard (Gao and Li (2013)). The literature in tackling CCMV can be classified into two categories, exact and heuristic algorithms. Although adopting different relaxation schemes, all exact algorithms invoke branch-and-bound algorithms to attain an optimality. Bienstock (1996) proposes to use a surrogate constraint to replace the cardinality constraint in the relaxation phase of the algorithm. Bertsimas and Shioda (2009) develop an exact solution approach by using a convex relaxation resulted from ignoring the cardinality constraint. Such a relaxation leads to a formulation ready for Lemke's pivoting method. Li et al. (2006) consider in their study several discrete features in real trading, including the cardinality constraint. Utilizing the monotonicity of the zero norm corresponding to the cardinality constraint and other geometric characteristics of the quadratic objective contour, they develop some cutting schemes to reduce the feasible region successively in a branch-and-bound solution process. Different from the majority of the existing literature which has primarily focused on some direct relaxations of the cardinality constraint, Gao and Li (2013) recently consider modifying the objective function to some separable relaxations, which are immune to the hard cardinality constraint. Keeping the cardinality constraint preserves certain inherent features of the primal problem, thus enabling a combination of an analytical solvability of the cardinality constrained separable relaxations and their corresponding polynomial-time dual search algorithms. Gao and Li (2013) report that their BnB algorithm outperforms the CPLEX solver significantly for problems (without side constraints on portfolio) with a relatively large dimension (up to dimension 300).

In the category of heuristic algorithms, Blog et al. (1983) propose a dynamic programming heuristic which is applicable only to small-scale portfolio selection problems. Chang et al. (2000) consider heuristic methods for the mean-variance portfolio selection problem with a cardinality constraint and integer constraints. Crama and Schyns (2003) use simulated annealing to find the solution to complex portfolio selection models with side constraints, including cardinality constraint. Shaw et al. (2008) use the Lagrangian relaxation to construct some lower bounds for such problems. Adopting a different solution concept, Xie et al. (2008) adopt a randomized algorithm to identify good feasible solutions to CCMV problem with some quality guarantee. Recently, Mokhtar et al. (2014) give a comprehensive review of the algorithms in solving the CCMV problem.

Examining the existing literature on cardinality constrained portfolio selection, it is surprising that none of the models utilizes structural information from the market in the selection process to help choose candidate assets into the portfolio. We believe that incorporating financial analysis with a formal optimization formulation will reduce computational burden, empower the analytical ability of the algorithm and produce more meaningful outcomes. To achieve this overall goal, we will i) characterize different risky assets by factor models using market data, ii) cluster similar risky assets into groups in accordance with their loading coefficients in the factor model, and iii) select representative(s) from individual groups to form a portfolio which satisfies the cardinality constraint and achieves a mean-variance efficiency. Different from Gao and Li (2013) which is an exact algorithm, we propose in this paper a heuristic algorithm. However, the heuristics adopted in this paper are scientific-based by invoking and integrating factor models in finance, clustering analysis in computer science and mixed integer programming models in operations research, thus powerful and effective in computing a fast solution to CCMV.

We consider in this paper the following cardinality-constrained mean-variance (CCMV) portfolio selection problem:

$$\begin{aligned} (P)\quad \min_{x} \quad & \sigma^2 = x'Qx \\ \text{Subject to}: \quad & \mathbf{1}'x = 1, \\ & r'x \geq \bar{r}, \\ & \sum_{i=1}^{n} \delta(x_i) = s, \\ & x \geq 0, \end{aligned}$$

where $r = (r_1, \cdots, r_n)'$ is the expected return vector of the n risky securities, $Q = \{Q_{i,j}\}|_{i,j=1}^{n} \in \mathbb{S}_{+}^{n}$ is the covariance matrix of the n risky securities which is positive definite, $x = (x_1, \cdots, x_n)'$ is the vector of portfolio weights, $\bar{r}$ is the preassigned return level which the investor would like to attain, $\mathbf{1}$ is the n-dimensional all-one vector, and $\delta(\cdot)$ is the indicator function, i.e., $\delta(a) = 1$ if $a \neq 0$ and $\delta(a) = 0$, otherwise. In the above CCMV portfolio selection problem, i) the objective $x'Qx$ is the variance of the portfolio return which measures the investment risk under decision x, and ii) the constraint $x \geq 0$ implies that short selling is not allowed. Solving problem (P) with $\bar{r}$ varying from its minimum level (the return level corresponding to the minimum variance policy) to infinity yields the efficient frontier in the mean-variance plane under the cardinality constraint.

