Expert Systems With Applications 186 (2021) 115732

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Portfolio formation and optimization with continuous realignment: A suggested method for choosing the best portfolio of stocks using variable length NSGA-II

Ramen Pal [a], Tamal Datta Chaudhuri [b], Somnath Mukhopadhyay [a,*]

[a] Department of Computer Science and Engineering, Assam University, Silchar, Assam, India
[b] Department of Economics, Centre for Knowledge Ideas and Development Studies, Kolkata, West Bengal, India

ARTICLE INFO

Keywords:
Vertical and horizontal clustering
Markowitz principle
Dynamic realignment
Benchmark return
Variable length NSGA-II
Realistic genetic operator

ABSTRACT

In this paper, we introduce a method of portfolio formation through vertical and horizontal clustering. The clustering algorithm incorporates sufficiently diversified number of stocks in the portfolio with exposure limits on each stock. On obtaining the near Pareto optimal portfolios by using the proposed variable-length Non-dominated Sorting based Genetic Algorithm (NSGA-II), quarter-wise weights of each portfolio's constituent stocks are determined through the proposed single objective Genetic Algorithm (GA) based Markowitz model. This enables dynamic realignment of the portfolios and can incorporate the macroeconomic environment of the time. The performance of the portfolios is then compared with a benchmark portfolio. Our results show that returns from each of our portfolios, dynamically realigned each quarter, have been able to beat the benchmark index return over our study's time period. The performance of the clustering algorithm is validated with 4 well-known clustering algorithms.

1. Introduction

The literature on portfolio optimization has examined four aspects of the problem, namely:

i. Choice of stocks in the portfolio
ii. Specification of the objective function/s, the optimization problem, and the constraints
iii. The algorithm to determine the weights of the stocks in the portfolio
iv. Testing the performance of the portfolio

Regarding (i), the literature has either chosen a set of stocks in terms of market capitalization, or from the companies' reputation, or their fundamentals aggregated through some method. Some fundamental indicators considered are earnings per share (EPS), price to earnings ratio (P/E), dividend payout ratio(PR), price to sales ratio (P/S), debt–equity ratio (DER), price to book value per share (P/B), current ratio (CR), price to cash flow ratio (P/CF), gross or net profit margin(PM) and accounts receivable turnover ratio(ART).

The literature is quite rich concerning (ii) and has extended the basic Markowitz model significantly. Instead of a single objective function, multiple objective functions have been considered. These range from maximizing returns, minimizing variance, maximizing skewness, minimizing kurtosis, minimizing semi-variance, maximizing the relative strength index, and minimizing the P/E ratio. The constraints considered are no short selling, cardinality, liquidity, capital requirements, and Value at Risk (VaR).

In (iii), the algorithms used include multi objective evolutionary algorithms (MOEAs) (Deb et al., 2002), particle swarm optimization (PSO) (Kennedy & Eberhart, 1995), ant colony optimization (ACO) [(Dorigo et al., 2006)], bacterial foraging optimization (BFO) (Dasgupta et al., 2009), artificial bee colony (ABC) (Karaboga & Basturk, 2008), cat swarm optimization (CSO) (Shu-Chuan et al., 2006), invasive weed optimization (IWO) (Karimkashi & Kishk, 2010), bat algorithm (BA) (Yang & Gandomi, 2012) and fireworks algorithm (FA) (Tan & Zhu, 2010). Use of fuzzy variables has also been adopted in the literature. For a review of the literature, one can refer to Okkes Ertenlice (Ertenlice & Kalayci, 2018), Mehmet Anil Akbay (Kalayci et al., 2019; Liagkouras & Metaxiotis, 2015), and Masmoudi (Masmoudi & Abdelaziz, 2017).

For (iv), the results of running the algorithms on similar data sets have been compared. Given the above four aspects of portfolio optimization, our paper focuses on (i) and (iv). It is designed to aid fund managers and financial advisers and does not focus on short-term and long-term individual investors' goals. Based on a single metric,

* Corresponding author.
E-mail addresses: ramen.pal673@gmail.com (R. Pal), tamal5302@yahoo.com (T.D. Chaudhuri), som.cse@live.com (S. Mukhopadhyay).

https://doi.org/10.1016/j.eswa.2021.115732
Received 30 November 2020; Received in revised form 28 July 2021; Accepted 4 August 2021
Available online 16 August 2021

along the lines of Mukhopadhyay and Datta Chaudhuri (Mukhopadhyay & Chaudhuri, 2019), the paper proposes a methodology for choosing stocks through vertical and horizontal clustering. Imposing restrictions on the number of stocks in each portfolio, near Pareto optimal clusters are generated. The cardinality constraint considered in the literature has been taken care of in the initial choice of stocks. The weights are determined using a single objective function. Each portfolio is then rebalanced each quarter by adjusting the portfolios' weights and overall performance. It is then checked whether the optimal portfolios can generate returns superior to the market index. This is in line with the Capital Asset Pricing Model and gives us an idea about α. Our method also enables us to determine the best portfolio in terms of performance.

The cardinality constraint is taken care of in the clustering algorithm. Liquidity is ensured through the initial choice of the master set of companies, from which the near Pareto optimal clusters are generated. These companies are consistently profit making, dividend-paying companies and are highly liquid with significant daily turnover. To avoid the issue of illiquidity, small-cap stocks and start-ups have not been included in the sample. The paper is aimed to help fund managers and financial analysts as it covers the following aspects, namely:

1. Vertical and Horizontal Clustering using Multi-Objective Optimization algorithm.
2. Choice of a single metric for clustering.
3. Training the clustering algorithm to have a sufficiently diversified number of stocks.
4. Obtaining near Pareto Optimal Portfolios.
5. Determining Quarter-wise Weights of the Constituent Stocks for each Portfolio through Markowitz principle.
6. Realigning the Portfolios Every Quarter and Determining the Quarterly Returns.
7. Comparing each Portfolio Returns with Nifty Returns as the benchmark.
8. Determining the Best Portfolio.

1.1. Literature review

In order to incorporate inherent uncertainties and solution robustness in the context of portfolio optimization, Quintana et al. (2017) proposed a robustness-based S-metric selection evolutionary multi-objective optimization algorithm (R-SMS-EMOA). The framework allows for the implementation issue, where there can be temporary deviations from the target portfolio. The results show that solutions are robust concerning (i) implementation risk and (ii) estimation risk. The paper considers simultaneous maximizing returns and minimizing risk and considers three additional constraints: cardinality, the total number of assets that can be considered, and upper and lower limits to the weights. A robustness measure is defined as an average of the ratios of deviations. The framework's validation was through an experiment on assets belonging to eight financial indices over fifteen years. The results are evaluated in terms of three metrics: stability, implementation risk, and extreme risk compared to a standard optimization framework.

Kaucic et al. (2019) propose a portfolio optimization framework where they maximize returns, minimize semi-variance, and put restrictions on capital loss. The paper uses evolutionary NSGA-II and SPEA 2 and the data sets are obtained from five indices and their constituent companies. Three optimization problems are solved using Mean-SV, Mean-CVar, and Mean-CVar-SV. The metrics for comparing the algorithms are Schott's spacing metric, generalized spread metric, inverted generation distance (IGD), and hypervolume.

Mehlawat and Gupta (2014) incorporate short-term returns, long-term returns, and liquidity in the portfolio in the portfolio optimization problem. The investor maximizes short term return and long term returns such that they exceed a desirable value of returns with some probability. Further, they obtain liquidity that is greater than some value of liquidity with some credibility. They impose upper and lower bounds to the weights and also upper bound to portfolio variance. Their model is a credibility-based fuzzy chance-constrained multi-objective model, and they apply a real coded genetic algorithm(RCGA) to solve the portfolio optimization problem. They experiment on seven assets and conclude that high expectations of short-term returns, long term returns, and high liquidity are associated with a high tolerance for portfolio risk. The results are also compared with Safety-First and Value-At-Risk-Based Portfolio Selection Models.

Mehlawat et al. (2018) proposed a multi-objective portfolio selection problem where the objectives included mean, variance, skewness, kurtosis, and efficiency. Efficiency score was derived from a DEA framework where the inputs were leverage, price/earnings ratio, and beta and the outputs were asset turnover ratio, earnings per share, and earnings per share growth rate. The proposed model's constraints included bounds on investment in individual assets, full utilization of funds, and no short selling of assets. There were nine assets in the portfolio. The framework's performance was measured with NIFTY based on returns.

Rezaei Pouya et al. (2016) solves a multi-objective portfolio optimization problem using invasive weed optimization based on fifty top companies of the Teheran Stock Exchange. The mean–variance framework has used additional objectives like the P/E ratio and experts' recommendation on stocks' future performance. They have compared their results with that of the Particle Swarm Optimization algorithm and the Reduced Gradient Method. Their framework transforms a multi-objective-portfolio selection model into a single-objective programming model using fuzzy normalization.

Wang et al. (2018) use the Sharpe ratio and Value-at-Risk ratio in fuzzy environments for portfolio optimization. The solutions are obtained for stocks from the New York Stock Exchange, and it is found that the portfolio using the Sharpe ratio is more diversified than the Value at Risk Ratio model. Finally, the performance of their proposed FMOPSO is compared with that of IMOPSO and TV-MOPSO. The same experiment is also conducted with twelve securities from different industries from the China Shanghai Stock Exchange.

Mitra Thakur et al. (2018) considered ten ratios, namely earnings per share, price to earnings ratio, dividend payout ratio, price to sales ratio, debt–equity ratio, price to book value per share, current ratio, price to cash flow, net profit margin and accounts receivable turnover ratio and then experts were asked through a questionnaire about their relative importance by the Delphi method. The experiment was conducted with stocks listed in the Bombay Stock Exchange, and of the thirty companies, the top ten stocks were considered in an ant colony optimization algorithm. The portfolio optimization problem was formulated with maximization of excess returns divided by semi-variance, subject to constraints on returns, skewness, and variance, without short selling.

Fuzzy mean–semi variance and DEA cross efficiency was incorporated in a fuzzy multi-objective portfolio selection model in Chen et al. (2020). The paper maximizes expected return and the Sharpe ratio, minimizes semi variance with cardinality constraint, and also limits the weights. It also maximizes the Sharpe ratio. A firefly algorithm was used in the optimization process. The inputs consisted of asset utilization, liquidity, leverage ratios, output, profitability, and growth ratios for the DEA. The results for stocks selected from Iran's stock exchange were compared to those derived from a possibilistic mean–semi variance portfolio model and were found to be superior.

In the multi-objective portfolio selection model of Li and Xu (2013), risk, returns, and liquidity are considered in the objective function. Further, experts' experience and judgment and investors' subjective attitudes about securities' future returns are incorporated in the model. With a sample of stocks, they demonstrate the sensitivity of their results with the investors' optimism and pessimism.

In their paper, Darsha Panwar and Srivastava (2018) divides the investors into three categories, namely high return-seeking, high risk-seeking, and liquidity seeking. From the responses, a clustering methodology is applied to each of the categories, and the set of companies are arrived at. The portfolio optimization problem uses different objective functions, including risk and return, a technical indicator like the Relative Strength Index, and ratios like the PEG ratio and the earnings yield. A fuzzy multi-objective linear programming optimization is performed. The multi-objective problem is converted to a single objective problem using a "weighted adaptive approach" in which AHP calculates the weights. The results are compared to those arrived at in Gupta et al. (2010).

