INDEX FUND PORTFOLIO OPTIMIZATION USING HEURISTIC ALGORITHM

Viswarup Misra

INDEX FUND PORTFOLIO OPTIMIZATION USING HEURISTIC ALGORITHM

Report submitted to

Indian Institute of Technology, Kharagpur

for the award of the degree

of

Master of Technology

in Financial Engineering

by

Viswarup Misra

under the guidance
of

Prof. Geetanjali Panda
Department of Mathematics
Indian Institute of Technology, Kharagpur

INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR
MAY,2016

DECLARATION

I certify that

a. The work contained in this report is original and has been done by me under the guidance of my supervisor.
b. The work has not been submitted to any other Institute for any degree or diploma.
c. I have followed the guidelines provided by the Institute in preparing the report.
d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Signature of the Student

CERTIFICATE

This is to certify that the Dissertation **Report** entitled, "**Index fund portfolio optimization using heuristic algorithm**" submitted by **Mr. Viswarup Misra** to Indian Institute of Technology, Kharagpur, India, is a record of bonafide project work carried out by him under my supervision and guidance and is worthy of consideration for the award of the degree of Master of Technology in Financial Engineering.

____

Superviser

Date:

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGMENT

I would like to express my humble indebtedness and deep sense of sincere gratitude to my project supervisor Dr. Geetanjali Panda, Professor, Department of Mathematics for her supervision, constructive guidance, nurturing encouragement and support at every stage of my work. I would also like to thank our program coordinator Prof. Prabina Rajib for her efforts.

VISWARUP MISRA
11AG3FP09
Department of Agricultural & Food Engineering
IIT Kharagpur

Date: 2nd May, 2016

Abstract

This study proposes a portfolio optimization scheme with the help of heuristic algorithm for index fund management. Index fund is one of popular strategies in portfolio management that aims at matching the performance of the benchmark index such as the S&P 500 in New York and the FTSE 100 in London as closely as possible. This strategy is taken by fund managers particularly when they are not sure about outperforming the market and adjust themselves to average performance. Recently, it is noticed that the performances of index funds are better than those of many other actively managed mutual. The main objective of this study is to report that index fund could improve its performance greatly with the proposed GA portfolio scheme, which will be demonstrated for index fund designed to track Nifty 50 and BSE S&P 500.

Keywords: Index fund, Genetic Algorithm, Nifty 50, BSE S&P 500, optimization.

Chapter 1 Introduction

Index funds are popular investment tools being used in modern portfolio management. Index funds are designed to mimic the behavior of the given benchmark market indices (e.g. the S&P 500 in New York, the FTSE 100 in London, the KOSPI 200 in Seoul, etc.). Thus, index funds are generally regarded as relatively stable and efficient investment tool compared with other mutual funds (Jensen, 1968; Sharpe, 1966). The index fund strategy is based on the concept of the passive investment management. There are several interesting papers reporting the superior performance of the index funds compared with other actively managed portfolios (Elton, Gruber, & Blake, 1996; Gruber, 1996; Malkiel, 1995). In addition to the performances of index funds in terms of risk and return, index funds are also considered cost effective investment tool in the capital market (Hogan, 1994).

Fund management involves the investment of many millions of dollars (or their equivalent) in equities (stocks, shares) of companies quoted on the world's stock markets. The funds for investment are typically provided by pension contributions, insurance premiums and savings. The objective for fund managers is to provide a combination of capital growth and income over the medium to long-term. The basic strategies adopted by fund managers can be broadly classified as

(a) Active management, where the fund managers have a high degree of flexibility and attempt to ‘‘pick winners'', stocks whose values are going to outperform (over time) other stocks. The assumption underlying this strategy is that fund managers can, through their expertise and judgement, add value though choosing high performing stocks and/or by the timing of their buy/sell decisions.

(b) Passive management, where the fund managers have much less flexibility and their role is to conform to a closely defined set of criteria. Common criteria are that the fund should achieve approximately the same return as a specified market index (such as the S&P 500 in New York, or the FTSE 100 in London) through investment in an appropriately selected set of stocks that are present in the index.

Market indices that can be tracked are not only available for national stock markets, but are also available to reflect regional stock market performances (e.g. S&P Europe 350) and even global performance (e.g. FTSE All-World Index).

Active and passive fund management strategies have their respective strengths and weaknesses:

(a) Active management has high fixed costs (associated with payments to the management team) and the frequent trading involved in stock picking means high transaction costs.

If all goes well these costs will be offset by the returns obtained.

(b) Passive management has lower fixed costs and lower transaction costs, but has the disadvantage that if the market (as represented by the index) falls, so inevitably will the return obtained from an index fund.

In active management an investor is exposed to both market and company risk, whilst in passive management an investor is only exposed to market risk.

In recent years, both in the USA and in Europe, passive management (and in particular index tracking) has been receiving a much higher profile for two reasons:

(1) Historical empirical analysis has revealed that: (a) Whilst the best of the actively managed funds outperform the market in any particular year, over the long-term the majority of such funds do not (e.g. in the UK in 1998 only one quarter of actively managed funds outperformed their comparative index over a five year period).(b) An actively managed fund which outperforms the market one year may fail to do so in subsequent years (e.g. in the UK many funds that performed well in 1992 had fallen to bottom quartile positions by 1998).

(2) As stock markets (and so their indices) have historically risen in the long-term it has become clear that reasonable returns can be obtained without incurring the extra risks associated with active management.

Consider now a passively managed fund that intends to track a single given index. This fund could purchase all of the stocks that make up the index (in appropriate quantities) and so perfectly reproduce the index. This approach is known as full (or complete) replication. Whilst simple, both conceptually and computationally, full replication has a number of disadvantages:

(1) Certain stocks that are in the index may be held (proportionally) in very small quantities. For example if the number of distinct stocks included in the index is large this inevitably happens. This may be administratively inconvenient and, because there may be a restricted market for such stocks, they can be expensive to buy.

(2) When the underlying composition of the index is revised the holdings of all stocks will typically need to be changed to reflect their new weightings in the index. Index revisions can

occur for a number of reasons. For example, one company may grow sufficiently large to merit inclusion in the index, and this may involve excluding another company from the index. Mergers are another reason why indices are revised.

1.1 GENETIC ALGORITHM

A genetic algorithm (GA) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. The algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution.

One can apply genetic algorithm to solve problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non differentiable, stochastic, or highly nonlinear.

In a genetic algorithm, a population of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible.

The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

A typical genetic algorithm requires: a genetic representation of the solution domain, a fitness function to evaluate the solution domain. A standard representation of each candidate solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. Tree-like representations are explored in genetic programming and graph-form representations are explored in evolutionary programming; a mix of both linear chromosomes and trees is explored in gene expression programming.

Once the genetic representation and the fitness function are defined, a GA proceeds to initialize a population of solutions and then to improve it through repetitive application of the mutation, crossover, inversion and selection operators.

Initialization

The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Often, the initial population is generated randomly, allowing the entire range of possible solutions (the search space). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

Selection

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as the former process may be very time-consuming.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. For instance, in the knapsack problem one wants to maximize the total value of objects that can be put in a knapsack of some fixed capacity. A representation of a solution might be an array of bits, where each bit represents a different object, and the value of the bit (0 or 1) represents whether or not the object is in the knapsack. Not every such representation is valid, as the size of objects may exceed the capacity of the knapsack. The fitness of the solution is the sum of values of all objects in the knapsack if the representation is valid, or 0 otherwise.

Genetic operators

The next step is to generate a second generation population of solutions from those selected through a combination of genetic operators: crossover (also called recombination), and mutation.For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated. Although reproduction methods that are based on the use of two parents are more "biology inspired", some research suggests that more than two "parents" generate higher quality chromosomes.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions. These less fit solutions ensure genetic diversity within the genetic pool of the parents and therefore ensure the genetic diversity of the subsequent generation of children.

It is worth tuning parameters such as the mutation probability, crossover probability and population size to find reasonable settings for the problem class being worked on. A very small mutation rate may lead to genetic drift (which is non-ergodic in nature). A recombination rate that is too high may lead to premature convergence of the genetic algorithm. A mutation rate that is too high may lead to loss of good solutions, unless elitist selection is employed.

Termination

This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

- A solution is found that satisfies minimum criteria.
- Fixed number of generations reached.
- Allocated budget (computation time/money) reached.
- The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results.
- Manual inspection.