By introducing a sufficiently large positive number M and an n-dimensional $0-1$ vector, problem (P) can be reformulated as the following quadratic mixed integer programming problem,

$$\begin{aligned} (\bar{P})\quad \min_{x} \quad & \sigma^2 = x'Qx \\ \text{Subject to}: \quad & \mathbf{1}'x = 1, \\ & r'x \geq \bar{r}, \\ & 0 \leq x_i \leq z_i M, \quad i = 1, \cdots, n. \\ & \sum_{i=1}^{n} z_i = s, \\ & z = \{0,1\}^n, \\ & x \geq 0. \end{aligned}$$

Although problem $(\bar{P})$ can be solved by some commercial softwares, such as CPLEX, and has been widely investigated in the literature, the computational efficiency is still not up to a satisfaction, especially for large scale problems. In the financial market, it is important to observe the principle of timeliness. It is the goal of this paper to develop a novel fast algorithm with a relatively high accuracy.

A salient feature of our research is an integration of the factor modeling scheme in finance with clustering analysis algorithm in computer science. Such a combination endows us a capability in reducing significantly the search complexity in CCMV. According to the arbitrage pricing theory (Ross (1976)), market risk can be classified into systematic and nonsystematic risks. It is well known that the nonsystematic risk can be eliminated by diversification, while the systematic risk cannot. The systematic risk is contributed by a set of random factors which can be viewed as the driving force of market movement. The price movement of each individual is then primarily dictated by the movements of the factors according to its factor loading coefficients. Classifying different securities into different groups according to their factor loading coefficients can be achieved using a data mining technique called clustering. The primary purpose of clustering analysis is to generate a meaningful partition of a set of variables into groups, according to their characteristics, such that the intra-cluster similarity is maximized and the inter-cluster similarity is minimized (See, e.g., Chen et al. (1996)). Cluster analysis has been well developed and widely applied for data processing in computer science and other fields as well. For example, Craighead and Klemesrud (2007) develop a clustering algorithm to select initial stocks in their portfolio according to the time series data of the stocks. Tola et al. (2008) adopt a correlation based clustering algorithm iteratively to build up a stable filtered correlation matrix. In our study, following establishment of a factor model, securities will be grouped by a clustering algorithm according to their factor loading coefficients. Intuitively, securities in the same group will perform similarly in a market driven by these identified factors. Then, constructing a portfolio by representative securities from individual groups would have a capability to mimic the performance of the entire market. It is obvious that, in our analysis, the selection of a factor model plays a key role in determining the features of securities according to different factors, which is a basis for clustering.

The paper is organized as follows. After presenting the problem formulations in this section and motivating our research, we will present our solution approach in the next section with detailed description of each of its components. In Section 3, we report the numerical results from our experiments and provide some explanations of the outcomes. We conclude the paper in Section 4.

2. NOVEL SOLUTION APPROACH

Our solution principle in dealing with CCMV is simple yet very logical. We first extract prominent features of all risky assets from market data. If two risky assets possess similar features, we tend to believe that they would perform similarly in the market. Next, we can partition all the risky assets into s groups according to their degree of similarity. Finally, our strategy becomes evident: We select one representative from each group to join the portfolio. Realizing the above solution scheme requires clear answers to the following questions: i) What will be representative features with which the performance of an individual risky asset in the market can be clearly characterized? ii) How to measure the closeness of different risky assets such that the smaller the measure difference, the more similar market performance between the two? iii) How to partition the entire pool of risky assets into s groups according to their closeness? and iv) How to identify a representative from each group to form a portfolio such that a mean-variance efficiency is achieved under the cardinality constraint?

