Applying Natural Computing to Stock Market Portfolio Management

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

in

Computer/IT Engineering

by

Bharatkumar Vikrambhai Chawda

129990907001

under supervision of

Dr. Jayeshkumar M. Patel

GUJARAT TECHNOLOGICAL UNIVERSITY
AHMEDABAD

July 2020

Applying Natural Computing to Stock Market Portfolio Management

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

in

Computer/IT Engineering

by

Bharatkumar Vikrambhai Chawda

129990907001

under supervision of

Dr. Jayeshkumar M. Patel

GUJARAT TECHNOLOGICAL UNIVERSITY
AHMEDABAD

July 2020

© Bharatkumar Vikrambhai Chawda

DECLARATION

I declare that the thesis entitled **"Applying Natural Computing to Stock Market Portfolio Management"** submitted by me for the degree of Doctor of Philosophy is the record of research work carried out by me during the period from **November 2012** to **July 2020** under the supervision of **Dr. Jayeshkumar M. Patel**, Associate Professor MCA Program, Acharya Motibhai Patel Institute of Computer Studies, Ganpat University and this has not formed the basis for the award of any degree, diploma, associateship and fellowship, titles in this or any other University or other institution of higher learning.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis. I shall be solely responsible for any plagiarism or other irregularities if noticed in the thesis.

Signature of the Research Scholar: ………………………………………..

Name of Research Scholar: **Bharatkumar Vikrambhai Chawda**

Date: 21/07/2020 Place: **Vallabh Vidyanagar**

CERTIFICATE

I certify that the work incorporated in the thesis entitled **“Applying Natural Computing to Stock Market Portfolio Management”** submitted by **Bharatkumar Vikrambhai Chawda** was carried out by the candidate under my supervision/guidance. To the best of my knowledge: (i) the candidate has not submitted the same research work to any other institution for any degree/diploma, Associateship, Fellowship or other similar titles (ii) the thesis submitted is a record of original research work done by the Research Scholar during the period of study under my supervision, and (iii) the thesis represents independent research work on the part of the Research Scholar.

Signature of Supervisor: ……………………………….

Name of Supervisor: **Dr Jayeshkumar M. Patel**

Date: 21/07/2020 Place: **Mehsana**

Course-work Completion Certificate

This is to certify that **Bharatkumar Vikrambhai Chawda** (Enrolment no. 129990907001) is a PhD scholar enrolled for the PhD program in the branch **Computer/IT Engineering** of Gujarat Technological University, Ahmedabad.

(Please tick the relevant option(s))

☐ He has been exempted from the course-work (successfully completed during M.Phil. Course)

☐ He has been exempted from Research Methodology Course only (successfully completed during M.Phil. Course)

☑ He has successfully completed the PhD course work for the partial requirement for the award of PhD Degree. His performance in the course work is as follows –

Grate Obtained in Research Methodology (PH001)	Grade Obtained in Self Study Course (Core Subject) (PH002)
AB	AB

Signature of Supervisor: …………………………………

Name of Supervisor: **Dr Jayeshkumar M. Patel**

Date: 21/07/2020 Place: **Mehsana**

Originality Report Certificate

It is certified that PhD Thesis titled **"Applying Natural Computing to Stock Market Portfolio Management"** submitted by **Bharatkumar Vikrambhai Chawda** has been examined by me. I undertake the following:

a. The thesis has significant new work/knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.
b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as Author own work.
c. There is no fabrication of data or results which have been compiled/analyzed.
d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.
e. The thesis has been checked using **https://www.scanmyessay.com** (copy of originality report attached) and found within limits as per GTU Plagiarism Policy and instructions issued from time to time (i.e. permitted similarity index <10%).

Signature of the Research Scholar: ………………………………………..

Name of Research Scholar: **Bharatkumar Vikrambhai Chawda**

Date: 21/07/2020 Place: **Vallabh Vidyanagar**

Signature of Supervisor: …………………………………

Name of Supervisor: **Dr Jayeshkumar M. Patel**

Date: 21/07/2020 Place: **Mehsana**

Viper Plagiarism Report

Thesis - 129990907001 Bharat.pdf **scanned May 28,**

Overall Score

9%

5.0% International Journal of Intelligent Systems an…
http://www.mecs-press.org/ijisa/ijisa-v9-n1/v9n1-5.ht

3.3% Stock Market Portfolio Management: A Walk
https://studylib.net/doc/8730228/stock-market-portfo

0.3% Modified Moth Search Algorithm for Portfolio …
https://link.springer.com/chapter/10.1007%2F978-98

0.1% Modeling and Simulation in Engineering, Econ…
https://b-ok.cc/book/2742625/c387e9

0.1% Portfolio selection problems in practice - arXiv ...
https://mafiadoc.com/portfolio-selection-problems-in-

A Hybrid Genetic Algorithm for a Two-Stage S…

(Note: The first two sources shown in this report are published work of author himself.)

Ph. D. THESIS Non-Exclusive License to GUJARAT TECHNOLOGICAL UNIVERSITY

In consideration of being a PhD. Research Scholar at GTU and in the interests of the facilitation of research at GTU and elsewhere, I, **Bharatkumar Vikrambhai Chawda** having Enrollment No. **129990907001** hereby grant a non-exclusive, royalty-free and perpetual license to GTU on the following terms:

a) GTU is permitted to archive, reproduce and distribute my thesis, in whole or in part, and/or my abstract, in whole or in part (referred to collectively as the "Work") anywhere in the world, for non-commercial purposes, in all forms of media;

b) GTU is permitted to authorize, sub-lease, sub-contract or procure any of the acts mentioned in paragraph (a);

c) GTU is authorized to submit the Work at any National / International Library, under the authority of their "Thesis Non-Exclusive License";

d) The Universal Copyright Notice (©) shall appear on all copies made under the authority of this license;

e) I undertake to submit my thesis, through my University, to any Library and Archives. Any abstract submitted with the thesis will be considered to form part of the thesis.

f) I represent that my thesis is my original work, does not infringe any rights of others, including privacy rights, and that I have the right to make the grant conferred by this non-exclusive license.

g) If third party copyrighted material was included in my thesis for which, under the terms of the Copyright Act, written permission from the copyright owners is required, I have obtained such permission from the copyright owners to do the acts mentioned in paragraph (a) above for the full term of copyright protection.

h) I retain copyright ownership and moral rights in my thesis and may deal with the copyright in my thesis, in any way consistent with rights granted by me to my University in this non-exclusive license.

i) I further promise to inform any person to whom I may hereafter assign or license my copyright in my thesis of the rights granted by me to my University in this non-exclusive license.

j) I am aware of and agree to accept the conditions and regulations of PhD including all policy matters related to authorship and plagiarism.

Signature of the Research Scholar: ………………………………………..
Name of Research Scholar: **Bharatkumar Vikrambhai Chawda**
Date: 21/07/2020 Place: **Vallabh Vidyanagar**

Signature of Supervisor: …………………………………
Name of Supervisor: **Dr Jayeshkumar M. Patel**
Date: 21/07/2020 Place: **Mehsana**

Seal:

Thesis Approval Form

The viva-voce of the PhD Thesis submitted by **Bharatkumar Vikrambhai Chawda** (Enrollment No. **129990907001**) entitled **"Applying Natural Computing to Stock Market Portfolio Management"** was conducted on Tuesday, 21st July 2020 at Gujarat Technological University.

(Please tick any one of the following options)

☑ The performance of the candidate was satisfactory. We recommend that he be awarded the PhD degree.

☐ Any further modifications in research work recommended by the panel after 3 months from the date of first viva-voce upon request of the Supervisor or request of Independent Research Scholar after which viva-voce can be re-conducted by the same panel again. (briefly specify the modifications suggested by the panel)

☐ The performance of the candidate was unsatisfactory. We recommend that he should not be awarded the PhD degree. (The panel must give justifications for rejecting the research work)

- -
Supervisor: **Dr. Jayesh M. Patel**

- -
External Examiner 1: **Dr. Manu Sood**

- -
External Examiner 2: **Dr. P.V. Virparia**

Acknowledgement / Dedication

In the words of Dr Alexander Elder, writing this part of the thesis is like having a dessert. I have an opportunity to express my gratitude to all those who helped me directly or indirectly in this long journey of more than seven years.

First of all, I would like to take this opportunity to express my immense gratitude towards each teacher of my life whose efforts, guidance, encouragement and blessings give me the ability to accomplish this research work.

This research work incorporates two different domains – finance (stock market portfolio management) and artificial intelligence (natural computing algorithms). I am thankful to my colleague Prof. B.I. Patel, due to whom I got interested in the stock market and Dr A.M. Panchal with whom I have shared many brainstorming sessions which helped to articulate my ideas more clearly regarding the portfolio management. I am also thankful to my friend Dr N.M. Sureja who introduced me to the field of natural computing – a simple yet one of the most effective ways to solve some of the most complex real-world problems.

The success of this work would not be possible without the guidance of my supervisor Dr J.M. Patel and DPC members Dr H.B. Bhadka and Dr C.K. Kumbharana. I am heartily grateful to them for their valuable suggestions that have improved the quality of my work, and, their support that helped to achieve this milestone.

I am thankful to my colleagues Prof. Paresh Tank and Prof. Lata Bhavnani who spent their valuable time from their overloaded schedules to read this entire thesis. They caught several oversights and helped me to rectify them, which might have been ignored completely otherwise.

I am also grateful to various experts who reviewed my research papers in different journals. They detected defects and pointed me out towards shortcomings in my work at different stages. These corrections are priceless and have helped me, perhaps, more than anything else in sharpening my work. I am thankful to all of such experts.

Apart from the research work, the PhD involves some non-obvious works too like travelling and communication. I have had great support in such activities whenever

required during this journey of research work. I am thankful to all those people, who helped me and made my tasks easier, including my friends – Dr Sanjay Patel, Prof. Sohil Gambhir, my cousins – Kalpesh, Sagar and Ajay, my late uncle Dinesh Chawda.

I am thankful to the Charutar Vidya Mandal and my institute B. & B. Institute of Technology, Vallabh Vidyanagar for providing me permission and support to do this research work. With this, I also express my gratitude towards the staff members of my institute for providing me motivation, support and a friendly environment. I also acknowledge the assistance of the PhD section, Gujarat Technological University.

In this entire endeavor, my Lappy (laptop) has remained a loyal companion since the beginning of this research work – without a single crash or hiccups. On many occasions, algorithm testing has run over several hours, and, this device has been utilized with full exhaustion. The support of this Lappy will remain in memory forever.

The chain of my gratitude would be incomplete without my deepest and sincere gratitude to my family members who have sacrificed the most in this endeavor. I am grateful to my parents for their blessings and best wishes all the time. I cannot express my feelings in words for my wife Saral who has stood with me in this entire journey, carrying out many responsibilities on her own, postponing many dreams time to time, and eagerly waiting for this work to be accomplished. I also cannot forget the cheerful distractions provided by my daughter Dhruvi, and particularly by my son Dhyanam, whose mischievous adventures have been proved to be the most challenging.

I am thankful to each of my friends and relatives who have helped me in this journey. I would also like to apologize to those whose names I could not mention here.

In the end, I would like to dedicate this work to all Part-Time Research Scholars like me. Friends, I have lived through the busyness of family and job related duties to be carried out along with research activities. It is not an easy task to balance both of these in any sense. I can understand and feel your pain. But, keep walking step by step. Milestone will be achieved – sooner or later.

- Bharat V. Chawda

Abstract

"Money must be invested to earn more money" to avoid erosion in the value of money. The stock market is one of the prime investment options. Though it can give lucrative returns over a period, an investment in stock market is risky. Due to the dynamic nature of the financial market, future returns to be realized from investments in the stock market are uncertain – making stock market portfolio management tricky. Stock market portfolio management deals with finding the optimum way of investing a particular amount in a set of stocks. To achieve this, portfolio management battles with dual goals – maximize returns, minimize risk, or both – making it multi-objective optimization problem.

Natural computing algorithms draw inspiration from the phenomena in the natural world. These algorithms have shown great potential and flexibility in solving real-world complex optimization problems. This research work applies natural computing to manage the stock market portfolio. A novel natural computing algorithm – **Winnowing Algorithm** – has been proposed for the same whose design draws inspiration from the real world winnowing process. This process is used to separate heavier and lighter components from a mixture by the help of the wind. The cardinality constrained mean-variance model has been used as the mathematical model to represent various aspects related to the stock market portfolio.

The research work has been carried out in three phases. The first phase focuses on the optimization of weights for given stocks to be included in a portfolio. A small dataset, provided in the literature, has been used to test the performance of the algorithm. The second phase focuses on the selection of stocks and their weights to construct a portfolio. Five different datasets, given in the OR-Library and representing the stock markets of five different countries, have been used to assess the reliability and scalability of the algorithm. Obtained experimental results are compared and analyzed with other state-of-the-art algorithms. The third phase focuses on the optimization of the portfolio over a period. The real world stock data derived from the National Stock Exchange of India have been used in this phase. The performance of the portfolio is compared and analyzed against the performance of the NIFTY50 - an index of the National Stock Exchange of India. Conclusions derived from this analysis prove the efficiency and reliability of the Winnowing Algorithm to manage the stock market portfolio successfully.

Table of Contents

List of Abbreviations

Abbreviation	Meaning
Δ-metric	Spread Metric
ABC	Artificial Bee Colony
AE	Algorithmic Effort
AET	Average Execution Time
BA	Bat Algorithm
BFO	Bacterial Foraging Algorithm
CCMV	Cardinality Constrained Mean-Variance
EP	Efficient Point
ER	Error Ratio
EXR	Expected Return
FA	Firefly Algorithm
FWA	Fireworks Algorithm
FY	Financial Year
GA	Genetic Algorithm
ICPSO	Improved Particle Swarm Optimization
IHS	Improved Harmony Search
IPSO-SA	Improved PSO Hybridized with SA
KH	Krill Herd
MED	Mean Euclidean Distance
MedPE	Median Percentage Return Error
MPE	Mean Percentage Deviation Error
MRE	Mean Return Error
NCA	Natural Computing Algorithm
NSE	National Stock Exchange
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
PSO	Particle Swarm Optimization
PSO-MP	PSO Hybridized with Mathematical Programming
SM	Scout Mechanism
S-metric	Spacing Metric
SR	Sharpe Ratio
TR	Turnover Ratio
uFA	Upgraded Firefly Algorithm
VaR	Value-at-Risk
VRE	Variance of Return Error
WA	Winnowing Algorithm
WA-PC	Winnowing Algorithm adapted for Portfolio Construction
WA-PWO	Winnowing Algorithm adapted for Portfolio Weight Optimization

List of Symbols

Symbol	Description
N	Total number of available stocks to construct portfolio
K	Total number of stocks to be included in a portfolio
σ_{ij}	Covariance between asset *i* and *j*
w_i	Weight associated with stock *i*
r_i	Expected return of asset *i*
r	Expected return of the portfolio
λ	Risk aversion parameter, or, Risk-return trade-off coefficient
z_i	If $z_i = 1$, asset *i* is present in portfolio; if $z_i = 0$, asset *i* is absent in portfolio
K_L	Lower bound on the number of stocks to be included in a portfolio
K_U	Upper bound on the number of stocks to be included in a portfolio
w_{min}	Floor constraint, or, Lower bound on the weight of stocks
w_{max}	Ceiling constraint, or, Upper bound on the weight of stocks
C_M	Class or Sector of similar stocks
$Wmin_m$	Lower bound on the weight of a class/sector
$Wmax_m$	Upper bound on the weight of a class/sector
l_i	Lot size of asset *i*
c_i	Buying price of asset *i*
Δ	Threshold for trading constraint
w_i'	Previous weight of stock *i*
B_i	Maximum buying threshold
S_i	Maximum selling threshold
r_{it}	Return at time *t* for asset *i*
n	Total number of returns below expected return
R_p	Mean return of the portfolio *p*
R_f	Return of benchmark index or risk-free return
σ_p	Standard deviation of a portfolio *p*
T	Time horizon
P	Pool or population size
γ	Replacement factor
S	Stagnation counter
Δw_{max}	Maximum fraction of weight used to update solutions
w_{ij}	Weight of stock j in i^{th} portfolio
η	Sum of excess weight for stocks in i^{th} portfolio
φ	Sum of deficient weight for stocks in i^{th} portfolio

$\boldsymbol{\varepsilon}$	Sum of w_{ij} where $w_{ij} < w_{max}$
$\boldsymbol{\delta}$	Sum of w_{ij} where $w_{ij} > w_{min}$
$\Delta \boldsymbol{w}$	Series of K random fractional weights
$\boldsymbol{\theta}$	Determines whether to exploit solution or to explore search space
$\boldsymbol{\lambda_{max}}$	Maximum allowed value for λ for a specific dataset
$\boldsymbol{v_j}$	Variance of the return for the portfolio j
$\boldsymbol{r_j}$	Mean of the return for the portfolio j
$\boldsymbol{\eta}$	Total number of portfolios in an obtained efficient frontier
$\boldsymbol{v_j^o}$	Variance of the return for the portfolio j in the obtained efficient frontier
$\boldsymbol{r_j^o}$	Mean of the return for the portfolio j in the obtained efficient frontier
e	Minimum Euclidean distance between obtained portfolio and standard efficient frontier
$\boldsymbol{v_e^s}$	Variance of the return for the nearest portfolio in the standard efficient frontier
$\boldsymbol{r_e^s}$	Mean of the return for the nearest portfolio in the standard efficient frontier

List of Figures

List of Tables

CHAPTER 1

Introduction

This chapter commences by providing an overview of the research problem along with the attempted solution. The motivation behind the research work, associated challenges, hypothesis and objectives of the research work are discussed next. After this, the original contributions of the thesis and organization of the thesis are given.

1.1 Overview

According to the theory of 'Time Value of Money', money must be invested to earn more money [1]. Idle cash loses value over a period due to factors like inflation. Higher inflation increases the general price level of goods and services [2]. As a result, the purchasing power of the money goes down with time. Money should grow at a rate at least equal to the rate of inflation to keep its value afloat. A wise investment must beat inflation to fulfil this purpose. Investment decisions bring around two vital questions.

- In which assets to invest?
- In which proportion to invest?

These two questions build the base for portfolio management. The research work, presented in this thesis, addresses stock market portfolio management – considering it as a research problem to be solved. Natural computing is used to obtain answers for these questions – recognizing it as a solution to the research problem.

1.1.1 Stock Market Portfolio Management

For investment, money is dispensed among a few resources, called assets. Portfolio refers to such allocation of wealth (or, money in hand) among several assets [3]. Many assets are available to make investments. Some of the familiar assets are real estate, gold/silver, metals, commodities, stocks, mutual funds, derivative instruments, bonds, fixed deposits, and post office savings.

These assets can be classified into two categories: risky assets and risk-free assets [4]. If the return to be realized in the future is uncertain for an asset, it is referred to as a risky asset. For example, consider that an investor purchases the stock of some company with the intention to hold it for a year from today. The return to be obtained by this investor after a year is uncertain. It depends on the stock price of that company at the time of sale and the dividends paid by the company. Thus, stocks are risky assets.

In contrast, if the return to be realized in the future is known with certainty today for an asset, it is referred to as a risk-free (or, risk-less) asset. For example, consider that an investor makes a fixed deposit in a bank for time duration of one year. The return to be obtained by this investor after a year is certain. It depends on the interest rate provided by that bank for fixed deposits. Thus, a fixed deposit in a bank is a risk-free asset.

Stock market portfolio refers to the allocation of wealth among stocks of several companies. Stocks are referred to as shares also. The stock market portfolio can yield good returns over a long period. Stock market portfolio management deals with finding the optimum way of investing a particular amount of money in a given set of stocks. The stock market portfolio management aims to achieve dual primary goals. The first is to maximize the return, reward or profit, and, the second is to minimize the risk.

Managing stock market portfolio involves two significant tasks: Portfolio Construction and Portfolio Optimization. *Portfolio Construction* selects ‘specific stocks and their weights’ from a large pool of available stocks to make the initial investment. The prices of stocks vary over a period due to the dynamic nature of the financial market. The portfolio must be adjusted to reflect these changes. So, once an initial portfolio is constructed, *Portfolio Optimization* continuously rebalances this portfolio by buying new stocks in a portfolio, selling existing stocks of the portfolio, or both.

Based on these primary tasks, the research work presented in the thesis has been divided into three different phases. The first phase is aimed to optimize only the weights of pre-selected stocks to be included in a portfolio. The second phase is intended to construct a portfolio by selecting stocks and their weights. The third phase is targeted to optimize portfolio by continuously rebalancing it.

1.1.2 Natural Computing Algorithms

It is said that – "*Nature is the best teacher.*" Nature is there since millenniums. Many natural elements have survived or evolved against hard complexities of the real world in these years. They have provided over-the-top solutions to complex real-world problems by applying simple approaches in systematic manners – proving their ability in tackling such issues successfully. This aspect has remained the core motivation behind natural computing algorithms and has inspired many researchers to mimic the nature in technology.

Natural Computing Algorithms (NCAs) are computer algorithms whose design draws inspiration from phenomena in the natural world [5]. NCAs are also known as Nature-Inspired Algorithms or Clever Algorithms. NCAs have provided optimal solutions to a large number of complex real-world problems within a reasonable time durations. As a result, natural computing has established itself as an essential field of computing – encompassing a wide range of real-world applications.

NCAs work in an iterative manner. They apply two prime aspects in each iteration: Exploration and Exploitation [6]. *Exploration* relates to generating diverse solutions to explore search space as broadly as possible. It helps to avoid local minima (or maxima). *Exploitation* relates to improving the quality of generated solutions by applying local search or other means. It aims to find a better solution compared to the existing solution. NCAs keep exploring search space and exploiting generated solutions iteration by iteration until an optimal solution is found.

There are four crucial factors [7] behind the success and acceptance of NCAs among broad category of researchers: 1) These algorithms are relatively *simple*. They use simple concepts derived from nature. 2) These algorithms are *flexible*. They can be applied to different problems without significant structural changes in the algorithm. Researchers only need to know how to represent their problem for NCAs. 3) These algorithms use *derivation-free mechanisms*. They find optimal solutions stochastically without applying derivation in contrast to gradient-based optimization approaches to help researchers to avoid expensive (and sometimes unknown) derivation methods. 4) These algorithms can *avoid local optima (either minima or maxima)*. The stochastic nature of NCAs can explore the entire search space extensively – preventing them from sticking in local solutions. These four features provide NCAs superiority over traditional algorithms such as exact methods involving logical or mathematical programming.

A novel natural computing algorithm, named *Winnowing Algorithm*, has been presented in this thesis. The design of this algorithm is inspired by the real world winnowing process which is used to separate heavier and lighter particles from a mixture by the help of wind. This algorithm has been designed and implemented to manage the stock market portfolio.

1.2 Motivation

Stock prices are dynamic. Their performance depends upon many factors. Investment in some stocks may give positive returns. Investment in some stocks may provide negative returns. This fact can be observed from the annual returns given in Table 1.1 for the Indian stock market.

Table 1.1 Top Five Gainer & Loser Stocks of NIFTY50

Rank	FY 2015-16		FY 2016-17		FY 2017-18	
	Company	**Return (%)**	**Company**	**Return (%)**	**Company**	**Return (%)**
1	EICHERMOT	20.41	HINDALCO	122.91	BAJFINANCE	51.07
2	BAJAJ-AUTO	18.69	IOC	95.65	HINDUNILVR	45.88
3	ZEEL	13.28	YESBANK	79.23	MARUTI	45.38
4	ULTRACEMCO	11.76	MARUTI	62.57	TECHM	39.76
5	HEROMOTOCO	11.58	TATASTEEL	51.96	RELIANCE	31.55
0	**NIFTY50**	**- 8.79**	**NIFTY50**	**18.86**	**NIFTY50**	**9.69**
5	TATAMOTORS	-29.59	WIPRO	-9.05	BOSCHLTD	-20.42
4	ONGC	-30.26	DRREDDY	-12.84	DRREDDY	-21.33
3	HINDALCO	-31.82	INFY	-15.93	SUNPHARMA	-28.12
2	IDEA	-39.50	SUNPHARMA	-16.18	TATAMOTORS	-30.20
1	BHEL	-51.72	IDEA	-22.31	LUPIN	-49.25

This table represents the top five gainer and loser stocks of NIFTY50 for the three financial years (FYs) 2015-16, 2016-17 and 2017-18. The mid-line in this table with NIFTY50 represents the performance of the overall Indian Stock Market in terms of NIFTY50 index. The required comparative data for this table has been calculated from the data collected from the NSE Equity Bhavcopy [8].

Observe that, irrespective of the performance of the overall market, some stocks give good returns while some stocks give poor returns. For example, the market was down in FY 2015-16 with -8.79% return, but EICHERMOT gave positive returns of 20.41%. In contrast to this, the market was up by 18.86% in FY 2016-17, but IDEA gave negative returns of -22.31%.

Another observation to be noteworthy is the fact that no stock is consistent in performing good as well as bad. For example, HINDALCO has given a negative return of -31.82% during FY 2015-16. However, the same stock has remained top gainer during FY 2016-17 with huge positive returns of 122.91%.

It can be observed from this table that the right selection of stocks along with their weights can provide excellent returns. Also, the performance of stocks needs to be tracked continuously. Moreover, whenever required, the portfolio should be rebalanced to optimize performance. In a nutshell, it can be inferred from these observations that the management of the stock market portfolio possesses high scope of the research.

1.3 Challenges

The stock market portfolio aims to maximize returns and minimize risks. However, stock performance depends upon many factors including company fundamentals, scope of business, government policies, monetary policies of central banks, inflation, crude oils prices, etc. So it becomes challenging to predict the performance of the stock and to select stocks which can give better returns in the future consistently.