1.2 Ant colony optimization

ACO is a part of the larger field of swarm intelligence in which scientists study the behavior pattern of bees, termites, ants and other social insects in order to simulate processes (Bell & McMullen, 2004). ACO was inspired by the foraging behavior of real ants (Dorigo et al. 1999). Ants randomly explore their surrounding using the pheromone trail (a chemical substance that can be sensed by other ants). Ants can follow various paths but in the long run, owing to reinforcement of the pheromone trail on the shortest paths, only the shortest paths remain in the use.

ACO replicates the behavior of ants, adding some features to make it more attractive for the computer implementation (Rizzoli et al. 2004). Ants construct a solution visiting a series of nodes on the graph. They select the move from one node on the graph to another, along an edge, according to two attributes: trails deposit on the edge and the attractiveness of the move.

The attractiveness of the move, from node to node, is computed according to a heuristic that express the *a priori* desirability of the move. In a shortest path problem, the desirability can be the inverse of the distance.

The pheromone trail level depends on the pheromone level, and it represents a dynamic indication *a posteriori* of its goodness.

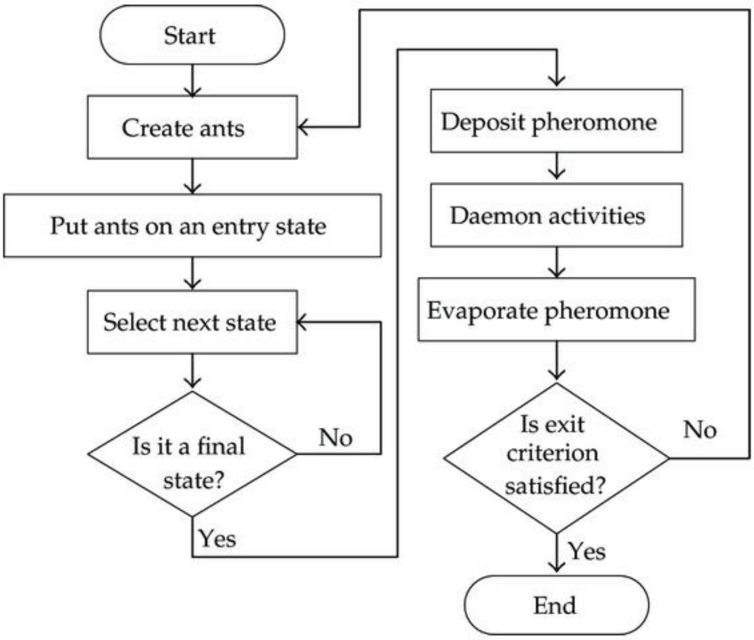

Fig. 1 Ant colony optimization flowchart

1.3 Gradient Descent

It is a first order optimization algorithm. The "gradient descent method", to find the minimum of f(x) starts from an initial point say x, then iteratively takes a step along the gradient (scaled by a stepsize), until convergence.

Output: some x hopefully minimizing f

4. repeat

5. x ← x − α ∂f(x) ∂x >

6. until Δx < θ for 10 iterations in sequence

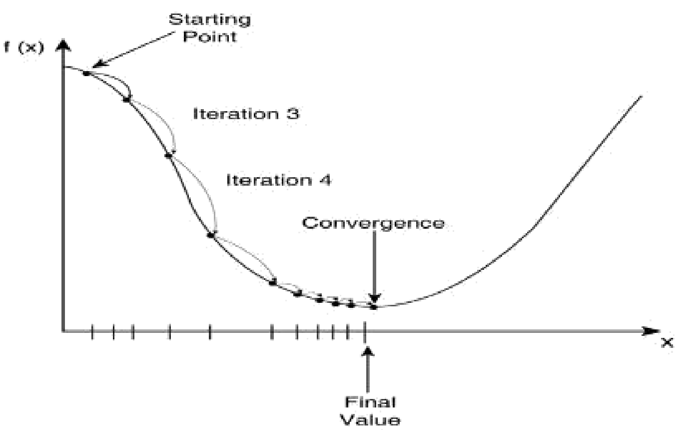

Fig. 2 Gradient Descent Method

1.4 CNX NIFTY and BSE S&P 500

The CNX Nifty, also called the Nifty 50 or simply the Nifty, is National Stock Exchange of India's benchmark stock market index for Indian equity market. Nifty is owned and managed by India Index Services and Products (IISL), which is a wholly owned subsidiary of the NSE Strategic Investment Corporation Limited. IISL had a marketing and licensing agreement with Standard & Poor's for co-branding equity indices until 2013. The 'CNX' in the name stands for 'CRISIL NSE Index'.

BSE Limited constructed a new index, christened S&P BSE-500, consisting of 500 scrips w.e.f. August 9, 1999. The changing pattern of the economy and that of the market were kept in mind while constructing this index. S&P BSE 500 index represents nearly 93% of the total market capitalization on BSE. S&P BSE 500 covers all 20 major industries of the economy. In line with other S&P BSE indices, effective August 16, 2005 calculation methodology was shifted to the free-float methodology.

The CNX Nifty covers 22 sectors of the Indian economy and offers investment managers exposure to the Indian market in one portfolio. The CNX Nifty index is a free float market capitalisation weighted index. The base period for the CNX Nifty index is November 3, 1995, which marked the completion of one year of operations of *National Stock Exchange Equity Market Segment*.

The CNX Nifty is a well-diversified 50 stock index accurately reflecting overall market conditions. The reward-to-risk ratio of CNX Nifty is higher than other leading indices, making it a more attractive portfolio hence offering similar returns, but at lesser risk.

Selection of the index set is based on the following criteria:

Liquidity: For inclusion in the index, the security should have traded at an average impact cost of 0.50 % or less during the last six months, for 90% of the observations. Impact cost is the cost of executing a transaction in a security in proportion to its index weight, measured by market capitalization at any point in time. This is the percentage mark-up suffered while buying/selling the desired quantity of a security compared to its ideal price -- (best buy + best sell)/2.

Float -Adjusted Market Capitalization: Companies eligible for inclusion in the CNX Nifty must have at least twice the float-adjusted market capitalization of the current smallest index constituent.

Float: Companies eligible for inclusion in the CNX Nifty should have at least 10% of its stock available to investors (float). For this purpose, float is stocks which are not held by the promoters and associated entities (where identifiable) of such companies.

Domicile: The company must be domiciled in India and trade on the NSE.

Eligible Securities: All common shares listed on the NSE (which are of equity and not of a fixed income nature) are eligible for inclusion in the CNX Nifty index. Convertible stock, bonds, warrants, rights, and preferred stock that provide a guaranteed fixed return are not eligible.

Other Variables: A company which comes out with an IPO is eligible for inclusion in the index if it fulfills the normal eligibility criteria for the index -- impact cost, float-adjusted market capitalization and float -- for a three-month period instead of a six-month period.

Timing of Changes: The index is reviewed semi-annually, and a four-week notice is given to the market before making any changes to the index constituents.

Additions: The complete list of eligible securities is compiled based on the float - adjusted market capitalization criteria. After that, the liquidity (impact cost) and float - adjustment filters are applied to them, respectively. The top ranking companies form the replacement pool. The top stocks, in terms of size (float-adjusted market capitalization) are, then, identified for inclusion in the index from the replacement pool.

Deletions: Stocks may be deleted due to mergers, acquisitions or spin-offs. Otherwise, as noted above, twice a year a new eligible stock list is drawn up to review against the current constituents. If this new list warrants changes in the existing constituent list, then the smallest existing constituents are dropped in favor of the new additions.