Saranya and Prasanna (2014) proposed a portfolio optimization model where the goals are maximization of returns and skewness and minimization of variance and kurtosis. A polynomial goal program was run on 108 sample portfolios constructed from non-normal stocks from the BSE 200 stocks. Three experiments were run consisting of the mean–variance model, mean–variance–skewness model, and the mean–variance–skewness–kurtosis model. The sample and out of sample returns of the portfolios of all the models across bullish, bearish, and crisis periods were examined to observe the consistency in model performance.

Wei Yue and Dai (2015) considered portfolio optimization using mean–variance–skewness and mean–variance–skewness-entropy (MVS-E) frameworks. They used a multi-objective fuzzy portfolio selection model with transaction cost and liquidity and compared its performance with MOEA/D and NSGA-II with Shanghai Stock Exchange Market data. Simulation results showed that their algorithm resulted in better portfolio diversification.

Macedo et al. (2017) incorporated technical indicators like Relative Strength Index (RSI), Moving Average Convergence/Divergence (MACD), Contrarian Bollinger Bands (CBB), and Bollinger Bands (BB) in a mean semi-variance portfolio optimization framework. The portfolio optimization exercise was conducted for four different sets of countries with varying degrees of development. The study compared the results of the NSGA II and SPEA II algorithms.

In Meghwani and Thakur (2017), a mean–variance problem is formulated for portfolio optimization with cardinality, pre-assignment, budget, quantity (floor and ceiling), and round-lot constraints. The following models have been used, namely NSGA-II, SPEA2, GWASFGA, and PESA-II. An empirical study is performed with assets from seven indices worldwide. The algorithms' relative performance is compared based on measures like generational distance, inverted generational distance, hypervolume, diversity, and ϵ-indicator.

One of the paper's objectives by Paiva et al. (2019) was to build an algorithm that would enable the classification of stocks by their potential to reach pre-determined daily returns. For this, Support Vector Regression was used for forecasting. A mean–variance portfolio problem was formulated with those stocks from the Sao Paulo Stock Exchange Index (Ibovespa) that reached a stipulated profit goal. Besides SVM + MV, two other models were used; namely, SVM + 1/N and Random + 1/N for portfolio generation and the results were compared with the performance of Ibovespa.

To reduce the computational time involved in solving portfolio optimization problems with many assets, Qu et al. (2017) considered two asset pre-selection procedures that consider return and risk and those assets whose returns are negatively correlated. For optimization, a Normalized Multi-objective Evolutionary Algorithm based on Decomposition (NMOEA/D) is used, and the results are compared with those of other algorithms used in this area of research.

Strumberger et al. (2016) applied the hybridized bat algorithm to a standard Mean–Variance constrained portfolio optimization problem. Portfolio variance was minimized, subject to returns exceeding a minimum benchmark. To test the robustness of the approach, a comparative analysis was made with results of other swarm intelligence algorithms and three variants of the genetic algorithm.

Yuanyuan Zhang and Guo (2018) proposed a new algorithm for portfolio optimization problem namely MOEA/D-CP based on MOEA/D. This utilizes a new weight vector generation approach such that different assets get sufficient weightage in the portfolio. Their results show that the MOEA/D-CP performs better than algorithms based on MOEA/D.

As an extension of their previous work, Dreżewski and Doroz (2017) propose an agent-based co-evolutionary algorithm for multi-objective portfolio optimization. They apply their algorithm to two different phases of the stock market in Warsaw to test for robustness. They consider one period of relative bullishness and one period of severe bearish condition when the index lost many points. The results were compared to those models applying the genetic algorithm, co-evolutionary algorithm, and trend-following algorithm to the same data sets.

Kumar and Mishra (2017) present a co-variance guided Artificial Bee Colony algorithm for portfolio optimization. They conducted five different exercises of portfolio optimization for five different markets across the world. For efficiency, they use the measures GD and SP. Their results show one relationship between risk and return. Portfolios with high risk have also shown higher returns.

Liagkouras and Metaxiotis (2017) incorporate more constraints in a Mean–Variance portfolio optimization framework and apply a multi-objective evolutionary algorithm (MOEA) to obtain solutions. The constraints are no short selling, cardinality, floor and ceiling, class constraint, and pre-assignment constraint. Experiments are conducted from existing data sets, and the results are compared based on the basis of GD, IGD, HV, and ϵ-indicator.

Liagkouras (2018) has formulated a three-dimensional encoding multi-objective evolutionary algorithm for portfolio optimization with different objectives like maximizing returns, minimizing variance, minimizing semi-variance, minimizing mean absolute deviation, minimizing value at risk, and minimizing conditional value at risk. They establish that their algorithm can solve such problems in minimum computing time. They empirically test their algorithm with data from assets from four different stock market indices worldwide.

In the same vein of reducing computational time and generating efficient solutions, a hybrid multi-objective evolutionary algorithm (MODE-GL) is proposed in Lwin et al. (2017). The objective function consists of optimizing mean returns and VaR of the portfolio under constraints such as cardinality, quantity, pre-assignment, round-lot, and class. The proposed algorithm is compared against NSGA-II and SPEA using data on financial assets from S & P 100 and S & P 500 indices.

Ni et al. (2017) present a variant of the PSO algorithm in a mean–variance problem of portfolio optimization with cardinality constraints and lower and upper bounds on exposure. They test their framework with data from four stock market indices worldwide and use measures like mean Euclidean distance, the variance of return error, mean return error, and contribution (%).

Ban et al. (2018) introduces performance-based regularization (PBR) for both mean–variance and mean-conditional value-at-risk (CVaR) problems in portfolio optimization. Out of sample performance was computed for eight portfolio allocation strategies, namely SAA, PBR only on the objective, PBR only on the constraint, PBR on both the objective and the constraint, L1 regularization, L2 regularization, Equally weighted portfolio, and Global minimum CVaR portfolio.

Jalota and Thakur (2017) combined the BEX-PM genetic algorithm with Repair Mechanism for portfolio optimization. The constraints considered were cardinality, budget and upper and lower bounds. Mean percentage error is used as a metric to compare between algorithms used on data set of assets belonging to worldwide stock market indices.

Kalayci et al. (2017) present an artificial bee colony algorithm to solve a cardinally constrained portfolio optimization problem with feasibility enforcement and infeasibility toleration. The algorithm was tested on data sets of developed and developing economies. The results were tested with measurement metrics like MEAPE, MEDPE, MINPE, MAXPE, VRE-I, MRE-I, MEUCD, VRE-II, and MRE-II.

Li et al. (2021) integrate sentiment aware variables with the returns distribution of the Shanghai Security Composite Index (SSCI) to forecast volatility and understand better jump intensity and jump size. Their methodology is expected to forecast volatility and extreme events better. Data on market sentiment reflected through attention, sentiment, and disagreement is obtained from textual analysis of comments on SSCI and Deep Learning Models are applied for analysis. This is embedded in a GARCH jump model for forecasting.

Koratamaddi et al. (2021) modifies the mean–variance approach to a portfolio optimization problem by including market sentiment variables obtained from social media platforms. They hypothesize that investors, besides past movement in stock prices, are influenced by these sentiment variables which are present in conventional news media and social media. Their data set consists of 30 stocks from the Dow Jones Industrial Average Index and they use a Deep Reinforcement Learning algorithm for analysis.

Combining indicators of technical analysis of stock price movements and fundamental analysis of companies, Picasso et al. (2019) present a stock price prediction framework with a portfolio of stocks listed in NASDAQ100 index. The technical indicators chosen include MACD, Bollinger Band, RSI, Moving Averages and Stochastic Oscillators. The methodology includes use of Random Forest, SVM and feed forward neural network.

Fund managers have different styles in stock selection and it is represented by the parameters they use to choose stocks in their portfolio. Li and Wu (2021) combine technical analysis indicators with sentiment analysis to identify different styles using clustering techniques. Stock price prediction exercise is then undertaken within each market style. They observe that an approach using market style outperforms one without market style.

Dang et al. (2018) use a deep learning approach to stock price prediction using sentiment analysis from financial news. They propose a two-stream Gated Recurrent Unit Network and Stock2Vec - a sentiment word embedding trained on financial news dataset and Harvard IV-4.

Valle-Cruz et al. (2021) integrates sentiments expressed through Twitter data with technical and fundamental analysis indicators to understand investor reaction to COVID-19 pandemic and H1N1 pandemic. They observed that the most influential Twitter accounts during the period of the pandemic were The New York Times, Bloomberg, CNN News and Investing.com.

Xing et al. (2019) propose a sentiment-aware volatility forecasting model by extending the VRNN model by integrating sentiment signals from social media. The authors suggest that stock prices move based on companies' fundamental performance and based on events and market sentiment. Thus, both sentiment variables and returns series feature in their forecasting model.

Zhang et al. (2018) use market sentiment, extracted from multiple sources, and investigate their joint impacts on stock price movements through a coupled matrix and tensor factorization framework. They feed this information to a predictive model to improve predictive efficiency.

Zhang et al. (2020) propose a CEEMD-PCA-LSTM hybrid deep learning framework for forecasting stock prices. While CEEMD can decompose the fluctuations or trends of different scales of time series, PCA eliminates the redundant information. These features are then subsequently fed into LSTM networks.

Qiu et al. (2019) use two decomposition methods, DWT and EMD, to decompose time movement of ten technical indicators. Subsequently, random vector functional link network (RVFL) and support vector regression (SVR) are applied on the decomposed series to generate stock price forecasts. The results are compared with existing predictive frameworks.

The rest of the paper is organized as follows. Detailed discussion on the proposed method is presented in Section 2. Results and discussion are presented in Section 3. Section 4 concludes the paper.

2. Proposed methodology

In this section, we discuss the proposed method in detail. The flow diagram of the methodology is presented in Fig. 1. The methodology is presented in Algorithm 1. It is a two-step process. In the first step, multiple different length portfolios are optimized by executing our proposed variable length and multi-objective clustering technique in two interdependent steps. In the next step, weights for each stock in every portfolio for different quarters are optimized using our proposed Genetic Algorithm (GA) based Markowitz model. During the optimization of portfolios, first, date-wise or vertical clustering is done to optimize multiple sets of centroids by using average returns per unit of risk for all companies on each date. For each date, the set of centroids with maximum silhouette score is preserved for the next step. Silhouette score ($Silhouette$) is a standard criteria for determining the quality of any clustering algorithm's internal performance. A high score (near to 1) indicates that the observations are well segmented, a lower score (near to −1) indicates that the observations are misclassified, and a score near to 0 indicates that the clusters are overlapped. It is calculated by using Eq. (1).

$$Silhouette = \sum_{i=1}^{n} \left\{ \frac{inter_{C_i} - intra_{C_i}}{max(inter_{C_i}, intra_{C_i})} \right\} \tag{1}$$

Here, $inter$ defines the inter-cluster distance and $intra$ defines the intra-cluster distance for clusters (C). After that, horizontal clustering is done using all these preserved centroids over dates to optimize multiple different length portfolios. We then extracted quarter-wise returns per unit of risk for all stocks in each optimized portfolio and computed the variance–covariance matrix, the portfolio variance, and portfolio returns. Finally, the weights for all stocks in each portfolio for 40 different quarters (for 10 years) are optimized using the proposed GA-based Markowitz framework. We discuss the proposed clustering method and the weight optimization technique in Sections 2.1 and 2.2 respectively.