2.1 Step 1: Characterizing risky assets by factor model

We discriminate humans by their heights, weights, hair color, or all of them. In our situation, we rely on factor models (Luenberger (1998)), which have been well established in finance, to characterize risky assets. Under the arbitrage pricing theory (APT), the market movement is driven by a set of factors. As the return of a given risky asset is basically determined by a linear combination of these factors, the loading coefficients for various factors dictate the performance of this specific risky asset, thus reasonably serving as the essential features of the asset. More specifically, we assume that the market performance of any risky asset is governed by the following m-factor model,

$$r_i = a_i + \sum_{j=1}^{m} b_{ij} f_j + \epsilon_i, \quad i = 1, 2, \ldots, n, \tag{1}$$

where r_i is the return of security i, f_j is the random return of factor j, $j = 1, 2, \ldots, m$, a_i is the intercept, b_{ij} is the factor loading that measures the sensitivity of security i to factor j, and ϵ_i is the "local" error term with a zero mean. Theoretically, each error term should be independent of any factor and the errors of other securities.

From the above factor model, we have the expressions of the mean of r_i, $\bar{r_i}$, and the covariance between r_i and f_j, $\mathrm{cov}(r_i, f_j)$, in terms of the mean of the factors, $\bar{f_j}$, $j = 1$, …, m, and the covariance matrices among factors, σ_{f_k,f_j}, $i, j = 1$, …, m,

$$\bar{r_i} = a_i + \sum_{j=1}^{m} b_{ij} \bar{f_j},$$

$$\mathrm{cov}(r_i, f_j) = \sum_{k=1}^{m} b_{ik} \sigma_{f_k,f_j}.$$

For a given factor model, all the coefficients a_i and b_{ij}, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$, can be calculated by regression. Let us consider an $(m+1)$-dimensional vector feature space with the intercept and m factor loadings as its coordinates. We can thus project each risky asset into this feature space at the following point,

$$security_{(i)} \longrightarrow (a_i, b_{i1}, b_{i2}, \ldots, b_{im}),$$

which can be viewed as the feature vector of asset i.

Without any doubt, choice of different factors affects the outcome of mapping various assets into the feature space. Theoretically, any factor can be selected as long as the characteristic based on this factor is meaningful. We can even generate some artificial factors to fit the security data. Most importantly, each factor should be representative, and the loading coefficients of different risky assets to that factor should be distinctive such that the feature vectors can be distinguished in the further processing.

Generally, we can consider three different kinds of factors: i) macroeconomic factors, such as Gross National Product(GNP), consumer price index, and unemployment rate; ii) statistical factors, such as the Hang Seng index and the S&P500 index; and iii) fundamental factors, such as the firm size, dividend yield, and book-to-market ratio. In this paper, we use the Hang Seng Composite Industrial Index as factors in our analysis, including Hang Seng Comp. Energy Index, Hang Seng Comp. Materials Index, Hang Seng Comp. Industrial Goods Index, Hang Seng Comp. Consumer Goods Index, Hang Seng Comp. Services Index, Hang Seng Comp. Telecommunications Index, Hang Seng Comp. Utilities Index, Hang Seng Comp. Financials Index, Hang Seng Comp. Properties & Construction Index, Hang Seng Comp. Information Technology Index, and Hang Seng Comp. conglomerates index, the total 11 factors. In general, we witness a positive correlationship of securities in the same industry and believe that the different trends among various industries contain a grain of truth. These 11 indexes can represent 11 different aspects of the market while they are meaningful to consider. As a result, feature vectors calculated by regression are sufficiently scattered and are thus suitable to be clustered. Furthermore, these kinds of security indexes are authorized, normalized, and easily found.

2.2 Step 2: Grouping risky assets using clustering algorithm

After endowing each asset a feature vector, the next question is how to group together assets with similar feature vectors, such that the entire pool of assets is partitioned into s groups based on their similarity. The smaller the norm of the difference between a pair of feature vectors, the higher degree of similarity of these two assets. It is easy to conclude that the cluster analysis well serves the purpose of partitioning and grouping. The purpose of applying clustering in this research is to avoid inclusion of multiple risky assets in the portfolio that behave similarly in the market (i.e., multiple risky assets with similar feature vectors).