Regardless of whether better performance is considered, to select K stocks out of a pool of N accessible stocks, there are N!/(K!(N-K)!) different ways. So, to pick ten stocks out of 100 accessible stocks, there are around 10^{13} unique permutations. Complexity does not stop here. The explanation is: if the selection of stocks falls in a discrete space, decision of their weights in a portfolio will fall in a continuous space. The weight of an individual stock in a portfolio can be anything from as low as 0.1 to as high as 1. With either increase in N or K, it turns out to be practically unfeasible to assess each combination of stocks and their weights in a reasonable period.

Some constraints are added to make a stock market portfolio more realistic. For example, the floor and ceiling constraint binds the weights of selected stocks within lower and upper limits. Cardinality constraint restricts a total number of stocks to be included in a portfolio to some maximum limit. Addition of such constraints brings portfolio management under the category of NP-Hard problems and makes traditional deterministic algorithms insufficient to provide optimal solutions within reasonable time duration.

Stock market portfolio management requires an in-depth knowledge of the finance, computer science and mathematics – making the problem of stock market portfolio management interdisciplinary problem. Also, stock market portfolio management aims to accomplish two targets – maximize or boost return and minimize or limit risk – which makes it a multi-objective combinatorial optimization problem.

Overall, managing stock market portfolio is a challenging task for researchers as well as investors and fund managers.

1.4 Hypothesis

The hypothesis behind the research work presented in this thesis is given here:

- Natural computing algorithm can be applied to manage the stock market portfolio, so that –
    - The return is maximized for a given risk, or, the risk is minimized for a given return.
- Appropriately selected stocks and their weights can outperform the overall stock market.

This hypothesis infers that stock market portfolio managed using natural computing algorithm can yield higher returns with comparatively lower risk in context to the overall stock market.

1.5 Objectives

The research work presented in this thesis aims to achieve the following primary objectives:

- To study and understand the natural computing along with stock market portfolio management.
- To develop an algorithm based on natural computing –
  - To select *K* stocks out of *N* stocks along with their respective weights, so that,
  - The return is maximized for a given risk, or,
  - The risk is minimized for a given return.
- To investigate the performance of the developed algorithm.

1.6 Original Contribution by the Thesis

The original contributions of this research work can be summarized as below:

- A novel natural computing algorithm – Winnowing Algorithm (WA) – has been designed and implemented to manage the stock market portfolio. The underlying winnowing process, based on which this algorithm is inspired, has been explained.
- In the first phase of the research work, WA has been adapted to optimize weights of the stocks to be included in a portfolio. The algorithm has been tested on the dataset of five fixed assets, given in the literature, to investigate the performance. Obtained results have been compared and analyzed with other state-of-the-art algorithms.
- In the second phase of the research work, WA has been adapted to construct a portfolio by selecting stocks and their weights from an available pool of stocks. The algorithm has been tested on five different datasets given in OR-Library. Obtained experimental results have been compared and analyzed with other state-of-the-art algorithms to investigate the reliability and scalability of the algorithm.
- In the third phase of the research work, WA has been adapted to optimize the

portfolio over a period. The real world stock data obtained from the National Stock Exchange (NSE) of India has been used in this phase. To evaluate the effectiveness of the portfolios constructed by WA, returns given by different portfolios have been compared with the return of the NIFTY50 which is an index representing well diversified 50 stocks of the NSE of India.

- Updating weights of stocks is an inherent part of the portfolio weight optimization as well as portfolio construction. During this process, it is necessary to normalize weights of stocks so that they conform to the various real-world constraints. This thesis presents a novel approach to normalize weights of stocks which is faster and more efficient than other methods available in the literature.

1.7 Organization of the Thesis

This thesis has been organized into seven different chapters. A brief overview of each chapter is given below.

Chapter one is an introductory chapter. Overview of the stock market portfolio management along with natural computing algorithms is presented at the beginning of the chapter. This overview is followed by the discussion regarding the motivation behind the research work, challenges associated with portfolio management, hypothesis behind the research work and objectives of the research work. In the end, original contributions to be made in this thesis are discussed.

Chapter two provides a theoretical background for the stock market portfolio management. It also provides a literature review of natural computing algorithms as well as their applications to portfolio management. It discusses topics such as portfolio management approaches, risk management, Markowitz's classical mean-variance model and enhancements to this model. Mathematical formulation of the research problem in the form of stock market portfolio management is provided next. This follows the summarized list of various natural computing algorithms. In the end, a review of most relevant research papers, in the context of stock market portfolio management using natural computing algorithms, is given.

Chapter three introduces the winnowing algorithm – a novel natural computing algorithm used to manage the stock market portfolio in this research work. The chapter begins with an explanation of the real world winnowing process, which is the inspiration behind the design of the winnowing algorithm. The pseudo code of the winnowing algorithm is presented with a brief description. In the end, the core steps of the winnowing algorithm are discussed in the context of portfolio management.

Chapter four presents the first phase of the research work and focuses on portfolio weight optimization. The chapter begins with the discussion of the standard dataset on which the performance of the algorithm is tested. After this, the adaptation of the winnowing algorithm to optimize portfolio weights is explained along with implementation details. Obtained experimental results are presented next. In the end, the performance of the winnowing algorithm is evaluated by comparing obtained results with other state-of-the-art algorithms.

Chapter five is dedicated to the second phase of this research work. It focuses on portfolio construction. The chapter begins with the discussion of the standard datasets provided in the OR-Library. Performance of the algorithm is tested using these datasets. After this, the adaptation of the winnowing algorithm to select stocks and their weights to construct a portfolio is explained along with implementation details. Obtained experimental results are presented next. In the end, the performance of the winnowing algorithm is evaluated by comparing obtained results with other state-of-the-art algorithms.

Chapter six is focused on the third phase of the research work and discusses portfolio optimization. The chapter begins with the discussion of the real-world dataset of the National Stock Exchange of India used to test the performance of the portfolios. This follows the description of static portfolios and performance issues associated with them. Later, dynamic portfolios are discussed which are allowed to change stocks and weights over a while. In the end, obtained experimental results are examined for the comparative analysis purposes, and the ability of the winnowing algorithm is tested to manage the real-world stock market portfolio.

Finally, **Chapter seven** presents conclusion and future work.

CHAPTER 2

Theoretical Background and Literature Review

This chapter provides an academic background for the stock market portfolio and literature review of natural computing algorithms as well as their applications to portfolio management. It discusses topics such as portfolio management approaches, risk management, Markowitz's mean-variance model and enhancements to this model. This discussion leads to the mathematical formulation of the research problem used in this thesis. This follows the summarized list of various natural computing algorithms. Along with this, review of most relevant research papers, in the context of stock market portfolio management using natural computing algorithms, is presented in this chapter.

2.1 Portfolio Management Approaches

Three different approaches (or strategies) are in use to manage stock market portfolio: Active Management, Passive Management, and Hybrid Management [9][10]. These approaches are discussed in the following subsections.

2.1.1 Active Management

Active management [11] approach is known as the traditional approach. The core belief behind active management is that financial markets are not sufficiently efficient. And, it is possible to outperform the overall market by skilful investment in individual stocks. In this approach, market inefficiencies are exploited by buying undervalues stocks or short selling overvalued stocks. Timing the market, i.e., when to purchase and when to sell, is extremely critical for the better performance of this methodology. Frequent trading might be involved in this approach which increases transaction costs and reduces effective returns. If a fund is managed by using this approach, the associated fees will be higher compared to passively managed funds. Decisions regarding when to buy or sell a stock are taken based on fundamental, technical analysis, or both [12].

Fundamental analysis searches for the real value of a stock using various factors. Some of the notable components utilized by the fundamental analyst are retained earnings, earnings yield, price-earning (PE) ratio, government approaches, company fundamentals, etc. Difference to this, Technical analysis [13], [14] looks for the perceptions of the real value of a stock utilizing factors such as movement of historical price & volumes. For this reason, charts of historical price and volume data are utilized alongside different technical indicators. Some of the notable indicators utilized are Moving Average (MA), Moving Average Convergence and Divergence (MACD), Relative Strength Index (RSI), On Balance Volume (OBV), and so forth.

In India, most of the stockbrokers follow this approach for selecting a portfolio for their clients.

2.1.2 Passive Management

Passive management [15] approach is also known as a modern approach. The core belief behind passive management is that financial markets are efficient. And, it is impossible to outperform the overall market or to time the market consistently. Passive management strives to mimic the performance of the benchmark in terms of risk and return. The benchmark can be a stock market index such as SENSEX or NIFTY50. Portfolio gets good diversification, low management fees, low turnover, and so, low transaction costs in this approach. Passive management is often referred to as index tracking. A passively managed fund is called an indexed exchange traded fund (ETF). Investors interested in passive management can buy one or more such ETFs. Passive management applies two different ways to match the performance of an index: full replication and partial replication [9].

In full (or complete) replication, all the stocks comprising an index are purchased in the same proportion as in the index. This method produces a perfect replica of the index, but, the size of the portfolio, i.e., the total number of stocks in a portfolio, increases. This may also increase transaction cost comparatively. Contrast to this, in partial replication, only a subset of stocks comprising an index is purchased. Stocks can be selected such that each sector can have representation in the portfolio, or, those that have the best chance of good performance. This method reduces transaction costs but introduces a tracking error – the measure of the deviation of the chosen portfolio from the index.

2.1.3 Hybrid Management

Hybrid Approach applies the mixed strategy of active management and passive management. The core belief behind the hybrid management is that it is possible to outperform the overall market by systematically investing in a subset of stocks comprising an index. This approach attempts to achieve higher returns compared to that of the overall market similar to active management. It also tries to reduce transactions, and so transaction costs, similar to passive management.

The Mean-Variance model developed by the Nobel Prize winner Harry Markowitz [16], [17] can be used for this approach. This model represents modern portfolio theory and helps to optimize risk and return of a portfolio. Section 2.3 describes this model in greater detail.

This research work is focused on the hybrid approach to manage the stock market portfolio. The NIFTY50, index of the National Stock Exchange (NSE) of India, is used as a benchmark index.

2.2 Risk Analysis

Stocks are risky assets. Stock price does not remain constant and varies with time. Performance of a stock depends upon many factors. So, returns that can be realised in the future by investing in stocks are uncertain. This uncertainty introduces risk. Analyzing risks associated with assets is helpful to make wiser plans for investments. Following subsections discuss risk elements and risk mitigation techniques.

2.2.1 Risk Elements

The elements of risk are the factors that cause variations in stock prices over a period. These elements can be broadly arranged in two classifications – Systematic risk and Unsystematic risk. The following Figure 2.1 depicts this classification.

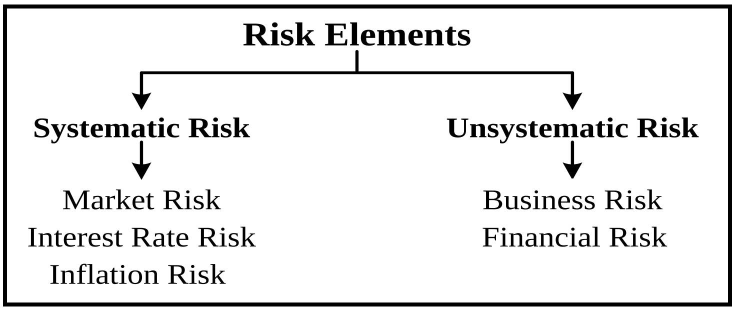


Figure 2.1 Risk Elements

The systematic risk [18], [19] comprises factors that are external to a company and macro in nature. It has an effect on the entire market by causing fluctuations in stock prices of all companies simultaneously. This risk is also referred to as aggregate risk or undiversified risk. It can be further subdivided into market risk, interest rate risk, and inflation risk. Market risk arises due to situations such as recession (or boom), political developments, and natural disaster. If the overall market falls, comparatively good stocks also fall as investors tend to follow the market direction. Interest rate risk arises due to changes in the interest rates from time to time. Such changes affect the borrowing power of the company. Inflation affects the purchasing power of money, thus, eroding the effective realized returns.

Systematic risk of some specific stock or portfolio can be known from its beta. A beta is a measure representing how volatile stock or a portfolio is compared to the overall market. A beta value more than one indicates higher systematic risk and less than one indicates lower systematic risk. If beta equals to one, systematic risk is the same as that of the market.

Unsystematic risk [20], [19], also known as idiosyncratic risk, incorporates factors that are internal to a company and micro in nature. It only affects a particular company or companies of a similar industry. This risk is also referred to as a specific risk, diversified risk, or residual risk. It can be further subdivided into business risk and financial risk. Business risk arises due to factors such as business cycles, technological changes, changes in customer preferences, the rise in competition. For example, the advent of cell phones with in-built cameras has affected the industry of digital cameras very severely. Financial risk, also known as leveraged risk, arises due to changes in the capital structure of the company. It can be expressed in terms of the debt-equity ratio. If a company has high debt, a large amount of the profit is spent on payment of interest of this debt. It reduces the actual profit of the company and so, real returns from the stock.

2.2.2 Risk Mitigation

The stock has two different kinds of risk associated with it: systematic risk and unsystematic risk. Total risk related to stock is a summation of these two risks.

Unsystematic risk can be reduced by including more than one stock in a portfolio. This strategy is known as risk diversification. Figure 2.2 given below delineates how unsystematic risk, thus total risk, can be decreased by increasing the number of uncorrelated stocks in a portfolio. Diversification applies the simple philosophy of '*Do not put all your eggs in one basket*'.

In contrast to unsystematic risk, systematic risk cannot be diversified by including multiple stocks in a portfolio. However, it can be mitigated by hedging and asset allocation. The least complex approach to hedge is to purchase derivative products, for example, puts. Systematic risk can also be reduced by allocating wealth among other assets in addition to stocks. If so, it is again diversification of a different kind – in a broader context.

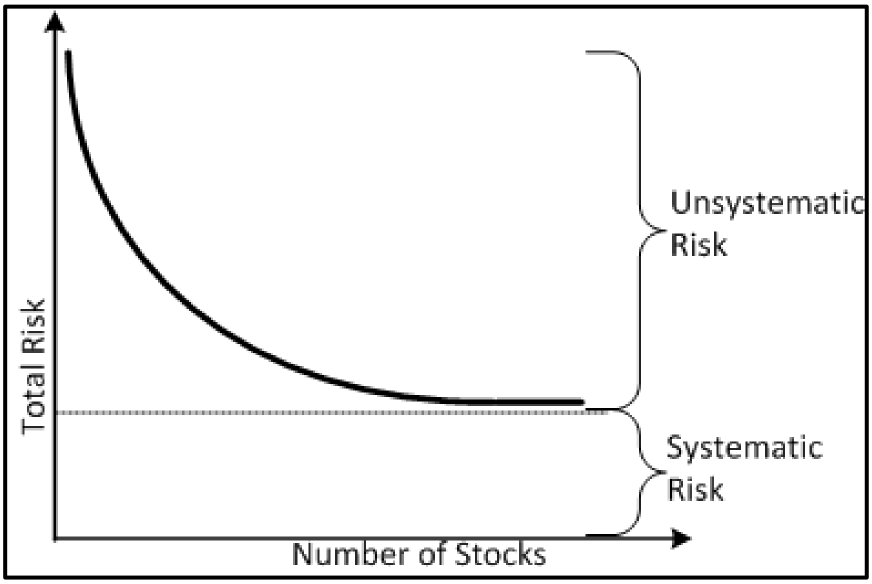


Figure 2.2 Risk Diversification

In the context of the stock market portfolio where investor only wants to invest in stocks, if the investor is not active in the derivatives market, systematic risk cannot be mitigated. Or say, all equity investors need to bear the systematic risk. So the only way available to reduce total risk is to target lower unsystematic risk by diversifying investment among more than one stock. In this case, the selection of stocks to construct portfolio becomes more crucial.

2.3 Markowitz's Model

Harry Markowitz introduced a simplified model to construct a portfolio. He is known as the father of the modern portfolio theory. He has also been awarded Nobel Memorial Prize (jointly) in Economic Science in 1990. The standard model given by Markowitz, efficient frontier, multi-objective versus single objective optimization in the context of portfolio management and criticisms of this model are discussed in this section.

2.3.1 The Standard Mean-Variance Model

The standard model [4], [16], [17] given by Markowitz is concerned with two properties of an asset: Risk and Return. This model is based on two-fold features desired by any investor: Ensure a certain return, and, minimize risk. This model focuses on the diversification of assets to mitigate risk. The essence of this model is that, instead of the risk of an individual asset, its contribution to the overall risk of a portfolio is more important. The expected return and risk are measured in terms of mean and variance of past returns. So, this model is also known as the Mean-Variance Model.

This model defines a portfolio as a vector of real numbers – representing the weight (i.e., the proportion with resepct to the total amount invested) corresponding to each asset available for the portfolio. The model then attempts to minimize the risk for the desired level of return and the other way around. *Risk of an asset* is defined as a variance (or standard deviation) of the past returns. The *expected return of an asset* is modelled as a random variable whose expected value is estimated from the past returns. The *expected return of the portfolio* is defined as a weighted average of the expected return of each asset in a portfolio. *Risk of the portfolio* is defined as the sum of the variances of the assets and covariance among the assets.

Mathematically this model can be represented as given below:

For N asset (or stock) problem,

$$Minimize\ Risk = \sum_{i=1}^{N}\sum_{j=1}^{N}\sigma_{ij}\ w_i\ w_j \tag{2.1}$$

Provided,

$$Return = \sum_{i=1}^{N} w_i \, r_i \geq r \tag{2.2}$$

$$\sum_{i=1}^{N} w_i = 1, \quad and \tag{2.3}$$

$$w_i \geq 0, \quad i = 1,2, \ldots, N \tag{2.4}$$

Where σ_{ij} is the covariance between asset *i* and *j*; w_i represents the weight assigned to asset *i*; r_i is the associated expected return of asset *i*; r is the expected return of the portfolio. Equation (2.3) specifies a budget constraint that the sum of the asset weights should be equal to 1 (i.e., 100%). Equation (2.4) specifies that weights should be positive, or in other words, no short selling is allowed.

2.3.2 Multi-Objective v/s Single Objective Optimization

Portfolio management is a multi-objective optimization problem and aims to achieve dual goals: maximize the return and minimize the risk. In practice, these two objectives are transformed into two different single objectives as given below:

- Maximize returns for a given level of risk, or,
- Minimize risk for a given level of return.

The first objective is relevant for a pre-selected risk level, while the second objective is applicable for a pre-selected return level. Both objectives simplify the process of constructing an *optimal portfolio*. Markowitz's standard mean-variance model aims to minimize risk for a given level of return. For this purpose, Equation (2.1) is considered as an objective function while Equation (2.2) is considered as a constraint.

An alternative way of transforming multi-objective optimization into a single objective is to introduce risk aversion parameter, or a risk-return trade-off coefficient, $\lambda \in [0, 1]$. Such type of mechanism has been used in [21], [22], [23], [24]. By using this parameter, the objective function of the model can be expressed as:

$$Maximize \;\; \lambda \sum_{i=1}^{N} w_i \, r_i - (1 - \lambda) \sum_{i=1}^{N} \sum_{j=1}^{N} \sigma_{ij} \, w_i \, w_j \tag{2.5}$$

With this Equation (2.5), if $\lambda = 1$, portfolio attempts to maximize return and ignores the risk entirely. In this case, the portfolio may have a single asset with a maximum expected return. If $\lambda = 0$, portfolio attempts to minimize risk and ignores the return completely. In this case, the portfolio may have several assets with low variances. Different values of λ, ranging from 0 to 1, can be used to construct different portfolios which suit the personal risk-return profiles of various investors.

2.3.3 Efficient Frontier

The expected return walks in parallel with the risk. High expected return brings high risk. If the risk is reduced, so reduces expected return. The standard mean-variance model [16], [17] builds this risk-return relationship for any feasible portfolio. A *feasible portfolio* is any portfolio that can be constructed from the available set of assets. The collection of every feasible portfolio is referred to as a *feasible set of portfolios*. Any feasible portfolio has associated with it is some risk and return. A feasible portfolio will simply referred to as a portfolio now onwards.

An *efficient portfolio* is a portfolio with the highest expected return among all feasible portfolios with the same level of risk. It is also referred to as a mean-variance efficient portfolio or non-dominated portfolio for a given level of risk. For different levels of risk, there can be different efficient portfolios. The collection of all efficient portfolios is referred to as the *efficient set*. It is also referred to as the *efficient frontier* because in graphical representation all the efficient portfolios lie on the boundary of the set of feasible portfolios having maximum return for a given level of risk. A portfolio which lies below the efficient frontier is called an *inefficient portfolio*. An *optimal portfolio* is an efficient portfolio that suits the risk-return profile of an investor. For different investors, optimal portfolios can be different.

The following Figure 2.3 depicts a feasible set of portfolios along with efficient frontier. Portfolios are plotted on a two-dimensional Risk-Return space. Darken circles represent efficient (non-dominated) portfolios. Empty circles represent inefficient portfolios. Efficient portfolios represent the highest expected return for a given level of risk. Similarly, they also represent the lowest risk for a given level of expected return.

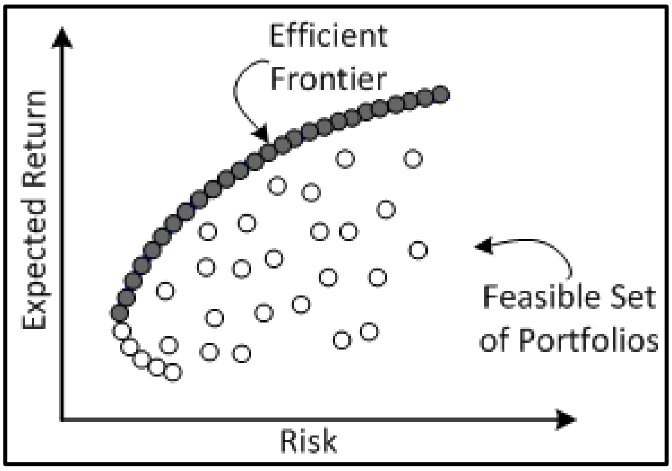


Figure 2.3 Efficient Frontier

Markowitz's standard mean-variance model is a multi-objective optimization model and attempts to maximize return for given risk and vice versa. If all possible portfolios are developed with considering above objectives, the result will be the entire efficient set of portfolios. But, different investors have different risk-return profiles. Also an individual investor will be keen on only a single optimal portfolio. So it will be costly in terms of time and effort to construct the entire efficient set first and then select an optimal portfolio.

In contrast to this, Equation (2.5) can be used as the optimization function. By resolving the problem now with different values of λ ranging from 0 to 1, an efficient frontier can be constructed similarly to Markowitz's standard model. An investor has to determine an optimal portfolio from this efficient frontier based on the personal risk-return profile. Alternatively, an optimal portfolio can be constructed directly by specifying some particular value of λ based on the risk-return profile of an investor.

2.3.4 Criticisms to the Standard Model

Markowitz's model formulates a pioneering methodology for the modern portfolio theory. Though, direct use of this model to manage real-world portfolio raises some critical issues. These criticisms are as follows:

- This model assumes a multivariate normal distribution for the rate of return of assets. But in practice, these distributions are asymmetric, and so, this assumption does not hold [25].

- This model only concerns for either risk or return [26]. It cannot provide portfolio by considering both of them. In other words, a portfolio with specific risk and return cannot be provided by this model. Also, investors may not be apparent with their expected returns and risk tolerances – two essential elements to construct portfolios with this model.

- The primary input data in this model are expected return and variance of returns of assets. In a simple approach, arithmetic mean of past return is considered as an expected return for an asset. A similar method is used to calculate variance. Now, if enough historical data of an asset is not available, it is difficult to estimate these values accurately [22]. In contrast to this, if the historical time duration is too long, this model gives the same influence to the earlier data as to the recent data [26]. The fact that recent data contains more information regarding the developments in the future price of an asset is ignored in this model. Also, the selection of the length of the actual duration is challenging and depends on the discretion of the investor.

- This model derives expected return and risk based on historical performance of the asset. But, the performance of a stock cannot be consistent over a longer time duration. Thus, to assume that, if a stock has performed either good or bad in the past, it will continue to do so in future as well is wrong.

- A risk measure used in this model, variance, equally accounts for upward and downward deviation in context to the mean value [25]. But in reality, only downward deviation represents loss (or risk). The upward deviation is not harmful and should be avoided in calculating risk measure.

- This model does not incorporate some aspects of real-world trading including a maximum number of assets to be included in a portfolio, transaction and management costs, short selling, minimum lot size, and preferences over various asset classes. Due to these reasons, this model is considered too simplistic to be used in the real world to manage portfolio [27].

- This model depends on the efficient market hypothesis [28]. It assumes that all investors are rational and risk-averse. It also believes that all information is available to all investors at the same time. But this cannot be true always. In reality, the market may contain insider trading, better-informed investors, etc.

- Correlations between stocks are assumed to be fixed and constant forever which is also not possible for real-world stocks.

2.4 Markowitz's Model – Enhancements

Criticisms discussed in the previous section shows the inability of the Markowitz's standard mean-variance model to be used for portfolio management in the real world. These criticisms have encouraged researchers to propose modifications to this model to adapt it to real-world requirements. Researchers have mainly emphasized on two aspects: 1) Considering real-world constraints, and, 2) Providing alternative risk measures. These measures have been summarized in [25] and [27].

2.4.1 Additional Constraints

Markowitz's model initially considered only two constraints: Budget constraint and No short-selling. The following other constraints can be looked at as per requirements to adapt this model for real-world portfolio management.