2.1. Proposed method for clustering of the stock market data

Our proposed multi objective and variable length NSGA-II based clustering algorithm is presented in Algorithm 2. We have considered the NSGA-II of Deb et al. (2002), and modified it to adopt the variable length behavior. In this technique, first the initial population with n number of randomly taken different length solutions are encoded. Each solution contains randomly taken return per unit of risk values (centroids) from the input dataset. A portfolio is treated well diverse and less biased if it comprises with of $10-15$ stocks. Thus, we have limited the length of each solution within the range mentioned above. Then fitness of each solution from the initial population is evaluated by using inter-cluster distance (Eq. (2)) and standard deviation (Eq. (3)).

$$\delta(M_1, M_2) = \max\left\{ eucid(X,Y) \right\}; \quad \text{where, } X \in M_1 \ \& \ Y \in M_2 \tag{2}$$

The complete linkage distance (δ) is the maximum inter-cluster distance (Euclidean) between any two clusters ($M_1 \& M_2$) in a set of clusters Dawyndt et al. (2005). $Euclid$ defines the Euclidean distance. It is calculated by using the function defined in (Mukhopadhyay et al., 2020).

$$\sigma(M) = \sqrt{\frac{\sum_{j=1}^{n} X_j - \bar{X}}{n}}; \quad \text{where, } X_j \in M \tag{3}$$

Standard deviation (σ) measures of dispersion of all observations in a cluster (Park & Jun, 2009). Here, $\bar{X}$ is the mean of all (n) observations in a cluster (M). A clustering algorithm is treated as efficient if it can generate clusters with minimum σ and maximum δ. So, our proposed method performed clustering by maximizing δ and minimizing σ. After the fitness evaluation of chromosomes we have executed the non-dominated sorting algorithm of Deb et al. (2002) to compute the fronts

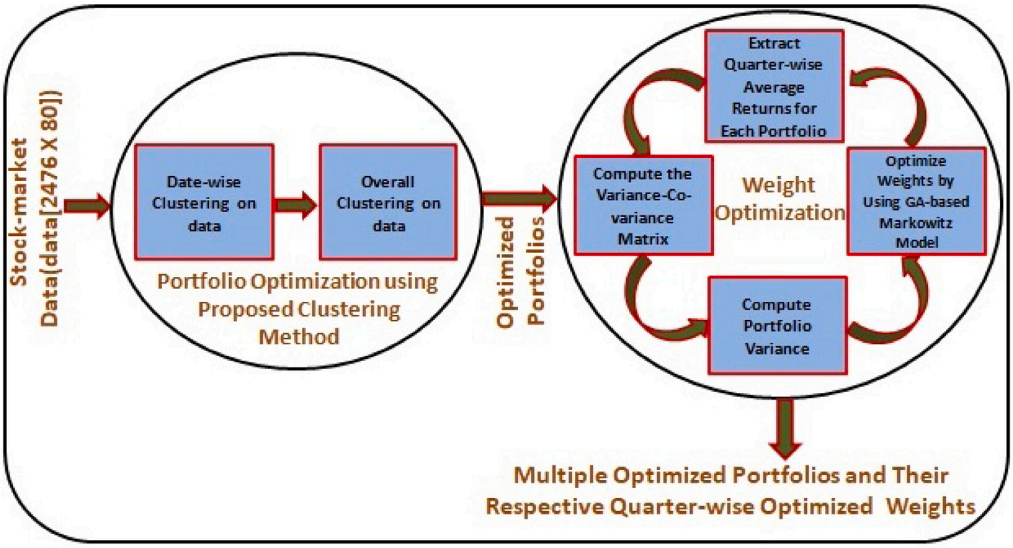

Stock-market Data(data[2476 X 80])
Date-wise Clustering on data
Overall Clustering on data
Portfolio Optimization using Proposed Clustering Method
Optimized Portfolios
Extract Quarter-wise Average Returns for Each Portfolio
Compute the Variance-Co-variance Matrix
Weight Optimization
Optimize Weights by Using GA-based Markowitz Model
Compute Portfolio Variance
Multiple Optimized Portfolios and Their Respective Quarter-wise Optimized Weights

Fig. 1. Proposed method for portfolio optimization and determination of quarter-wise weights.

Algorithm 1: Portfolio Optimization Using Variable length Multi Objective Non Dominated Sorting based Genetic Algorithm.

```
Input: Stock Market Data($data[2476 \times 80]$), Number of years($yr$), Number of generation($gen$), Number of chromosomes($n$)
Output: Set of Portfolios($pf_{optimized}$), Quarter-wise Weights for each portfolio($pf_{Weights}$)
Begin;
/* Date(row) wise clustering on $data$ using NSGA-II based optimization initiated */
for $i = 1$ to 2476 do
    $return = data[i \times 80]$ ;    // Return on date $i$
    Perform the clustering on $return$ by using Algorithm 2 and store the $front - 1$ solutions in $newpool$ ;
    Compute the silhouette score ($SC$) for each solution in $newpool$ by using Eq. (1);
    Preserve the solution with the lowest $SC$ in a repository($date_{sol}$);
end
/* Clustering on the preserved date wise solutions($date_{sol}$) using NSGA-II based optimization initiated */
$return = date_{sol}$;
Perform the clustering on $return$ by using Algorithm 2 and store the $front - 1$ solutions in $pf_{optimized}$ ;
$pf$= total number of portfolios in $pf_{optimized}$ ;
$quarter = yr \times 4$;
/* Optimizing the weights for portfolios using GA-based Markowitz model initiated */
for $q = 0$ to $qr$ do
    for $p = 0$ to $pf$ do
        Date-wise weighted average returns for each company in $p$ are extracted and preserved it in $data$;
        The variance–covariance matrix ($Cov_{mat}$) is calculated by using $data$;
        Portfolio variance is calculated based $Cov_{mat}$;
        Weights are optimized by using the Algorithm 5 and preserved it in $pf_{Weights}$.;
    end
end
End;
```

(ranking) of solutions. For the first iteration of the algorithm, we considered Binary Tournament (tournament size = 2) Selection (BTS) strategy to generate a mating pool with n number of chromosomes. BTS selects one best solution between 2 solutions based on their rank. We have taken the random selection approach, in case two solutions in a tournament have same rank. In the rest of the iteration of the algorithm crowded-BTS is considered to generate the mating pool. In this strategy a solution with the highest crowding distance value is selected between two solutions with the same rank. Crowding distance function proposed by Deb et al. (2002) is considered here.

The solutions in the selected mating pool are reproduced by using our proposed genetic operators. Our proposed different length enabled real-valued crossover technique is presented in Algorithm 3. Crossover is one of the important genetic operators in biological evolution. In reality, genetic sequences in parent chromosomes does not change drastically in the child chromosomes. Random crossover algorithms do not retain these properties mentioned above. So, these algorithms suffered from poor convergence rate and failed to generate the fittest child (comparing with parent) in most cases. This research has proposed a realistic, different length enabled, and real-valued crossover technique to address the issue mentioned above. A graphical representation of the flow of execution of a crossover operation is presented in Fig. 2. In this technique, first, the common genetic sequences between the parent chromosomes are identified and preserved. Any uncommon gene sequence is represented as Crossover Window (CW). Second, total number of CW's in each chromosomes are computed and represented as $count_{c_1}$ and $count_{c_2}$ respectively. Third, the chromosome (say c_1) with the lowest number of CW(say $count_{c_1}$) is identified. All the sequences in that chromosome are represented in an order ($CW_1, CW_2, \dots, CW_n$) according to their appearances (either left to right or right to left order). Fourth, these CW's are replaced with randomly chosen CW's from the other one. It is done to preserve the variable length property and increase the possibility of increasing the reproduced chromosomes' fitness. The probability of executing a crossover operation should be a higher value. So, we have considered 0.8 as the threshold probabilistic value in our algorithm.

The proposed mutation technique is represented in Algorithm 4. We have designed the mutation technique to allow the new stock to be a centroid or eliminate a stock from the chromosome. A mutation operation is represented graphically in Fig. 3. In this technique, a gene (g) is mutated by randomly selecting a return per unit of risk information($temp$) of any new stock from the input dataset. If the corresponding stock of the $temp$ is already present (in the form of

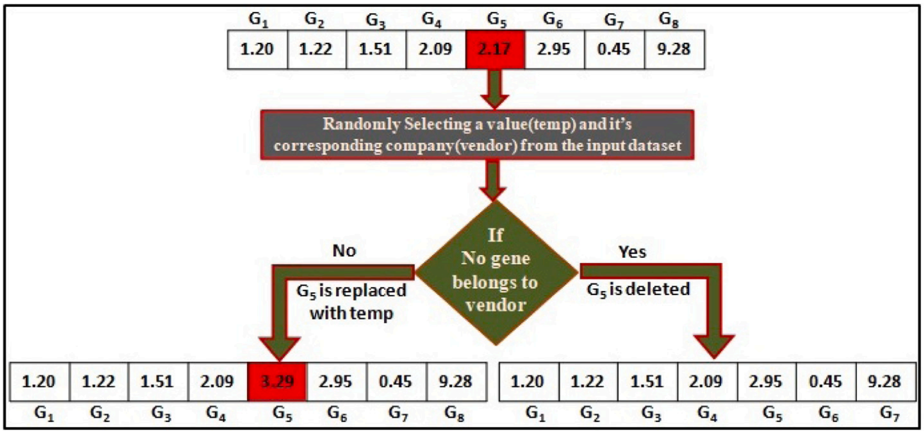


Fig. 2. Flow diagram of the proposed crossover algorithm.

Algorithm 2: Proposed method for NSGA-II based clustering of stock data

```
Input: $return$, $gen$, $n$
Output: $Front - 1$ solutions
Begin;
i=1;
Initial population with $n$ number of variable length ($len$)
chromosomes are encoded by randomly selecting return values
from $return$ ;                                   // $10 \leq len \leq 15$
for $g = 1$ to $gen$ do
    Fitness values ($fit_1$ and $fit_2$) for each chromosome ($i$) are
    computed by using Eqs. (1) and (2) respectively;
    Non-dominated sorting is performed to assign rank to each
    solution. if $g = 1$ then
        $n$ number of solutions are selected from the population to
        create the mating pool by using Binary Tournament
        Selection ($BTS$) strategy;
    end
    else
        Crowding distance for each solution is computed.;
        $n$ number of solutions are selected from the population to
        create the mating pool by using Crowded Binary
        Tournament Selection ($CBTS$) strategy;
    end
    Crossover is performed on the mating pool by using
    Algorithm 2;
    The mutation is performed on the mating pool by using the
    Algorithm 3;
    The solutions from the population and the mating pool are
    combined to get the combined population;
    Non-dominated sorting is performed on the combined
    population to compute the ranks for the solutions;
    Crowding distance is calculated for each solution;
    $CBTS$ strategy is applied to the combined population to get
    a population with $n$ number of solutions;
end
Rank-1 solutions are returned to the calling method;
End;
```

other return values) in the selected chromosome, then the gene(g) is eliminated from that chromosome. This is done to maintain the variable length property of the algorithm and to check that the other centroids present in that chromosome are sufficient to give better clustering results. The chance of a gene to get mutated should be a very less value

Algorithm 3: Proposed crossover technique on real-coded and variable length chromosomes

```
Input: Mating pool($mpool$), $n$
Output: Updated mating pool($mpool$)
Begin;
for $c = 1$ to $\frac{n}{2}$ do
    From the $mpool$, two chromosomes ($C_1$ and $C_2$) are randomly
    selected;
    A probabilistic value $C_p$ is randomly selected;
    if $C_p \leq 0.8$ then
        Common genetic sequences or genes between the $C_1$ and
        $C_2$ are identified and preserved;
        $count_{C_1}$= number of uncommon gene sequences or genes
        in $C_1$;
        $g_{C_1}$= $count_{C_1}$ number of gene sequences in $C_1$;
        $count_{C_2}$= number of uncommon gene sequences or genes
        in $C_2$;
        $g_{C_2}$= $count_{C_2}$ number of gene sequences in $C_2$;
        if $count_{c_1} \geq count_{C_2}$ then
            $count_{C_2}$ number of gene sequences(irrespective the
            number of genes in each sequence) in $C_2$ are
            uniformly swapped with the $count_{C_2}$ number of gene
            sequences in $C_1$;
        end
        else
            $count_{C_1}$ number of gene sequences(irrespective the
            number of genes in each sequence) in $C_1$ are
            uniformly swapped with the $count_{C_1}$ number of gene
            sequences in $C_2$;
        end
    end
end
End;
```

compared to crossover. Thus we have considered 0.1 as the thresholded mutation probability.