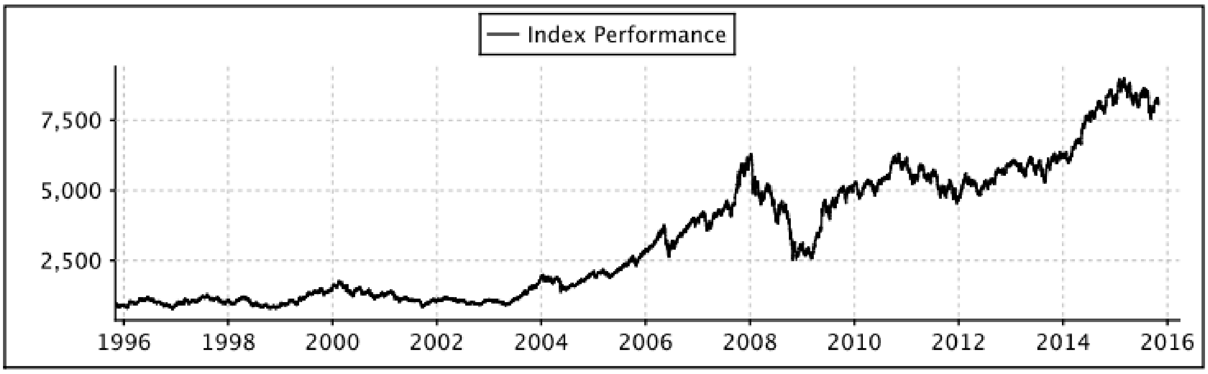

Fig. 3 Index performance

Sector	Weight (%)
FINANCIAL SERVICES	31.11
IT	16.43
ENERGY	10.94
CONSUMER GOODS	9.81
AUTOMOBILE	9.43
PHARMA	8.10
CONSTRUCTION	4.02
CEMENT & CEMENT PRODUCTS	2.85
METALS	2.76
TELECOM	2.22
SERVICES	0.93
MEDIA AND ENTERTAINMENT	0.78
INDUSTRIAL MANUFACTURING	0.63

Table 1. Sector Representation

COMPANY'S NAME	Weight(%)
INFOSYS LTD.	7.88
HDFC BANK LTD.	7.55
HOUSING DEVELOPMENT FINANCE CORPORATION LTD.	6.90
ITC LTD.	6.53
ICICI BANK LTD.	5.59
RELIANCE INDUSTRIES LTD.	5.44
TATA CONSULTANCY SERVICES	4.44
L&T	4.02
SUN PHARMA	3.37
AXIS BANK	2.78

Table 2. Top 10 constituents by weightage

Objective

This study proposes a genetic algorithm (GA) and Ant Colony Optimization (ACO) portfolio scheme for the index fund optimization. The scheme exploits GA and ACO and provides the optimal selection through the use of similarity index or by using fundamental variables like beta, average trading amount, and average market capitalization. These fundamental variables are well-known core factors frequently used in analyzing and forecasting the stock market. Roughly speaking, the GA portfolio scheme consists of two steps. First, the stocks for the index fund are selected. Second, the relative weights of the selected stocks are optimized.

Chapter 2 Literature Review

Genetic algorithm has been used in quite a few occasions for portfolio optimization. Some of the salient works are enlisted herewith:

2.1. Modern portfolio theory provides a well-developed paradigm to form a portfolio with the highest expected return for a given level of risk tolerance. Markowitz (1952, 1959), a creator of modern portfolio theory, originally formulated the fundamental theorem of mean–variance portfolio framework, which explains the trade-off between mean and variance each representing expected returns and risk of a portfolio, respectively. Although Markowitz's theory uses only mean and variance to describe the characteristics of return, his theory about the structures of a portfolio became a cornerstone of modern portfolio theory (Fama, 1970; Hakansson 1970, 1974; Merton, 1990; Mossin, 1969). After mean–variance portfolio theory, there was an enormous progress on portfolio theory and practice which include various practical applications introduced in portfolio formulation. Recently, it is found that low-cost passively managed index funds actually deliver the highest risk-adjusted returns in each category of mutual funds (Bogle, 1998).

Index fund management is a stock-allocation strategy equipped with index tracking skill which attempts to replicate the behavior of a given benchmark index. Index funds usually do not include every stock comprising the index. However, they are designed to copy the benchmark index with relatively small number of stocks, which can be easily managed and controlled in the capital market. Thus, the performance of the index fund critically depends on how well the index tracking skill replicates the benchmark index with only a subset of the stocks. Index tracking, of course, involves tracking error (TE) which is measured by TE volatility, the sum of the deviations of returns of the replicating portfolio from the benchmark index. When fund managers formulate an index fund, they try to minimize the TE volatility level since it would produce as close as possible returns to the benchmark returns (Clarke, Krase, & Statman, 1994; Sharpe, 1971; Konno & Yamazaki, 1991). In general, there are several types of TE measures available, i.e. quadratic, linear and absolute, among which quadratic measure are preferred since it possesses a number of desirable statistical properties (Roll, 1992).

2.2 Roll 1992, Suggested the importance of volatility of tracking error for fund managers. Volatility of tracking error is defined as the variance of the difference between the returns of the portfolio and benchmark. Lesser the volatility, better the portfolio tracks the benchmark. We can also incorporate β constraint and define a new objective function to minimize TEV(Tracking error volatility). TEV is a quadratic error function, though difficult to comprehend it is considered to be superior to other tracking error functions.

2.3 Beasley et. al., 1995 defines the index tracking problem is the problem of reproducing the performance of a stock market index, but without purchasing all of the stocks that make up the index. Their formulation of the problem explicitly includes transaction costs (associated with buying or selling stocks) and a limit on the total transaction cost that can be incurred. Their formulation also includes a constraint limiting the number of stocks that can be purchased. An evolutionary heuristic (population heuristic) is presented for the solution of the index tracking problem.

2.4 Orito et al. 2007 suggests that Index fund optimization is one of portfolio optimizations and can be viewed as a combinatorial optimization for portfolio managements. It is well known that an index fund consisting of stocks of listed companies on a stock market is very useful for hedge trading if the total return rate of a fund follows a similar path to the rate of change of a market index. In this paper, they propose a method that consists of a genetic algorithm and a heuristic local search on scatter diagrams to make linear association between the return rates and the rates of change strong. A coefficient of determination is adopted as a linear association measure of how the return rates follow the rates of change. Then they apply the method to the Tokyo Stock Exchange.

2.5 Many real world problems lead to a non-convex research space, example cardinality constraints which limit the number of different assets in a portfolio or suggests a minimum buy in threshold. Evolutionary algorithm is used to come up with convex subsets of the set of all feasible portfolios, solve critical line algorithm for each subset and merge the partial solution to form solution of original non convex problem (K Deb, 2009). The paper defines the 5-10-40 constraint and builds an objective function according to it. They then develop an enveloped based multi objective evolutionary algorithm to get the solutions.

Chapter 3 Materials and Methods

3.1 Stock price dataset & Computational Tool

The data consists of stock prices, volume traded and number of outstanding stocks for each of the listed companies in Nifty & BSE 500. The data was obtained from Bloomberg terminal. Amount traded was found by multiplying volume traded and stock price, whereas market capitalization was calculated by multiplying stock price with number of outstanding shares. Rstudio was used to code for the genetic algorithm and ant colony optimization.

Date	ACC	ACC	ADSEZ	ADSEZ	ACEM	ACEM
10/30/2015	1379.6	267835	296.25	3452020	206.95	2865884
10/29/2015	1385.4	186117	300.1	4636277	207.45	4392554
10/28/2015	1401.05	244447	300.65	6847533	208.75	3866274
10/27/2015	1385.8	187936	311.3	3153230	205.4	4472757
10/26/2015	1393.75	181166	316.35	1780770	208.65	874211
10/23/2015	1400.2	282679	314.15	1688767	208.95	1604359
10/21/2015	1378.95	466471	314.2	2346810	209.2	1299281
10/20/2015	1379.3	338148	309.85	5899730	209.95	2407020
10/19/2015	1395.95	395406	317.7	1535982	210.05	3187045
10/16/2015	1393.7	170840	323.7	1728846	212.15	3865670
10/15/2015	1371.25	197527	329	2170298	209.65	2993217
10/14/2015	1376.25	248256	324.5	1812411	209.95	2170530
10/13/2015	1352.2	234953	323.55	1375903	209.95	2682919

Table 3. Sample dataset of Nifty 50.

3.2 Methodology adopted for optimization

3.2.1. The proposed scheme is based on three fundamental variables: portfolio beta, trading amount, and market capitalization. These three variables are frequently used in portfolio management area, among which portfolio beta is especially the most important variable (Chang, 2004; Keim, 1999).

3.2.2. Portfolio beta is defined as the correlation between the company returns and the industry returns. It is well known that the portfolio beta measures portfolio volatility relative to the benchmark index or the capital market. Indeed, if a portfolio is well chosen such that returns of portfolio and benchmark index are highly correlated, then portfolio beta becomes the volatility ratio between the portfolio and the benchmark index approximately. Thus, portfolio beta usually implies the portfolio sensitivity relative to the fluctuations of the market index. For further discussions of our portfolio scheme, denote portfolio beta for the j^{th} stock among stocks that comprises a market index m by β_j, i.e.

$$\beta_j = \frac{Cov(r_j, I_m)}{Var(I_m)}$$

3.2.3 Portfolio scheme for index fund management

Let n and l be the numbers of stocks for the benchmark index and its index fund portfolio, respectively (l<n).