- **Cardinality Constraint:**

Cardinality constraint [21] restricts the total number of assets selected in a portfolio. A mathematical formulation of this constraint introduces a binary variable z_i. If $z_i = 1$, asset i is present in portfolio and $w_i > 0$. If $z_i = 0$, asset i is absent in portfolio and $w_i = 0$. This constraint has two versions. The *exact* version, given in Equation (2.6), imposes that the portfolio must have exactly K assets. The *soft* version, given in Equation (2.7), imposes lower bound (K_L) and/or upper bounds (K_U) on this number. This constraint facilitates portfolio management and helps to reduce its management costs.

$$\sum_{i=1}^{N} z_i = K \quad and \quad z_i = 0 \; or \; 1 \qquad (2.6)$$

$$K_L \le \sum_{i=1}^{N} z_i \le K_U \quad and \quad z_i = 0 \; or \; 1 \qquad (2.7)$$

- **Floor and Ceiling Constraint:**

Floor and Ceiling constraint [29] imposes lower bounds and/or upper bounds on the weight of each asset selected in a portfolio. The floor constraint represents a lower bound (w_{min}). It restricts negligible allocations of capital to chosen assets in a portfolio. Such restriction limits the number of stocks in a portfolio which helps to reduce administrative and transaction costs. The ceiling constraint represents the upper bound (w_{max}). It prevents too large allocation of capital to a single asset in a portfolio. Thus such capital is shared among several assets and helps to minimize risk by maintaining diversification in true essence. Mathematically these constraints can be represented as

$$w_{min} \leq w_i \leq w_{max}, \quad i = 1,2,\dots,N \tag{2.8}$$

- **Class/Sector Weight Constraint:**

Class/Sector weight constraint [30], [31] specifies lower and/or upper bounds on the weight of class or sector of assets. For example, stocks from the automobile industry may represent one sector. This constraint is similar to the floor-ceiling constraint. The difference is: it restricts the sum of weights of assets belonging to the same sector between some lower and upper bound instead of the weight of individual assets. Mathematically this constraint can be expressed as,

$$Wmin_m \leq \sum_{i \epsilon Cm} w_i \leq Wmax_m, \quad m = 1, 2, \dots, M \tag{2.9}$$

Where M is the set of classes or sectors C_1, C_2,…, C_M ; $Wmin_m$ is lower bound for class M; w_i represents the weight assigned to asset i and, $Wmax_m$ is upper bound for class M.

- **Short Sales Constraint:**

Markowitz's standard model does not permit assets to have negative values for weights restricting short selling of assets. Yet in reality, some markets permit investors to sell assets that are not yet possessed by them in anticipation of price falling. This is referred to as short selling [32]. It requires assets to have negative values for weights. This can be expressed as given below. It replaces constraint given by Equation (2.4).

$$w_i \epsilon R, \quad i = 1,2,\dots,N \tag{2.10}$$

- **Round Lot (Minimum Lot) Constraint:**

Round lot, or say Minimum lot, constraint [33] specifies that buying and selling of assets must be done in a multiple of the minimum transaction lots. Lot size varies from asset to asset. Such lots are also referred to as a round. Some markets, such as the Japanese one, and most of the derivative markets follow such type of concept for trading. With this constraint, weight w_i of an asset i in a portfolio can be expressed as,

$$w_i = \frac{l_i \, c_i}{\sum_{i=1}^{N} l_i \, c_i}, \quad i = 1, 2, \dots, N \tag{2.11}$$

where w_i is the weight, l_i is lot size, and, c_i is the buying price for asset i.

- **Trading Constraint:**

Trading constraint [34] specifies that if there is a change in weight w_i of asset i, the difference between current weight w_i and previous weight w_i' must be greater or equal to some threshold Δ. This constraint can be expressed as,

$$| w_i - w_i^{'} | \geq \Delta_i \quad or \ | w_i - w_i^{'} | = 0, \quad i = 1, 2, \dots, N \tag{2.12}$$

- **Turnover Constraint:**

Turnover constraint [30] specifies that the sum of the absolute change between current weight w_i and previous weight w_i' must be less than the maximum turnover ratio TR. This constraint is particularly useful when considering a multi-period investment horizon. The following equation represents this constraint.

$$\sum_{i=1}^{N} | w_i - w_i^{'} | \leq TR \tag{2.13}$$

In a similar way, various constraints for purchasing and selling are proposed in [34] as given beneath.

$$\max(w_i - w_i^{'}, 0) \leq B_i, \quad i = 1, \dots, N \tag{2.14}$$

$$\max(w_i^{'} - w_i, 0) \leq S_i, \quad i = 1, \dots, N \tag{2.15}$$

where B_i represents the maximum buying threshold, and S_i represents the maximum selling threshold.

2.4.2 Alternative Risk Measures

Markowitz's standard mean-variance model considered a variance as a risk measure. Variance equally considers variations in returns above and below expected returns as a risk. From an investor's perspective, only below expected return variations are harmful. And so, variance as a risk measure remained at the centre among criticisms of this model. Other risk measures, except variance, are discussed here.

- **Semi-Variance / Downside Risk:**

Semi-variance is an average of the squared deviations of returns that are below the expected return [35]. Semi-variance accounts only downside movements of the return distribution. So, this measure is also known as downside risk. The mathematical formula for semi-variance of asset *i* can be expressed as

$$Semi\text{–}\ variance = \frac{1}{n} \sum_{r_t < \mu}^{N} (r_{it} - r_i)^2 \tag{2.16}$$

Where r_{it} is a return at time *t* for asset *i*; r_i is the expected return of asset *i*; and *n* is a total number of returns below expected return.

- **Value-at-Risk (VaR):**

Value-at-Risk (VaR) [36] measures the worst expected loss over a given time period under normal market conditions at a given level of confidence. VaR is measured in three variables: a potential loss, likelihood (or confidence level) of that loss, and the time horizon. As explained in [37], consider a portfolio with 1-day VaR of $1 million at the 99% confidence level. This means that there is a 1% (derived as 100% – 99%) chance that the value of the portfolio will lose by $1 million or more during one day. A simple explanation, as well as calculation of VaR, is presented in [38].

- **Conditional Value-at-Risk (CVaR):**

Conditional Value-at-Risk (CVaR) [39] is an extension of VaR and can be derived by taking a weighted average between the VaR and losses exceeding VaR. It is also known as Mean Excess Loss, Tail VaR, or Mean Shortfall. VaR finds a minimum level of loss that can be expected. CVaR finds the expected (or mean) loss, given that the losses have exceeded VaR. So, the value of CVaR will be greater than or equal to that of VaR. CVaR helps investors to know the extent of risk. CVaR has been optimized in [40], [41]. The detailed comparative analysis between VaR versus CVaR is given in [42].

- **Sharpe Ratio (SR):**

Sharpe ratio (SR) [43], [44] is a measure of the excess return per unit of total risk of a portfolio. Excess return is also referred to as risk premium. It helps to calculate the risk-adjusted return. The return will be better with higher Sharpe ratio with the same unit of risk. The mathematical calculation for Sharpe ratio can be given as

$$Sharpe\ Ratio = \frac{(R_p - R_f)}{\sigma_p} \tag{2.17}$$

where p is the portfolio, R_f is a return of benchmark index or risk-free return, R_p is a mean return of the portfolio p, and σ_p is a standard deviation of a portfolio p. The numerator part of this equation, i.e., R_p - R_f, represents excess return.

- **Mean-Absolute Deviation (MAD):**

Mean-Absolute Deviation (MAD) is the average of absolute deviations from a mean value. Let $\{x_1, x_2, \ldots x_n\}$ be the data set, and μ is the mean value of this data set. MAD can be calculated mathematically as given by following formulae.

$$Mean\text{–}\ Absolute\ Deviation = \frac{1}{n}\sum_{i=1}^{n} |\ x_i - \mu\ | \tag{2.18}$$

A portfolio optimization model given in [45] uses MAD as a risk measure. In this model, the risk of a portfolio is defined as

$$\frac{1}{T}\sum_{t=1}^{T} \left|\sum_{i=1}^{N} w_i\,(r_{it} - r_i)\right| \tag{2.19}$$

where N represents total number of assets; T represents time horizon; w_i represents weight of asset i; r_{it} represents return of asset i at time t; and r_i represents mean return of asset i. In this model, covariance among assets is not required to measure risk. Thus, making it easier to deal with this model compared to Markowitz’s model. Similar to variance, MAD also accounts upward and downward deviations in equal proportion.

- **Mean Semi-Absolute Deviation (MSAD):**

Mean Semi-Absolute Deviation (MSAD) combines the concept of semi-variance with MAD to overcome the drawback of later. A portfolio selection model, given in [46], uses MSAD as a risk measure. In this model, the risk of the portfolio is defined as

$$\frac{1}{T}\sum_{t=1}^{T} \left| \min\left(0, \sum_{i=1}^{N} w_i\,(r_{it} - r_i)\right) \right| \tag{2.20}$$

where N represents total number of assets; T represents time horizon; w_i represents weight of asset i; r_{it} represents return of asset i at time t; r_i represents mean return of asset i.

2.5 Mathematical Formulation of the Research Problem

The research work presented in this thesis aims to manage the stock market portfolio. The enhanced mean-variance model which incorporates additional constraints such as cardinality constraint and floor-ceiling constraint has been used for this purpose. This model is known as the Cardinality Constrained Mean-Variance (CCMV) model. The mathematical formulation of this model can be given as below:

$$Portfolio\ Expected\ Return = \sum_{i=1}^{N} w_i\, r_i \tag{2.21}$$

$$Portfolio\ Risk = \sum_{i=1}^{N}\sum_{j=1}^{N} \sigma_{ij}\ w_i\, w_j \tag{2.22}$$

$$Maximize\ \ \lambda \sum_{i=1}^{N} w_i\, r_i - \ (1-\lambda)\sum_{i=1}^{N}\sum_{j=1}^{N} \sigma_{ij}\ w_i\, w_j \tag{2.23}$$

subject to,

$$\sum_{i=1}^{N} w_i = 1, \tag{2.24}$$

$$\sum_{i=1}^{n} z_i = K, \qquad z_i = 0\ or\ 1, \quad and \tag{2.25}$$

$$0 \le w_{min}\, z_i \le w_i \le w_{max}\, z_i \le 1, \ \ i = 1,2,\ldots,N \tag{2.26}$$

Where,

- N is the total number of stocks available to construct a portfolio
- w_i is the weight associated with stock i
- r_i is the associated expected return of stock i
- σ_{ij} is the covariance between stock i and j
- λ is the risk-return trade-off coefficient
- z_i determines the presence of stock i in a portfolio; If $z_i = 1$, stock i is present in the portfolio. If $z_i = 0$, stock i is absent in portfolio
- K is the total number of stocks to be included in a portfolio
- w_{min} is the lower bound on the weight of stocks
- w_{max} is the upper bound on the weight of stocks

Equation (2.24) specifies a budget constraint that the sum of the asset weights should be equal to 1 (i.e., 100%). Equation (2.25) specifies exact cardinality constraint and restricts the total number of assets to be included in a portfolio to K. Equation (2.26) specifies floor and ceiling constraints and restricts the weight of each asset to be included in a portfolio in the range of lower bound (w_{min}) and upper bound (w_{max}). This equation also specifies that weights should be positive, or in other words, no short selling is allowed.

Note that, Equation (2.23) is the objective function to be maximized in this model. Practically it cannot provide the perfect balance between portfolio return and risk as portfolio risk (variance) possesses smaller values compared to return. Based on the risk measure, it can be skewed towards either portfolio return or risk. So the selection of the λ should be made with utmost care.

2.6 Natural Computing Algorithms – A Summary

Similar to nature itself, natural computing algorithms have remained successful in providing optimal solutions to a large number of complex real-world problems within a reasonable time durations. Features like simplicity, flexibility, derivation free mechanisms and avoidance of local optima have provided NCAs superiority over traditional algorithms. Even though NCAs have widespread acceptance and use, according to *No Free Lunch* theorem [47], no single NCA is best suited as a solution to all the optimization problem. This fact has forced researchers to either develop new algorithms or to enhance existing ones.

Consequently, researchers have developed a number of various natural computing algorithms and their variations over the last couple of decades as listed out in Table 2.1 given below. NCAs have been listed in chronological order according to their proposal time in this table. This table also provides widely used abbreviations of NCAs, natural phenomena behind their inspirations, and names of researchers.

Table 2.1 Natural Computing Algorithms – A Summary

NCA & Abbreviation		Underlying Natural Phenomena	Author(s)	Refer
Genetic Algorithm	GA	Natural selection process that mimics biological evolution	Holland	[48], [49], [50]
Simulated Annealing	SA	Cooling process of molten metal	Kirkpatrick, Gelatt, Vecchi	[51], [52]
Memetic Algorithm	MA	Cultural revolution	Moscato	[53]
Ant Colony Optimization	ACO	Foraging behaviour of ants	Dorigo, Colorni	[54]
Genetic Programming	GP	Extension of GA – Solution is represented as a tree with a variable length	Koza	[55]
Particle Swarm Optimization	PSO	Flocking behaviour of birds	Kennedy, Eberhart	[56]
Differential Evolution	DE	Genetic evolution with mutation as an arithmetic combination of individuals	Storn, Price	[57]
Bacterial Evolutionary Algorithm	BEA	Microbial evolution phenomenon with the gene transfer operation	Nawa, Furuhashi	[58]
Artificial Immune System	AIS	Human immune system	Dasgupta	[59][60]
Evolution Strategies	ES	Adaption and evolution using natural selection	Beyer, Schewefel	[61]
Bacterial Foraging Optimization	BFO	Foraging behaviour of bacteria	Passino	[62]
Fish Swarm Algorithm	FSA	Schooling behaviour of fish	Li, Shao, Qian	[63]
Shuffled Frog Leaping Algorithm	SFLA	Frog leaping on stones in a pond	Eusuff, Lansey	[64], [65]
Social Cognitive Optimization	SCO	Human social cognition	Xie, Zhang	[66]
Invasive Weed Colony Optimization	IWCO	The ecological process of weed colonization and distribution	Mehrabian, Lucas	[67]

Table 2.1 Natural Computing Algorithms – A Summary (Continued)

NCA & Abbreviation		Underlying Natural Phenomena	Author(s)	Refer
Artificial Bee Colony	ABC	Foraging behaviour of bees	Karaboga, Basturk	[68], [69]
Group Search Optimization	GSO	Searching behaviour of animals and their group living theory	He, Wu, Saunders	[70]
Central Force Optimization	CFO	The metaphor of the gravitational kinematics and particle motion in a gravitational field	Formato	[71], [72]
River Formation Dynamics	RFD	How rivers are formed	Rabanal, Rodriguez, Rubio	[73], [74]
Intelligent Water Drops	IWD	Actions and reactions among water drop in a river	Shah-Hosseini	[75]
Roach Infestation Optimization	RIO	Social behaviour of cockroaches	Havens, Spain, Salmon, Keller	[76]
Monkey Search	MS	Mountain climbing process of monkeys	Zhao, Tang	[77]
Biogeography-Based Optimization	BBO	Distribution of species in nature over time and space	Simon	[78]
League Championship Algorithm	LCA	Competition of sports teams in a league championship	Kashan	[79]
Glowworm Swarm Optimization	GSO	The behaviour of glowworms – the capability to change the intensity of luciferin emission	Krishnanand, Ghose	[80]
Bumble Bees Mating Optimization	BBMO	Mating behaviour of bumble bees	Marinakis, Marinaki, Matsatsinis	[81]
Hunting Search Optimization	HSO	Group hunting behaviour of animals such as lions and wolves	Oftadeh, Mahjoob	[82]
Firefly Algorithm	FA	Flashing behaviour of fireflies	Yang	[83]
Harmony Search	HS	Improvisation process of musicians	Yang	[84]
Paddy Field Algorithm	PFA	Reproduction of plant populations	Premaratne, Samarabandu, Sidhu	[85]
Gravitational Search Algorithm	GSA	Low of gravity and resultant mass interactions	Rashedi, Nezamabadi-Pour, Saryazdi	[86]
Cuckoo Search	CS	Breeding behaviour of the cuckoo – laying colour-pattern mimicked eggs in nests of other birds	Yang, Deb	[87], [88]
Bat Inspired Approach	BIA	Echolocation behaviour of bats	Yang	[89]
Fireworks Algorithm	FA	Explosion processes of fireworks and mechanisms for maintaining the diversity of sparks	Tan, Zho	[90]
Plant Propagation Algorithm	PPA	Propagation of the plants, particularly Strawberry plants	Salhi, Fraga	[91]

Table 2.1 Natural Computing Algorithms – A Summary (Continued)

NCA & Abbreviation		Underlying Natural Phenomena	Author(s)	Refer
Collective Animal Behavior	CAB	Collective behaviour of different animal groups such as swarming, milling, migrating in aligned groups	Cuevas, González, Zaldivar, Pérez-Cisneros, García	[92]
Water Cycle Algorithm	WCA	Real world water cycle among transpiration/evaporation, condensation, precipitation	Eskandar, Sadollah, Bahreininejad, Hamdi	[93]
Krill Herd	KH	Herding behaviour of krill individuals	Gandomi, Alavi	[94]
Bacterial Colony Optimization	BCO	The behaviour of E. Coli bacteria at different development stages in their life cycle	Niu, Wang	[95]
Lion's Algorithm	LA	Social behaviour of lions that helps to keep themselves strong	Rajakumar	[96]
Stem Cells Optimization	SCO	Reproduction behaviour of stem cells	Taherdangkoo, Yazdi, Bagheri	[97]
Blind Naked Mole-Rats Algorithm	BNMR	Social behaviour of Mole-Rats	Shirzadi, Bagheri	[98]
Flower Pollination Algorithm	FPA	Fertilization/Pollination process of flowers	Yang	[99], [100]
Black Hole	BH	Star swallowing behaviour of black holes	Hatamlou	[101]
Cuttlefish Algorithm	CA	Mechanism of colour changing behaviour adopted by the cuttlefish	Eesa, Abdulazeez, Orman	[102]
Mine Blast Algorithm	MBA	The concept of a mine bomb explosion	Sadollah, Bahreininejad, Eskandar, Hamdi	[103]
Social Spider Optimization	SSO	Simulation of cooperative behavior of social spiders	Cuevas, Cienfuegos, Zaldívar, Pérez-Cisneros	[104]
Spider Monkey Optimization	SMO	Foraging behaviour of spider monkeys based on fission-fusion	Bansal, Sharma, Jadon, Clerc	[105]
Animal Migration Optimization	AMO	The behaviour of animals during migration from one location to another location	Li, Zhang, Yin	[106]
Bird Mating Optimizer	BMO	Mating strategies of birds	Askarzadeh	[107]
Forest Optimization Algorithm	FOA	Seeding procedure of the trees in a forest	Ghaemi, Feizi-Derakhshi	[108]
Grey Wolf Optimizer	GWO	The leadership hierarchy and hunting mechanism of grey wolves	Mirjalili, Mirjalili, Lewis	[7]
Vortex Search Algorithm	VSA	Vortex (swirl) pattern due to vertical flow of affected fluids	Doğan, Ölmez	[109]
Water Wave Optimization	WWO	Propagation, refraction, and breaking phenomena of shallow water waves	Zheng	[110]
Elephant Herding Optimization	EHO	Herding behaviour of elephant groups	Gai-Ge Wang	[111]
Raven Roosting Optimization	RRO	Social roosting and foraging behaviour of common raven	Brabazon, Cui, O'Neill	[112]

2.7 Natural Computing and Portfolio Weight Optimization

The weight of the stock in a portfolio represents the proportion of the amount invested in that particular stock. Weight optimization for the stocks to be included in a portfolio has a significant effect on the performance of the portfolio. Portfolio Weight Optimization is also referred to as weight selection of stocks. It comprises the first phase of the research work presented in this thesis. Note that portfolio weight optimization only focuses on optimizing weights of already selected stocks and does not apply cardinality constraint. If no constraints exist, traditional quadratic programming can be used with Markowitz's mean-variance model to generate portfolios. Though, many researchers have focused only on the weight optimization of stocks for initial assessment of the appropriateness of the newly proposed methods.

Sefiane and Benbouzian [113] has applied Genetic Algorithm (GA) [48],[49],[50] to portfolio weight optimization with three different crossover operators: single-point crossover, two-point crossover, and arithmetic crossover. Performance of the algorithm has been evaluated on a simple dataset. This dataset contains historical annual returns of five stocks (names not specified) for a period of five years from 2007 to 2011. This dataset has been used as a benchmark dataset to provide a comparative analysis of the performance by other researchers. Among three variations of GA applied here, arithmetic crossover yields better results compared to the other two varieties.

Tuba et al. [114] have applied the Firefly Algorithm (FA) [83][115] to portfolio weight optimization. The number of fireflies, i.e., population size, is kept 40. Two different variations have been used as the maximum number of allowed iterations – 6000 and 8000 – as termination criteria. This results in total 240000 (40*6000) and 320000 (40*8000) objective function evaluations respectively. Experimental results suggest improvement in FA behaviour with a higher number of assessments yielding better results.

Bacanin et al. [116] have applied Artificial Bee Colony (ABC) [68] to the portfolio optimization problem. The number of bees is kept 40. The algorithm was allowed to run for a total of 6000 iterations, resulting in total 240000 (40*6000) objective function evaluations.

Tuba et al. [117] have applied the Krill Herd (KH) algorithm [94] to portfolio weight optimization. The number of Krill agents, i.e., population size, is kept 40. Two different variations have been used as the maximum number of allowed iterations – 6000 and 8000 – as termination criteria. This results in total 240000 (40*6000) and 320000 (40*8000) objective function evaluations respectively. Experimental results suggest improvement in the performance of the algorithm with a higher number of evaluations yielding better results.

Bacanin and Tuba [118] have applied the Fireworks Algorithm (FWA) [90] to portfolio weight optimization. The number of fireworks is kept 5. The number of sparks is selected 50. Three different total numbers of iterations – 50, 4000 and 50000 – have been used as termination criteria. Experimental results suggest improvement in the performance of the algorithm towards a higher number of iterations yielding better results.

Strumberger et al. [119] have applied Bat Algorithm (BA) [89], hybridized with scout mechanism (SM) of Artificial Bee Colony (ABC) algorithm, to solve the portfolio weight optimization problem. The population size is kept 60. The total number of maximum iterations allowed is kept 4000 to terminate the algorithm. This results in 240000 (60*4000) objective function evaluations.

The same problem formulation and dataset as given in [113] have been used to assess the performance of these algorithms. Comparative analysis of experimental results presented in these papers shows that the best value of the objective function obtained by BA-SM outperforms all other approaches. In contrast to this, the average and worst value of the objective function obtained by FWA outperforms all other approaches. In other words, no single algorithm obtains good results accurately and consistently. This indicates that there is a window to further research in this area to find better results.

Also, the approaches applied in these papers only focuses on generating a single portfolio which maximizes return and minimizes risk. But in reality, different investors will have different risk-return profiles, and single portfolio cannot suit to all of them. So instead of generating a single portfolio, the algorithm should be able to create entire efficient frontier having optimal portfolios with different risk-return profiles. The winnowing algorithm presented here focuses on portfolio weight optimization in the first phase of the thesis and attempts to generate an efficient frontier with the same dataset used by these researchers.

2.8 Natural Computing and Portfolio Construction

Along with weight optimization, selection of stocks plays a crucial role in the performance of the portfolio. Portfolio construction, i.e., selection of stocks along with their weights, comprises the second phase of this research work and attempts to build the efficient frontier. Many researchers have applied various NCAs on constrained portfolios to obtain an efficient frontier.

Chang et al. [21] provided a pioneer work in applying NCAs to cardinality constrained portfolio construction. They proved that, in the presence of constraints such as cardinality and floor-ceiling, the efficient frontier becomes discontinuous. With such constraints, the problem of portfolio construction becomes NP-Hard in nature and cannot be solved by traditional deterministic methods [120]. And, it becomes necessary to apply techniques such as NCAs to solve such problems.

Chang et al. [21] applied GA, Tabu Search and SA to the cardinality constrained portfolios on five different datasets. These datasets contain stocks ranging from 31 to 225 and represent five different global indices such as Hang Seng, DAX 100, FTSE 100, S&P 100, and Nikkei 225 as given in OR-Library [121]. Each of these datasets corresponds to the weekly prices from March 1992 to September 1997. They contain data such as a number of stocks in a dataset, mean return of the individual stocks, the standard deviation of returns, and the correlation between them along with unconstrained efficient frontiers. The obtained efficient frontiers for all of these datasets have been compared with the unconstrained efficient frontiers.

Results obtained from these datasets have been used as a benchmark to evaluate the performance of other algorithms by researchers. Different researchers have used different multi-objective optimization performance metrics to compare obtained results such as – mean Euclidean distance (MED), variance of return error (VRE), mean return error (MRE), median percentage return error (MedPE), mean percentage deviation error (MPE), number of efficient points (EP), expected return (EXR), variance of return (VR), spread metric (Δ-metric), spacing metric (S-metric), algorithmic effort (AE), error ratio (ER), and Sharpe ratio (SR).

Cura [24] applied PSO to optimize portfolio for cardinality constrained mean-variance (CCMV) model. An arrangement algorithm (AA) is presented to ensure that generated portfolios are feasible portfolios and follow various constraints. A c-value is proposed to felicitate decision making to add or remove stock in a portfolio. A c-value represents a proportion between mean return and mean risk concerning risk-return aversion parameter (λ). Three different performance metrics – MED, VRE, and MRE – have been used to compare the performance of this algorithm.

Mozafari et al. [120] extended the work of Cura [24] and applied improved PSO hybridized with SA (IPSO-SA) to optimize portfolio for CCMV model. PSO searching capabilities have been improved in terms of convergence time and precision. For this purpose, two methods have been incorporated to update velocity of a particle – inertia weight mechanism and constriction factor approach. SA is embedded in this improved PSO to avoid convergence in local optima. An arrangement algorithm [24] is used to satisfy various constraints and to ensure that generated solutions are feasible.