The three most commonly used classes of clustering algorithms are: partitioning clustering, density-based clustering, and hierarchical clustering (see, e.g., Han et al. (2011)). Partitioning clustering, including k-means, PAM, and CLARANS, aims to partition n points into s clusters so as to minimize the intra-cluster sum of squared errors. It performs better when clusters are hyper-ellipsoidal or globular in similar sizes. The main advantages of this algorithm are its simplicity and high speed, which allows its running on large data sets. However, it is sensitive to outliers and the final clusters depend on the initial assignments. Density-based clustering, including DBscan, is based on the assumption that the density within genuine clusters is uniform and much larger than the density of the interval. Hierarchical clustering, including CURE, ROCK, and CHAMELEON, seeks to build up successive clusters using previously established clusters. This kind of clustering renders a clear hierarchical tree, but the computation cost is high when a large data set is dealt with. Because our data set is relatively small and the dimension is low in terms of computation, we decide to adopt a relatively simple partitioning clustering algorithm called k-means in this research. We state the clustering procedure in our algorithm as follows.

```
k-means Algorithm
begin
randomly select $s$ observations from data set as the initial centroids
repeat
    form $s$ clusters by assigning all observations to the closest centroid
    recompute the centroid of each cluster
until
    the centroids no longer change
end
```

In the above partitioning clustering approach, each point is assigned to the cluster with the closest centroid. The initial assignment thus plays a very important role in this algorithm. In general, the whole procedure needs to be repeated several times, with each repetition having a new set of initial cluster centroid positions, and the finally chosen result should have a minimum intra-cluster sum of squared errors. The centroid is typically calculated by the mean of the points in each cluster. In this research, we measure the closeness by Euclidean distance. A concern with this algorithm is that an empty cluster may occur. If a cluster loses all its member observations, we will create a new cluster using the point furthest from its centroid.

2.3 Step 3: Identifying representative(s) from individual groups and forming the optimal portfolio

After we group risky assets using cluster analysis, the remaining task is to select representatives from individual groups to form the best portfolio (in terms of its mean and variance) that satisfies the cardinality constraint. To provide our algorithm more flexibility, we consider options with that i) k, the total number of groups generated from clustering analysis, may not be equal to s, the given cardinality, and ii) the number of representative(s) from an individual group could be different from one.

The n available securities as a whole form the market. What we do here in this research is to decompose the entire market into k clusters, within each of which all constituents possess similar features. Theoretically, a joint presence of representatives from all such "homogeneous" groups can somehow replicate the whole market, thus eliminating "performance redundancy" when involving all risky assets in the portfolio. Even though constituents in each cluster are "indistinct" in concept, we still seek to find the one with the best performance (jointly with other representatives) when evaluation the entire portfolio. Therefore, we consider adding the clustering result as pre-grouping constraints to problem $(\bar{P})$, with a primary purpose of reducing the size of the combinatorial choices.

Let the entire pool of risky assets be partitioned into k groups by clustering algorithm with I_i being the index set of cluster i, $i = 1, \cdots, k$, such that

$$I_i \cap I_j = \emptyset, \quad \text{for } i \neq j,$$
$$\cup_{i=1,\cdots,k} I_i = \{1, 2, \cdots, n\}.$$

We now consider the following revised version of problem $(\bar{P})$ by incorporating grouping constraints,

$$
\begin{aligned}
(\hat{P}) \quad & \min_{x} \ \sigma^2 = x'Qx, \\
\text{Subject to:} \quad & \mathbf{1}'x = 1, \\
& r'x \geq \bar{r}, \\
& 0 \leq x_i \leq z_i M, \quad i = 1, \cdots, n. \\
& \alpha_j \leq \sum_{i \in I_j} z_i \leq \beta_j, \quad j = 1, \cdots, k, \\
& \sum_{i=1}^{n} z_i = s, \\
& z = \{0, 1\}^n, \\
& x \geq 0,
\end{aligned}
$$

where all α_j and β_j, $j = 1, \cdots, k$, are nonnegative integers. In the above problem formulation $(\hat{P})$, while the portfolio has to satisfy the cardinality constraint, the number of representative assets selected from cluster j can vary from α_j to β_j, $j = 1, \cdots, k$.