Further, let c_k (k=1,2,.,l) be the serial code of k^{th} stock to be included in the index fund portfolio. In other words, index fund portfolio set is Ω_p={c1,c2,.,cl} which is to be selected from the entire n stocks. Let s denote the number of industry sectors comprising the benchmark index and d_i the number of stocks comprising i^{th} industry sectors (i.e. $\sum_{i=1}^{s} d_i$ = n). In addition, for each j^{th} stock of i^{th} industry sector (j=1,2,.,d_i and i=1,2,.,s), suppose the portfolio beta is given by $\beta_{i(j)}$ where the subscript i(j) is used to stress dependence of j on i. For that specific stock, let $r_{i(j)(t)}$, $A_{i(j)(t)}$ and $M_{i(j)(t)}$ denote rate of return, trading amount, and market capitalization at time t, respectively.

Define priority $P_{i(j)} = v_1\{B_{i(j)}\}^{-1} + v_2 A_{i(j)} + v_3 M_{i(j)}$

Where $A_{i(j)}$ represent average trading amount in the period a_0 to a_o+T

$M_{i(j)}$ represent average market capitalization in the period a_0 to a_o+T

$B_{i(j)}$ is the beta.

Step1

Select industry sector i_k having the largest amount of market capitalization

i.e $mc_{i_k(k)} = max_{i=1,2....s} mc_{i(k)}$.

For the selected industry sector i_k calculate $P_{i_k(j)}$ for j= 1,2,……,$d_{i(k)}$ and choose the stock $i_k(j_k)$ with highest priority and add that stock to the portfolio and remove from the selected industry sector i_k. Now update the market capitalization without the stock $i_k(j_k)$ and repeat the steps until l stocks for the index fund are obtained.

Step2

For index fund portfolio established by $\emptyset_p = \{c_1, c_2, c_3 \ldots . c_l\}$ let w_k^m be the market capitalization of c_k scaled by entire market capitalization.

Now derive the optimal weights minimizing the given equation

$$Q_{w1,w2,....wl} = \sum_{k=1}^{l} (w_k - w_k^m)^2 \, \sigma_k^2$$

Or optimizes the selected stocks by GA by maximizing

$$R^2 = \left(\frac{Cov(X,Y)}{\sqrt{Var(X),Var(Y)}}\right)^2$$

On the 1st generation of the GA, we generate the initial population at random. In the crossover, we apply the standard two-point crossover for exchanging the partial structure between the two chromosomes and repair to a probability distribution via renormalization. In the mutation, we apply the standard two point mutation for replacing the partial structure of the selected chromosomes with a new random value in [0, 1] and repair to a probability distribution via renormalization. After making offspring, we apply an elitism method of one chromosome based on the fitness value. Finally the GA is broken off on the last generation. After executing the GA, we select one chromosome with the highest coefficient of determination R^2. We use a mutation rate of 0.1, selection rate of 0.5 and crossover rate of 0.5.

3.3 Selection of stocks by using of Similarity Index

It is possible to formulate an optimization problem that using some similarity measure, selects only a given number of representative stocks. Generally for such an optimization problem a constraint is used to restrict the number of assets in the tracking portfolio is present.

Hence, the problem is an integer programming problem, which is NP hard. It is possible to solve such problems using algorithms for meta heuristic optimization. There is also an approach that avoids solving an integer programming problem. The simplest version is to solve the optimization problem to find the minimum tracking error, using all of the assets in the index universe. As a result, the weights are computed for every asset. Then, only the assets with the largest weights are included in the tracking portfolio. The number of assets selected is equal to the size of tracking portfolio desired. This approach can lead to poor solutions if there is a sector or region with many stocks that are highly correlated.

The main disadvantage of optimization methods that look for stocks that are very similar to the index is the low level of diversification in the resulting tracking portfolio. The optimal tracking portfolio may contain stocks that are very similar to the index and, as a consequence, these stocks will be similar to each other, which reduce the diversification of the portfolio. There are two ways of avoiding this problem. The first is to add additional constraints, which control the diversification of the portfolio. For example, it is possible to add sector constraints, which specify that the sectorial structure of the portfolio must mimic that of the index, or a portfolio concentration constraint, which limits the weights that can be assigned to individual stocks. The second possibility is to filter out similar stocks before carrying out the optimization.

To the problem we are addressing an integer programming problem is formulated as follows and solved by Genetic Algorithm and Ant Colony Optimization.

Max $z = \sum_{i \epsilon N} \sum_{j \epsilon N} \rho_{ij}\, x_{ij}$ (1)

s.t

$\sum_{j \epsilon N} y_j = q$ (2)

$\sum_{i \epsilon N} x_{ij} = 1 \ \forall\ i \epsilon N$ (3)

$x_{ij} \leq y_j \ \forall\ i \epsilon N; \forall\ j \epsilon N$ (4)

$x \in \{0,1\}^{NXN}$ (5)

$y \in \{0,1\}^{N}$ (6)

Let be N the set of all n stocks included in the market index. $\rho_{ij} \in [-1,1]$ be the similarity measure between stocks $i \epsilon N$ and stock $j \epsilon N$ in the index. Let x_{ij} be a binary variable indicating whether the stock i is represented by the stock j in the index fund or not. Similarly, binary variable y_j is 1 if the stock is selected in the index fund and 0 otherwise.

	ACC	ADSEZ	ACEM	APNT	AXSB	BJAUT	BOB
ACC	1	0.059595	0.909395	-0.01637	0.507322	-0.45889	-0.01764
ADSEZ	0.059595	1	0.148125	0.597103	0.460857	-0.26677	-0.13896
ACEM	0.909395	0.148125	1	0.118564	0.688037	-0.38767	-0.06862
APNT	-0.01637	0.597103	0.118564	1	0.399259	-0.13939	0.02498
AXSB	0.507322	0.460857	0.688037	0.399259	1	-0.33403	-0.45501
BJAUT	-0.45889	-0.26677	-0.38767	-0.13939	-0.33403	1	0.307352
BOB	-0.01764	-0.13896	-0.06862	0.02498	-0.45501	0.307352	1

Table 4. Sample correlation(similarity) matrix

3.4 Cardinality constraints

$$Minimize\ E\ (tracking\ error) \qquad (1)$$

$$\sum_{i=1}^{N} z_i = K \qquad (2)$$

$$\varepsilon_i z_i \leq \frac{V_{iT} x_i}{C} \leq \delta_i z_i \quad \text{i= ,...., N} \qquad (3)$$

$$x_i \geq 0 \quad \text{i= ,...., N} \qquad (4)$$

$$z_i \in [0,1] \quad \text{i= ,...., N} \qquad (5)$$

$$C_{trans} \leq \gamma C \qquad (6)$$

$$\sum_{i=1}^{N} V_{iT} x_i = C - C_{trans} \qquad (7)$$

X_i - the number of units of stock *i* (*i*=1,...,*N*) that we choose to hold in the tracking portfolio

Z_i - 1 if any of stock *i* (*i*=1,...,*N*) is held in the tracking portfolio, 0 otherwise

ε_i be the minimum proportion of the tracking portfolio that must be held in stock *i*(*i*=1,...,*N*) if any of stock *i* is held.

δ_i be the maximum proportion of the tracking portfolio that can be held in stock *I*

V_{it} be the value of one unit of stock *i* (*i*=1,...,*N*)

C_{trans} the total transaction cost involved

C – cash invested

Chapter 4 Result and Discussion

In this section let's see the results obtained for different scenarios. The analysis is performed from Nifty (50) data developing an index of 20 stocks. Computational work that has been performed in mentioned in Appendix.