After the reproduction (crossover and mutation) operation, the fitness evaluation of all chromosomes is done in the reproduced mating pool with n number of chromosomes is done. It is then combined with the population (set of solutions before the selection operation) with n number of chromosomes to get a combined mating pool with $2 \times n$ number of chromosomes. Subsequently, non dominated sorting, and the crowded BTS operations are executed again to generate a new population with n number of chromosomes for the next iteration. On

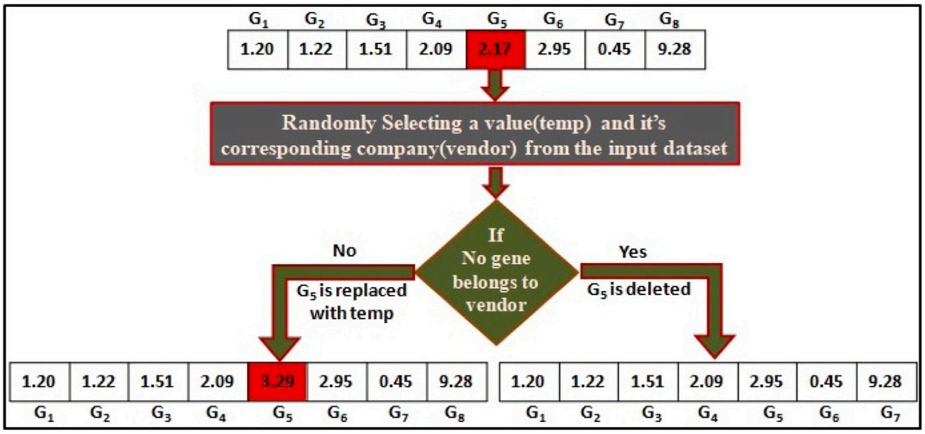


Fig. 3. Flow diagram of the proposed mutation algorithm.

Algorithm 4: Proposed mutation technique on real-coded and variable length chromosomes

```
Input: Mating pool($mpool$), $n$, return
Output: Updated mating pool($mpool$)
Begin;
for $c = 1$ to $\frac{n}{2}$ do
    From the $mpool$, one chromosome($C_1$) is randomly selected;
    for Each gene($g$) in $C_1$ do
        A probabilistic value $M_p$ is randomly selected;
        if $M_p \le 0.1$ then
            $temp$= a randomly selected value from $return$;
            $vendor$= the company associated with the $temp$;
            if there is no gene(value) corresponds to the $vendor$ then
                $g$ is replaced with $temp$;
            end
            else
                $g$ is deleted form $C_1$;
            end
        end
    end
end
End;
```

completion of the execution, all the $rank - 1$ solutions are selected as the set of optimized solutions.

2.2. Proposed method for weight optimization of stock in a portfolio

Markowitz model optimize the weights of all stocks in a portfolio that gives the highest return in respect of a predefined level of risk or lowest risk corresponding to a predefined level of return. We have incorporated this Markowitz model in our proposed GA based weights optimization algorithm and is presented in Algorithm 5. We have optimized the weights for each portfolio in 40 different quarters by minimizing the portfolio variance. The calculation of portfolio variance is done by using Eq. (4).

$$\sigma^2 = W^T \times \sum \times W \tag{4}$$

The portfolio variance is represented by using σ^2. W are the weights of stocks in a portfolio. W^T is used to define the transpose of the weight matrix(W). $\sum$ is used to define the variance–covariance matrix. The weight(W_i) of a stock(i) is limited within the range 0.002 to 0.15. The required return (θ) is considered as $\frac{0.15}{250}$. We have incorporated the condition that multiplication $W^T[n \times 1]$ with the returns of a stock (R[$n \times 1$]) will always equal the value of θ and the summation of weights of all stocks is always equal to 1. In this algorithm, the initial population with n number of chromosomes is encoded using probabilistic values. The value of each gene (weights) is randomly selected within the range of 0.002 to 0.15. The length of each solution is the same as the length of the input portfolio. The fitness value of each chromosome is evaluated by using Eq. (4). The mating pool with n number of chromosomes is selected using the roulette wheel selection strategy. In this technique, the chance of a chromosome for selection is proportionate with it's fitness value (Hong Qu & Alexander, 2013). The solutions in the selected mating pool are reproduced by using the single-point crossover and mutation technique. Our proposed mutation procedure is presented in Algorithm 6.

Algorithm 5: GA based Markowitz model for weight optimization

```
Input: Portfolio($p$), $data$, $cov_{mat}$, Number of generation($gen$),
       Number of chromosomes($n$)
Output: Optimized weights($W_{optmized}$)
Begin;
Initial population($i_{pop}$) with $n$ number of chromosomes is
encoded by using the probabilistic values($W$). The value of each
gene is selected within the range 0.002 to 0.15 ; // $\sum_{i=1}^{m} W_i = 1$, ∀
Chromosomes
for $g = 1$ to $gen$ do
    The fitness of each chromosome is computed by using
    Equation 5;
    Roulette wheel selection strategy is applied on $i_{pop}$ to
    constitute a mating pool($m_{pop}$) with $n$ number of
    chromosomes;
    Single point crossover is applied on $m_{pop}$; // $0.002 \le$ value
    of each gene $\le 0.15$
    Mutation is done on $m_{pop}$ by using the Algorithm 6; // $0.02 \le$
    value of each gene $\le 0.15$
    Elite selection is made to preserve 1 elite solution.
end
End;
```

In this operation, a randomly taken value is added to or subtracted from a gene. The unbiased selection of operation (addition or subtraction) is made by using a randomly taken Boolean value(opt). The addition is performed when, $opt = 0$. Otherwise, subtraction is performed. The mutation on the chromosome's overall effect is quantified by subtracting the total probabilistic value ($\sum gene$) from 1. It is represented by using $adjust$. The chromosome is then adjusted by adding $adjust$ with the gene with the lowest weight (probabilistic value) if $adjust$ is a positive value. Otherwise, $adjust$ is subtracted from the gene with the highest weight (probabilistic value). The flow diagram of the

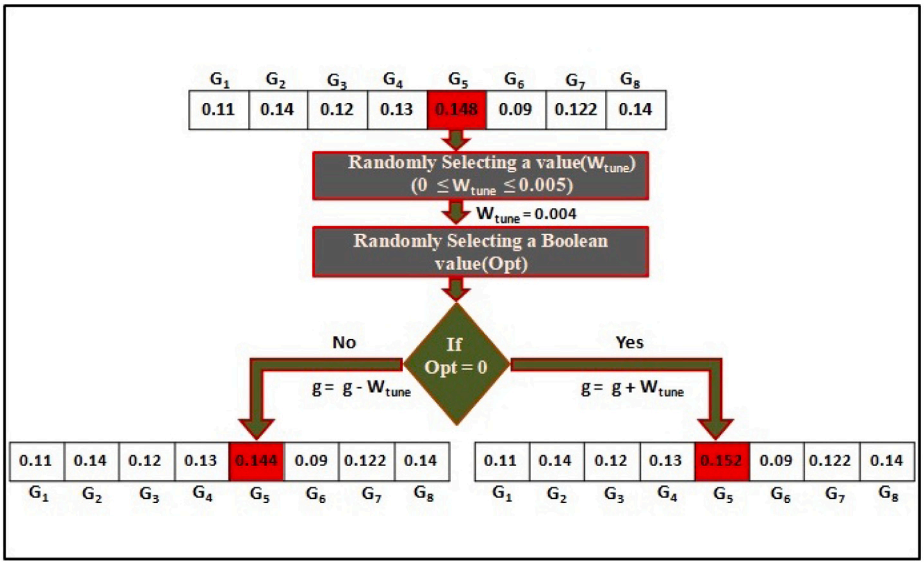


Fig. 4. Flow diagram of the proposed mutation algorithm during the weight optimization of portfolios.

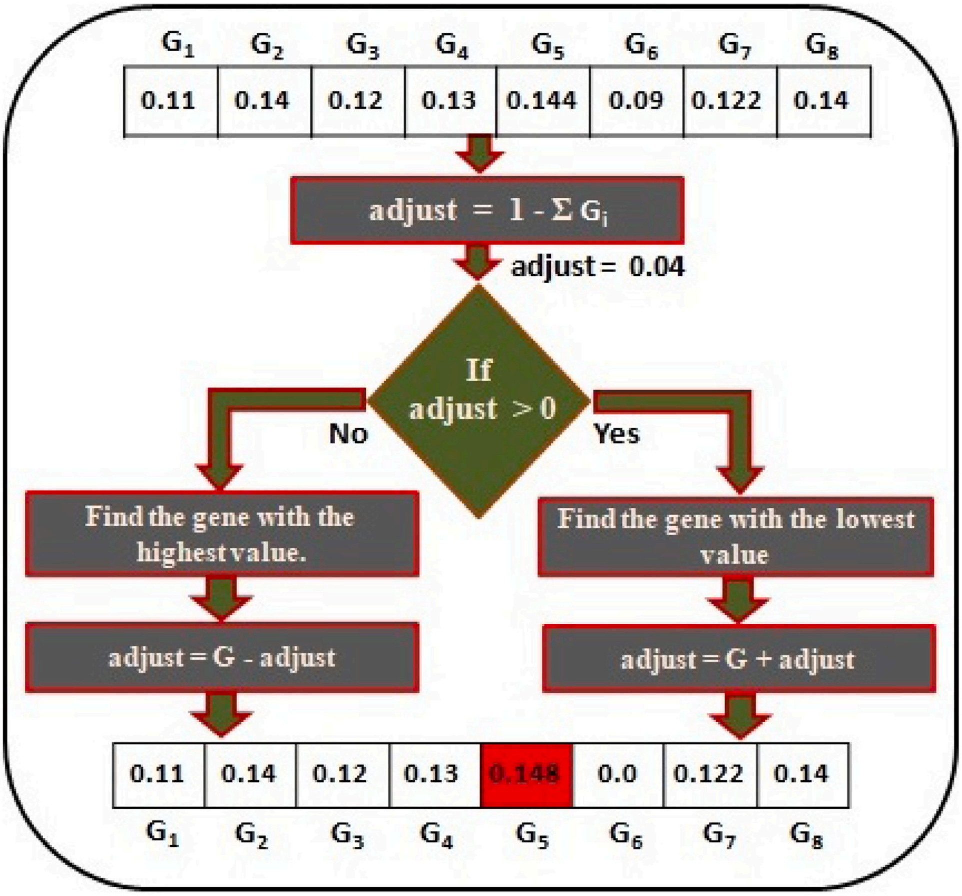


Fig. 5. Flow diagram of the weight adjustment of process in mutation.

proposed mutation technique and it's weight adjustment technique are presented in Figs. 4 and 5 respectively. After reproduction, the elite selection is made to preserve one elite solution in the mating pool for the next generation. This algorithm is executed for 200 generations. Finally, the lowest fitness value solution is selected as optimal weights for the stocks in that portfolio.