Bacanin and Tuba [122] applied modified FA for CCMV portfolio optimization problem with entropy diversity constraint. Entropy is a measure of the portfolio diversification. Entropy constraint is used here to ensure that the diversity of a portfolio is not too low. An arrangement algorithm [24] is used to ensure that generated solutions are feasible. The original FA has been improved by introducing *Abandonment Threshold (AT)* to overcome the low exploration power of FA in early iterations. If a particular solution cannot be improved in *AT* number of attempts, it is considered exhausted and replaced by the randomly generated new solution. In contrast to early iterations, exploration is almost not required in late iterations when the right part of the search space is reached. A control parameter, *Exploration Breakpoint (EBP)* is introduced to determine whether to trigger exploration or not. Test results justified the inclusion of entropy constraint in CCMV model as well as modifications made to original FA.

Salahi et al. [123] applied improved PSO (ICPSO) and improved HS (IHS) algorithm for CCMV model. PSO algorithm is enhanced by using modifications in inertial weight and learning coefficients. HS algorithm is enhanced by modifying harmony memory consideration rate, pitch adjustment rate and bandwidth used in the algorithm. An arrangement algorithm [24] is used to ensure that generated solutions are feasible. Test results indicate that IHS gives slightly better results compared to ICPSO in terms of performance metrics. Also, IHS is proved much faster compared to ICPSO.

Experimental results obtained by above-discussed algorithms have been analyzed using three different performance metrics – MED, VRE, and MRE. The proposed winnowing algorithm has also been applied to CCMV model on the same datasets to generate efficient frontiers in the second phase of this thesis. Performance has been compared in terms of MED, VRE, and MRE with results obtained by these algorithms to assess the effectiveness and robustness of the winnowing algorithm.

Other researchers have also applied various NCAs on the same datasets, but they have used different performance metrics. For example –

- Xu et al. [124] applied CRO with two different variations – canonical CRO and super-molecule based CRO – to solve portfolio selection problem and compared performance using Sharpe Ratio, expected return and variance of the return.

- Mishra et al. [125] applied non-dominated sorting genetic algorithm-II (NSGA-II) and multi-objective particle swarm optimization (MOPSO) to tackle the same problem and compared performance using MED and S-metric.

- Chen et al. [126] proposed ABC algorithm and evaluated performance using MED, Δ-metric and algorithmic effort.

- Cui et al. [127] applied PSO hybridized with mathematical programming (PSO-MP) and assessed performance using MedPE and MPE.

- Tuba and Bacanin [128] applied upgraded FA (uFA) to select portfolio and compared performance using MedPE and MPE.

- Mishra et al. [129] proposed multi-objective BFO (MOBFO) to tackle the same problem and used MED, Δ-metric, S-metric, and ER as performance evaluation metrics.

Table 2.2 given below summarizes various NCAs applied to the constrained portfolio construction problem to obtain an efficient frontier.

Table 2.2 Natural Computing and Portfolio Construction

Researcher(s)	NCA and Variation	Model	Performance Metrics	Refer
Chang, Meade, Beasley, Sharaiha	GA, TS, SA	CCMV	MedPE, MPE, EP	[21]
Cura	PSO	CCMV, AA, c-value	MED, VRE, MRE	[24]
Mozafari, Tafazzoli, Jolai	IPSO-SA	CCMV, AA	MED, VRE, MRE	[120]
Bacanin, Tuba	mFA	CCMV, Entropy Diversity Constraint, AA	MED, VRE, MRE	[122]
Salahi, Daemi, Lotfi, Jamalian	ICPSO, IHS	CCMV, AA	MED, VRE, MRE	[123]
Xu, Lam, Li	CRO, S-CRO	CCMV with Non-Negative Objective Function	SR, ER, VR	[124]
Mishra, Panda, B. Majhi, R. Majhi	NSGA-II, MOPSO	Prediction based Mean-Variance model	MED, Δ-metric	[125]
Chen, Liang, Liu	ABC	CCMV	MED, Δ-metric, AE	[126]
Cui, Cheng, Bai	PSO-MP	Extended Mean-Variance	MedPE, MPE	[127]
Tuba, Bacanin	uFA	CCMV, AA	MedPE, MPE	[128]
Mishra, Panda, R. Majhi	MOBFO	CCMV	MED, Δ-metric, ER, S-metric	[129]

The next chapter introduces winnowing algorithm. The underlying winnowing process from which the design of the winnowing algorithm has been inspired is also explained.

CHAPTER 3

Winnowing – Process and Algorithm

Winnowing algorithm, a novel natural computing algorithm, is introduced in this chapter. The design of the winnowing algorithm is inspired based on the real world winnowing process. The chapter begins with an explanation of this winnowing process. The working principle of this process is explained with an example followed by the critical observations of this process. In the next section, the pseudo code of the winnowing algorithm is presented with a brief description. Core steps of the winnowing algorithm are also discussed in the context of portfolio management.

3.1 Winnowing Process – A Real World Phenomena

The meaning of the word 'to winnow' is 'to blow away'. Winnowing is a separation method. It separates heavier and lighter components from the mixture with the help of wind. Winnowing process is in use since ancient times. It is widely used by farmers and housewives to remove impurities from the useful products such as chaffs from wheat, a husk from rice, dust from grain, and bran from flour.

3.1.1 Working Principle and an Example

The working principle behind winnowing is straightforward. When a mixture having heavier and lighter components is dropped from height and wind is allowed to flow through this falling mixture, heavier and lighter components will form separate heaps on the ground. The gravitational force of earth will attract them towards the ground making them fall downwards. But compared to heavier particles, lighter components will be carried away by the force of the wind – creating a separate heap at some distance away.

Components can be separated from these two different heaps. Figure 3.1 demonstrates the working of the winnowing process.

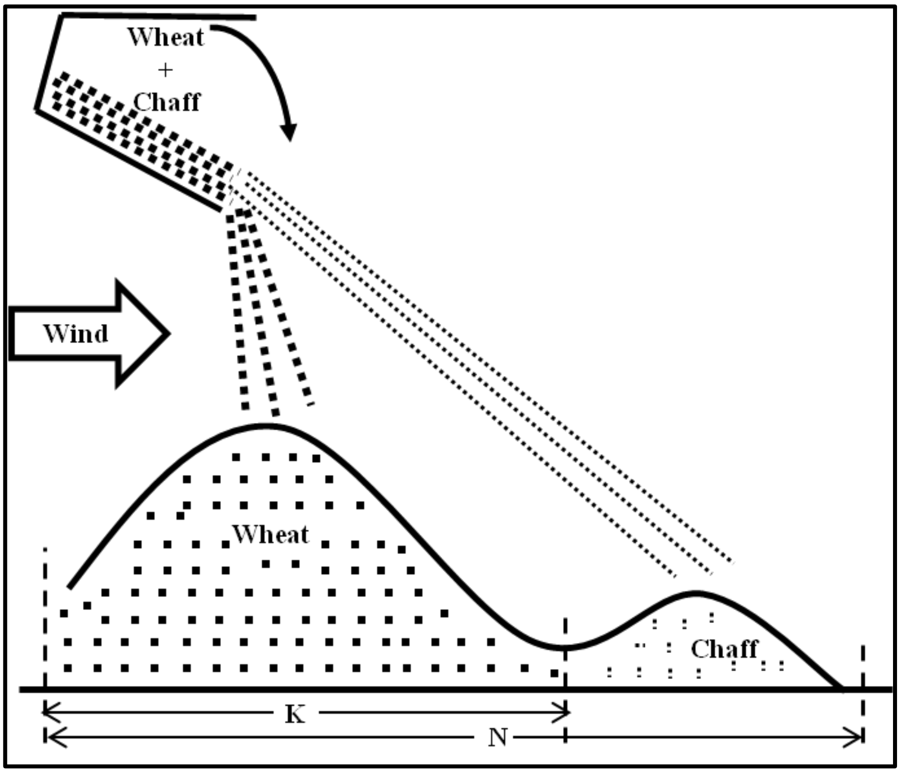


Figure 3.1 Winnowing Process

Their mixture is dropped from some height to separate the chaff from the wheat. In this mixture, chaff is impurity and lighter in weight, while wheat is a useful product and heavier in weight. Due to the gravitational force of the earth, both will fall on the ground. But, the chaffs will be carried away due to their lighter weights and fall at some distance away – forming a separate heap. The effect of the flow of wind will be comparatively less on the wheat due to their heavier weights. They will fall vertically downward forming a different heap. From these separate heaps, chaffs are removed, and the wheat is collected.

This process is repeated until all chaffs are separated from the wheat. Winnowing is followed by manual separation most of the time. In this case, remaining chaffs are found manually (by hand) from the wheat and removed. Such type of manual separation further improves the quality of the separation process.

Note that in the winnowing process, an informal inference can be made that, heavier particles on the ground attract other heavier particles and lighter particles on the ground attract other lighter particles.

The reason behind the wide acceptance and popularity of the winnowing across the world is its simplicity. Winnowing applies a simple mechanism yet it has established itself as an effective separation method. In its basic form, it utilizes natural forces fully and does not require any machines. In the absence of natural wind, the flow of the wind can be generated using a fan. With industrialization, farm machines like thresher have been developed to facilitate farming based on the concepts of winnowing.

3.1.2 Key Observations

Some critical observations of the winnowing process are as below:

- Winnowing is a separation method. It is used to collect/select desirable components from the mixture and to remove/reject undesirable components of the mixture.

- Winnowing is an iterative process. It is repeated as per requirement, or until separation completes.

- The distribution of desirable components (*K*) such as wheat and undesirable components *(N-K)* such as impurities depend upon their proportion in the mixture, their respective weights, the height from which the mixture is dropped, and the force of the flow of the wind.

- Impurities get removed from the mixture iteration by iteration. Consequently, the relative proportion of the desirable components in the remaining mixture keeps increasing. Or say, the relative portion of the undesirable components in the remaining mixture keeps decreasing. Thus, the quality of the mixture keeps increasing iteration by iteration.

- Manual separation is followed to improve the quality of the mixture further.

- The process continues until the separation completes, i.e., all desirable components are collected, or, all undesirable components are removed. In other words, the process continues until no further improvement in the quality of the remaining mixture is possible.

These observations serve as the basis for the proposed Winnowing Algorithm.

3.2 Winnowing Algorithm

This section presents the winnowing algorithm (WA). It begins with a description of the pseudo code for this algorithm and later explains the core steps of this algorithm in the context of the portfolio management.

3.2.1 Winnowing Algorithm Pseudo Code

The pseudo code for the Winnowing Algorithm (WA) is given in Figure 3.2

```
Initialize parameters
Construct initial solutions
While (true)
{
   Update each solution
        Update solution weights
            Generate random fractional weights
            Remove impurities (unwanted weights)
            Add to stock weights
            Normalize weights
        Evaluate the solution
        Improve solution using local search
   Update pool of solutions
        Replace worse solutions with the best solution
   Check for termination criteria, exit if true
}
```

Figure 3.2 Pseudo code for Winnowing Algorithm

Winnowing process is simple in nature, and so the winnowing algorithm is. This algorithm itself requires very few parameters. It begins with the initialization of these parameters. Parameters include Pool or population size (*P*), Replacement factor (γ), Stagnation counter (*S*), and Maximum fraction of weight used to update solutions (Δw_{max}). After initializing parameters, a pool (population) of probable solutions is constructed. For this purpose, *P* (number of) feasible solutions are generated. After constructing such kind of pool, the algorithm will progress iteratively.

Similar to other NCAs, WA also attempts to explore search space as broadly as possible and to exploit generated solutions to improve their quality in each iteration. For this, solutions are updated based on the winnowing process. Updated solutions are evaluated to assess their quality. These solutions are further improvised by applying some local search – resembling a manual search followed by winnowing process in the real world. After updating specific solutions, a pool of solutions is updated by replacing worse solutions with the best solution. This step improves the overall quality of the entire pool. It also increases the chances of having better solutions in the next iteration of the algorithm. At the end of each iteration, a termination criterion is checked to determine further progress of the algorithm. Termination criteria can be maximum time duration allowed to execute the algorithm, a maximum number of iterations, or stagnation in the best solution.

The steps – update each solution, and, update the pool of solutions – are repeated iteratively until termination criteria occur. With each iteration, the quality of the solutions and quality of the pool will keep increasing until an optimal solution is found.

3.2.2 Winnowing Algorithm in context of Portfolio Management

The description provided here attempts to give insights into the winnowing algorithm in the context of portfolio management. For this purpose, various steps of the winnowing algorithm are explained from the perspective of portfolio management.

- **Initialize parameters:**

The winnowing algorithm begins with initializing various parameters. This algorithm itself requires only four parameters: Pool or population size (P), Maximum fraction of weight used to update solutions (Δw_{max}), Stagnation counter (S), and Replacement factor (γ). These parameters are referred to as *algorithm-specific* parameters. Apart from these, other parameters are required to manage a portfolio. These parameters are referred to as *problem-specific* parameters. They include Total number of available stocks (N), Cardinality constraint, or, total stocks to be included in a portfolio (K), Risk-return trade-off coefficient (λ), Floor constraint or Lower bound on the weight of a stock (w_{min}), and, Ceiling constraint or Upper bound on the weight of a stock (w_{max}). Selection of values for these parameters has been described in the next chapters.

- **Construct Initial solutions:**

A pool (population) of probable solutions is constructed. In the context of portfolio management, each solution is a feasible portfolio. It can be represented as a vector of real numbers representing weights of stocks. The K stocks can be selected randomly out of the pool of given N stocks to construct initial solutions. These selected stocks can initially be given either equally distributed weights, or, randomly generated weights between lower bound (w_{min}) and upper bound (w_{max}).

- **Update Each Solution:**

This step updates each portfolio for the dual purpose of exploiting a generated solution and exploring a search space. This procedure begins by adding randomly generated fractional weights to the weights of stocks included in a portfolio. Stocks can be selected either based on their influence in a portfolio or randomly. Updated weights may violate various constraints. So they are normalized to confine them to such constraints. After this, each solution is evaluated to determine its quality in terms of the objective function. The quality of the portfolio is further improved by applying local search.

- **Update Pool of Solutions:**

A pool of solutions is updated by replacing worse solutions with the best solution. For this purpose, γ (in percentage) worse portfolios are found from the entire pool based on their objective function values. They are replaced by the best solution. This step improves the overall quality of the whole pool. It also increases the chances of having better solutions in the next iteration of the algorithm.

- **Check for Termination Criteria:**

The algorithm continues its execution until the termination criteria met. This step checks the termination criterion to terminate the algorithm. Stagnation in the best solution is considered as a termination criterion in the proposed work. If the best solution cannot be improved for S number of consecutive iterations, it is believed that there is no further scope of improvement. The current best solution is regarded as an optimal solution, and, the algorithm terminates.

The steps – update each solution, and, update the pool of solutions – are repeated iteratively until termination criteria occur. In the context of the problem of portfolio management, in each step, the WA attempts to include appropriate stocks in a portfolio as well as to increase the relative weight of the better stocks in a portfolio. This phenomenon is similar to the real world winnowing process which attempts to increase the relative proportion of the desirable components in the given mixture in successive iterations.

3.3 The Work Ahead

Chapter four presents the first phase of the research work and focuses on portfolio weight optimization. The winnowing algorithm is adapted to optimize portfolio weights in this chapter. The adaptation of the winnowing algorithm to optimize portfolio weights is explained along with implementation details. Obtained experimental results are presented next. In the end, the performance of the winnowing algorithm is evaluated by comparing obtained results with other state-of-the-art algorithms.

Chapter five is dedicated to the second phase of this research work. It focuses on portfolio construction. The winnowing algorithm is adapted to construct portfolio in this chapter. The adaptation of the winnowing algorithm to select stocks and their weights to construct a portfolio is explained along with implementation details. Obtained experimental results are presented next. In the end, the performance of the winnowing algorithm is evaluated by comparing obtained results with other state-of-the-art algorithms.

Chapter six is focused on the third phase of the research work and discusses portfolio optimization. The chapter begins with the discussion of the real-world dataset of the National Stock Exchange of India used to test the performance of the portfolios. This follows the description of static portfolios and performance issues associated with them. Later, dynamic portfolios are discussed which are allowed to change stocks and weights over a while. In the end, obtained experimental results are examined for the comparative analysis purposes, and the ability of the winnowing algorithm is tested to manage the real-world stock market portfolio.

CHAPTER 4

Phase I: Portfolio Weight Optimization

Investment in stock markets needs to answer two vital questions:

- In which stocks to invest?
- In which proportion to invest?

The first phase of the research work presented in this chapter has evolved around the second question. It solely focuses on the optimization of the weight of given stocks to be included in a portfolio. In this chapter, the winnowing algorithm is adapted to optimize portfolio weights or say, weights of the stocks to be included in a portfolio. This adapted winnowing algorithm will be referred to as WA-PWO henceforth.

The chapter begins with the description of a simple dataset which has been used to test the performance of the algorithm. Next section describes the adaptation of winnowing algorithm for portfolio weight optimization in detail. This section also presents the algorithm used to normalize portfolio weights in this research work. In the end, experimental results are presented, and comparative analysis is discussed.

4.1 Dataset

A simple dataset, given in [113], consisting of five stocks has been used to assess the performance of the WA-PWO. The same dataset has been used by several researchers to evaluate the performance of their algorithms. This dataset provides historical annual returns for five different stocks for a period of five years from 2007 to 2011.

Table 4.1 provides these annual returns along with their mean (average) values. Table 4.2 represents the variance-covariance matrix for these stocks. The italic values across the diagonal in this table represent variances for that specific stock. This matrix has been derived by applying MATLAB function *cov* to the return values given in Table 4.1.

Table 4.1 Standard Dataset with 5 Stocks

Year	Stock 1	Stock 2	Stock 3	Stock 4	Stock 5
2007	-0.15	0.29	0.38	0.18	-0.10
2008	0.05	0.18	0.63	-0.12	0.15
2009	-0.43	0.24	0.46	0.42	0.15
2010	0.79	0.25	0.36	0.24	0.10
2011	0.32	0.17	-0.57	0.30	0.25
Mean	0.1160	0.2260	0.2520	0.2040	0.1100

Table 4.2 Variance-Covariance Matrix for the Standard Dataset

Stock	Stock 1	Stock 2	Stock 3	Stock 4	Stock 5
Stock 1	*0.2173*	-0.0042	-0.0669	-0.0116	0.0133
Stock 2	-0.0042	*0.0025*	0.0106	0.0030	-0.0057
Stock 3	-0.0669	0.0106	*0.2225*	-0.0389	-0.0299
Stock 4	-0.0116	0.0030	-0.0389	*0.0407*	0.0035
Stock 5	0.0133	-0.0057	-0.0299	0.0035	*0.0168*

4.2 Adapting WA for Portfolio Weight Optimization

The adapted winnowing algorithm to optimize portfolio weights (WA-PWO) is given in Figure 4.1. The explanation regarding the implementation of WA-PWO is given in following sub-sections. The stocks referred in the example provided here are based on the standard dataset with five stocks described in section 4.1 in this chapter.

```
# Initialize parameters
N, K, λ, w_min, w_max, P, γ, S, Δw_max

# Construct initial solutions
For i = 1:P
        For j = 1:K
                w_ij ← Random weight between w_min and w_max
        End For
        Normalize weights                                    # Figure 4.2
End For

# Iterative loop
Counter = 0
While (true)
        # Update each solution
        For i = 1:P
                        # Update solution weights
                Δw ← Generate K random weights between 0 and Δw_max
                Sort Δw and remove trailing weights          # Equation (4.2)
                For j = 1:K
                        w_ij = w_ij + Δw_j                   # Figure 4.3
                End For
                Normalize weights                            # Figure 4.2

                        # Evaluate solution
                f_i ← Objective function value for i^th solution   # Equation (2.23)

                        # Improve solution using Local Search
                Select random stocks α and β where α, β ϵ [1, K]
                w_ij' ← New solution by exchanging w_iα and w_iβ in w_ij
                f_i' ← Objective function value for new solution   # Equation (2.23)
                If (f_i' > f_i)
                        Replace w_ij with w_ij' and f_i with f_i'
                End If
        End For

                # Update pool of solutions
        For i = 1: γ                                         # γ in percentage
                Select the worst solution in a pool
                Replace it with the best solution
        End For

                # Check for stagnation, exit if true
        Solution_best ← Best solution with maximum f_i, i = 1…P
        If (No improvement in Solution_best), Counter = Counter + 1
        Else    Counter = 0
        End If
        If (Counter > S), Exit, End If
End While
```

Figure 4.1 WA-PWO Algorithm

4.2.1 Parameters

Table 4.3 summarizes various parameters required for the WA-PWO.

Table 4.3 Parameter Set for WA-PWO

Parameter	Meaning	Value
N	Total available stocks	5
K	Cardinality Constraint (Total stocks in a portfolio)	5
λ	Risk-return trade-off coefficient	[0.0:0.1:1.0]
w_{min}	Lower bound on the weight of a stock (Floor constraint)	0.01
w_{max}	Upper bound on the weight of a stock (Ceiling constraint)	0.50
P	Pool size	20
γ	Replacement factor (%)	10
S	Stagnation counter	200
Δw_{max}	Maximum fraction of weight used to update portfolio	0.001

The parameters N, K, λ, w_{min}, and w_{max} are portfolio-specific parameters. Their values depend upon the stock market as well as preferences of the individual investors as discussed below.

- The value of N is kept five as the dataset used here contains five stocks.
- This algorithm only optimizes weights of selected stocks, and all the five stocks of the given dataset are included in a portfolio. So, the value of K is also kept 5.
- Different values of λ in the range of *0.0 to 1.0* with a step size of *0.1*, each value at a time, has been used to generate an efficient frontier of the optimal portfolios for different risk-return combinations.
- The w_{max} determines the number of dominating stocks (in terms of weights) in a portfolio. For example, if w_{max} = 1.0, it allows one stock to dominate among all other stocks of a portfolio. If w_{max} = 0.5, it allows two stocks to dominate among other stocks of a portfolio. In this work, it is kept as 0.50 allowing two stocks to dominate over other stocks.
- The w_{min} avoids the negligible weight of stocks in a portfolio. In other words, it ensures relevant participation of stocks in a portfolio. It is kept 0.01 – representing at least 1% participation for each of the stocks in a portfolio.

The other parameters P, γ, S, and Δw_{max} are algorithm-specific parameters. Their values have been empirically determined as discussed below.

- Practical observations indicate that WA-PWO fails to converge to an optimal solution if P is less than 10. Performance of WA-PWO improves with an increase in P to some extent. If P is kept higher than 20, no significant improvement is found in the performance of the algorithm. It unnecessarily increases execution time. So, P is kept as 20.

- The γ percentage worse solutions of pool P are replaced by the best solution in each iteration of the WA-PWO to update the pool of solutions. As γ is 10 and P is 20, two worse solutions are found based on the objective function value and replaced with the best solution.

- The algorithm improves the quality of the best solution iteration by iteration. It has been observed that if there is no improvement in the best solution for 140 or more consecutive iterations, then there will not be any significant improvement in the best solution for further iterations also. Considering this observation as well as to accommodate few exceptional cases (if any), the stagnation counter S is kept 200.

- Selection of Δw_{max} affects convergence time for the WA-PWO as well as the quality of the obtained optimal solution. For example, if Δw_{max} is kept high (around 0.10), the algorithm gives a faster result, but solution sticks in local minima and the optimal solution could not be found. In contrast, if Δw_{max} is kept very small (around 0.0001), the algorithm progresses smoothly and finds the optimal solution. But, it unnecessarily increases required iterations to get convergence. So a balanced value, which is around 10% of the w_{min}, in this case, 0.001, has been selected for Δw_{max}. Portfolio quality is smoothly improved iteration by iteration without being stuck in local minima. It also prevents increment in unnecessary iterations.

4.2.2 Initial Solutions

Each solution is a feasible portfolio. It can be represented as a vector of real numbers specifying weights of selected stocks to be included in a portfolio. The WA-PWO only aims to optimize the weights of selected stocks. So, all the five stocks of the standard dataset are assigned randomly generated weights in the range of w_{min} and w_{max} to construct the initial solution. The vector of such five weights represents a single solution, or say portfolio. Based on the pool size, total *P* solutions are constructed initially in this manner.

As weights are generated randomly, they may violate the constraints such as budget constraint given in Equation (2.24) and floor-ceiling constraint given in Equation (2.26). These weights must be normalized to confine them to such constraints. For this purpose, an algorithm named *Weight-Normalizer* is proposed as given in Figure 4.2. This algorithm is simpler and faster compared to the arrangement algorithm presented in [24].

In this algorithm, let w_{ij} be the weight of stock j in i[th] portfolio, η be the sum of excess weight for stocks in i[th] portfolio, ϕ be the sum of deficient weight for stocks in i[th] portfolio, ε be the sum of w_{ij} where $w_{ij} < w_{max}$ and δ be the sum of w_{ij} where $w_{ij} > w_{min}$. If the weight for asset j in portfolio i, w_{ij}, exceeds upper bound or goes below lower bound, it will be adjusted as follows:

$$w_{ij} = \begin{cases} w_{ij} + \frac{w_{ij}}{\varepsilon}\eta & if\ w_{ij} < w_{max} \\ w_{max} & if\ w_{ij} > w_{max} \\ w_{ij} - \frac{w_{ij}}{\delta}\varphi & if\ w_{ij} > w_{min} \\ w_{min} & if\ w_{ij} < w_{min} \end{cases} \tag{4.1}$$

Table 4.4 represents a sample pool of four portfolios (i.e., initial solutions) along with their expected return, risk and objective function values. The CCMV model presented in section 2.5 is used to calculate expected return, risk and objective function values. This table also shows average values of expected return, risk and objective function for the entire pool. Also note that to keep this explanation simple, the values for *P*, λ, Δw_{max} and γ have been kept 4, 1.0, 0.10 and 25% respectively.