Compared to the original problem formulation $(\bar{P})$, in which a combinatorial number, $\binom{n}{s} = \frac{n!}{s!(n-s)!}$, of choices can be considered as candidates to satisfy the cardinality constraint, the revised problem formulation $(\hat{P})$ only has $\prod_{j=1}^{k}(\beta_j - \alpha_j + 1)$ choices, a number much much less than $\binom{n}{s}$. This dimension reduction facilitates significantly the computational process for large-scale portfolio selection problems.

Varying the value of $\bar{r}$ yields the mean-variance efficient frontier under cardinality constraint and grouping constraints from clustering. Generally speaking, the efficient frontier of $(\hat{P})$ gets closer to that of $(\bar{P})$ when increasing $\beta - \alpha$ or k/s, or both. However, we need to consider the trade-off between the solution accuracy and computational time.

3. NUMERICAL RESULTS FROM COMPUTATIONAL EXPERIMENTS

We present in this section computational experiments using data of 679 stocks between Jan. 3, 2000 and Dec. 28, 2009 from the Hong Kong market. The computer we use to conduct the experiments is an Inter(R) Core(TM) 2 Quad CPU Q9400 @ 2.66GHz, 2.87GB of RAM. We carry out the clustering algorithm using Matlab, and compare our computational performance with CPLEX 12.2, a commercial optimization software, both in terms of running time and accuracy. For CPLEX 12.2 implementation, after several tries, we found that $M = 10$ is large enough for calculation.

We use the notation "$n - s - \bar{r}$" to denote a specific problem with a problem size n, cardinality size s, and pre-given return level $\bar{r}$. The three levels of pre-given return, "high", "mid", and "low", indicate that we set $\bar{r}$ as $\min(r) + \frac{4}{5}(\max(r) - \min(r))$, $\min(r) + \frac{1}{2}(\max(r) - \min(r))$, and $\min(r) + \frac{1}{5}(\max(r) - \min(r))$, respectively. We also set an upper limit of 3600 CPU seconds for calculation; that is, we terminate the calculation if an algorithm does not find an optimal solution within 3600 CPU seconds and record the incumbent as an approximate solution.

Table 1. Experiment results using CPLEX

	"200-10-mid"			"200-10-low"		
Prob	CPU	Node	OptV	CPU	Node	OptV
1	27.22	23262	0.001683	3600	2728991	0.000279
2	93.64	76290	0.001237	7.99	5232	0.000195
3	3600	3089187	0.000550	57.11	51587	0.000152
4	310.82	264206	0.000500	15.77	12457	0.000215
5	108.25	81068	0.000122	2.41	1862	0.000114
6	1589.55	1294729	0.000888	47.42	44686	0.000153
7	77.13	62093	0.000918	42.80	41813	0.000143
8	542.64	434630	0.000667	2.11	1429	0.000154
9	155.71	130758	0.000971	50.36	43017	0.000236
10	110.14	97073	0.001645	71.44	59603	0.000265
Avg	661.52	555330	-	389.74	299068	-

Table 2. Experiment 1 for "200-10-mid" using our new algorithm with $k = 10$, $\alpha = 1$ and $\beta = 1$

	$k = 10$, $s = 10$, $\alpha = 1$ and $\beta = 1$					
Prob	CPU1	CPU2	Node	Ratio	Acc	OptV
1	0.24	4.11	3554	0.1510	1.1294	0.001901
2	0.28	18.05	32015	0.1927	1.2395	0.001534
3	0.26	60.36	474076	0.0168	1.1550	0.000635
4	0.27	13.38	16826	0.0430	1.2873	0.000643
5	0.27	24.17	23709	0.2233	1.8777	0.000230
6	0.26	25.80	36079	0.0162	1.3366	0.001186
7	0.23	14.42	112850	0.1870	1.1431	0.001050
8	0.27	46.44	75867	0.0856	1.1606	0.000774
9	0.24	23.49	37260	0.1508	1.0960	0.001064
10	0.26	4.88	8111	0.0443	1.0469	0.001722
Avg	0.26	23.51	82035	0.1111	1.2472	-

Table 3. Experiment 2 for "200-10-mid" using our new algorithm with $k = 20$, $\alpha = 0$ and $\beta = 1$