3. Results and discussion

In this section, we present the experimental results. The proposed method is implemented and tested in the system with the following configuration:

i. Editor: Spyder

Table 1
Distribution of companies across near Pareto optimal Portfolios.
Source: Result derived from the clustering output.

P1	P2	P3	P4	P5	P6	P7	P8	P9
Biocon	Biocon	Shree Cement	Hindalco	Hindalco	Biocon	Hindalco	JSW	Biocon
Naveen Fluorine	Naveen Fluorine	Dr. Reddy's	Britannia	Britannia	Naveen Fluorine	Britannia	Naveen Fluorine	Naveen Fluorine
Godrej Consumer Products	Godrej Consumer Products	Tata Motors	Kansai Nerolac	Kansai Nerolac	Godrej Consumer Products	Kansai Nerolac	Bajaj Finserv	Godrej Consumer Products
Colgate Palmolive	Colgate Palmolive	MRF	Tata Motors	Tata Motors	Thermax	Tata Motors	Colgate Palmolive	Colgate Palmolive
Siemens	Berger Paints	Britannia	Glenmark	Glenmark	Siemens	Glenmark	Pidilte	Berger Paints
HUL	Tata Motors	Bajaj Auto	Bajaj Auto	Bajaj Auto	Emami	Bajaj Auto	HUL	Emami
Graphite	Emami	United Spirits	United Spirits	United Spirits	Hero Motocorp	Grasim	Graphite	Kansai Nerolac
ICICI Bank	Aarti Industries	Bharat Forge	Bharat Forge	Bharat Forge	Kansai Nerolac	Voltas	Emami	United Spirits
Emami	Kansai Nerolac	Asian Paints	Asian Paints	Asian Paints	Grasim	Godrej Consumer Products	Kansai Nerolac	Bharat Forge
Aarti	Grasim	Bajaj Finserv	Bajaj Finserv	Thermax			Grasim	Asian Paints
Kansai Nerolac	LIC Housing Finance	Grasim	Grasim	Grasim			Supreme Industries	Bajaj Finserv
Grasim		ITC	Supreme Industries	Supreme Industries				Grasim
LIC Housing Finance								Supreme Industries

Algorithm 6: Proposed mutation technique on real-coded chromosomes during weight optimization of portfolios

```
Input: Mating pool($mpool$), $n$
Output: Updated mating pool($mpool$)
Begin;
for $c = 1$ to $\frac{n}{2}$ do
    From the $mpool$, one chromosome($C_1$) is randomly selected;
    for Each gene(g) in $C_1$ do
        A probabilistic value $M_p$ is randomly selected;
        if $M_p \leq 0.1$ then
            A random value($W_{tune}$) is randomly selected;
            // $0 \leq W_{tune} \leq 0.005$
            $opt$=randomly selecting a Boolean value;
            if $opt = 0$ then
                $g$ is mutated by adding $W_{tune}$ with it;
            end
            else
                $g$ is mutated by subtracting $W_{tune}$ from it;
            end
        end
    end
    $adjust$=$1 - \sum(genes)$;
    if $adjust \neq 0$ then
        if $adjust > 0$ then
            The gene($g$) with the smallest value(weight) is identified;
            $g = g + adjust$;
        end
        else
            The gene($g$) with the highest value(weight) is identified;
            $g = g - adjust$;
        end
    end
end
End;
```

ii. Programming language: Python 3.7.3
iii. Operating System: Windows-10
iv. RAM: 12 GB

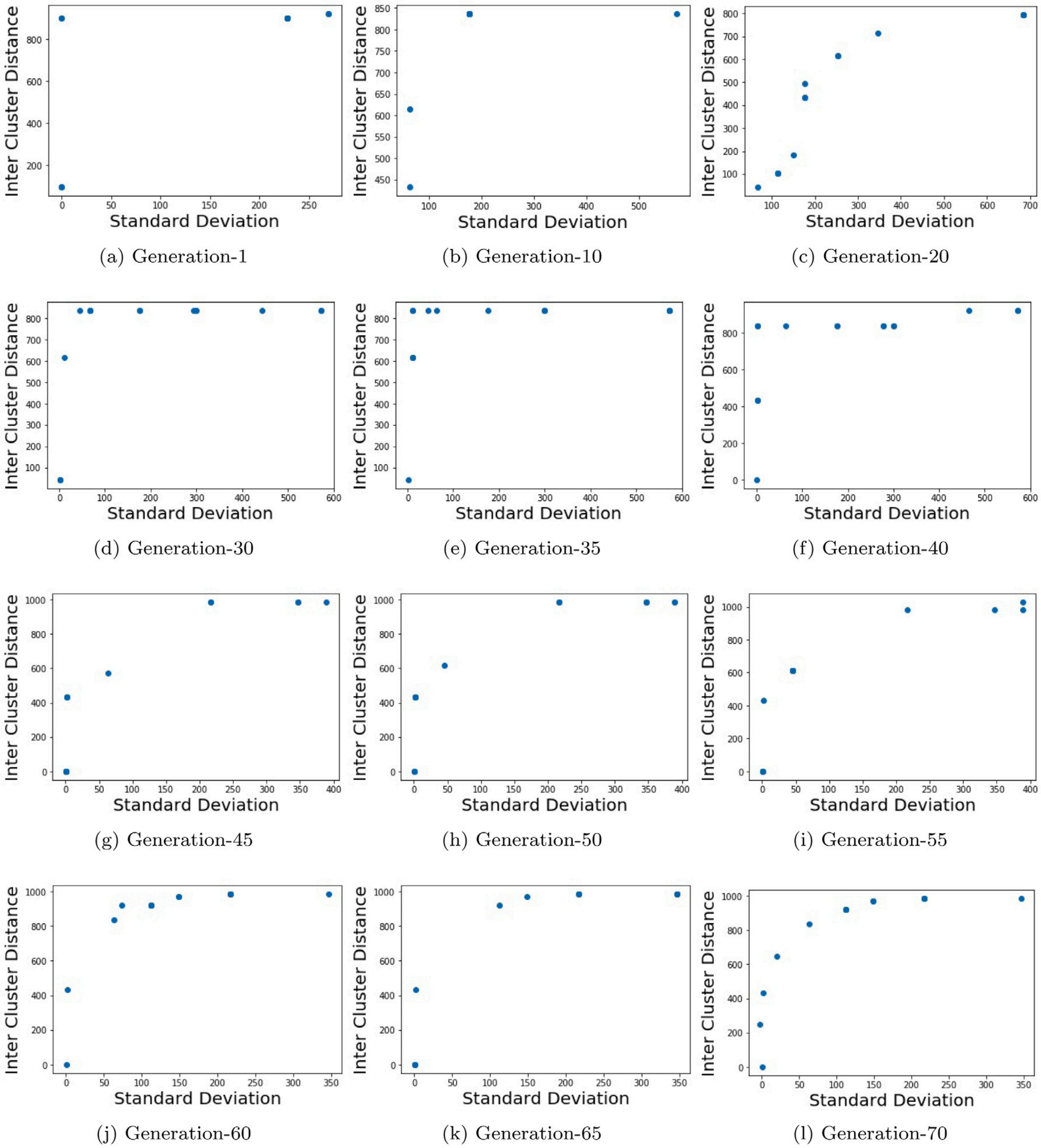
v. Processor: Intel(R)-$Core^{TM}$-i5-L16G7-3.0 GHz.

Our dataset consists of 80 Indian companies that are listed on the National Stock Exchange (NSE) and Bombay Stock Exchange (BSE). These are consistently profit-making, dividend-paying, large-cap companies that are highly liquid and have a large daily turnover. Data on prices of these stocks have been collected from 4.1.2010 to 31.1.2020 from the Metastock database. The proposed method is applied to this dataset to optimize multiple portfolios in a single execution. The algorithm is iterated for 70 generations. To have proper diversification of companies in a portfolio, in the clustering algorithm itself, the number of companies belonging to a portfolio was restricted by the constraint $10 \leq N_i \leq 14$ where N_i is the number of companies in the ith portfolio. This is known as the cardinality constraint and is imposed in the portfolio optimization stage in the literature. We have incorporated this constraint by considering the upper limit of the length of all chromosomes as 14 and the lower limit as 10. The implication of the constraint is to have diversity and not have funds concentrated in a few stocks only. On the other hand, we do not want too many stocks in a portfolio as we know that risk cannot be reduced by increasing the number of stocks beyond a point. The size of the initial mating pool was 20. Generation wise $front-1$ solutions obtained during the vertical clustering are represented in Fig. 6. We can observe that the $front-1$ solutions target the Pareto front throughout the generations. Finally, nine well-diversified $front-1$ solutions or portfolios are obtained after the 70th iteration of the algorithm.

The distribution of companies across these portfolios are given in Table 1. We can observe that all the portfolios satisfied the aforementioned cardinality constraint. The performance of the algorithm during clustering is measured by using Davies Bouldin (DB) index (Pakhira et al., 2004), Dunn index (DI) (Rivera-Borroto et al., 2012) and Silhouette Score (SS) (LaHaye et al., 2019). We have also measured the performance of K-Means (Kanungo et al., 2002), Fuzzy-C-Means (Bezdek et al., 1984), ISODATA (Ball & Hall, 1965) and DBSCAN (Birant & Kut, 2007) algorithms on this dataset. It is presented in Table 2. We can observe that the proposed method outperforms all these clustering algorithms in terms of DI, DB and SS.

In the portfolio optimization exercise, we have imposed an additional restriction that the maximum exposure to a stock cannot exceed 15% of the total funds deployed. This is consistent with the exposure norms laid down by the Reserve Bank of India (RBI) for lending to companies.

Our near Pareto optimal portfolios have representation from different industries like the FMCG sector, the auto sector, the capital goods sector, the metal sector, the financial services sector, and the pharma sector. The portfolios are well diversified. Further, our clustering exercise, conducted over a considerable period, has repeatedly



(a) Generation-1

(b) Generation-10

(c) Generation-20

(d) Generation-30

(e) Generation-35

(f) Generation-40

(g) Generation-45

(h) Generation-50

(i) Generation-55

(j) Generation-60

(k) Generation-65

(l) Generation-70

Fig. 6. $Front - 1$ solutions over the 70 iterations obtained during the execution of the proposed method.

Table 2
Comparison of the internal clustering performance between K-Means, FCM, ISODATA, DBSCAN and proposed method.