Table 4.4 Initial Solutions for WA-PWO

Portfolio	W_1	W_2	W_3	W_4	W_5	Return	Risk	Objective Function
P_1	0.2027	0.1887	0.2132	0.0959	0.2995	0.1724	0.0112	0.1724
P_2	0.1691	0.1911	0.1909	0.2258	0.2231	0.1815	0.0078	0.1815
P_3	0.1428	0.1409	0.3088	0.1202	0.2873	0.1824	0.0149	0.1824
P_4	0.0525	0.3141	0.3273	0.0700	0.2362	0.1998	0.0188	0.1998
					Average	0.1840	0.0132	0.1840

Algorithm: Weight-Normalizer (W_i)

W_i: weight of selected stocks for an i^th solution

While (true)
  $\psi = \sum_{j=1...K} w_{ij}$ # total weights

  If $\psi < 1$, $w_{ij} = w_{ij} + (1-\psi)/K$, where j = 1,2,...$K$

  If $\psi > 1$, $w_{ij} = w_{ij} / \psi$, where j = 1,2,...K

  $\eta = \sum_{j=1...K} \max(0, w_{ij} - w_{max})$ # sum of excess weights

  $\phi = \sum_{j=1...K} \max(0, w_{min} - w_{ij})$ # sum of deficient weights

  If (η=0 and ϕ=0)
    Exit Algorithm
  End If

  If ($\eta \neq 0$)   # if the solution has excessive weights

    $\varepsilon = \sum_{j=1...K} w_{ij}$, if $w_{ij} < w_{max}$

    For j=1:K
      If ($w_{ij} < w_{max}$), $w_{ij} = w_{ij} + w_{ij} * \eta / \varepsilon$
      Else, $w_{ij} = w_{max}$
      End If
    End For

  End If

  If ($\phi \neq 0$)   # if the solution has deficient weights

    $\delta = \sum_{j=1...K} w_{ij}$, if $w_{ij} > w_{min}$

    For j=1:K
      If ($w_{ij} > w_{min}$), $w_{ij} = w_{ij} - w_{ij} * \phi / \delta$
      Else, $w_{ij} = w_{min}$
      End If
    End For

  End If

End While

Figure 4.2 Weight-Normalizer Algorithm

4.2.3 Updating Solutions

This step is the heart of the winnowing algorithm. Individual solutions are updated in this step to achieve primary aspects of any natural computing algorithms – exploitation and exploration. For this purpose, solutions are first updated by modifying weights of stocks in a portfolio. These updated solutions are evaluated. Solutions are further improved by applying local search. These steps have been explained below.

- **Update Solution Weights**

This step updates the weight of stocks. For this purpose, small fractional weights are added to the weight of stocks in a portfolio. For this purpose, a series of K random fractional weights (Δw) is generated in the range of 0 to Δw_{max}. This series is sorted in descending order. This operation is similar to blowing the wind through the falling mixture of the winnowing process. Due to this sorting, lighter weights will move towards the end of the series and heavier weights will move towards the starting of the series.

In the winnowing process, lighter components are considered as impurities and removed. In this algorithm also, lighter weights are considered as impurities and removed. For this purpose, such fractional weights from a series of random fractional weights (Δw) are truncated to zero. The total number of trailing fractional weights to be truncated depends upon K and w_{max} as given in Equation (4.2). For example, if w_{max} = 0.50, it means two stocks can dominate other stocks in a portfolio. So, if K = 5, remaining 3 fractional weights ($K - 2 = 5 - 2 = 3$) in Δw are truncated.

$$\text{Rejection} = K - 1/w_{max} \tag{4.2}$$

Table 4.5 explains the selection of Δw. Note that, for explanatory purpose, the value of Δw_{max} is considered 0.10 in this example. So, five random values are generated between 0 and 0.10 to obtain a series of fractional weights Δw.

Table 4.5 Selection of Δw for WA-PWO

Operation	Δw_1	Δw_2	Δw_3	Δw_4	Δw_5	Winnowing Action
Fractional Weights (Randomly Generated)	0.0485	0.0393	0.0671	0.0741	0.0520	Throw mixture
Sorted	0.0741	0.0671	0.0520	0.0485	0.0393	Blow wind
Tail Terminated (Δw)	0.0741	0.0671	0.0000	0.0000	0.0000	Reject Impurities

Stock weights are updated by adding fractional weights (Δw) to the weight of stocks in a portfolio. Two different approaches are used to select stocks for updating their weights so that either generated solutions can be exploited or search space can be explored. The pseudo-code given in Figure 4.3 provides the procedure to select one of these two approaches.

```
For each solution to be updated, toss a coin
If it is head,
    Exploit the solution
Else
    Explore the search space
End If
```

Figure 4.3 Pseudo code for selecting an approach for updating solution weights

Tossing a coin has a 50% probability of getting either head or tail and gives a fair chance to both of these approaches. To implement this, a random number, $\theta \in [0, 1]$ is generated. If $\theta < 0.5$, it is considered as a head and generated solutions are exploited. If $\theta \geq 0.5$, it is viewed as a tail and search space is explored.

- *Exploit the solution:* this approach attempts to exploit the solution, i.e., to improve the quality of the solution. For this purpose, stocks are selected to update their weights based on their influence in the portfolio. In a winnowing process, it can be said that heavier particles attract other heavier particles and lighter particles attract other lighter particles. Similarly, in this approach, stocks having comparatively higher weights are considered as more influential stocks in a portfolio. And, they draw more weights.

- *Explore the search space:* In contrast to the above approach, to explore the available search space of feasible portfolios, stocks are selected to update their weights purely in a random manner. In this approach, the updated solution can be either better or worse compared to the original solution. But, it helps the winnowing algorithm to avoid local minima.

It is possible that the updated weight of stocks may violate constraints related to weights as per the CCMV model. So, to confine these weights according to various constraints, these updated weights are again normalized using Weight-Normalizer algorithm. Table 4.6

explains various operations to update the weight of stocks for a single portfolio P_1 constructed initially. Note that, in this example, it is assumed that θ is less than 0.5.

Table 4.6 Updating Weights of a Portfolio for WA-PWO

Parameter	w_1	w_2	w_3	w_4	w_5	Description
Initial Weights (w)	0.2027	0.1887	0.2132	0.0959	0.2995	Initial Portfolio (P_1)
Δw	0.0741	0.0671	0.0000	0.0000	0.0000	Fractional Weights (Table 4.5)
$\theta < 0.5$	---	---	0.0671	---	0.0741	Exploit Solution
$w' = w + \Delta w$	0.2027	0.1887	0.2803	0.0959	0.3736	Updated Weights (Sum = 1.1412)
Normalized Weights	0.1776	0.1654	0.2456	0.0840	0.3274	Weight-Normalizer Algorithm
Updated Portfolio	Expected Return = 0.17303; Risk = 0.01165					Updated return & risk

Observe that, weights of stock 3 and stock 5 have been increased in the updated portfolio. Compared to these two stocks, the weights of other stocks have decreased. The expected return and risk have been increased for the updated portfolio. As the $\lambda = 1.0$ in this example, the objective function of the CCMV model attempts to maximize return and ignores the risk entirely. In this case, portfolios with higher expected returns are considered better portfolios. So, it can be inferred that with an increase in weights of more important stocks, the solution quality improves.

Table 4.7 represents the pool of four portfolios with updated weights at the end of this step. Note that with changes in weight of stocks, the expected return and risk of portfolios have also been changed. Also, the higher average of the expected return for a pool, compared to that of given in Table 4.4, shows improvement in the overall quality of the solutions.

Table 4.7 Solutions after Updating Weights for WA-PWO

Portfolio	W_1	W_2	W_3	W_4	W_5	Return	Risk	Objective Function
P_1	0.1776	0.1654	0.2456	0.0840	0.3273	0.1730	0.0117	0.1730
P_2	0.1546	0.2066	0.1746	0.2601	0.2041	0.1841	0.0069	0.1841
P_3	0.1250	0.1233	0.3347	0.1052	0.3117	0.1825	0.0172	0.1825
P_4	0.0461	0.3359	0.3492	0.0615	0.2074	0.2046	0.0226	0.2046
					Average	0.1861	0.0146	0.1861

In the winnowing process, the relative proportion of the desirable components keeps increasing in the remaining mixture iteration by iteration. Similarly, in winnowing algorithm, the relative weight of the desirable stocks keeps increasing in the longer term – resulting in better quality of the solutions.

- **Evaluate Solution**

Each solution is evaluated to determine its quality in terms of expected return, risk and objective function value. For this purpose, Equation (2.23) of the CCMV model has been used as an objective function. As this model attempts to maximize an objective function, solutions with higher objective function values are considered better solutions.

- **Improve Solution using Local Search**

To further improve the quality of the solution, a simple approach has been used as a local search. In this approach, weights of two randomly selected stocks are exchanged to get a new solution. The resultant solution is evaluated and compared with the original solution. If the new solution has better objective function value, it replaces the original solution in a pool. Else, it is discarded, and the original solution is kept as it is.

Table 4.8 represents resultant solutions after applying local search. In this example, solution P_2 has been improved due to the exchange of weights between stock 1 and stock 5. Similarly, solution P_3 has been improved by exchange of weights between stock 2 and stock 5. This step also improves the average objective function value of the entire pool.

Table 4.8 Improved Solutions after Applying Local Search for WA-PWO

Portfolio	W_1	W_2	W_3	W_4	W_5	Return	Risk	Objective Function
P_1	0.1776	0.1654	0.2456	0.0840	0.3273	0.1730	0.0117	0.1730
P_2	*0.2041*	0.2066	0.1746	0.2601	*0.1546*	0.1844	0.0094	0.1844
P_3	0.1250	*0.3117*	0.3347	0.1052	*0.1233*	0.2043	0.0203	0.2043
P_4	0.0461	0.3359	0.3492	0.0615	0.2074	0.2046	0.0226	0.2046
					Average	0.1916	0.0160	0.1916

4.2.4 Updating Pool

The γ percentage worse solutions of the entire pool are replaced with the best solution. As γ is 25% in this example, one worst solution P_1 is replaced by the best solution P_4. Table 4.9 represents the resultant solutions. The higher average of the objective function value in this table compared to Table 4.8 indicates improvement in the overall quality of the pool. It also creates better chances to have a broader exploration of the search space towards the direction of optimal solutions in the following iterations.

Table 4.9 Solutions after Updating Pool for WA-PWO

Portfolio	W_1	W_2	W_3	W_4	W_5	Return	Risk	Objective Function
P_1	0.0461	0.3359	0.3492	0.0615	0.2074	0.2046	0.0226	0.2046
P_2	0.2041	0.2066	0.1746	0.2601	0.1546	0.1844	0.0094	0.1844
P_3	0.1250	0.3117	0.3347	0.1052	0.1233	0.2043	0.0203	0.2043
P_4	0.0461	0.3359	0.3492	0.0615	0.2074	0.2046	0.0226	0.2046
					Average	0.1995	0.0187	0.1995

4.2.5 Termination Criteria

Stagnation in the best solution has been considered as termination criteria for the WA-PWO. A solution with the maximum objective function value is considered as the best solution and represents the best portfolio. If the best solution cannot be improved for *S* number of iterations consecutively, it will be considered as an optimal portfolio and algorithm will stop. A counter initialized with zero is used to implement this strategy. If there is no improvement in the current best solution compared to the previous best solution, counter is incremented by one. Else, it is reset to zero. If the counter becomes greater than *S*, the current best solution will be considered as an optimal solution and algorithm terminates.

Again, each solution is updated, the entire pool is updated, and the termination criterion is checked. These steps are repeated iteratively until the termination criterion is met.

It has been observed that, in the initial iterations, wrong stocks may get more weights downgrading the quality of some solutions. The reason behind this is a random distribution of initial weights across five stocks. But, with the progress of WA-PWO, such solutions are either improved by local search or replaced by the best solution.

As the algorithm progresses, more important stocks with higher weights keep attracting more weights resulting in better quality of solutions. In other words, the WA-PWO attempts to increase the relative weight of the desirable stocks in a portfolio iteration by iteration. This is similar to the real world winnowing process which attempts to increase the relative weight of the desirable components in the given mixture in successive iterations.

4.3 Results and Discussion

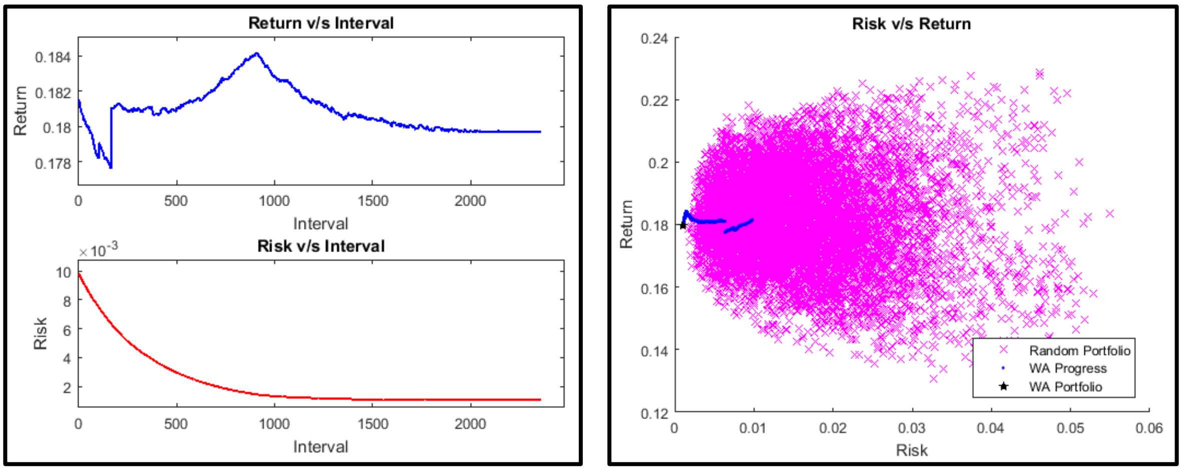
This section discusses the obtained experimental results and comparative analysis of the performance of the WA-PWO with other NCAs. The WA-PWO has been tested on the standard dataset discussed in section 4.1 . The algorithm has been applied to this dataset with different values of λ ranging from 0.0 to 1.0 with a step size of 0.1 – resulting in 11 different portfolios. The algorithm has been implemented using MATLAB R2015b. It has been executed on a system having Intel(R) Core(TM) i5-2410M processor @2.30GHz with 3 GB RAM and 32-bit Windows 7 operating system. For each different value of λ, the algorithm has been executed 10 times separately to obtain best, average and worst results.

4.3.1 Experimental Results

Figure 4.4 shows the progress of the WA-PWO to find the optimal solution for the $\lambda = 0.0$. Figure 4.4 (a) provides charts for expected return v/s iterations and risk v/s iterations. Note that if $\lambda = 0.0$, the algorithm will only attempt to minimize risk avoiding return. This figure illustrates how the algorithm proceeds to minimize the risk of a portfolio in successive iterations. Observe that after initial iterations, as the risk reduces the expected return also reduces. Figure 4.4 (b) displays the best portfolios (blue dots) found progressively along with the optimal portfolio (black star). This figure also displays randomly generated portfolios (pink cross) to provide an idea about the distribution of feasible solutions across a probable search space.

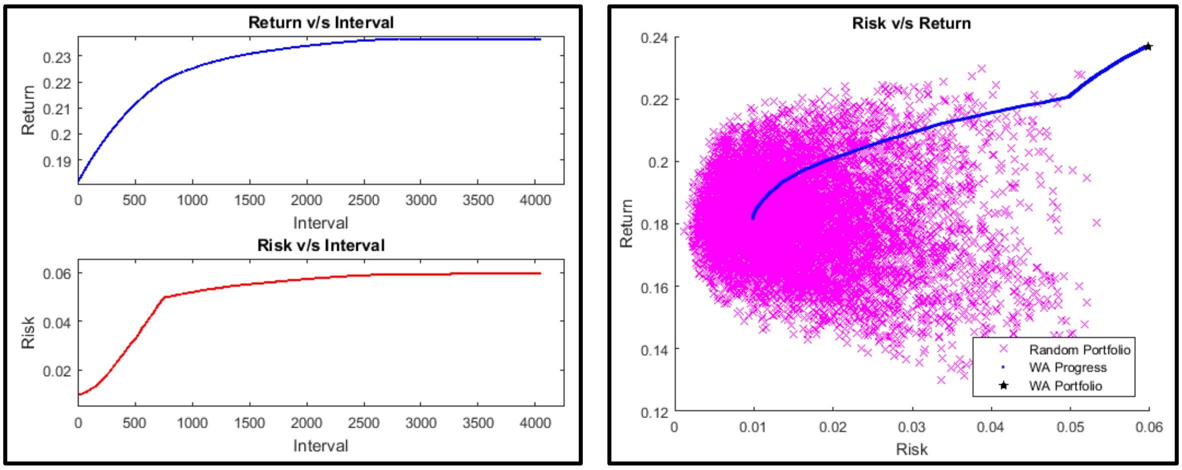
Similarly, Figure 4.5 shows the progress of the WA-PWO to find the optimal solution for the $\lambda = 1.0$. Figure 4.5 (a) represents how the algorithm proceeds to maximize expected return in successive iterations. Observe that with an increase in expected return, the risk also increases. Figure 4.5 (b) displays the best portfolios (blue dots) found progressively along with the optimal portfolio (black star) and randomly generated portfolios (pink cross).

Figure 4.6 displays optimal portfolios obtained with different values of λ in a graphical manner. These portfolios represent the efficient frontier obtained by WA-PWO along with randomly generated 10000 different portfolios. In the context of random portfolios, it can be observed from this figure that each WA-PWO portfolio yields the best expected return for the specific risk values.



(a) Risk, Return v/s Interval Graph **(b) Risk v/s Return Graph**

Figure 4.4 WA-PWO Progress for λ = 0.0



(a) Risk, Return v/s Interval Graph **(b) Risk v/s Return Graph**

Figure 4.5 WA-PWO Progress for λ = 1.0

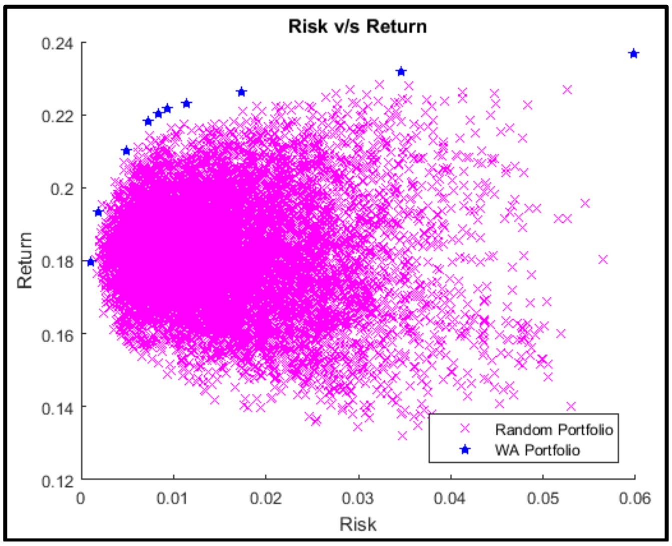


Figure 4.6 Efficient Frontier for Standard Dataset

Table 4.10 and Table 4.11 summarize the obtained experimental results for the standard dataset with 5 stocks with different values of λ. Table 4.10 provides best, average and worst values for the expected return and risk of individual portfolios derived over 10 different runs of the WA-PWO. This table also provides average iterations and average time (in seconds) taken by the algorithm to find optimal solutions. Table 4.11 provides optimal weights for the best portfolios for each different λ values along with best portfolios in terms of expected return and risk.

Table 4.10 Risk and Return for Standard Dataset with 5 Stocks

λ	Expected Return			Risk (Variance)			Average	
	Best	Average	Worst	Best	Average	Worst	Iterations	Time (s)
0.0	0.179690	0.179690	0.179690	0.001063	0.001063	0.001063	2438.10	2.56
0.1	0.193222	0.193227	0.193241	0.001814	0.001814	0.001816	2259.50	2.34
0.2	0.210166	0.210172	0.210178	0.004872	0.004873	0.004875	1760.40	1.77
0.3	0.218205	0.218229	0.218234	0.007281	0.007291	0.007293	3090.70	3.32
0.4	0.220399	0.220395	0.220395	0.008380	0.008377	0.008377	3168.60	3.31
0.5	0.221518	0.221508	0.221465	0.009311	0.009301	0.009259	3202.60	3.36
0.6	0.223195	0.223043	0.222835	0.011400	0.011179	0.010882	3342.90	3.49
0.7	0.226263	0.226362	0.226644	0.017377	0.017621	0.018315	3700.10	3.96
0.8	0.231661	0.231671	0.231702	0.034656	0.034696	0.034822	2869.70	3.05
0.9	0.236520	0.236486	0.236368	0.059757	0.059680	0.059424	3666.00	3.94
1.0	0.236520	0.236489	0.236432	0.059758	0.059688	0.059562	3878.70	4.29

Table 4.11 Optimized Portfolio Weights for Standard Dataset with 5 Stocks

λ	Optimized weights for the best portfolios					Return	Risk
	w_1	w_2	w_3	w_4	w_5		
0.0	0.01564	0.50000	0.05187	0.04501	0.38748	0.179690	0.001063
0.1	0.02728	0.50000	0.07810	0.14860	0.24601	0.193222	0.001814
0.2	0.04182	0.50000	0.11079	0.27853	0.06885	0.210166	0.004872
0.3	0.02516	0.50000	0.13248	0.33236	0.01000	0.218205	0.007281
0.4	0.01000	0.50000	0.15040	0.32960	0.01000	0.220399	0.008380
0.5	0.01000	0.50000	0.17372	0.30629	0.01000	0.221518	0.009311
0.6	0.01000	0.50000	0.20864	0.27136	0.01000	0.223195	0.011400
0.7	0.01000	0.50000	0.27256	0.20744	0.01000	0.226263	0.017377
0.8	0.01000	0.50000	0.38502	0.09499	0.01000	0.231661	0.034656
0.9	0.01000	0.47001	0.49999	0.01000	0.01000	0.236520	0.059757
1.0	0.01000	0.47000	0.50000	0.01000	0.01000	0.236520	0.059758

Observe the results presented in the above tables. With the increase in the value of λ, the expected return and so risk associated with portfolio increases. With λ=0.0, the algorithm attempts to minimize risk only ignoring the expected return completely. With λ=1.0, the algorithm attempts to maximize expected return only ignoring the risk entirely. Also, note that portfolio risk is considered in the form of variance.

The results presented in the above tables indicate that portfolio returns are higher values compared to portfolio variances for this dataset. Due to this reason, the objective function given by Equation (2.23) gets skewed towards portfolio returns and fails to provide the perfect balance between expected return and risk of portfolios. The results for the λ = 0.9 and λ = 1.0 show this fact. The algorithm starts ignoring risk and focuses on maximizing expected return with λ = 0.9 rather than λ = 1.0 as shown in the above tables.

The study of Table 4.10 shows minor variations in the best and worst results of expected return and risk with respect to average results. It proves that the WA-PWO is able to find accurate optimal solutions in a consistent manner. The minor variations in the best and the worst results also show that the algorithm is capable of avoiding local optima and can converge to the optimal solution in each run of the algorithm. This algorithm also requires less number of iterations compared to peer NCAs. Along with this, the low values of the average time taken to find optimal solutions presented in this table show that the WA-PWO is able to find optimal solutions in reasonable time durations.

4.3.2 Comparative Analysis

Other researchers have applied various NCAs such as GA, FA, ABC, KH, FWA, and BA to the same dataset. Optimal portfolios given as a vector of real values representing stock weights are provided in their research paper. The expected return and risk values for these optimal portfolios have been calculated again using Equation (2.21) and Equation (2.22) to maintain the homogeneity across various results. Only the best results obtained by all these algorithms have been considered to make comparisons here. The cost of the portfolio has been calculated based on the following Equation (4.3) as given in [119]. The portfolio with the lower cost will be considered a better portfolio.

$$Cost = Risk + \frac{1}{Return} \tag{4.3}$$

Table 4.12 shows the comparative results of WA-PWO with all other NCAs applied to the standard dataset with 5 stocks with similar portfolio model.

Table 4.12 Comparative Results for WA-PWO

Algorithm	Return	Risk	Cost	Reference
GA	0.222235	0.031749	4.531490	[113]
FA	0.216894	0.026044	4.636591	[114]
ABC	0.216390	0.030388	4.651674	[116]
KH	0.214742	0.030585	4.687336	[117]
FWA	0.233192	0.124476	4.412788	[118]
BA	0.233076	0.125621	4.416067	[119]
WA-PWO (λ = 0.8)	0.231661	0.034656	**4.351314**	Winnowing Algorithm
WA-PWO (λ = 1.0)	**0.236520**	**0.059758**	**4.287732**	Winnowing Algorithm

It can be observed from the above table that WA-PWO produces better results compared to peer NCAs. This algorithm could find a moderately low-risk portfolio with λ = 0.8 which has better cost value compared to other NCAs. With λ = 1.0, this algorithm has generated a portfolio having the best result in terms of return and cost. These results are shown in bold faces in the above table. These results establish the competency of the winnowing algorithm against other NCAs.

Other NCAs discussed here are focused on generating a single optimal portfolio. But in reality, such kind of single portfolio cannot be suitable to all investors having different risk-return profiles. The WA-PWO could successfully generate an efficient frontier of the optimal portfolios for different risk-return combinations.