1. Based on Simple selection method which emphasize on the criteria of market capitalization, trading volume and portfolio beta, the selected stocks obtained after implementing the above algorithm are-

[1] "AXSB" "CIPLA" "DRRD" "GRASIM" "HMCL" "HUVR" "TCS" "UTCEM"
[9] "YES" "LT" "MSIL" "ONGC" "PWGR" "RIL" "SBIN" "KMB"
[17] "ICICIBC" "IDEA" "IIB" "HDFC"

2. Application of Ant Colony Optimization method to the formulated problem as mentioned in 3.2.1

19. "AXSB" "BHEL" "BHARTI" "BOS" "COAL" "DRRD" "GRASIM" "HNDL" "CAIR" "TCS"

[11] "UTCEM" "WPRO" "Z" "SUNP" "PWGR" "PNB" "RIL" "IIB" "INFO" "HDFC"

3. The weights obtained for the corresponding companies using GA after selecting stocks through simple selection method are -

[1] 0.072676638 0.039279743 0.024136214 0.021266181 0.046622894
0.00613042 [7] 0.092754684 0.105280957 0.014543032 0.072264372
0.035404768 0.0641121 [13] 0.125739527 0.031731200 0.094564650
0.012002396 0.017405014 0.0473102 [19] 0.008248308 0.068526632

Maximum value of fitness function comes out to be : 0.85

4. The weights obtained for the corresponding companies using GA to select stocks and assign weigths are

7.578302e-02 1.286819e-02 7.006219e-02 8.106577e-05 1.074383e-01 4.057809e-02
1.360494e-01 5.960780e-02 5.037078e-02 7.691661e-02 2.907309e-02 5.389594e-02
2.614412e-02 1.367372e-02 7.487054e-02 2.086327e-02 6.600049e-02 5.628974e-02
3.919323e-03 2.551438e-02

Maximum fitness value comes out to be : 0.92

5. The weights obtained for the corresponding companies using genetic algorithm for weights and ACO for selection -

0.048058	0.036233	3.69E-05	0.093327	0.056856	0.072625
0.059135	8.55E-07	0.078333	0.046853	7.22E-08	0.11525
0.056864	0.059359	3.47E-07	0.061327	0.0075765	0.063298
0.062505	1.12E-06				

Maximum correlation comes out to be : 0.87

6. The weights obtained for the corresponding companies using Gradient Descent and selecting stocks through genetic algorithm method are

0.032039	3.36E-06	0.0030233	0.0046917	0.10136	0.0097751
0.00042583	0.19408	0.20841	0.0084202	0.004656	0.014578
0.092007	0.028583	0.056322	0.13572	0.049599	0.01879
0.0086629	0.028854				

Maximum correlation comes out to be : 0.92

Regarding the weights idea drawn is that we can use various solution methods but a heuristic method never guarantees an optimal solution wherein a mathematical model ensures the optimality. In this study we found by doing an empirical analysis that Gradient Descent suggested a good solution at the same time we say it for certainty that it’s the optimal one too.(not like a heuristic method where we can only say that the solution is near optimal)