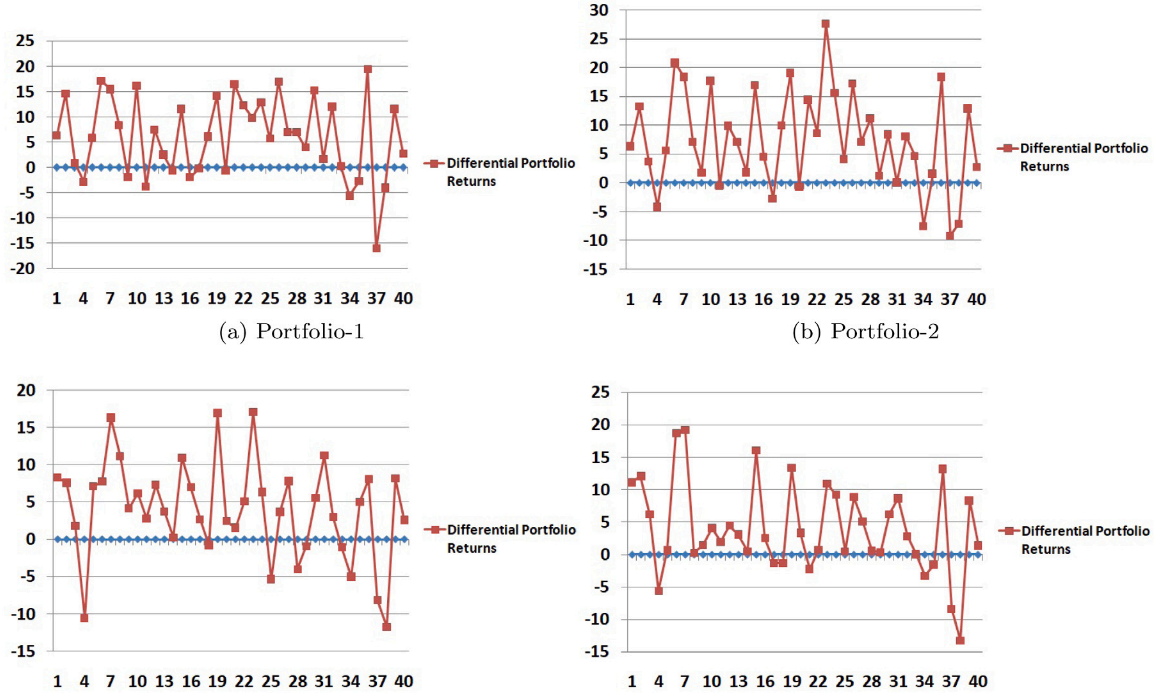
Algorithm	Dunn score	DB score	Silhouette score
K-Means	0.92637827430	0.42148876270	0.6718640320
ISODATA	1.04148900235	0.2003821979	0.691539792
FCM	1.03189799184	0.19422189210	0.752039410
DBSCAN	1.213874820	0.1953826328	0.769290461
Proposed method	**1.351879012895**	**0.0602403915**	**0.922601228520**

chosen a few companies in different portfolios. This implies that these companies, across industries, can be chosen by fund managers as they have emerged from the algorithm's optimality requirements. It can also be checked that analysts have recommended these stocks for acquisition purposes.

From the point of view of the performance of the portfolios, shown in Tables 3 and 4, over the entire period, all our 9 portfolios have generated returns greater than investing in the benchmark portfolio, Nifty. It is graphically represented in Figs. 7 and 8. This indicates that all our portfolios, generated from the clustering algorithm, have performed better than a benchmark portfolio.

From Fig. 9 we can observe that Portfolio 2 has performed the best as it has generated the highest returns among the near Pareto optimal portfolios. Portfolio 9 is the next best portfolio. Continuous rebalancing of the constituent stocks through the realignment of the weights has made this possible. This continuous realignment of weights is of practical use and has not been performed in the literature. However, we can observe that in some quarters (highlighted with green color in Table 3), Nifty, the benchmark portfolio, has generated greater returns than some of our portfolios.

Fig. 10 presents the evolution of the Indian stock market index (Nifty) during the time period of the study. While an upward trend



(a) Portfolio-1

(b) Portfolio-2

(c) Portfolio-3

(d) Portfolio-4

Fig. 7. Quarter-wise differential portfolio returns (DIF) over Nifty returns for portfolio 1 to 4.

is observed, there has been periods of sharp decrease in the index (market corrections). This is more precisely reflected in Figure Fig. 11 which presents the movement in quarterly average Index (Nifty) returns (AVNR) and quarterly average Index (Nifty) volatility (AVNV) from 4/1/2010 to 31/1/2020. It may be observed that in quarters of high volatility in index returns, the index returns have been negative. If we relate this observation with our results in Table 3, we can see that in quarters 7, 8, 15, 22, 23, 25, 28, 36 and 39 (marked by arrow in Figure Fig. 11) where quarterly volatility in index returns is high, all our portfolios, with quarterly realignment, has generated greater returns than the index. This further validates our approach that the method of portfolio choice and realignment has been successful during volatile markets. The index volatility has been on the lower side in the quarters where the index has generated better returns than all the portfolios or some of them.

The movement in the market index represents overall market sentiment and reflects the macroeconomics of the time. The interested reader can check that the increase in market volatility in the beginning period of the study, starting from 2010, reflected a reduction in government expenditure, which had gone up significantly to boost the Indian economy after the sub-prime crisis. An increase in market volatility in the later period, starting from around 2018, reflects the slowing down of the Indian economy, which continued till the beginning of the pandemic, and got accentuated thereafter.

Our framework of portfolio optimization with horizontal and vertical clustering and dynamic realignment of weights, besides being efficient, is realistic and incorporates changes in the macroeconomic environment. This would be of help to fund managers. For further improvement, in our future work, we will not only be re-balancing the weights, but also choose the constituent stocks through a dynamic process.

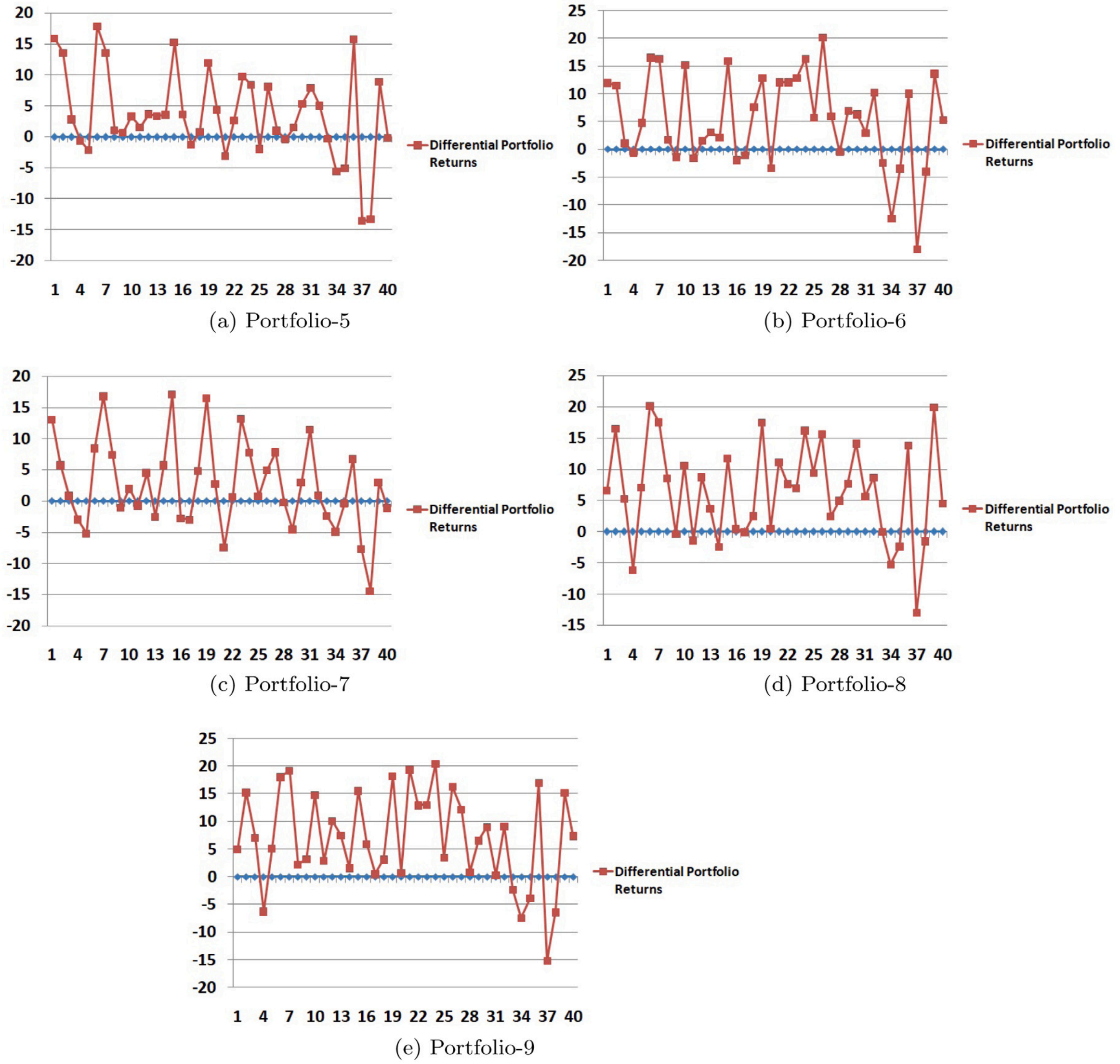
4. Conclusion & future scope

The literature on multi-objective portfolio optimization has primarily focused on the complexities of multiple objectives and multiple constraints. They have provided answers through novel algorithms that have given efficient solutions based on various criteria they have used. Some have emphasized reduced computing time; some have made comparisons with similar approaches; some have calculated GD, IGD, Hyper volume, and epsilon metrics. In contrast, some have considered results in vis-a-vis performance of benchmark portfolios. At times similar data sets have been used, while some have considered countrywide stock market data.

While the formulation of the objective functions and the constraints are indeed novel, at times, some of the objectives like minimization of the P/E multiple, or RSI, or the difference between portfolio variance and mean may be debated upon. In the Indian context, where many public sector undertakings are listed and traded, they all have low P/E ratios but are not there in any portfolio of mutual funds (except PSU funds). Low P/E can signal undervaluation but can also signal a lack of future growth.

Our contribution in this field of portfolio optimization lies in forming the initial portfolios through a method of automatic horizontal and vertical clustering. Based on a single metric, returns per unit of risk, clustering is done on the sample set of companies for each day, and then again clustering is done on the cluster centers, thus obtained over the entire period of ten years. This is done through a novel clustering algorithm where a cardinality condition and exposure limits are imposed on each stock. It may be mentioned that the existing literature incorporates these constraints in the optimization stage. We have proposed a new approach to designing a variable-length NSGA-II. We have also proposed a robust and realistic crossover and mutation technique. The above optimization exercise generated nine near Pareto optimal portfolios.