	$k = 20$, $s = 10$, $\alpha = 0$ and $\beta = 1$					
Prob	CPU1	CPU2	Node	Ratio	Acc	OptV
1	0.37	4.83	3314	0.1774	1.0666	0.001795
2	0.33	12.44	9567	0.1328	1.0568	0.001308
3	0.32	49.95	23806	0.0139	1.0419	0.000573
4	0.37	9.33	6457	0.0300	1.0933	0.000546
5	0.36	11.20	7239	0.1035	1.2695	0.000155
6	0.35	171.80	128662	0.1081	1.1013	0.000977
7	0.37	14.22	9606	0.1844	1.0586	0.000972
8	0.38	5.98	4112	0.0110	1.0395	0.000693
9	0.34	18.16	14250	0.1166	1.0598	0.001029
10	0.40	12.41	10262	0.1126	1.0412	0.001713
Avg	0.36	31.03	21728	0.0990	1.0828	-

For each specific problem, we generate 10 cases in our computational experiments and take the average as the outcome. Due to the page limit, we only report in this paper the results for problems with n equal to 200 and $\bar{r}$ being set at "mid" and "low". In the tables below, the column 'CPU' refers to the execution time of CPU in seconds, the column 'Node' records the number of nodes visited in the enumerating tree, and the column 'OptV' gives the objective value attained by the algorithm.

Table 1 presents the computational results for problems "200-10-mid" and "200-10-low" by CPLEX. Tables 2, 3, 4 and 5 report the computational results by using our proposed algorithm, more specifically, by using the industry index as factors for the factor model and using the "k-means" clustering algorithm for grouping. When applying our method, we try different sets of parameter setting for k, α, and β, and we report only two such settings in this paper due to the page limit. In Tables 2,

Table 4. Experiment 1 for “200-10-low” using our new algorithm with $k = 10$, $\alpha = 1$ and $\beta = 1$

	$k = 10$, $s = 10$, $\alpha = 1$ and $\beta = 1$					
Prob	CPU1	CPU2	Node	Ratio	Acc	OptV
1	0.24	71.44	85902	0.0198	1.4158	0.000396
2	0.28	3.75	2687	0.4696	1.4946	0.000291
3	0.26	99.16	141904	1.7363	1.4337	0.000218
4	0.27	4.14	2610	0.2626	1.3396	0.000288
5	0.27	0.91	142	0.3768	1.7750	0.000202
6	0.26	6.09	4917	0.1285	1.1845	0.000182
7	0.23	6.72	5536	0.1570	1.2466	0.000178
8	0.27	1.44	1224	0.6818	1.4114	0.000218
9	0.24	5.00	4767	0.0993	1.1757	0.000278
10	0.26	23.80	28135	0.3331	1.4002	0.000372
Avg	0.26	22.24	27782	0.4265	1.3877	-

Table 5. Experiment 2 for “200-10-low” using our algorithm with $k = 20$, $\alpha = 0$ and $\beta = 1$

	$k = 20$, $s = 10$, $\alpha = 0$ and $\beta = 1$					
Prob	CPU1	CPU2	Node	Ratio	Acc	OptV
1	0.37	82.92	79258	0.0230	1.1166	0.000312
2	0.33	0.99	324	0.1234	1.3515	0.000263
3	0.32	12.17	8189	0.2131	1.3991	0.000213
4	0.37	2.16	1203	0.1367	1.2308	0.000265
5	0.36	1.08	444	0.4479	1.4959	0.000170
6	0.35	0.64	190	0.0135	1.2546	0.000192
7	0.37	9.75	6322	0.2278	1.5522	0.000222
8	0.38	1.08	317	0.5111	1.3336	0.000206
9	0.34	12.11	3352	0.2405	1.2088	0.000285
10	0.40	33.49	7576	0.4687	1.1583	0.000307
Avg	0.36	15.64	10718	0.2406	1.3102	-