The WA-PWO presented in this chapter has focused only on optimizing weights of the selected stocks in a portfolio. Obtained experimental results provide motivation to apply winnowing algorithm to construct a portfolio, i.e., to select stocks as well as their weights, rather than only optimizing stock weights. The next chapter presents the winnowing algorithm in the context of portfolio construction.

CHAPTER 5

Phase II: Portfolio Construction

Revisit the two vital questions regarding investment discussed earlier:

- In which stocks to invest?
- In which proportion to invest?

The previous chapter focused only on the second question and attempted to optimize the weight of already selected stocks. This chapter attempts to answer both of these questions and represents the second phase of the research work. In this chapter, the winnowing algorithm is adapted to construct a portfolio. *Portfolio Construction* selects 'specific stocks and their weights' from a large pool of available stocks to make the initial investment. This adapted winnowing algorithm will be referred to as WA-PC henceforth.

The chapter begins with the description of a dataset which has been used to test the performance of the algorithm. Next section describes the adaptation of winnowing algorithm for portfolio construction in detail. In the end, experimental results are presented, and comparative analysis is discussed.

5.1 Dataset

The WA-PC algorithm has been tested on five different datasets given in the OR-library [121] to assess its performance. Results obtained from these datasets have been used as a benchmark to evaluate the performance of several researchers. Table 5.1 provides a list of data sources for these datasets. These datasets represent stocks ranging from 31 to 225 and represent five different global indices such as Hang Seng, DAX 100, FTSE 100, S&P 100, and Nikkei 225.

Table 5.1 OR-Library Datasets

Sr. No.	Data Source (Index and Country)	No. of Stocks
1	Hang Seng (Hong Kong)	31
2	DAX 100 (Germany)	85
3	FTSE 100 (UK)	89
4	S&P 100 (USA)	98
5	Nikkei 225 (Japan)	225

Each of these datasets corresponds to the weekly prices from March 1992 to September 1997. These datasets contain data such as a number of stocks in a dataset, mean return of the individual stocks, the standard deviation of returns, and the correlation between them along with *unconstrained* efficient frontiers. The covariance between stocks, derived from this correlation matrix, has been used to calculate risk for the portfolios.

5.2 Adapting WA for Portfolio Construction

The adapted winnowing algorithm to construct portfolio (WA-PC) is given in Figure 5.1. The explanation regarding the implementation of WA-PC is given in following sub-sections.

5.2.1 Parameters

Table 5.2 given below summarizes various parameters required for the WA-PC. Note that parameter values given in this table are not optimal and used only for the explanation purpose of WA-PC in this section.

Table 5.2 Parameter Set for WA-PC (Not optimized)

Parameter	Meaning	Value
N	Total available stocks	31
K	Cardinality Constraint (Total stocks in a portfolio)	5
λ	Risk-return trade-off coefficient	1.0
w_{min}	Lower bound on the weight of a stock (Floor constraint)	0.01
w_{max}	Upper bound on the weight of a stock (Ceiling constraint)	0.50
P	Pool size	4
γ	Replacement factor (%)	25
S	Stagnation counter	200
Δw_{max}	Maximum fraction of weight used to update portfolio	0.10

```
# Initialize parameters
        N, K, λ, w_min, w_max, P, γ, S, Δw_max
# Construct initial solutions
For i = 1:P
        For j = 1:K
                A_ij ← Randomly selected unique stock ϵ [1,N]
                w_ij ← Random weight between w_min and w_max
        End For
        Normalize weights                                  # Figure 4.2
End For

# Iterative loop
Counter = 0
While (true)
        # Update each solution
        For i = 1:P
                        # Update solution weights
                Δw ← Generate K random weights between 0 and Δw_max
                Sort Δw and remove trailing weights        # Equation (4.2)
                For j = 1:K
                        w_ij = w_ij + Δw_j                 # Figure 4.3
                End For
                Normalize weights                          # Figure 4.2
                        # Evaluate solution
                f_i ← Objective function value for i^th solution   # Equation (2.23)
                        # Improve solution using Local Search
                Select two different random stocks α ϵ A_i and β ϵ [1, N]
                A_i' ← New solution by replacing α with β in A_i
                f_i' ← Objective function value for new solution   # Equation (2.23)
                If (f_i' > f_i)
                        Replace A_i with A_i' and f_i with f_i'
                End If
        End For

                # Update pool of solutions
        For i = 1:γ                                        # γ in percentage
                Select the worst solution in a pool
                Replace it with the best solution
        End For

                # Check for stagnation, exit if true
        Solution_best ← Best solution with maximum f_i, i = 1…P
        If (No improvement in Solution_best), Counter = Counter + 1;  Else Counter = 0; End If
        If (Counter > S), Exit, End If
        If (Counter = S/2), Δw_max = max(0.0001, Δw_max / 10), End If     # Update Δw_max
End While
```

Figure 5.1 WA-PC Algorithm

The parameters N, K, λ, w_{min}, and w_{max} are portfolio-specific parameters. Their values depend upon the stock market as well as preferences of the individual investors. The stocks referred in this example belong to the Hang Seng dataset listed in Table 5.1. The $N = 31$ suggests total available stocks in this dataset. The $K = 5$ suggests that the algorithm is considered to construct portfolio selecting five stocks among available 31 stocks. The $\lambda = 1$ suggests that the algorithm aims to maximize the expected return and ignores the risk entirely. The $w_{min} = 0.01$ specifies lower bound on the weight of stocks and ensures relevant participation of all K stocks in a portfolio. The $w_{max} = 0.5$ specifies upper bound on the weight of stocks and allows two stocks to dominate other stocks in a portfolio.

The other parameters P, γ, S, and Δw_{max} are algorithm-specific parameters. For the sake of simplicity of the example given here, non-optimized hypothetical values have been considered for these parameters as given in Table 5.2. Note that with γ being 25% and P being 4, one worst solution out of a pool of four solutions is replaced with the best solution in each iteration of the algorithm. This algorithm uses dynamic values for Δw_{max} in contrast to the fixed value used in WA-PWO algorithm as discussed in the section 5.3.1 .

5.2.2 Initial Solutions

Each solution is a feasible portfolio. It can be represented as a vector of real numbers specifying weights of selected stocks to be included in a portfolio. Two different vectors are used to represent solutions to keep this explanation simple: First, asset vector A_i represents selected stocks for the i^{th} solution. Second, weight vector w_i represents weights of selected stocks for the i^{th} solution. Initially, for each solution, K stocks are selected randomly out of N available stocks. Once K stocks are selected, their weights are selected randomly in the range of w_{min} and w_{max}. These weights may violate constraints such as budget constraint and floor-ceiling constraint. So, these weights are normalized to confine them to such constraints using *Weight-Normalizer* algorithm presented in Figure 4.2.

Table 5.3 represents a sample pool of four portfolios (i.e., initial solutions) along with their expected return, risk and objective function values. The CCMV model presented in section 2.5 is used to calculate expected return, risk and objective function values. This table also shows average values of expected return, risk and objective function for the entire pool.

Table 5.3 Initial Solutions for WA-PC

P	Asset Vector					Weight Vector					Return	Risk	Objective Function
	A_1	A_2	A_3	A_4	A_5	W_1	W_2	W_3	W_4	W_5			
P_1	3	4	5	11	25	.2269	.1960	.2621	.1242	.1908	.00484	.00165	.00484
P_2	2	17	20	25	26	.2958	.1640	.0265	.3717	.1419	.00309	.00136	.00309
P_3	3	18	22	26	28	.1033	.0319	.3552	.3545	.1551	.00290	.00088	.00290
P_4	8	14	15	17	21	.2121	.0883	.1625	.2896	.2476	.00276	.00126	.00276
										Average	.00340	.00128	.00340

5.2.3 Updating Solutions

This step is the heart of the winnowing algorithm. Individual solutions are updated in this step to achieve primary aspects of any natural computing algorithms – exploitation and exploration. For this purpose, solutions are first updated by modifying weights of stocks in a portfolio. These updated solutions are evaluated and then they are further improved by applying local search. These steps have been explained below.

- **Update Solution Weights**

This step is exactly the same as that of WA-PWO algorithm. Small fractional weights are added to the weight of stocks in a portfolio to update solution weights. For this purpose, a series of K random fractional weights (Δw) is generated in the range of 0 to Δw_{max}. This series is sorted in descending order so that lighter weights will move towards the end of the series. Similar to the winnowing process, lighter weights are considered as impurities and removed, i.e., truncated to zero. The total number of trailing fractional weights to be truncated depends upon K and w_{max} as given in Equation (4.2). For example, if $w_{max} = 0.50$, it means two stocks can dominate other stocks in a portfolio. So, if $K = 5$, remaining 3 fractional weights ($K - 2 = 5 - 2 = 3$) in Δw are truncated. Table 4.5 given in the previous chapter explains the selection of Δw.

Stock weights are updated by adding fractional weights (Δw) to the weight of stocks in a portfolio. Two different approaches are used to select stocks for updating their weights so that either generated solutions can be exploited or search space can be explored. The pseudo-code given in Figure 4.3 of the previous chapter provides the procedure to select one of these two approaches.

- *Exploit the solution:* For the purpose of exploiting the solution, i.e., to improve the quality of the solution, stocks are selected to update their weights based on their influence in the portfolio. In a winnowing process, it can be said that heavier particles attract other heavier particles and lighter particles attract other lighter particles. Similarly, in this approach, stocks having comparatively higher weights are considered as more essential stocks in a portfolio and they attract more weights.

- *Explore the search space:* In contrast to the above approach, to explore the available search space of feasible portfolios, stocks are selected to update their weights purely in a random manner. In this approach, the updated solution can be either better or worse compared to the original solution. But, it helps the winnowing algorithm to avoid local minima.

Table 4.6 of the previous chapter explains various operations to update the weight of stocks. Table 5.4 represents the pool of four portfolios with updated weights at the end of this step. Note that with changes in weight of stocks, the expected return and risk have also been changed. The higher average of the expected return for a pool, compared to that of given in Table 5.3, shows improvement in the overall quality of the solutions.

Table 5.4 Solutions after Updating Weights for WA-PC

P	**Asset Vector**					**Weight Vector**					**Return**	**Risk**	**Objective Function**
	A_1	**A_2**	**A_3**	**A_4**	**A_5**	**W_1**	**W_2**	**W_3**	**W_4**	**W_5**			
P_1	3	4	5	11	25	.2487	.1757	.2931	.1114	.1711	.00504	.00168	.00504
P_2	2	17	20	25	26	.3238	.1418	.0904	.3213	.1227	.00328	.00134	.00328
P_3	3	18	22	26	28	.0867	.0267	.3803	.3762	.1301	.00296	.00091	.00296
P_4	8	14	15	17	21	.1791	.0745	.2129	.3243	.2091	.00266	.00121	.00266
										Average	.00348	.00128	.00348

Observe that the objective function value for the portfolios P_1, P_2 and P_3 has increased suggesting improvement in these portfolios and for the portfolio P_4 has decreased suggesting decline in the portfolio. This shows that this step is helpful to exploit the solutions as well as to explore the search space.

In the winnowing process, the relative proportion of the desirable components keeps increasing in the remaining mixture iteration by iteration. Similarly, in winnowing algorithm, the relative weight of the desirable stocks keeps increasing in the longer term – resulting in better quality of the solutions.

- **Evaluate Solution**

Each solution is evaluated to determine its quality in terms of expected return, risk and objective function value. For this purpose, Equation (2.23) of the CCMV model has been used as an objective function. As this model attempts to maximize an objective function, solutions with higher objective function values are considered better solutions.

- **Improve Solution using Local Search**

To further improve the quality of the solution, a simple approach has been used as a local search. This approach is slightly different from the approach used in the WA-PWO. The primary objective behind applying local search here is to improve the solution by selecting better stocks.

For this purpose, two different stocks are selected in a random manner. The first stock α is selected from *K* stocks of the portfolio. The second stock β is selected from the pool of available *N* stocks. Note that β can also be a member of the portfolio but different from the stock α. The stock α is replaced by the stock β to get a new solution. The resultant solution is evaluated and compared with the original solution. If the new solution has better objective function value, it replaces the original solution in a pool. Else, it is discarded, and the original solution is kept as it is.

Table 5.5 represents resultant solutions after applying local search. In this example, solutions P_3 and P_4 have been improved by replacing the stock 3 with the stock 13 in solution P_3 and the stock 17 with the stock 18 in solution P_4. This step also improves the average objective function value of the entire pool.

Table 5.5 Improved Solutions after Applying Local Search for WA-PC

P	Asset Vector					Weight Vector					Return	Risk	Objective Function
	A_1	A_2	A_3	A_4	A_5	W_1	W_2	W_3	W_4	W_5			
P_1	3	4	5	11	25	.2487	.1757	.2931	.1114	.1711	.00504	.00168	.00504
P_2	2	17	20	25	26	.3238	.1418	.0904	.3213	.1227	.00328	.00134	.00328
P_3	*13*	18	22	26	28	.0867	.0267	.3803	.3762	.1301	.00322	.00089	.00322
P_4	8	14	15	*18*	21	.1791	.0745	.2129	.3243	.2091	.00269	.00141	.00269
										Average	.00356	.00133	.00356

5.2.4 Updating Pool

The γ percentage worse solutions of the entire pool are replaced with the best solution. As γ is 25% in this example, one worst solution P_4 is replaced by the best solution P_1. Table 5.6 represents the resultant solutions. The higher average of the objective function value in this table compared to Table 5.5 indicates improvement in the overall quality of the pool. It also creates better chances to have a broader exploration of the search space towards the direction of optimal solutions in the following iterations.

Table 5.6 Solutions after Updating Pool for WA-PC

P	Asset Vector					Weight Vector					Return	Risk	Objective Function
	A_1	A_2	A_3	A_4	A_5	W_1	W_2	W_3	W_4	W_5			
P_1	3	4	5	11	25	.2487	.1757	.2931	.1114	.1711	.00504	.00168	.00504
P_2	2	17	20	25	26	.3238	.1418	.0904	.3213	.1227	.00328	.00134	.00328
P_3	13	18	22	26	28	.0867	.0267	.3803	.3762	.1301	.00322	.00089	.00322
P_4	3	4	5	11	25	.2487	.1757	.2931	.1114	.1711	.00504	.00168	.00504
										Average	.00415	.00140	.00415

5.2.5 Termination Criteria

Stagnation in the best solution has been considered as termination criteria for this algorithm similar to the WA-PWO. A solution with the maximum objective function value is considered as the best solution and represents the best portfolio. If the best solution cannot be improved for *S* number of iterations consecutively, it will be considered as an optimal portfolio and algorithm will stop.

Again, each solution is updated, the entire pool is updated, and the termination criterion is checked. These steps are repeated iteratively until the termination criterion is met.

It is possible with WA-PC that wrong stocks may get selected initially or wrong stocks may get more weights resulting in poor quality of the solutions. But, with the progress of WA-PC, such solutions are either improved by local search or replaced by the best solution.

With the progress of the algorithm, the better stock becomes part of the portfolio, and more essential stocks with higher weights keep attracting more weights resulting in better quality of solutions. The WA-PC attempts to select desirable stocks in a portfolio along with increasing their relative weight in a portfolio iteration by iteration. This is similar to the real world winnowing process which attempts to increase the relative weight of the desirable components in the given mixture in successive iterations.

5.3 Results and Discussion

This section discusses the actual parameter values used along with obtained experimental results and comparative analysis of the performance of the WA-PC with other NCAs. The WA-PC has been tested on the five different datasets discussed in section 5.1 . The algorithm has been implemented using MATLAB R2015b. It has been executed on a system having Intel(R) Core(TM) i5-2410M processor @2.30GHz with 3 GB RAM and 32-bit Windows 7 operating system.

5.3.1 Parameters

The WA-PC has been applied to select stocks and their weights to construct portfolios. Various parameter values used for this purpose are listed in Table 5.7 given below.

Table 5.7 Parameter Set for WA-PC

Parameter	Meaning	Value
N	Total available stocks	31-225 (Table 5.8)
K	Cardinality Constraint (Total stocks in a portfolio)	10
λ	Risk-return trade-off coefficient	(Table 5.8)
w_{min}	Lower bound on the weight of a stock (Floor constraint)	0.01
w_{max}	Upper bound on the weight of a stock (Ceiling constraint)	1.00
P	Pool size	20
γ	Replacement factor (%)	20
S	Stagnation counter	200
Δw_{max}	Maximum fraction of weight used to update portfolio	[0.01, 0.001, 0.0001]

The parameters N, K, λ, w_{min}, and w_{max} are portfolio-specific parameters. The values for N vary from 31 to 225 according to the dataset being used as per Table 5.8. Different values of the λ for different datasets have been used to generate 51 different portfolios as per Table 5.8. Practical observations have found that the portfolio returns are higher values compared to portfolio risks (variances) and so, the objective function given by Equation (2.23) gets skewed towards portfolio returns. It starts maximizing expected return and ignores risk much earlier than λ reaches to 1.

Due to this reason, to provide the perfect balance between expected return and risk, different maximum values of λ (λ_{max}) have been used for different datasets rather than theoretical value 1. These values have been determined empirically and represented in Table 5.8. For example, the objective function starts ignoring portfolio risk in comparison to return of the portfolio around λ = 0.65 for the Hang Seng dataset. So, the maximum allowed value of λ is kept 0.65 with a step size of 0.013 (0.65/50) to generate 51 different portfolios. Also, the last value of λ in this series is replaced with 1.0 so that risk can be ignored completely.

Table 5.8 The Values of N and λ_{max} for Different Datasets of OR-Library

Sr. No.	Dataset	N	λ_{max}	Step Size
1	Hang Seng	31	0.65	0.0130
2	DAX 100	85	0.85	0.0170
3	FTSE 100	89	0.55	0.0110
4	S&P 100	98	0.88	0.0176
5	Nikkei 225	225	0.90	0.0180

The values for K, $w_{min,}$ and w_{max} have been kept 10, 0.1 and 1.0 respectively. Values for all of these parameters have been kept the same as used by other researchers. This is to maintain the homogeneity of results to make a comparative analysis. The other parameters P, S, γ, and Δw_{max} are algorithm-specific parameters. Their values have been determined empirically. The values of P, S, and γ have been kept 20, 200 and 20% respectively. For the value of the Δw_{max}, this algorithm uses a different approach compared to that of WA-PWO as discussed below.

Practical observations have found that the algorithm should have high exploration power in early iterations. In contrast to this, the algorithm reaches in the right part of the search space during late iterations and requires fine (small) exploration. In winnowing algorithm, the value of Δw_{max} specifies the proportion of the weight of stocks to be updated. By this, it affects the convergence time of the algorithm as well as the quality of the obtained optimal solution. With the high value of Δw_{max} (around 0.10), the algorithm gives a faster result, but solution sticks in local minima and the optimal solution could not be found. With a small value of Δw_{max} (around 0.0001), the algorithm progresses smoothly and finds the optimal solution but unnecessarily increases required iterations to get convergence. So a balanced value (0.001) was used in WA-PWO.

The WA-PC is allowed to vary values of Δw_{max} during its execution instead of using fixed static value throughout all the iterations. This algorithm begins with a higher value of Δw_{max} = 0.01 and reduces it by 10% to 0.001 whenever the current best solution achieves half stagnation. This value is further reduced to 0.0001 to have a fine (small) exploration of the search space during late iterations of the algorithm.

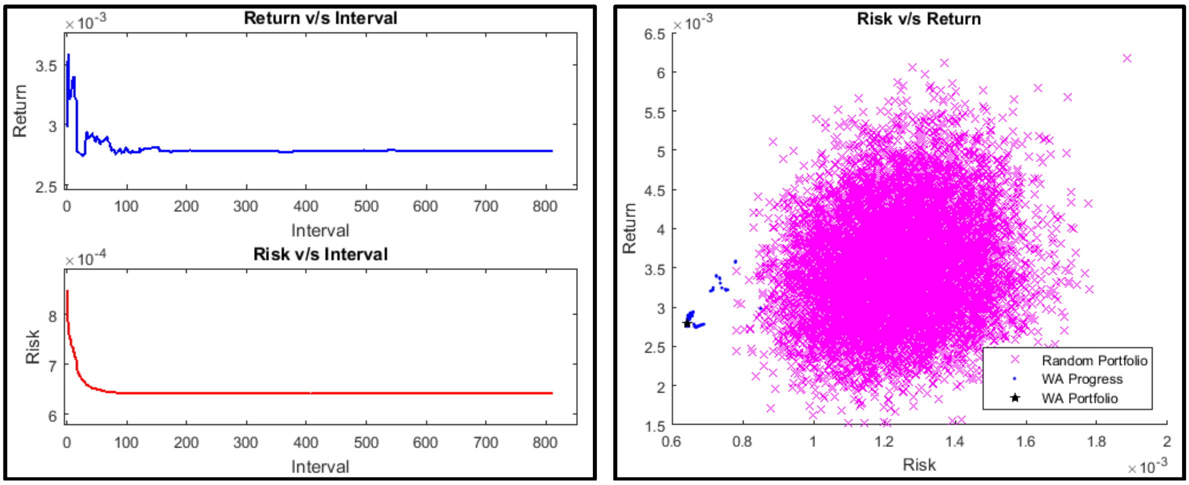
5.3.2 Experimental Results

The WA-PC has been tested on the five different datasets listed in Table 5.1. For each dataset, a total of 51 portfolios have been generated using different values of the λ as given in Table 5.8. For each of these different values of λ, the algorithm has been executed 10 times separately to obtain best, average and worst results.

Figure 5.2 shows the progress of the WA-PC to find the optimal solution for the λ = 0.0 for the Hang Seng dataset. Figure 5.2 (a) provides charts for expected return v/s iterations and risk v/s iterations. Note that if λ = 0.0, the algorithm will only attempt to minimize risk avoiding return. This figure illustrates how the algorithm proceeds to minimize the risk of a portfolio in successive iterations. Observe that, as the risk reduces the expected return also reduces. Figure 5.2 (b) displays the best portfolios (blue dots) found progressively along with the optimal portfolio (black star). This figure also displays randomly generated portfolios (pink cross) to provide an idea about the distribution of feasible solutions across a probable search space.

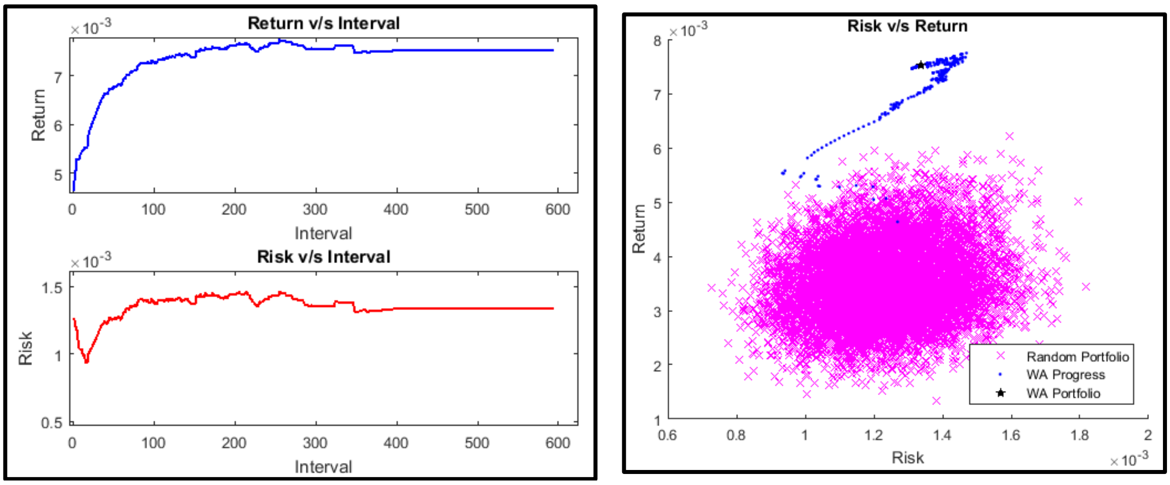
Figure 5.3 shows the progress of the WA-PC to find the optimal solution for the λ = 0.325 for this dataset. Here, λ_{max} being 0.65, the algorithm with λ = 0.325 attempts to construct a portfolio with balanced expected return and risk value. Figure 5.3 (a) represents how the algorithm proceeds in successive iterations. Figure 5.3 (b) displays the best portfolios (blue dots) found progressively along with the optimal portfolio (black star) and randomly generated portfolios (pink cross).

Similarly, Figure 5.4 shows the progress of the WA-PC to find the optimal solution for the λ = 1.0 for this dataset. Figure 5.4 (a) represents how the algorithm proceeds to maximize expected return in successive iterations. Figure 5.4 (b) displays the best portfolios (blue dots) found progressively along with the optimal portfolio (black star) and randomly generated portfolios (pink cross).