Our next contribution lies in realigning the weights of each stock in each of the portfolios, quarter-wise. We proposed a new approach of designing the Markowitz model involving a single objective function of minimization of portfolio variance, subject to a threshold level of portfolio return and no short selling. We perform this optimization quarter-wise to determine the weights to realign the portfolios (the weights of the nine portfolios for forty quarters of each portfolio



(a) Portfolio-5

(b) Portfolio-6

(c) Portfolio-7

(d) Portfolio-8

(e) Portfolio-9

Fig. 8. Quarter-wise differential portfolio returns (DIF) over Nifty returns for portfolio 5 to 9.

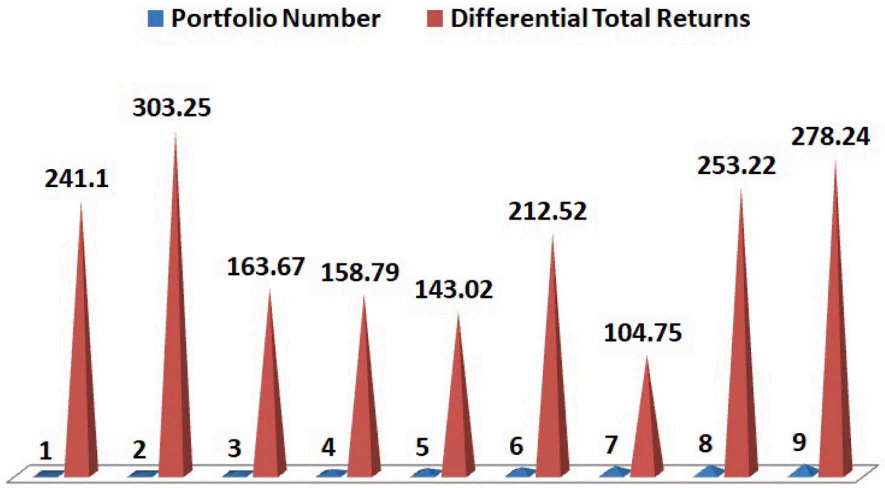


Fig. 9. Bar diagram to represent quarter-wise total excess returns over the entire period over Nifty returns for portfolio-1 to 9 (Assuming Rs. 1000 is invested in the portfolios in the beginning of each quarter. Figures in Rupees).

Table 3
Quarter-wise differential portfolio returns (DIF) over Nifty returns.
Source: Results from the dynamic portfolio optimization exercise.

Quarter	DIF 1	DIF 2	DIF 3	DIF 4	DIF 5	DIF 6	DIF 7	DIF 8	DIF 9
1	6.29	6.35	8.30	11.11	15.94	11.98	13.02	6.58	4.97
2	14.62	13.24	7.63	12.09	13.57	11.50	5.76	16.54	15.27
3	0.82	3.69	1.84	6.16	2.80	1.09	0.87	5.26	7.08
4	−2.86	−4.17	−10.52	−5.60	−0.68	−0.58	−2.95	−6.16	−6.24
5	5.88	5.58	7.14	0.66	−2.16	4.81	−5.22	7.07	5.11
6	17.18	20.84	7.83	18.74	17.86	16.54	8.47	20.15	18.03
7	15.48	18.41	16.35	19.17	13.59	16.34	16.77	17.52	19.16
8	8.33	7.12	11.16	0.26	1.05	1.67	7.41	8.57	2.27
9	−1.93	1.80	4.16	1.43	0.65	−1.40	−1.08	−0.35	3.22
10	16.25	17.70	6.24	4.07	3.31	15.21	1.97	10.64	14.82
11	−3.81	−0.53	2.82	1.97	1.52	−1.55	−0.73	−1.43	2.96
12	7.47	9.92	7.37	4.48	3.68	1.59	4.51	8.72	10.14
13	2.51	7.08	3.74	3.07	3.37	3.09	−2.51	3.64	7.45
14	−0.61	1.86	0.29	0.50	3.54	2.10	5.77	−2.46	1.63
15	11.59	16.93	11.00	16.03	15.27	15.93	17.07	11.69	15.54
16	−1.96	4.55	7.05	2.50	3.57	−1.92	−2.78	0.42	5.92
17	−0.22	−2.72	2.72	−1.34	−1.29	−1.02	−3.01	−0.10	0.54
18	6.10	9.97	−0.74	−1.31	0.80	7.65	4.81	2.48	3.14
19	14.14	19.06	16.97	13.37	11.96	12.85	16.50	17.50	18.25
20	−0.71	−0.66	2.47	3.31	4.33	−3.35	2.75	0.49	0.77
21	16.51	14.44	1.58	−2.28	−3.18	12.10	−7.39	11.10	19.38
22	12.33	8.64	5.17	0.75	2.60	12.10	0.64	7.63	12.91
23	9.78	27.59	17.11	10.96	9.75	12.92	13.20	6.97	13.03
24	12.89	15.58	6.38	9.21	8.37	16.36	7.75	16.19	20.41
25	5.67	4.09	−5.31	0.48	−2.00	5.73	0.80	9.37	3.51
26	16.94	17.26	3.72	8.83	8.08	20.16	4.95	15.67	16.32
27	7.01	7.15	7.86	5.13	0.96	5.98	7.82	2.43	12.16
28	6.92	11.25	−3.98	0.55	−0.44	−0.50	−0.19	4.93	0.84
29	3.96	1.27	−0.86	0.36	1.48	6.94	−4.54	7.66	6.57
30	15.28	8.44	5.54	6.20	5.37	6.32	3.02	14.13	9.00
31	1.61	0.08	11.30	8.67	7.90	2.95	11.48	5.57	0.28
32	12.03	8.07	3.03	2.80	5.03	10.30	0.93	8.67	9.16
33	0.25	4.68	−1.01	0.03	−0.29	−2.45	−2.42	−0.02	−2.30
34	−5.64	−7.51	−4.98	−3.28	−5.64	−12.44	−4.89	−5.28	−7.40
35	−2.74	1.57	5.06	−1.52	−5.13	−3.45	−0.37	−2.39	−3.84
36	19.46	18.41	8.09	13.17	15.77	10.05	6.75	13.83	17.01
37	−16.01	−9.11	−8.07	−8.41	−13.60	−17.97	−7.65	−12.97	−15.10
38	−4.05	−7.06	−11.69	−13.24	−13.36	−4.01	−14.38	−1.54	−6.35
39	11.58	12.94	8.23	8.31	8.86	13.63	2.96	19.96	15.20
40	2.74	9.47	2.67	1.40	−0.20	5.28	−1.12	4.54	7.39

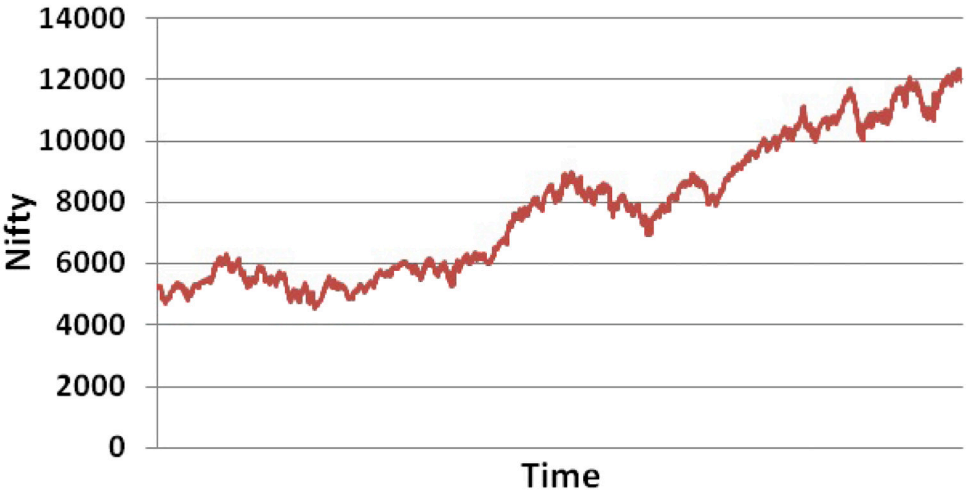


Fig. 10. Movement in the Indian stock market index (Nifty) during 4/1/2010 to 31/1/2020.

Table 4
Quarter-wise total excess returns over the entire period over nifty returns (Assuming Rs. 1000 is invested in the portfolios in the beginning of each quarter. Figures in Rupees).

DIF 1	DIF 2	DIF 3	DIF 4	DIF 5	DIF 6	DIF 7	DIF 8	DIF 9
241.10	303.25	163.67	158.79	143.02	212.52	104.75	253.22	278.24

are available on request). This methodology would greatly help fund managers.

The next obvious question that arises is why not also determine new sets of companies for each quarter. This way, both companies and weights would change. This is the subject matter of our ongoing work. We are also working on technical indicator based clustering for the choice of companies. As given in the introduction, we came

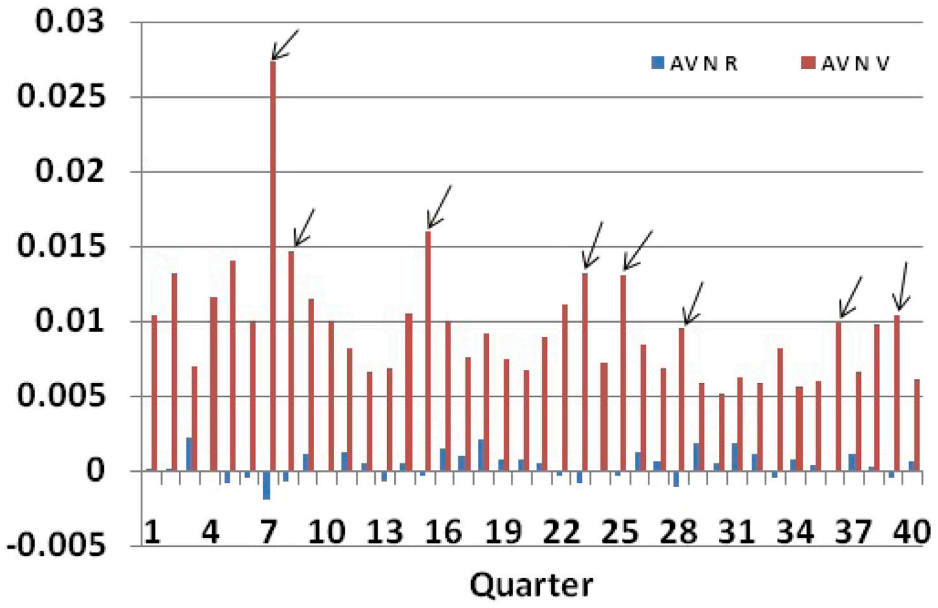


Fig. 11. Movement in quarterly average Index (Nifty) returns (AVNR) and average Index (Nifty) volatility (AVNV) during 4/1/2010 to 31/1/2020.

across papers that have considered technical indicators in the objective function. We want to introduce these indicators in the clustering stage itself.

CRediT authorship contribution statement

Ramen Pal: Methodology, Software, Data curation, Validation, Writing – original draft. **Tamal Datta Chaudhuri:** Conceptualization, Supervision, Data analysis, Writing – review & editing. **Somnath Mukhopadhyay:** Software, Visualization, Investigation, Formal analysis, Reviewing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Ball, G., & Hall, D. (1965). *Isodata: A novel method of data analysis and pattern classification: Technical report*, Menlo Park: Stanford Research Institute.

Ban, G.-Y., El Karoui, N., & Lim, A. E. B. (2018). Machine learning and portfolio optimization. *Management Science*, *64*(3), 1136–1154.