3, 4 and 5, the column ‘CPU1’ refers to the CPU time of carrying out clustering in Matlab, and ‘CPU2’ refers to the CPU time of solving $(\hat{P})$ using CPLEX. The column ‘Ratio’ compares the CPU time of our method and CPLEX which is measured by the ratio of ‘CPU2’/‘CPU’, where ‘CPU’ is the corresponding time for problem $(\bar{P})$ presented in Table 1. The smaller the number, the more time saving of our algorithm over CPLEX. We also give the accurate rate of the objective value of our methods in column ‘Acc’, which is measured by the ratio $\frac{our`OptV'}{CPLEX`OptV'}$. The closer to one from above, the better the quality of our algorithm. Generally speaking, our method always has the advantage over CPLEX in CPU time, while paying a price of solution quality. We can see from our numerical experiments that the solution of $(\hat{P})$ is increasingly closer to $(\bar{P})$ with an increase of $\alpha - \beta$ and k/s; however, the CPU time increases as well as a trade-off.

4. CONCLUSIONS

Financial decisions should be based on structural information from the market, thus being benefitted from utilizing financial insights. We have provided a complete answer to the list of four questions raised in the beginning of Section 2 by invoking solution concepts from factor model, clustering analysis and integer programming. Using the factor model approach, a fundamental market decomposition scheme, we extract essential features in characterizing individual risky assets. With the help of clustering algorithm in grouping risky assets, we attach to the mixed integer programming formulation of CCMV problem a family of pre-grouping constraints, thus reducing significantly the computational complexity of the primal problem. Although our preliminary numerical experiments display promising performance of our newly developed algorithm, when compared with CPLEX, the most powerful powerful commercial software package in solving mixed integer programming, more systematic tests are needed for more systematic comparison, thus reaching recommendation to improve and to refine our algorithm.

REFERENCES

Bienstock, D. Computational study of a family of mixed-integer quadratic programming problems. *Math. Programming*, 74, 121–140, 1996.

Bertsimas, D., Shioda, R. Algorithm for cardinality-constrained quadratic optimization. *Comput. Optim. Appl.*, 43, 1–22, 2009.

Blog, B., van der Hoek, G., Rinnooy Kan, A., Timmer, G. T. The optimal selection of small portfolios. *Management Sci.*, 29, 792–798, 1983.

Chang, T. J., Meade, N., Beasley, J. E., Sharaiha, Y. M. Heuristics for cardinality constrained portfolio optimisation. *Comput. Oper. Res.*, 27, 1271–1302, 2000.

Chen, M. S., Han, J. W., Yu, P. S. Data mining: An overview from a database perspective. *IEEE Trans. Knowledge and Data Engineering*, 8, 866–883, 1996.

Craighead, S., Klemesrud, B. Stock selection based on cluster and outlier analysis, Nationwide Financial, One Nationwide Plaza, Columbus, OH 43215, USA, 2007.

Crama, Y., Schyns, M.. Simulated annealing for complex portfolio selection problems. *European J. Oper. Res.*, 150, 546–571, 2003.

Gao, J. J., Li, D. Optimal cardinality constrained portfolio selection. *Oper. Res.*, 61, 745–761, 2013.

Han, J. W., Kamber, M., Pei, J. Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.

Li, D., Sun, X. L., Wang, J. Optimal lot solution to cardinality constrained mean-variance formulation for portfolio selection. *Math. Finance*, 16, 83–101, 2006.

Luenberger, D. G. Investment Science, Oxford University Press, New York, 1998.

Markowitz, H. M. Portfolio selection. *J. Finance*, 7, 77–91, 1952.

Mokhtar, M., Shuib, A., Mohamad, D. Mathematical programing models for portfolio optimization problem: A Review. *Int. J. Social, Human Sci. Eng.*, 8, 411–420, 2014.

Ross, S. A., The arbitrage theory of capital asset pricing. *J. Economic Theory*, 13, 341–360, 1976.

Shaw, D. X., Liu, S., Kopman, L. Lagrangian relaxation procedure for cardinality-constrained portfolio optimization. *Optim. Methods Softw.* 23, 411–420, 2008.

Tola, V., Lillo, F., Gallegati, M., Mantegna, R. N. Cluster analysis for portfolio optimization. *J. Economic Dynamics & Control*, 32, 235–258, 2008.

Xie, J., He, S., Zhang, S. Randomized portfolio selection with constraints. *Pacific J. Optim.*, 4, 89–112, 2008.