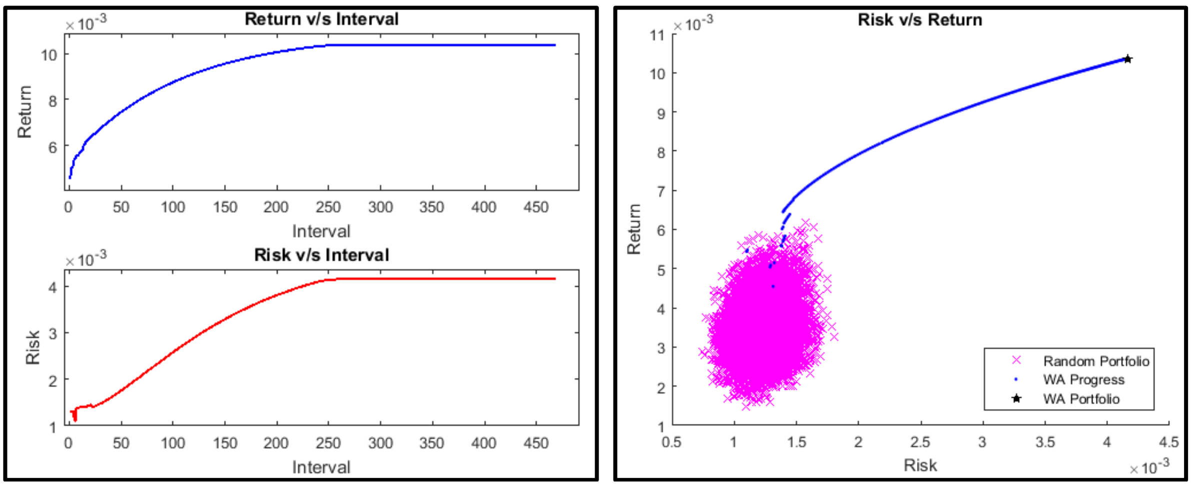
(a) Risk, Return v/s Interval Graph **(b) Risk v/s Return Graph**

Figure 5.2 WA-PC Progress for λ = 0.0



(a) Risk, Return v/s Interval Graph **(b) Risk v/s Return Graph**

Figure 5.3 WA-PC Progress for λ = 0.325



(a) Risk, Return v/s Interval Graph **(b) Risk v/s Return Graph**

Figure 5.4 WA-PC Progress for λ = 1.0

Table 5.9 to Table 5.13 summarizes the obtained experimental results for five different datasets of OR-Library. Results for eleven different λ values out of total 51 values have been presented in these tables for each of the datasets. These λ values are selected from 0.0 to λ_{max} with a step size of $\lambda_{max}/10$. For each of these λ values best, average and worst values of expected return and risk are given in these tables. Along with this, these tables also provide average iterations and average time (in seconds) taken by the algorithm to find optimal solutions.

Different optimal portfolios (total 51) obtained by the WA-PC are plotted in Figure 5.5 to Figure 5.9 for each of the five datasets. These portfolios represent the efficient frontiers for several datasets obtained using the WA-PC. These figures also show unconstrained efficient frontiers comprising of 2000 different portfolios given in [121] for each of the five datasets. These unconstrained efficient frontiers are referred to as standard efficient frontiers. The performance of the WA-PC is evaluated by comparing obtained efficient frontiers with these standard efficient frontiers as discussed in next sub-section.

The graphs depicted in Figure 5.5 to Figure 5.9 prove the ability of the WA-PC to generate efficient frontiers for different datasets successfully. As these datasets are comprised of a different number of stocks varying from 31 to 225, obtained results also show the robustness of the algorithm in finding optimal solutions.

Along with this, the study of results presented in Table 5.9 to Table 5.13 shows that there are minor variations in best and worst results with respect to average results for expected return and risks values. This proves the ability of the WA-PC to find accurate optimal solutions in a consistent manner. This also shows the ability of the algorithm to avoid local optima successfully. The smaller values of the average iterations and time taken to find optimal solutions show that the WA-PC is able to find optimal solutions in reasonable time durations.

Table 5.14 presents each of the 51 different portfolios obtained using the WA-PC for five different datasets. The best values for the expected return and risk are shown in this table for each of these portfolios. Note that this table summarizes various portfolios depicted in Figure 5.5 to Figure 5.9.

Table 5.9 Results for the Hang Seng Dataset

λ	Expected Return			Risk (Variance)			Average	
	Best	Average	Worst	Best	Average	Worst	Iterations	Time (s)
0.000	0.002784	0.002784	0.002783	0.000642	0.000642	0.000642	1029.50	1.05
0.065	0.004587	0.004580	0.004578	0.000700	0.000699	0.000699	959.30	1.24
0.130	0.005567	0.005566	0.005552	0.000803	0.000803	0.000801	903.60	1.18
0.195	0.006462	0.006464	0.006471	0.000978	0.000978	0.000980	709.10	0.91
0.260	0.007070	0.007067	0.007068	0.001151	0.001150	0.001150	589.30	0.72
0.325	0.007522	0.007489	0.007373	0.001339	0.001323	0.001271	602.80	0.74
0.390	0.008064	0.008061	0.008049	0.001643	0.001642	0.001638	637.40	0.82
0.455	0.008625	0.008626	0.008629	0.002054	0.002054	0.002057	822.00	1.07
0.520	0.009291	0.009291	0.009291	0.002693	0.002693	0.002693	853.40	1.15
0.585	0.009911	0.009911	0.009911	0.003448	0.003448	0.003448	819.20	1.04
1.000	0.010359	0.010359	0.010358	0.004161	0.004161	0.004160	493.80	0.65

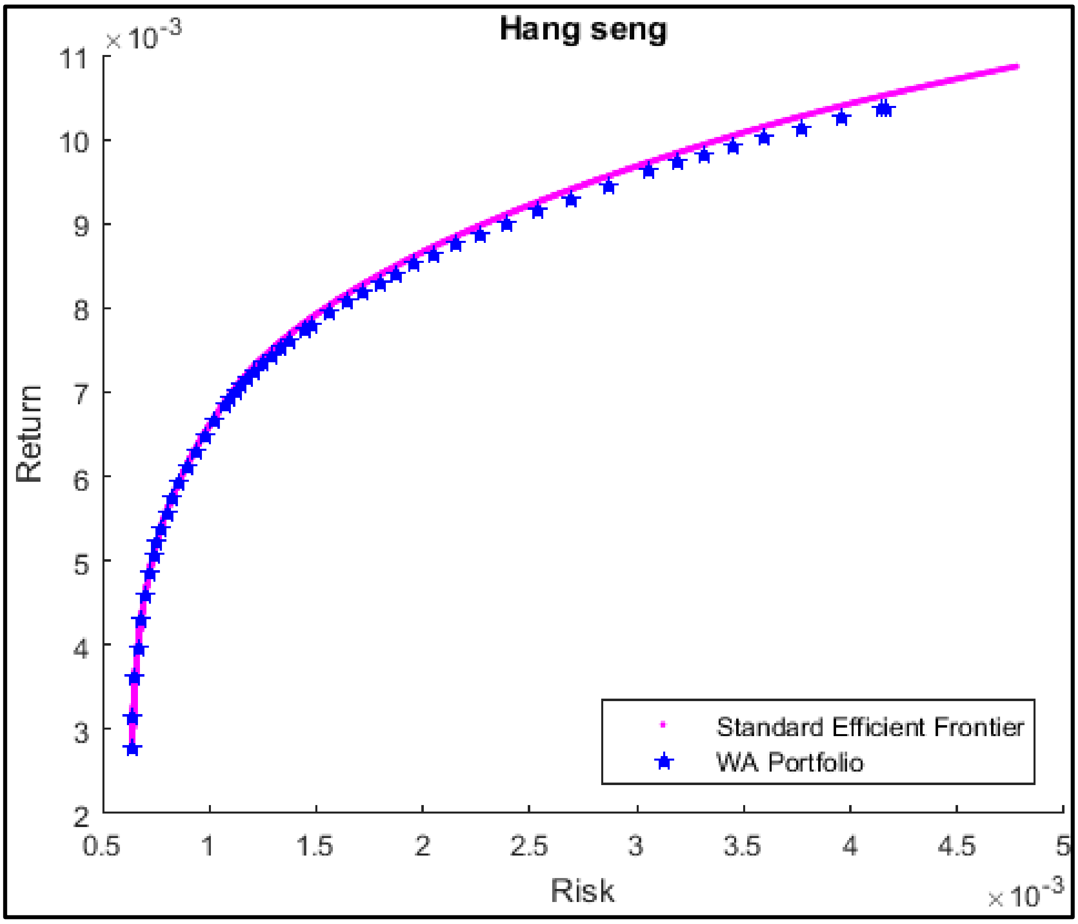


Figure 5.5 Efficient Frontiers for the Hang Seng Dataset

Table 5.10 Results for the DAX 100 Dataset

λ	Expected Return			Risk (Variance)			Average	
	Best	Average	Worst	Best	Average	Worst	Iterations	Time (s)
0.000	0.002144	0.002083	0.002042	0.000148	0.000148	0.000148	904.60	1.30
0.085	0.006380	0.006380	0.006380	0.000308	0.000308	0.000308	918.40	1.20
0.170	0.008012	0.008013	0.008021	0.000524	0.000524	0.000526	931.70	1.54
0.255	0.008462	0.008461	0.008461	0.000645	0.000645	0.000645	813.90	1.35
0.340	0.008606	0.008606	0.008606	0.000705	0.000705	0.000705	1023.00	2.11
0.425	0.008763	0.008763	0.008763	0.000804	0.000804	0.000804	985.40	1.86
0.510	0.008905	0.008905	0.008905	0.000924	0.000924	0.000924	789.90	1.34
0.595	0.008954	0.008954	0.008954	0.000985	0.000985	0.000985	750.80	1.50
0.680	0.009028	0.009028	0.009028	0.001119	0.001119	0.001119	820.20	1.41
0.765	0.009156	0.009156	0.009156	0.001463	0.001463	0.001463	776.00	1.29
1.000	0.009376	0.009376	0.009376	0.002394	0.002394	0.002394	508.20	0.77

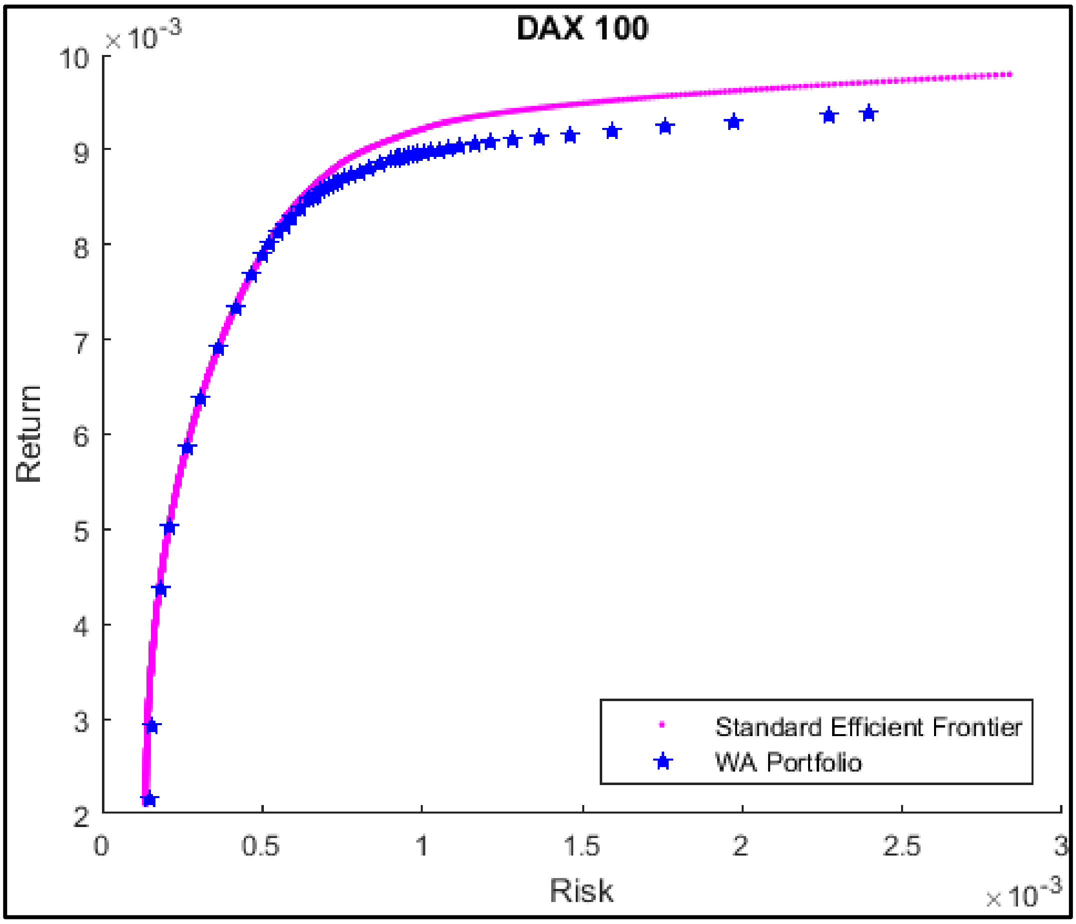


Figure 5.6 Efficient Frontiers for the DAX 100 Dataset

Table 5.11 Results for the FTSE 100 Dataset

λ	Expected Return			Risk (Variance)			Average	
	Best	Average	Worst	Best	Average	Worst	Iterations	Time (s)
0.000	0.002405	0.002439	0.002574	0.000206	0.000206	0.000207	1138.50	1.44
0.055	0.004385	0.004491	0.004489	0.000261	0.000269	0.000269	1159.40	1.43
0.110	0.005576	0.005558	0.005515	0.000357	0.000355	0.000350	1107.40	1.35
0.165	0.006215	0.006215	0.006216	0.000460	0.000460	0.000460	1286.40	1.71
0.220	0.006730	0.006729	0.006730	0.000580	0.000580	0.000581	729.80	1.15
0.275	0.006969	0.006968	0.006967	0.000658	0.000658	0.000658	716.20	1.13
0.330	0.007209	0.007209	0.007209	0.000763	0.000763	0.000763	774.60	1.24
0.385	0.007484	0.007484	0.007482	0.000917	0.000917	0.000916	964.50	1.96
0.440	0.007682	0.007682	0.007682	0.001052	0.001052	0.001053	920.00	1.83
0.495	0.007824	0.007824	0.007824	0.001177	0.001177	0.001177	957.70	1.67
1.000	0.007958	0.007958	0.007958	0.001322	0.001322	0.001324	515.20	0.83

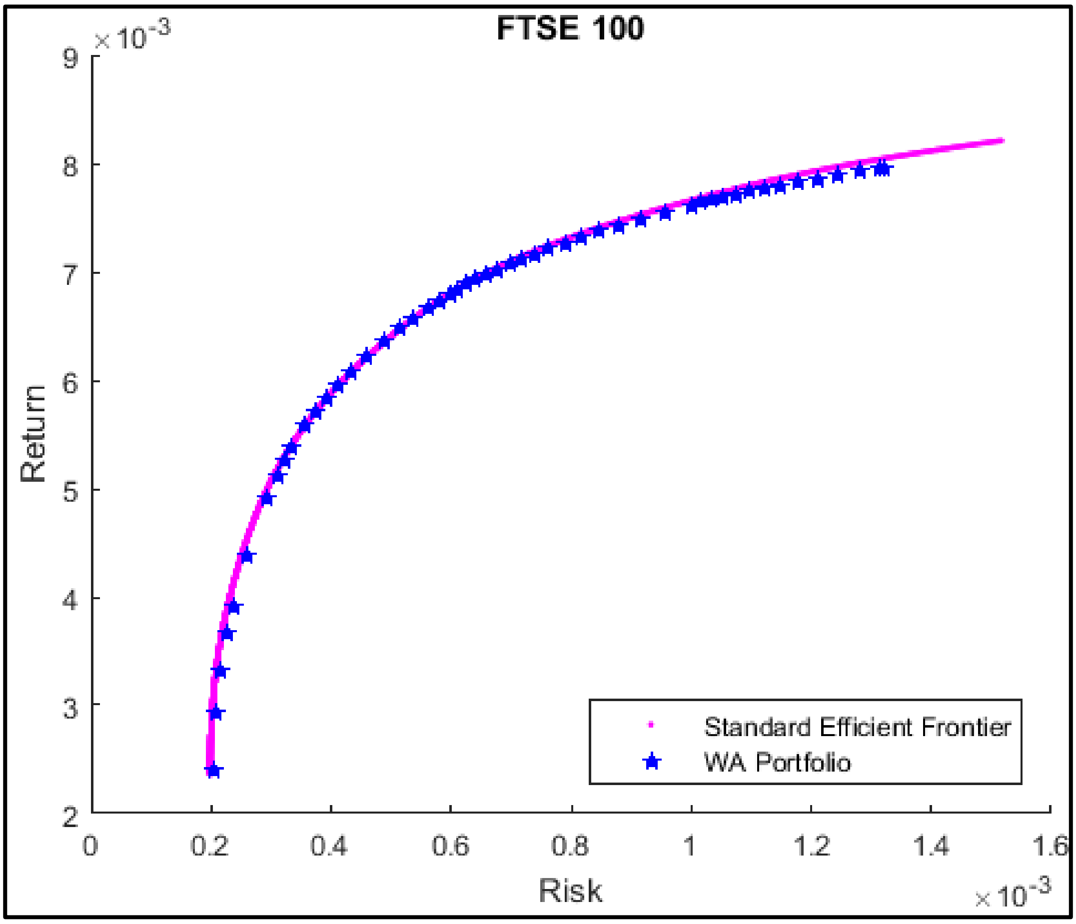


Figure 5.7 Efficient Frontiers for the FTSE 100 Dataset

Table 5.12 Results for the S&P 100 Dataset

λ	Expected Return			Risk (Variance)			Average	
	Best	Average	Worst	Best	Average	Worst	Iterations	Time (s)
0.000	0.001746	0.001795	0.001935	0.000133	0.000133	0.000135	1059.40	1.51
0.088	0.005314	0.005210	0.005019	0.000287	0.000277	0.000260	971.40	1.43
0.176	0.006671	0.006672	0.006680	0.000491	0.000491	0.000492	1000.40	1.46
0.264	0.007499	0.007498	0.007498	0.000724	0.000724	0.000724	1128.50	2.13
0.352	0.008092	0.008073	0.008050	0.000978	0.000968	0.000957	715.70	2.58
0.440	0.008267	0.008286	0.008309	0.001094	0.001111	0.001132	610.50	1.13
0.528	0.008499	0.008502	0.008524	0.001315	0.001319	0.001346	555.60	0.92
0.616	0.008704	0.008707	0.008705	0.001578	0.001583	0.001582	631.50	1.24
0.704	0.008755	0.008754	0.008753	0.001678	0.001678	0.001675	655.80	1.53
0.792	0.008848	0.008848	0.008848	0.001967	0.001967	0.001967	923.80	1.68
1.000	0.008957	0.008957	0.008957	0.002561	0.002561	0.002561	541.90	0.91

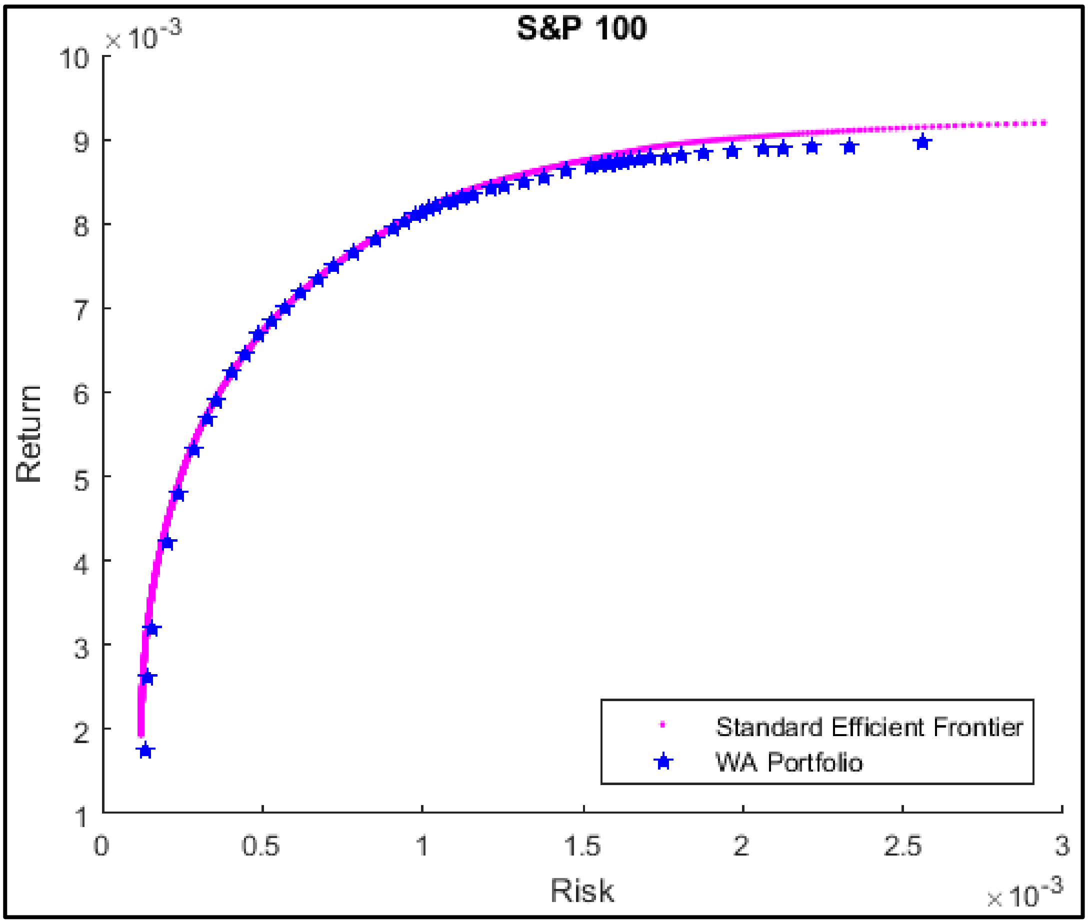


Figure 5.8 Efficient Frontiers for the S&P 100 Dataset

Table 5.13 Results for the Nikkei 225 Dataset

λ	Expected Return			Risk (Variance)			Average	
	Best	Average	Worst	Best	Average	Worst	Iterations	Time (s)
0.000	0.000082	0.000081	0.000074	0.000305	0.000305	0.000305	1226.90	2.29
0.090	0.002156	0.002155	0.002153	0.000405	0.000405	0.000404	1060.90	1.87
0.180	0.003296	0.003296	0.003298	0.000571	0.000571	0.000572	958.10	2.28
0.270	0.003413	0.003413	0.003411	0.000605	0.000605	0.000604	888.90	2.36
0.360	0.003501	0.003501	0.003498	0.000645	0.000645	0.000644	738.40	1.77
0.450	0.003566	0.003565	0.003573	0.000688	0.000688	0.000696	601.90	1.44
0.540	0.003640	0.003638	0.003633	0.000762	0.000760	0.000756	706.50	1.78
0.630	0.003727	0.003726	0.003732	0.000885	0.000885	0.000901	606.50	1.47
0.720	0.003819	0.003821	0.003818	0.001069	0.001077	0.001084	626.80	1.62
0.810	0.003845	0.003845	0.003845	0.001163	0.001163	0.001164	740.90	2.00
1.000	0.003904	0.003904	0.003903	0.001494	0.001494	0.001496	593.90	1.35