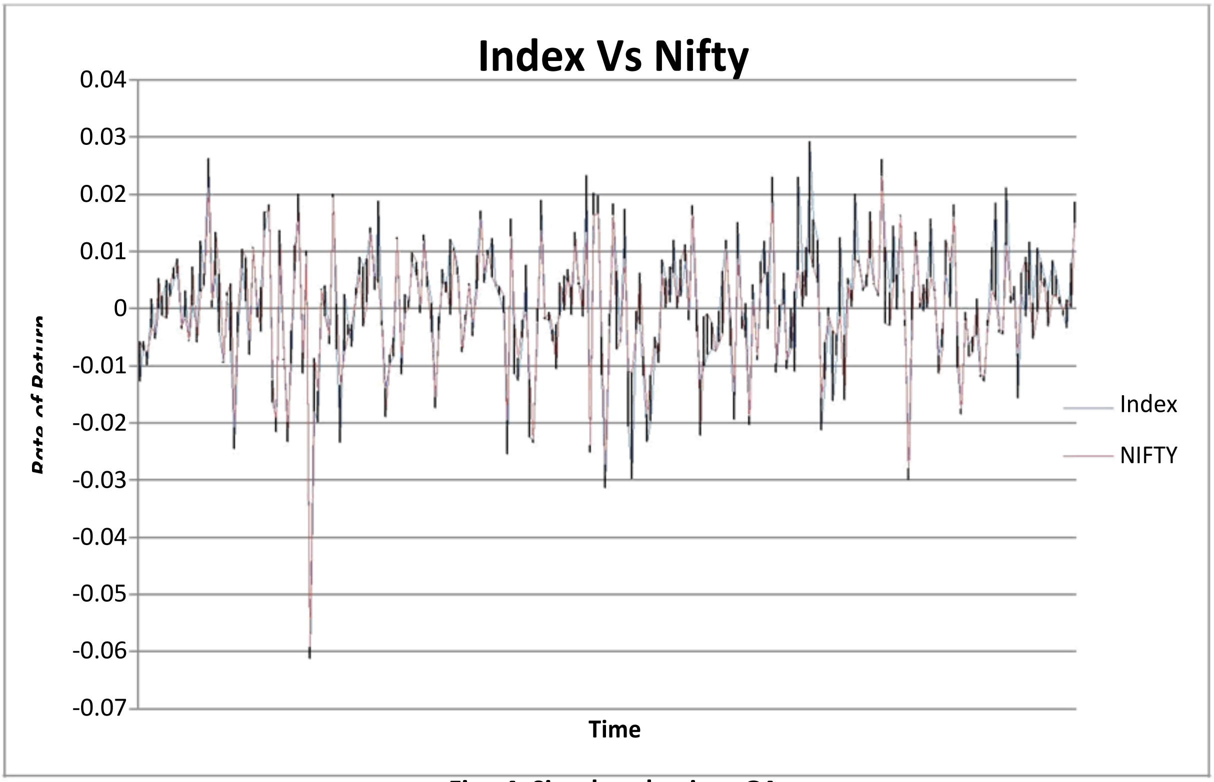

Fig. 4 Simple selection_GA

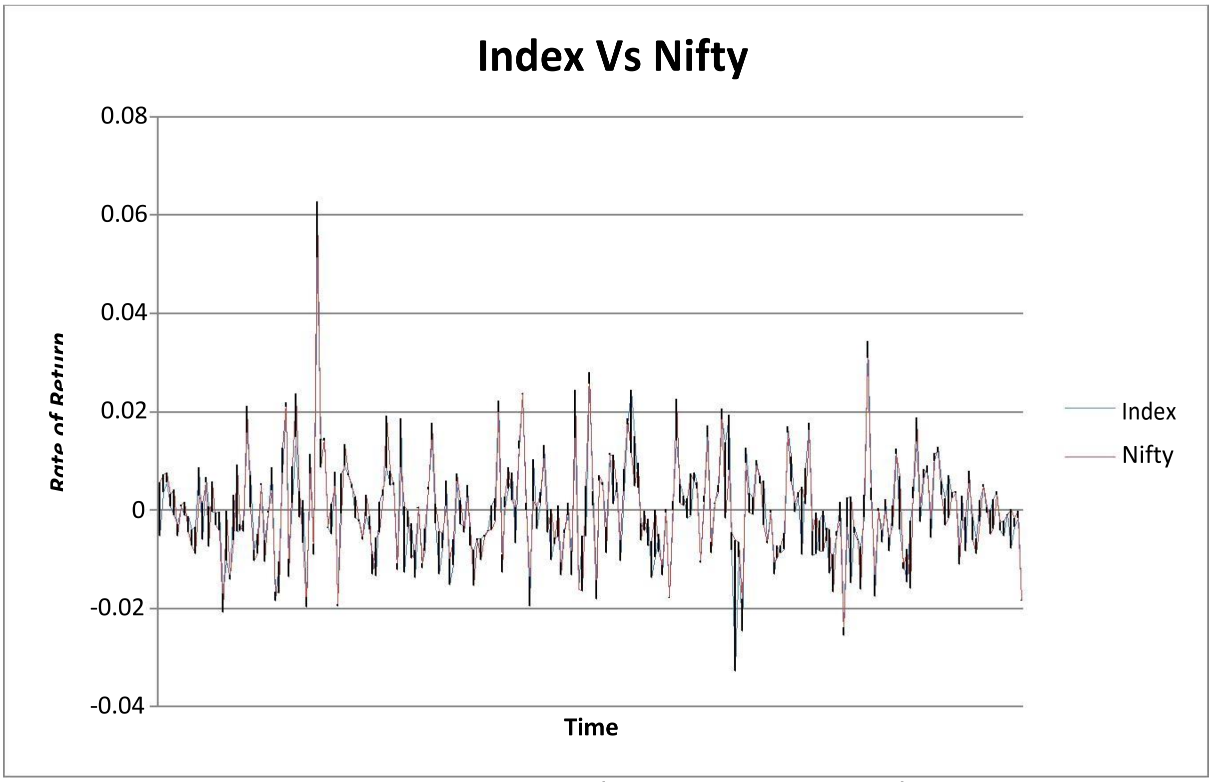

Fig. 5 Optimized selection (Ant Colony optimization)_GA

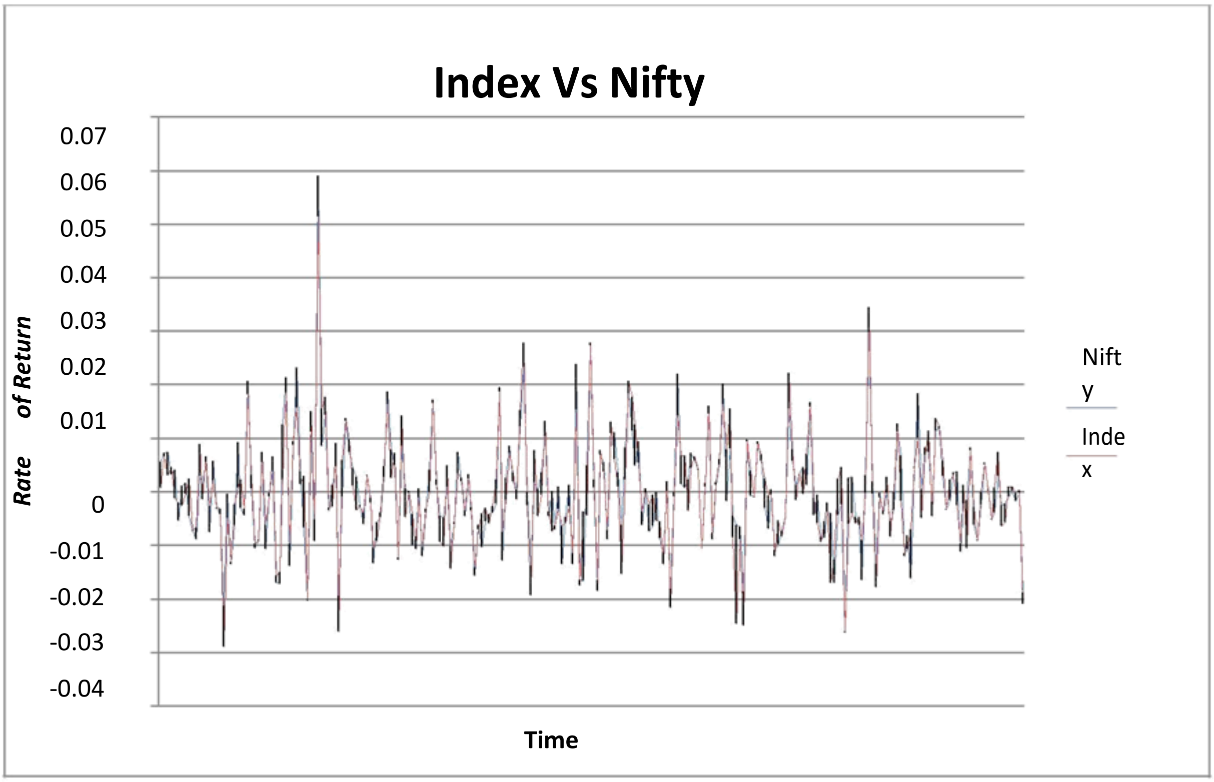

Fig. 6 Optimized Selection(GA)_ GA

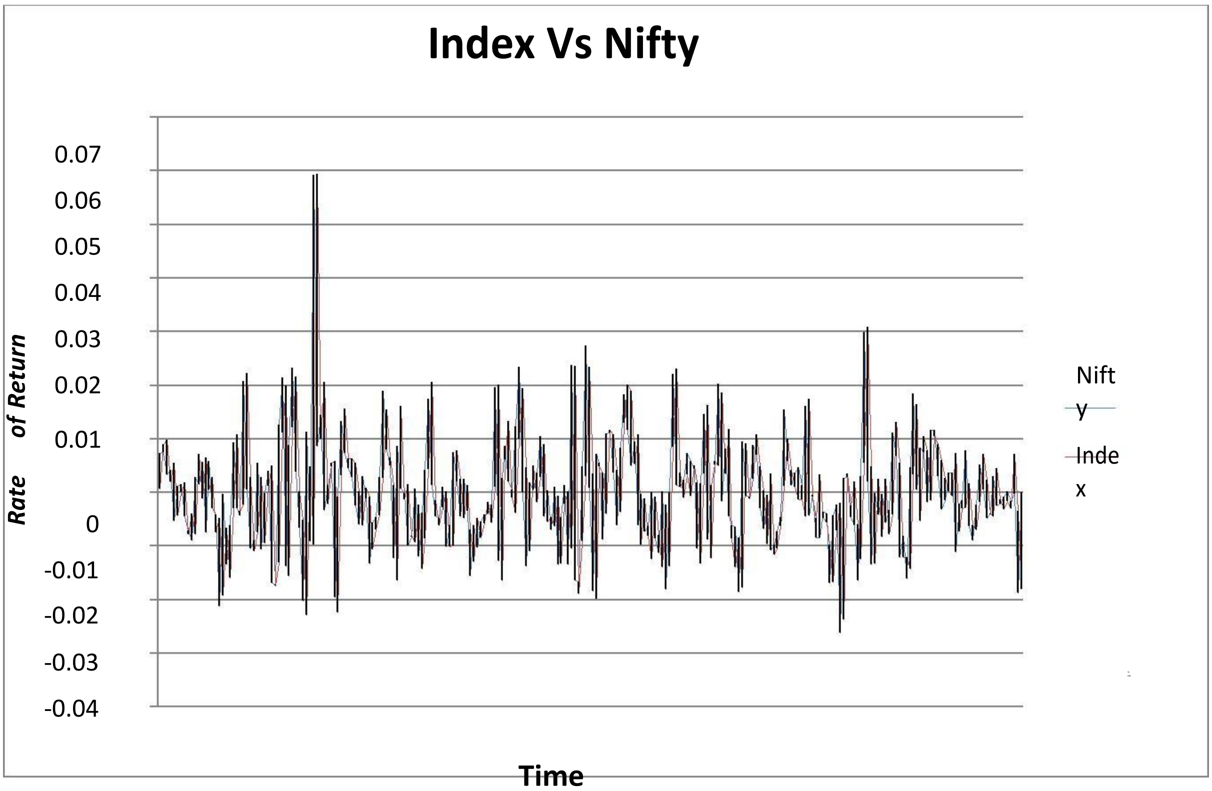

Fig. 7 Optimized selection (ACO)_Gradient Descent

BSE S&P 500 vs index fund (Correlation 0.94)

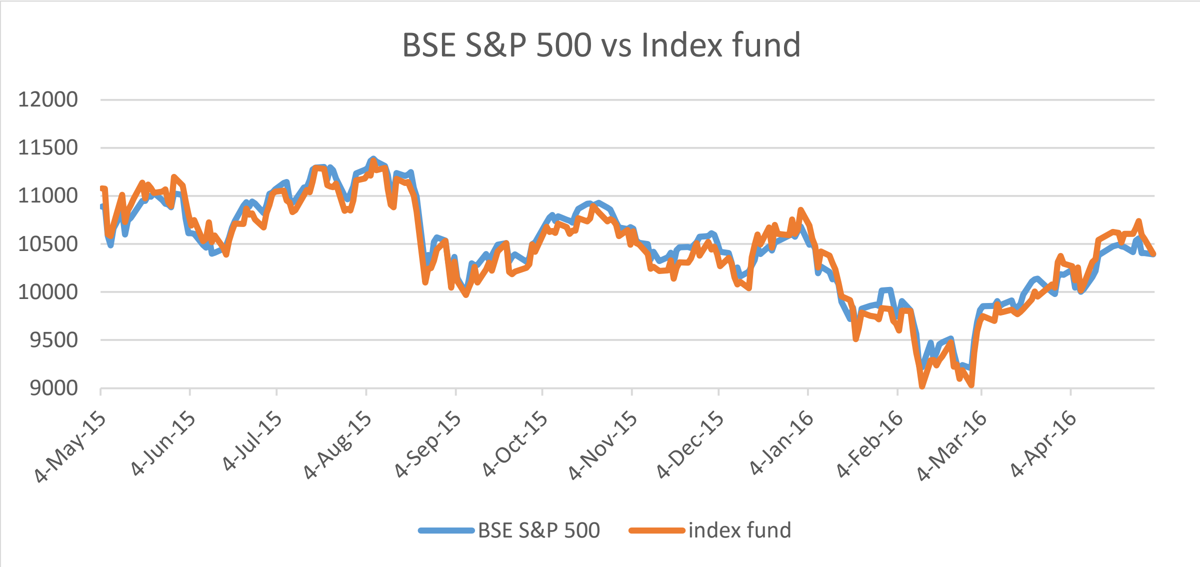

Fig. 8 Index fund tracking BSE S&P 500

The R^2 obtained for different scenarios is mentioned, which seems pretty good after seeing the plot of returns of the index vs the portfolio. The portfolio curve traces the index curve very closely, overlapping many a times.