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, *10*(2), 191–203.

Birant, D., & Kut, A. (2007). St-dbscan: An algorithm for clustering spatial–temporal data. *Data and Knowledge Engineering*, *60*(1), 208–221, Intelligent Data Mining.

Chen, W., Li, S., & Zhang, J. (2020). A comprehensive model for fuzzy multi-objective portfolio selection based on dea cross-efficiency model. *Soft Computing*, *24*, 2515–2526.

Dang, L. M., Sadeghi-Niaraki, A., Huynh, H. D., Min, K., & Moon, H. (2018). Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. *IEEE Access*, *6*, 55392–55404.

Darsha Panwar, M. J., & Srivastava, N. (2018). Optimization of risk and return using fuzzy multiobjective linear programming. *Advances in Fuzzy Systems*, *2018*.

Dasgupta, S., Das, S., Abraham, A., & Biswas, A. (2009). Adaptive computational chemotaxis in bacterial foraging optimization: An analysis. *IEEE Transactions on Evolutionary Computation*, *13*(4), 919–941.

Dawyndt, P., De Meyer, H., & Baets, B. D. (2005). The complete linkage clustering algorithm revisited. *Soft Computing*, *9*, 1433–7479.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, *1*(4), 28–39.

Dreżewski, R., & Doroz, K. (2017). An agent-based co-evolutionary multi-objective algorithm for portfolio optimization. *Symmetry*, *9*(9).

Ertenlice, O., & Kalayci, C. B. (2018). A survey of swarm intelligence for portfolio optimization: Algorithms and applications. *Swarm and Evolutionary Computation*, *39*, 36–52.

Gupta, P., Mehlawat, M. K., & Saxena, A. (2010). A hybrid approach to asset allocation with simultaneous consideration of suitability and optimality. *Information Sciences*, *180*(11), 2264–2285.

Hong Qu, K. X., & Alexander, T. (2013). An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. *Neurocomputing*, *120*, 509–517, Image Feature Detection and Description.

Jalota, H., & Thakur, M. (2017). Genetic algorithm designed for solving portfolio optimization problems subjected to cardinality constraint. *International Journal of System Assurance Engineering and Management*, *9*, 294–305.

Kalayci, C. B., Ertenlice, O., & Akbay, M. A. (2019). A comprehensive review of deterministic models and applications for mean–variance portfolio optimization. *Expert Systems with Applications*, *125*, 345–368.

Kalayci, C. B., Ertenlice, O., Akyer, H., & 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.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 881–892.

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (abc) algorithm. *Applied Soft Computing*, *8*(1), 687–697.

Karimkashi, S., & Kishk, A. A. (2010). Invasive weed optimization and its features in electromagnetics. *IEEE Transactions on Antennas and Propagation*, *58*(4), 1269–1278.

Kaucic, M., Moradi, M., & Mirzazadeh, M. (2019). Portfolio optimization by improved nsga-ii and spea 2 based on different risk measures. *Financial Innovation*, *5*(26).

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International conference on neural networks, Vol. 4*, (pp. 1942–1948).

Koratamaddi, P., Wadhwani, K., Gupta, M., & Sanjeevi, S. G. (2021). Market sentiment-aware deep reinforcement learning approach for stock portfolio allocation. *Engineering Science and Technology, An International Journal*, *24*(4), 848–859.

Kumar, D., & Mishra, K. (2017). Portfolio optimization using novel co-variance guided artificial bee colony algorithm. *Swarm and Evolutionary Computation*, *33*, 119–130.

LaHaye, N., Ott, J., Garay, M. J., El-Askary, H. M., & Linstead, E. (2019). Multi-modal object tracking and image fusion with unsupervised deep learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *12*(8), 3056–3066.

Li, S., Ning, K., & Zhang, T. (2021). Sentiment-aware jump forecasting. *Knowledge-Based Systems*, *228*, Article 107292.

Li, X., & Wu, P. (2021). Stock price prediction incorporating market style clustering. *Cognitive Computation*.

Li, J., & Xu, J. (2013). Multi-objective portfolio selection model with fuzzy random returns and a compromise approach-based genetic algorithm. *Information Sciences*, *220*, 507–521, Online Fuzzy Machine Learning and Data Mining.

Liagkouras, K. (2018). A new three-dimensional encoding multiobjective evolutionary algorithm with application to the portfolio optimization problem. *Knowledge-Based Systems*, *163*, 186–203.

Liagkouras, K., & Metaxiotis, K. (2015). Efficient portfolio construction with the use of multiobjective evolutionary algorithms: Best practices and performance metrics. *International Journal of Information Technology and Decision Making*, *14*(3), 535–564.

Liagkouras, K., & Metaxiotis, K. (2017). Handling the complexities of the multi-constrained portfolio optimization problem with the support of a novel moea. *Journal of the Operational Research Society*, *69*(10), 1609–1627.

Lwin, K. T., Qu, R., & MacCarthy, B. L. (2017). Mean-var portfolio optimization: A nonparametric approach. *European Journal of Operational Research*, *260*(2), 751–766.

Macedo, L. L., Godinho, P., & Alves, M. J. (2017). Mean-semivariance portfolio optimization with multiobjective evolutionary algorithms and technical analysis rules. *Expert Systems with Applications*, *79*, 33–43.

Masmoudi, M., & Abdelaziz, F. B. (2017). Portfolio selection problem: a review of deterministic and stochastic multiple objective programming models. *Annals of Operations Research*, *267*, 335–352.

Meghwani, S. S., & Thakur, M. (2017). Multi-criteria algorithms for portfolio optimization under practical constraints. *Swarm and Evolutionary Computation*, *37*, 104–125.

Mehlawat, M. K., & Gupta, P. (2014). Fuzzy chance-constrained multiobjective portfolio selection model. *IEEE Transactions on Fuzzy Systems*, *22*(3), 653–671.

Mehlawat, M. K., Kumar, A., Yadav, S., & Chen, W. (2018). Data envelopment analysis based fuzzy multi-objective portfolio selection model involving higher moments. *Information Sciences*, *460–461*, 128–150.

Mitra Thakur, G. S., Bhattacharyya, R., & Sarkar (Mondal), S. (2018). Stock portfolio selection using dempster shafer evidence theory. *Journal of King Saud University - Computer and Information Sciences*, *30*(2), 223–235.

Mukhopadhyay, S., & Chaudhuri, T. D. (2019). *Studies in computational intelligence*: *vol. 687*, *Advances in intelligent computing*. Singapore: Springer, chapter Different Length Genetic Algorithm-Based Clustering of Indian Stocks for Portfolio Optimization.

Mukhopadhyay, S., Paul, M., Pal, R., & De, D. (2020). Tea leaf disease detection using multi-objective image segmentation. *Multimedia Tools and Applications*, 1573–7721.

Ni, Q., Yin, X., Tian, K., & Zhai, Y. (2017). Particle swarm optimization with dynamic random population topology strategies for a generalized portfolio selection problem. *Natural Computing*, *16*(1), 31–44.

Paiva, F. D., Cardoso, R. T. N., Hanaoka, G. P., & Duarte, W. M. (2019). Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications*, *115*, 635–655.

Pakhira, M. K., Bandyopadhyay, S., & Maulik, U. (2004). Validity index for crisp and fuzzy clusters. *Pattern Recognition*, *37*(3), 487–501.

Park, H., & Jun, C. (2009). A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, *36*(2, Part 2), 3336–3341.

Picasso, A., Merello, S., Ma, Y., Oneto, L., & Cambria, E. (2019). Technical analysis and sentiment embeddings for market trend prediction. *Expert Systems with Applications*, *135*, 60–70.

Qiu, X., Suganthan, P. N., & Amaratunga, G. A. J. (2019). Fusion of multiple indicators with ensemble incremental learning techniques for stock price forecasting. *Journal of Banking and Financial Technology*, *3*(1), 33–42.

Qu, B. Y., Zhou, Q., Xiao, J. M., Liang, J., & Suganthan, P. N. (2017). Large-scale portfolio optimization using multiobjective evolutionary algorithms and preselection methods. *Mathematical Problems in Engineering*, *2017*.

Quintana, D., Denysiuk, R., Garcia-Rodriguez, S., & Gaspar-Cunha, A. (2017). Portfolio implementation risk management using evolutionary multiobjective optimization. *Applied Sciences*, *7*.

Rezaei Pouya, A., Solimanpur, M., & Jahangoshai Rezaee, M. (2016). Solving multi-objective portfolio optimization problem using invasive weed optimization. *Swarm and Evolutionary Computation*, *28*, 42–57.

Rivera-Borroto, O. M., Rabassa-Gutiérrez, M., Grau-Ábalo, R., Marrero-Ponce, Y., & Vega, J. (2012). Dunn's index for cluster tendency assessment of pharmacological data sets. *Canadian Journal of Physiology and Pharmacology*, *90*, 425–433.

Saranya, K., & Prasanna, P. (2014). Portfolio selection and optimization with higher moments: Evidence from the indian stock market. *Asia-Pacific Financial Markets*, *21*, 133–149.

Shu-Chuan, C., Tsai, P.-w., & Pan, J.-S. (2006). Cat swarm optimization. In Q. Yang, & G. Webb (Eds.), *PRICAI 2006: Trends in artificial intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Strumberger, I., Bacanin, N., & Tuba, M. (2016). Constrained portfolio optimization by hybridized bat algorithm. In *2016 7th international conference on intelligent systems, modelling and simulation (ISMS)* (pp. 83–88).

Tan, Y., & Zhu, Y. (2010). Fireworks algorithm for optimization. In *Advances in swarm intelligence* (pp. 355–364). Berlin, Heidelberg: Springer Berlin Heidelberg.

Valle-Cruz, D., Fernandez-Cortez, V., Lòpez-Chau, A., & Sandoval-Almazàn, R. (2021). Does twitter affect stock market decisions?financial sentiment analysis in pandemic seasons: A comparative study of h1n1 and covid-19. *Cognitive Computation*.

Wang, B., Li, Y., Wang, S., & Watada, J. (2018). A multi-objective portfolio selection model with fuzzy value-at-risk ratio. *IEEE Transactions on Fuzzy Systems*, *26*(6), 3673–3687.

Wei Yue, Y. W., & Dai, C. (2015). An evolutionary algorithm for multiobjective fuzzy portfolio selection models with transaction cost and liquidity. *Mathematical Problems in Engineering*, *2015*.

Xing, F. Z., Cambria, E., & Zhang, Y. (2019). Sentiment-aware volatility forecasting. *Knowledge-Based Systems*, *176*, 68–76.

Yang, X.-S., & Gandomi, A. (2012). Bat algorithm: A novel approach for global engineering optimization. *Engineering Computations*, *29*.

Yuanyuan Zhang, X. L., & Guo, S. (2018). Portfolio selection problems with markowitz's mean-variance framework: a review of literature. *Fuzzy Optimization and Decision Making*, *17*, 125–158.

Zhang, Y., Yan, B., & Aasma, M. (2020). A novel deep learning framework: Prediction and analysis of financial time series using ceemd and lstm. *Expert Systems with Applications*, *159*, Article 113609.

Zhang, X., Zhang, Y., Wang, S., Yao, Y., Fang, B., & Yu, P. S. (2018). Improving stock market prediction via heterogeneous information fusion. *Knowledge-Based Systems*, *143*, 236–247.