7. For Gradient Descent , correlation tends to 1 with the iteration number.

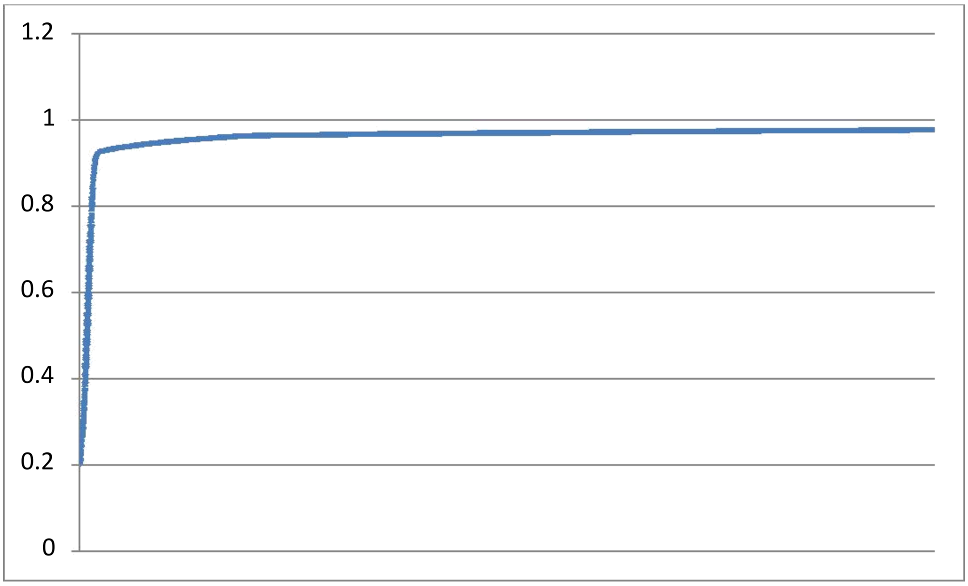

Fig. 9 Correlation Trends to one in gradient descent

8. The genetic algorithm can be tested for other parameters too, to obtain a better result. Different selection procedures, like roulette wheel can be used. The same applies to different ways of doing crossovers like single point crossover and double point. It goes same for mutation.

Chapter 5 Future Scope of Study

1. Research suggests that SPEA 2 (Strength Pareto Evolutionary Algorithm) is the best algorithm suited for portfolio optimization with 3 objectives followed by NSGA II.
2. Optimizing the weights in the index fund portfolio consisting considering various tracking error functions like absolute tracking error, quadratic tracking error, etc.
3. Checking for the tracking error and rebalancing the portfolio if tracking error falls beyond a threshold.
4. Optimizing an index fund based on considering excess return and tracking error together.

Chapter 6 References

1. **Beasley, J.E., Meade, N., Chang, T.J., 2003.** An evolutionary heuristic for the index tracking problem. *European Journal of Operational Research, Volume 148, Issue 3, 1 August 2003, Pages 621–643*

2. **Nigam, A., Agarwal, Y.K., 2013**. Ant colony optimization for index fund problem. *Journal of Applied Operational Research, 2013, 5(3), 96-104*

2. **Anagnostopoulos, K.P., Mamanis, G., 2010**.A portfolio optimization model with three objectives and discrete variables. *Computers & Operations Research 37 (2010) 1285–1297*

3. **Kyong, O.H., Kim, T.Y., Sungky, M., 2005**. Using genetic algorithm to support portfolio optimization for index fund management. *Expert Systems with Applications 28 (2005) 371–379*

1. **Karlow, D.,2012**. Comparisons and Development of Methods for Index Tracking.*Frankfurt School of Finance and Management*

2. **Cornuejols, G., & Tutuncu, R., 2007**. Optimization Methods in Finance*. Cambridge University Press*

3. **Bell, J., & McMullen, P. 2004**. Ant Colony Optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics , 41-48.*

4. **Orito, Y., Yamamoto, H., & Yamazaki, G. 2003.** Index fund selections with genetic algorithms and heuristic classifications*. Computers and Industrial Engineering , 97-109*

5. **Black, F., Litterman, R., 1992.** Global portfolio optimization. *Financial Analysts Journal*, Sep/Oct , pp. 28-43.

6. **Blume, M.E., Edelen, R.M., 2004.** S&P 500 indexers, tracking error, and liquidity. *The Journal of Portfolio Management*, 30, (3), pp. 37-46.

7. **Gourieroux, C., Jasiak, J., 2001.** *Financial econometrics: problems, models and methods.* Princeton

8. **Grauer, R.R., Shen, F.C., 2000.** Do constraints improve portfolio performance?. *Journal of Banking &Finance*, 24, (8), pp. 1253-1274.

Chapter 7 Appendix

7.1 R Code for Genetic Algorithm

```
library(ggplot2)
file<-read.csv('final.csv')
x<-read.csv('CNX NIFTY30-10-2014-29-10-2015.csv')
file$NIFTY<-rev(x$Close)
rm(x)
nshares<-20
vol<-c(187745356,2070951761,1551897421,959197790,2375296362,289314176,2304159598,2447600000,
    723084248,3997400102,31398900,1874852752,803090988,6316364400,170588515,1268477400,
    91866649,1406251338,2511458217,199687500,2064985760,2163868647,2406423348,1958727979,
    2887203602,2704629398,971215439,962337884,274406796,2964694239,2469894208,418107704,
    960448720,1830512584,930298999,449884842,621092384,302080060,8245464400,8555490120,
    5231589648,1963597490,3236439248,7566207932,1577684750,8026495071,5807681135,3599610386,
    592166408,2296944664)

ind<-c('CEMENT & CEMENT PRODUCTS','SERVICES','CEMENT & CEMENT PRODUCTS','CONSUMER GOODS',
    'FINANCIAL SERVICES','AUTOMOBILE','FINANCIAL SERVICES','INDUSTRIAL MANUFACTURING',
    'ENERGY','TELECOM','AUTOMOBILE','ENERGY','PHARMA','METALS','PHARMA','ENERGY',
    'CEMENT & CEMENT PRODUCTS','IT','FINANCIAL SERVICES','AUTOMOBILE','METALS',
    'CONSUMER GOODS','PHARMA','IT','AUTOMOBILE','ENERGY','METALS','IT',
    'CEMENT & CEMENT PRODUCTS','METALS','IT','FINANCIAL SERVICES','MEDIA & ENTERTAINMENT',
    'FINANCIAL SERVICES','CONSTRUCTION','PHARMA','AUTOMOBILE','AUTOMOBILE','ENERGY','ENERGY',
    'ENERGY','FINANCIAL SERVICES','ENERGY','FINANCIAL SERVICES','FINANCIAL SERVICES',
    'CONSUMER GOODS','FINANCIAL SERVICES','TELECOM','FINANCIAL SERVICES','IT')

ret<-file[seq(from=2,to=102,by=2)]
amount<-colMeans(file[seq(from=3,to=101,by=2)])
mshare<-colMeans(ret[-51])*vol

## Priority Function
priority<-function(w1,w2,w3)
{
  return<-(ret[-nrow(ret),]-ret[-1,])/ret[-1,]
  return<-scale(return,center = T,scale = F)
  s<-colSums(return^2)
  k<-sqrt((365-2)*(s[length(s)]))
  p<-(w1*s[-length(s)]/k)+(w2*amount)+(w3*mshare)
}

## Fitness Function
fitness<-function(pop,ret)
{
  fit<-rep(0,popsize)
  for (i in 1:popsize)
  {
    daily<-mapply(`*`,pop[i,],ret[-(nbits+1)])
    fit[i]<-cor(rowSums(daily),ret[nbits+1])
#       daily<-pop[i]-pop[i]*vol/sum(ret[i]*vol)

  }
  fit
}
```

```
## Initial Population
init<-function(popsize,nbits)
{
  pop<-matrix(runif(nbits*popsize),popsize)
  pop<-pop/rowSums(pop)
}

## Selection Function
select<-function(fit,keep)
{
  s<-1:length(fit)
  now<-sample(s,keep)
  df<-data.frame(pop=s[-now],fit=(fit[-now])^2)
  df<-df[order(df$fit),]
  df$cumfit<-cumsum(df$fit)/sum(df$fit)

  s[1:keep]<-s[now]
  for(i in (keep+1):length(fit))
  {
    s[i]<-df$pop[sum(runif(1)>df$cumfit)+1]
  }
  s
}

## Crossover Function
cross<-function(pop,s,keep)
{
  newpop<-pop[1:keep,]
  pop<-pop[-c(1:keep),]
  for (i in seq(from=1,to=(length(s)-keep),by=2))
  {
    a<-pop[i,]
    b<-pop[i+1,]
    x<-a
    y<-b
    r<-runif(length(a))>0.5
    x[r]<-b[r]
    y[r]<-a[r]
    pop[i,]<-x/sum(x)
    pop[i+1,]<-y/sum(y)
  }
  newpop<-rbind(newpop,pop)
}

## Result
result<-function(best)
{
  daily<-mapply(`*`,best,ret[-(nbits+1)])

  df<-data.frame(date=file$Date,index=rowSums(daily),nifty=ret[(nbits+1)])
  x<-df[-nrow(df),]
  df<-df[-1,]
  x$index<-(x$index-df$index)/df$index
  x$NIFTY<-(x$NIFTY-df$NIFTY)/df$NIFTY
  x$date<-as.Date(x$date, '%Y-%m-%d')
  print(ggplot(x)+geom_line(aes(date,index,color='red'))+
        ggtitle('Comparison of Index with 10')+geom_line(aes(date,NIFTY,color='blue')))
```

```
}

## Mutate Function
mutate<-function(newpop,mut)
{
  s<-sample(nrow(newpop),mut)
  now<-newpop[s,]

  newpop[s,]<-now

}

## Select top 20
prior<-priority(0.4,0.3,0.3)
mind<-data.frame(ind=ind,mshare=mshare,prior=prior)
for (i in 1:nshares)
{
  x<-aggregate(mshare~ind,mind,sum)
  selectcap<-x$ind[which.max(x$mshare)]
  y<-mind[mind$ind==selectcap,]
  selectshare<-rownames(y)[which.max(y$prior)]
  mind<-mind[rownames(mind)!=selectshare,]
}
selected<-setdiff(names(mshare),rownames(mind))

## GA Parameters
mutrate<-0.1
selection<-0.5

popsize<-500
nbits<-nshares
keep<-floor(selection*popsize)
mut<-floor(mutrate*popsize)

epochs<-50

## GA
iga<-1
pop<-init(popsize,nbits)
fit<-fitness(pop,ret)
bestfit<-max(fit)
best<-pop[which.max(fit),]

while(iga<=epochs)
{
  s<-select(fit,keep)
  newpop<-cross(pop,s,keep)
  # newpop<-mutate(newpop)

  fit<-fitness(newpop,ret)
  if (max(fit)>bestfit)
  {
    bestfit<-max(fit)
    best<-newpop[which.max(fit),]
  }
  pop<-newpop
  iga<-iga+1
```

```
   print(bestfit)
}
bestfit
result(best)
##
```

7.2 Matlab Code for gradient descent

```
function [cor, delW]=getDelW(x,Y,W)

for j=1:1:size(x,1) X(j,1)=sum(x(j,:).*W);

end

Nm=sum((X-mean(X)).*(Y-mean(Y))); Dn=norm(X-mean(X))*norm(Y-mean(Y)); cor=Nm/Dn;
delW_Nm=getDelW_Nm(x,Y,W); delW_Dn=getDelW_Dn(x,X,Y,W); delW=delW_Nm/Dn - Nm*delW_Dn/Dn^2;

end


function delW_Nm=getDelW_Nm(x,Y,W) for i=1:1:length(W)

delW_Nm(i)=sum((x(:,i)-mean(x(:,i))).*(Y-mean(Y)) ); end

end


function delW_Dn=getDelW_Dn(x,X,Y,W) for i=1:1:length(W)

delW_Dn(i)=norm(Y)/norm(X)*sum( (X-mean(X)).*( x(:,i)-mean(x(:,i)) ) ); end


%deltaW(k,n)=(f(k)-f_mean)*p(n)/Dn- Nm*(g(k)-g_mean)*p(n)/(Dn^3); clear all;

close all; xY=csvread('Data.csv'); Y=xY(:,end); x=xY(:,1:end-1); W=rand(1,size(x,2));


for loop=1:1:100000 if(mod(loop,100)==0)

disp(loop) ; end

W_loop{loop}=W; [cor(loop),delW]=getDelW(x,Y,W); W=W+0.0001*delW;

%         W=max(W,0);

W=W/sum(W);

end



%%


[max_cor max_i]=max(cor); W_max=W_loop{max_i}; for j=1:1:size(x,1)
```

```
X(j,1)=sum(x(j,:).*W_max);

end


X_c=X;

Y_c=Y;

X=X-mean(X);

X=X/norm(X);


Y=Y-mean(Y);

Y=Y/norm(Y);

figure(1)

subplot(2,1,1)

plot(X);

subplot(2,1,2)

plot(Y);


figure(2)

plot(X,'r') hold on plot(Y,'g')


%% ploy return X=X_c;

for i=1:1:(length(X)-1)
x_r(i) =(X(i+1)-X(i))/X(i); end


Y=Y_c;

for i=1:1:(length(Y)-1) y_r(i)=(Y(i+1)-Y(i))/Y(i);

end

figure(3) plot(x_r,'r'); hold on plot(y_r,'g');
```

7.3 R code Ant colony optimization

```
file<-read.csv('final.csv')
x<-read.csv('CNX NIFTY30-10-2014-29-10-2015.csv')
file$NIFTY<-rev(x$Close)
rm(x)
ret<-file[seq(from=2,to=100,by=2)]
corr<- cor(ret)



K<-10000
alpha<-1.5
beta<-0.5
rho<-0.15
LSC<-50
GSC<-50
stock_selection<-20

update_ph<-function(ph,rho,d_ph)
{
  ph<-(1-rho)ph+d_ph
}

generate_random<-function(K)
{
  Pk<-sample(c(rep(1,stock_selection),rep(0,(50-stock_selection))))
  for(i in 2:K)
  {
    Pk<-rbind(Pk,sample(c(rep(1,stock_selection),rep(0,(50-stock_selection)))))
  }
  Pk
}

fitness<-function(Pk)
{
  fit<-rep(0,K)
  for (i in 1:K)
  {
    try<-Pk[i,]
    try<-(sum(apply((try*corr), MARGIN=c(2), max)))/50
    fit[i]<-try*try
  }
  fit
}

localswap<-function(Pk)
{

}

Pk<-generate_random(K)
fit<-fitness(Pk)
max(fit)
```

7.4 Rcode GA stock selection

```
library(dplyr)
library(reshape2)
library(genalg)
d=nifty.stock.price
cor(d)
d_cor <- as.matrix(cor(d))
d_cor_melt <- arrange(melt(d_cor), value)
write.csv(d_cor, file = "correlation matrix.csv")

mydata<-data.frame(correlation.matrix)
mydata[-c(1), ]
mydata <- mydata[-c(1),]

stock.selection <- function(x) {
  x<- matrix(data=NA,nrow=50,ncol=1)
  i=as.integer(1)
  for (i in 1:50)
  {x[i,1] = 0 || 1}
  print(x)
  maximize
  sum(correlation.matrix*x)
 if(sum(y)=20)

}


woppa<-rbga.bin(size=10,
         suggestions=NULL,
         popSize=200, iters=1000,
         mutationChance=0.05,
         elitism=.2, zeroToOneRatio=10,
         monitorFunc=NULL, evalFunc=stock.selection,
         showSettings=TRUE, verbose=FALSE)

y<-kmeans(nifty.transpose, 20, iter.max = 50, nstart = 1,
          algorithm = c("Hartigan-Wong"), trace=FALSE)
# K-Means Cluster Analysis
fit <- kmeans(nifty.stock.price, 20)
# get cluster means
aggregate(nifty.transpose,by=list(fit$cluster),FUN=mean
```