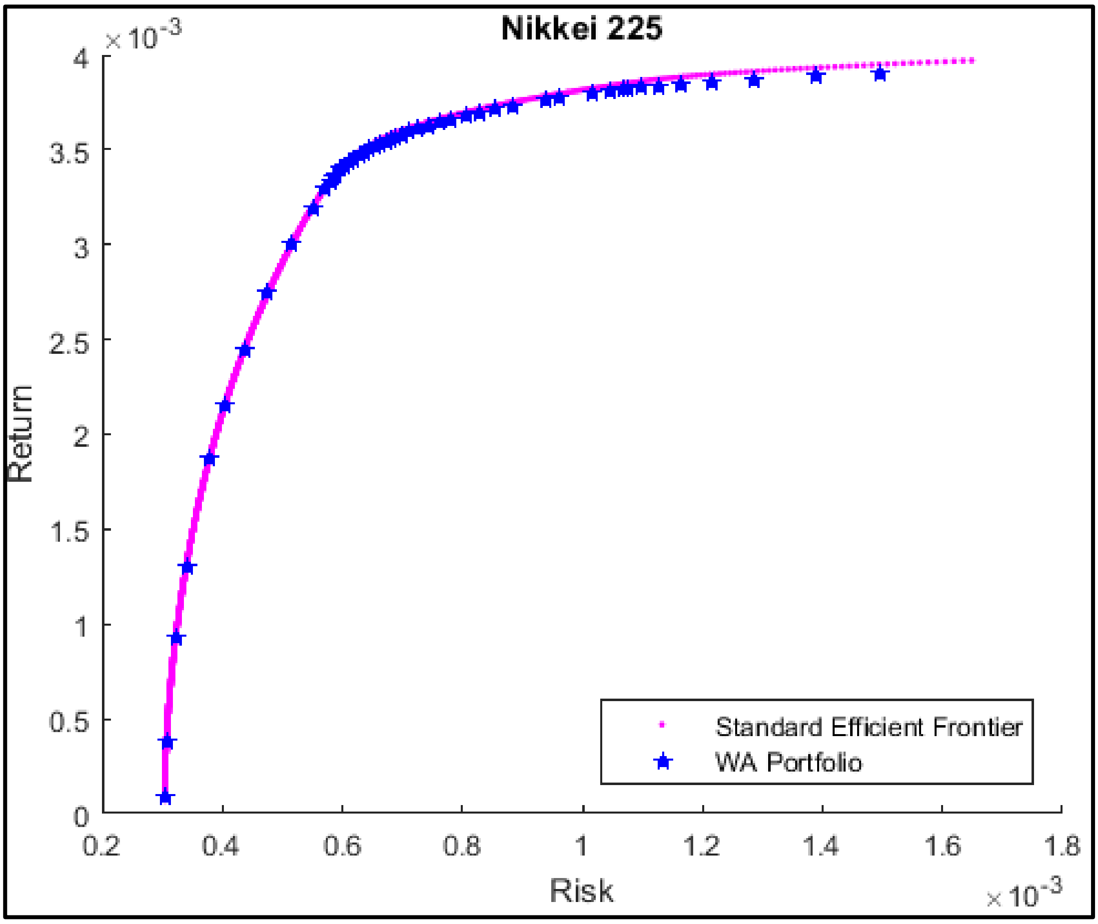


Figure 5.9 Efficient Frontiers for the Nikkei 225 Dataset

Table 5.14 Risk and Return Values of various Portfolios for OR-Library Datasets

P	Hang Seng		DAX 100		FTSE 100		S&P100		Nikkei225	
	Return	**Risk**	**Return**	**Risk**	**Return**	**Risk**	**Return**	**Risk**	**Return**	**Risk**
1	.002784	.000642	.002144	.000148	.002405	.000206	.001746	.000133	.000082	.000305
2	.003139	.000645	.002927	.000152	.002919	.000209	.002609	.000142	.000379	.000308
3	.003613	.000654	.004378	.000186	.003319	.000216	.003179	.000155	.000927	.000324
4	.003964	.000666	.005020	.000214	.003663	.000227	.004219	.000202	.001298	.000342
5	.004295	.000682	.005862	.000265	.003901	.000237	.004784	.000240	.001877	.000380
6	.004587	.000700	.006380	.000308	.004385	.000261	.005314	.000287	.002156	.000405
7	.004843	.000720	.006915	.000363	.004900	.000293	.005675	.000327	.002445	.000436
8	.005055	.000739	.007331	.000415	.005122	.000310	.005894	.000356	.002746	.000476
9	.005216	.000757	.007677	.000465	.005250	.000321	.006224	.000406	.002998	.000515
10	.005377	.000777	.007891	.000501	.005381	.000335	.006443	.000445	.003195	.000550
11	.005567	.000803	.008012	.000524	.005576	.000357	.006671	.000491	.003296	.000571
12	.005739	.000831	.008123	.000548	.005698	.000373	.006841	.000529	.003335	.000580
13	.005917	.000862	.008200	.000567	.005822	.000391	.007004	.000570	.003351	.000584
14	.006100	.000897	.008280	.000589	.005950	.000412	.007176	.000619	.003367	.000589
15	.006289	.000938	.008374	.000617	.006081	.000435	.007349	.000673	.003395	.000598
16	.006462	.000978	.008462	.000645	.006215	.000460	.007499	.000724	.003413	.000605
17	.006652	.001026	.008503	.000660	.006352	.000488	.007656	.000783	.003433	.000612
18	.006831	.001075	.008526	.000669	.006473	.000515	.007822	.000851	.003448	.000619
19	.006910	.001098	.008554	.000681	.006565	.000537	.007953	.000909	.003473	.000630
20	.006988	.001123	.008580	.000692	.006660	.000561	.008021	.000942	.003487	.000637
21	.007070	.001151	.008606	.000705	.006730	.000580	.008092	.000978	.003501	.000645
22	.007159	.001183	.008634	.000720	.006792	.000599	.008129	.001000	.003517	.000654
23	.007241	.001215	.008664	.000737	.006835	.000612	.008164	.001021	.003531	.000663
24	.007333	.001253	.008695	.000757	.006878	.000626	.008196	.001042	.003544	.000672
25	.007418	.001290	.008728	.000779	.006923	.000641	.008242	.001075	.003554	.000679
26	.007522	.001339	.008763	.000804	.006969	.000658	.008267	.001094	.003566	.000688
27	.007595	.001375	.008800	.000832	.007016	.000677	.008302	.001123	.003579	.000699
28	.007728	.001445	.008839	.000864	.007065	.000697	.008346	.001161	.003592	.000711
29	.007796	.001483	.008881	.000901	.007111	.000717	.008403	.001214	.003606	.000725
30	.007934	.001563	.008897	.000916	.007159	.000739	.008445	.001257	.003624	.000744
31	.008064	.001643	.008905	.000924	.007209	.000763	.008499	.001315	.003640	.000762
32	.008180	.001719	.008913	.000933	.007261	.000789	.008552	.001376	.003656	.000781
33	.008292	.001796	.008922	.000943	.007314	.000817	.008612	.001451	.003676	.000808
34	.008401	.001876	.008932	.000955	.007369	.000848	.008669	.001526	.003692	.000830
35	.008511	.001961	.008942	.000969	.007425	.000881	.008693	.001560	.003709	.000855
36	.008625	.002054	.008954	.000985	.007484	.000917	.008704	.001578	.003727	.000885
37	.008745	.002157	.008966	.001004	.007545	.000956	.008711	.001589	.003758	.000939
38	.008872	.002271	.008979	.001026	.007610	.000999	.008720	.001606	.003770	.000963
39	.009004	.002397	.008994	.001051	.007633	.001015	.008730	.001625	.003797	.001018
40	.009144	.002537	.009010	.001082	.007657	.001033	.008742	.001650	.003810	.001047
41	.009291	.002693	.009028	.001119	.007682	.001052	.008755	.001678	.003819	.001069
42	.009447	.002866	.009048	.001162	.007708	.001073	.008768	.001713	.003819	.001068
43	.009611	.003058	.009070	.001216	.007735	.001096	.008784	.001756	.003822	.001076
44	.009717	.003188	.009095	.001281	.007763	.001121	.008802	.001810	.003828	.001097
45	.009814	.003315	.009124	.001362	.007793	.001148	.008823	.001878	.003836	.001126
46	.009911	.003448	.009156	.001463	.007824	.001177	.008848	.001967	.003845	.001163
47	.010015	.003598	.009194	.001592	.007856	.001209	.008873	.002066	.003857	.001214
48	.010125	.003766	.009238	.001758	.007889	.001245	.008886	.002128	.003871	.001286
49	.010242	.003956	.009290	.001977	.007924	.001283	.008903	.002213	.003888	.001390
50	.010351	.004140	.009352	.002272	.007951	.001314	.008924	.002336	.003904	.001494
51	.010359	.004161	.009376	.002394	.007958	.001322	.008957	.002561	.003904	.001494

5.3.3 Comparative Analysis

The obtained efficient frontiers from WA-PC have been compared with the standard unconstrained efficient frontier given in OR-Library. Three different performance metrics have been used for this purpose: mean Euclidean distance (MED), the variance of return error (VRE) and mean return error (MRE) [24], [123].

Let the pair (v_i^s, r_i^s) (i=1,…,2000) represent the variance of the return and mean of the return of portfolios in the *standard* efficient frontier. Similarly, let the pair (v_j^o, r_j^o) (j=1,…,η) represent the variance of the return and mean value of the return of portfolios in the efficient frontier *obtained* using WA-PC. Here η is 51 and represents a total number of portfolios obtained by WA-PC for each of the datasets. Let (v_e^s, r_e^s) represent the nearest portfolio on the standard efficient frontier having minimum Euclidean distance from the obtained portfolio (v_j^o, r_j^o) where e can be found using the following equation.

$$e = \underset{i=1,\dots,2000}{\arg\min} \left(\sqrt{(v_i^s - v_j^o)^2 + (r_i^s - r_j^o)^2} \right) \quad j = 1, \dots, \eta \tag{5.1}$$

MED can be defined as:

$$MED = \left(\sum_{j=1}^{\eta} \sqrt{(v_e^s - v_j^o)^2 + (r_e^s - r_j^o)^2} \right) \text{ x } \frac{1}{\eta} \tag{5.2}$$

Similar to MED, VRE and MRE can be defined as:

$$VRE = \left(\sum_{j=1}^{\eta} 100\, |v_e^s - v_j^o| / v_j^o \right) \text{ x } \frac{1}{\eta} \tag{5.3}$$

$$MRE = \left(\sum_{j=1}^{\eta} 100\, |r_e^s - r_j^o| / r_j^o \right) \text{ x } \frac{1}{\eta} \tag{5.4}$$

Basically, these metrics give the measure of the distance or say closeness between obtained efficient frontier and the standard efficient frontier. The obtained efficient frontier should be as close as possible to the standard one, and the values of these metrics should be as small as possible. So, lower values of these metrics represent a higher quality of the obtained efficient frontier.

Table 5.15 shows the comparative results between standard efficient frontier and efficient frontier obtained using the WA-PC algorithm. Comparisons are given in terms of MED, VRE, and MRE for each of the five datasets. This table also shows the average execution time (AET, in seconds) for each of the dataset. Results of WA-PC have been compared with other NCAs such as GA, TS, SA, PSO, IPSO-SA, mFA and IHS. Results in bold fonts represent the best results. Results in italic fonts represent the second best results.

Note that MED values cannot be zero when VRE and MRE are non-zero. So, the results showing MED as zero from the literature are considered as precision point errors and have been ignored from the comparisons.

Table 5.15 Comparative Results for WA-PC

Dataset	Performance Metric	GA	TS [24]	SA	PSO	IPSO-SA [120]	mFA [122]	IHS [123]	WA-PC
Hang Seng (31 Stocks)	MED	0.0040	0.0040	0.0040	0.0049	0.0001	0.0003	0.0001	**0.000045**
	VRE	1.6441	1.6578	1.6628	2.2421	1.6388	**1.2387**	1.8044	*1.571941*
	MRE	0.6072	0.6107	0.6238	0.7427	0.6059	0.4715	0.6484	**0.329519**
	AET								0.914479
DAX 100 (85 Stocks)	MED	0.0076	0.0082	0.0078	0.0090	0.0001	0.0009	0.0001	**0.000117**
	VRE	7.2180	9.0309	8.5485	*6.8588*	**6.7806**	7.2569	7.3850	7.478892
	MRE	1.2791	1.9078	1.2817	1.5885	1.2770	1.3786	1.0493	**0.902337**
	AET								1.482187
FTSE 100 (89 Stocks)	MED	0.0020	0.0021	0.0021	0.0022	0.0000	0.0004	0.0000	**0.000017**
	VRE	2.8660	4.0123	3.8205	3.0596	2.4701	2.7085	3.2479	**1.773780**
	MRE	0.3277	0.3298	0.3304	0.3640	0.3247	0.3121	0.3203	**0.123284**
	AET								1.482096
S&P 100 (98 Stocks)	MED	0.0041	0.0041	0.0041	0.0052	0.0001	0.0003	0.0001	**0.000059**
	VRE	3.4802	5.7139	5.4247	3.9136	**2.6281**	3.6026	3.9024	*2.662963*
	MRE	1.2258	**0.7125**	0.8416	1.4040	0.7846	0.8993	0.9480	*0.725749*
	AET								1.458012
Nikkei 225 (225 Stocks)	MED	0.0093	0.0010	0.0010	0.0019	0.0000	0.0000	0.0000	**0.000015**
	VRE	1.2056	1.2431	1.2017	2.4274	0.9583	1.2015	1.6021	**0.853139**
	MRE	5.3266	0.4207	0.4126	0.7997	1.7090	0.4892	0.4037	**0.345217**
	AET								1.968306

It can be observed from this table that WA-PC outperforms all other NCAs in terms of MED. For VRE, this algorithm gives the best result for FTSE 100 and Nikkei 225 datasets while the second best result for Hang Seng and S&P 100 datasets. For MRE, this algorithm gives the best result for each dataset except for the S&P 100, for which it gives the second best result. Overall, WA-PC has provided better results compared to other NCAs.

These results prove the competency of the WA-PC algorithm compared to other NCAs in generating efficient frontier or say, in finding out optimal solutions. These results also show that the winnowing algorithm is reliable to provide consistently optimal results across global datasets. This helps to establish the superiority of the winnowing algorithm compared to other NCAs in the context of portfolio construction. Also, the small values of AET indicate the feasibility of the WA-PC to find optimal solutions within a reasonable time duration.

The WA-PC algorithm presented in this chapter has focused on portfolio construction. The performance of the algorithm has been assessed on the five global datasets. The algorithm proved its ability to find optimal solutions (portfolios) successfully in a consistent manner. In reality, stock prices vary over a period of time, and the portfolio must be adjusted to reflect these changes. So, once an optimal portfolio is constructed, its performance must be tracked, and the portfolio should be optimized whenever required. Experimental results obtained in this chapter provide motivation to apply winnowing algorithm to optimize stock market portfolio for the real world data. The next chapter presents the winnowing algorithm in the context of portfolio optimization.

CHAPTER 6

Phase III: Portfolio Optimization

Again, remember the two vital questions regarding investment in the stock market:

- In which stocks to invest? And, in which proportion to invest?

The second phase of this research work discussed in the previous chapter provided answers to these questions. The winnowing algorithm was adapted to construct portfolio – to select stocks and their weights from a pool of available stocks to make initial investments based on the risk-return profile of the investor. This algorithm suggests stocks as well as their proportions to make investments. Once the investment is made, the next vital questions in mind of an investor (or fund manager) will be:

- Should portfolio be allowed to remain as it is? If yes – up to how much time?
- Should portfolio be changed in the future? If yes – when? And, how?

This chapter attempts to answer these questions and represents the third phase of the research work. Stock prices vary over a period of time. So, the portfolio must be adjusted, or say the portfolio must be optimized, to reflect these changes. *Portfolio Optimization* tracks the performance of the portfolio and continuously rebalances it by buying new stocks in a portfolio and/or selling existing stocks of the portfolio.

While the previous two chapters focused on the performance of the winnowing algorithm, this chapter focuses on the performance of the portfolio constructed by the winnowing algorithm. The chapter begins with the description of a real-world dataset used in this chapter. Next two sections discuss static and dynamic portfolios. In the end, comparative analysis is discussed.

6.1 Dataset

The real world stock data derived from the National Stock Exchange (NSE) of India have been used in this phase. For this purpose, required historical data of the stocks which are constituent members of the NIFTY50 has been collected from the NSE Equity Bhavcopy [8]. The NIFTY50, commonly known as NIFTY, is an index representing well diversified 50 stocks of the NSE of India. The historical data for the index itself, i.e., NIFTY50 has been collected from [130].

The NSE Equity Bhavcopy is a compressed file having data in the form of comma separated values (CSV) for stocks. This file contains data such as open price, high price, low price, close price, last price, previous close price, total traded quantities, and values, total trades for stocks which have been traded on the exchange on a specific day. These data have been collected for the duration beginning from 1st April 2010 to 31st March 2019 – nine financial years (FYs). Table 6.1 provides a list of stocks for which these data have been collected. These stocks are either constituent members of the NIFTY50 or have remained members of the NIFTY50 in the past for the above given duration. Data presented in this table includes the symbol of the stock, date of the inclusion in the index or start date (i.e. 1st April 2010), and date of the exclusion from the index or end date (i.e. 31st March 2019). If any of the stocks has been a member of the index for multiple time periods, the last duration of such stocks has been considered in this dataset.

Obtained data has been preprocessed to accommodate effects of corporate actions such as a change in the symbol for the stock, stock splitting, and bonus issues. The historical prices of stocks have been adjusted for stock splitting and bonus issues according to the methodology given in [131].

For each stock and the index, daily return values have been calculated based on their closing prices for each day. These daily return values have been used to calculate mean returns, the standard deviation of returns and covariance of returns among stocks. Further, these data have been used to calculate the expected return and risk of the portfolio using the CCMV model discussed in section 2.5 .

Table 6.1 NIFTY50 Stocks for the duration from 01-APR-2010 to 31-MAR-2019

Symbol	Inclusion Date	Exclusion Date	Symbol	Inclusion Date	Exclusion Date
ABB	01/04/2010	01/10/2010	ITC	01/04/2010	31/03/2019
ACC	01/04/2010	29/09/2017	JPASSOCIAT	01/04/2010	28/03/2014
ADANIPORTS	28/09/2015	31/03/2019	JSWSTEEL	28/09/2018	31/03/2019
AMBUJACEM	01/04/2010	02/04/2018	KOTAKBANK	08/04/2010	31/03/2019
ASIANPAINT	27/04/2012	31/03/2019	LT	01/04/2010	31/03/2019
AUROPHARMA	01/04/2016	02/04/2018	LUPIN	28/09/2012	28/09/2018
AXISBANK	01/04/2010	31/03/2019	M&M	01/04/2010	31/03/2019
BAJAJ-AUTO	01/10/2010	31/03/2019	MARUTI	01/04/2010	31/03/2019
BAJAJFINSV	02/04/2018	31/03/2019	MCDOWELL-N	28/03/2014	19/09/2014
BAJFINANCE	29/09/2017	31/03/2019	NMDC	01/04/2013	28/09/2015
BANKBARODA	27/04/2012	29/09/2017	NTPC	01/04/2010	31/03/2019
BHARTIARTL	01/04/2010	31/03/2019	ONGC	01/04/2010	31/03/2019
BHEL	01/04/2010	31/03/2017	PNB	01/04/2010	01/04/2016
BOSCHLTD	29/05/2015	02/04/2018	POWERGRID	01/04/2010	31/03/2019
BPCL	01/04/2010	31/03/2019	RCOM	01/04/2010	27/04/2012
CAIRN	01/04/2010	01/04/2016	RELCAPITAL	01/04/2010	10/10/2011
CIPLA	01/04/2010	31/03/2019	RELIANCE	01/04/2010	31/03/2019
COALINDIA	10/10/2011	31/03/2019	RELINFRA	01/04/2010	27/09/2013
DLF	01/04/2010	27/03/2015	RPOWER	01/04/2010	27/04/2012
DRREDDY	01/10/2010	31/03/2019	SAIL	01/04/2010	28/09/2012
EICHERMOT	01/04/2016	31/03/2019	SBIN	01/04/2010	31/03/2019
GAIL	01/04/2010	31/03/2019	SIEMENS	01/04/2010	01/04/2013
GRASIM	25/03/2011	26/05/2017	STER	01/04/2010	28/09/2012
HCLTECH	01/04/2010	31/03/2019	SUNPHARMA	01/04/2010	31/03/2019
HDFC	01/04/2010	31/03/2019	SUZLON	01/04/2010	25/03/2011
HDFCBANK	01/04/2010	31/03/2019	TATAMOTORS	01/04/2010	31/03/2019
HEROMOTOCO	01/04/2010	31/03/2019	TATAMTRDVR	01/04/2016	29/09/2017
HINDALCO	01/04/2010	31/03/2019	TATAPOWER	01/04/2010	29/09/2017
HINDPETRO	29/09/2017	31/03/2019	TATASTEEL	01/04/2010	31/03/2019
HINDUNILVR	01/04/2010	31/03/2019	TCS	01/04/2010	31/03/2019
IBULHSGFIN	31/03/2017	31/03/2019	TECHM	28/03/2014	31/03/2019
ICICIBANK	01/04/2010	31/03/2019	TITAN	02/04/2018	31/03/2019
IDEA	27/03/2015	31/03/2017	ULTRACEMCO	28/09/2012	31/03/2019
IDFC	01/04/2010	29/05/2015	UPL	29/09/2017	31/03/2019
INDUSINDBK	01/04/2013	31/03/2019	VEDL	26/05/2017	31/03/2019
INFRATEL	01/04/2016	31/03/2019	WIPRO	27/09/2013	31/03/2019
INFY	01/04/2010	31/03/2019	YESBANK	27/03/2015	31/03/2019
IOC	31/03/2017	31/03/2019	ZEEL	19/09/2014	31/03/2019

6.2 Static Portfolios

Static portfolio follows a buy-and-hold strategy. Once it is constructed, no any changes are made either in selected stocks or in their weights. This section discusses static portfolios. This section attempts to emphasize the need for the performance tracking of the portfolio as well as its rebalancing over a period of time.

6.2.1 Parameters

The WA-PC algorithm discussed in the previous chapter has been applied to select stocks and their weights to construct static portfolios. Various parameter values used for this purpose are listed in Table 6.2 given above.

Table 6.2 Parameter Set to Construct Static Portfolio

Parameter	Meaning	Value
N	Total available stocks	50
K	Cardinality Constraint (Total stocks in a portfolio)	10
λ	Risk-return trade-off coefficient	[0.0, 0.7, 1.0]
w_{min}	Lower bound on the weight of a stock (Floor constraint)	0.01
w_{max}	Upper bound on the weight of a stock (Ceiling constraint)	1.00, 0.15
P	Pool size	20
γ	Replacement factor (%)	20
S	Stagnation counter	200
Δw_{max}	Maximum fraction of weight used to update portfolio	[0.01, 0.001, 0.0001]

The constituent member stocks of the NIFTY50 are used to construct static portfolios. As the NIFTY50 represents fifty different stocks, the value of N is 50 and represents available stocks. The K = 10 suggests that the algorithm will select 10 stocks to construct a portfolio. Three different kind of portfolios with $\lambda = 0.0$ (conservative portfolio), $\lambda = 0.7$ (moderate portfolio), and $\lambda = 1.0$ (aggressive portfolio) are considered here. The conservative portfolio will seek minimum risk irrespective of the return. The aggressive portfolio will seek maximum return irrespective of the risk. The moderate portfolio will seek a balance between the risk and the return avoiding extreme limits. The w_{min} and w_{max} specify that each stock can have 1% to 100% proportion in a conservative and moderate portfolio, while 1% to 15% proportion in an aggressive portfolio. The other parameters P, γ, S, and Δw_{max} are algorithm-specific parameters. The WA-PC algorithm uses the same values for these parameters as discussed in the previous chapter.

6.2.2 Experimental Results

Each of the above discussed portfolios is constructed on the date of 1st April 2011 which is the first day of FY 2011-12. Portfolios are constructed by selecting stocks and their weights based on the historical data of FY 2010-11. The mean value of the returns of the past one year is considered as an expected return for a specific stock. The same data are used to determine the covariance matrix of returns among stocks which is used to determine portfolio risk. In the real-world stock market, the standard deviation is preferred as a measure of risk instead of the variance. So here also, portfolio risk is considered in terms of standard deviation rather than a variance and can be calculated as a square root of the variance. The initial amount to be invested is considered Rs. 100000.

The performance of the individual portfolios is tracked and compared against the performance of the NIFTY50 beginning from 1st April 2011 to 31st March 2019. This duration will be referred to as the *test duration* henceforth. And it includes total 1981 trading days spanning over eight financial years. Note that transaction costs and dividend returns are ignored in this discussion while comparing the performance of portfolios for the sake of simplicity. Also, note that a stock cannot be bought in quantity with real numbers. The units (number of shares) that can be transacted in real-world equity markets are always positive integer numbers and cannot be real values. But to get the exact difference in the performance of the portfolio against the overall market, units are allowed to be real values in this discussion.

- **Conservative Portfolio**

Table 6.3 represents a conservative portfolio. This table gives a list of stocks included in a portfolio by the WA-PC algorithm along with their weights. The number of units of individual stock depends upon their weight in a portfolio, purchase price of the stock and investment amount as given in Equation (6.1).

$$Stock\ Units = \frac{Stock\ Weight * Investment\ Amount}{Stock\ Price} \tag{6.1}$$

This table shows the initial and final price along with the value of the individual stocks. The initial price refers to the opening price of stocks on the portfolio commencement date (01-APR-2011). The final price refers to the closing price of stocks on the last date (31-

MAR-2019). The last column shows returns (%) over a test period for stocks. The bottom part of this table presents similar kinds of data for the NIFTY50. Here, NIFTY50 has been considered as a stock and hypothetical portfolio has been considered to be made by investing the same initial amount of Rs. 100000/- in the NIFTY50 itself.

It can be observed from this table that NIFTY50 has given 99.21% return in the test duration while conservative portfolio has given 101.69% return. The difference in returns shows that conservative portfolio provides almost the same returns as that of the overall market. Their values have been plotted against the time duration in Figure 6.1 to obtain a better idea about how the portfolio performed against the NIFTY50 during the test duration. Daily portfolio values have been calculated based on the closing prices of individual stocks and NIFTY50 for the test duration. From these daily values, the portfolio values at the end of each fiscal quarter have been plotted in this graph.

- **Moderate Portfolio**

A moderate portfolio ($\lambda = 0.7$) attempts to achieve a balance between risk and return of a portfolio instead of only minimizing risk or only maximizing return. Table 6.4 presents this kind of portfolio. Notice the difference between selected stocks and their weights for this portfolio compared to that of a conservative portfolio. This portfolio has given 162.97% return in the test duration and successfully outperforms the overall market. The graphical representation of the portfolio values against time duration is given in Figure 6.2.

- **Aggressive Portfolio**

An aggressive portfolio ($\lambda = 1.0$) attempts to maximize return irrespective of risk values. Table 6.5 presents this kind of portfolio. The mean value of the returns of the past one year is considered as an expected return for a specific stock. The expected return of the portfolio is calculated based on expected returns and weights of individual stocks as given in Equation (2.21). This portfolio has given 100.27% return in the test duration and provides almost similar returns as that of the overall market. The graphical representation of the portfolio values against time duration is given in Figure 6.3.

Table 6.3 Static Conservative Portfolio v/s NIFTY50

Stock	Weight (%)	Units	Initial Price	Initial Total	Final Price	Final Total	Return (%)
Portfolio							
ACC	9.02	8.43	1070.00	9022.44	1666.30	14050.56	55.73
CIPLA	10.86	33.65	322.90	10864.32	528.90	17795.41	63.80
DRREDDY	9.22	5.61	1644.00	9219.17	2780.25	15590.99	69.11
GRASIM	7.38	14.86	496.80	7383.92	857.95	12751.68	72.70
HINDUNILVR	9.84	34.39	286.00	9835.63	1706.80	58697.37	496.78
INFY	9.36	23.16	404.39	9364.11	743.85	17224.69	83.94
POWERGRID	15.00	148.00	101.35	15000.00	197.90	29289.59	95.26
SIEMENS	8.78	9.97	881.00	8783.70	1128.80	11254.30	28.13
SUNPHARMA	8.45	38.23	221.15	8454.94	478.85	18307.25	116.53
TATAPOWER	12.07	91.10	132.51	12071.77	73.80	6723.24	-44.31
Total:	100.00			100000.00		201685.09	**101.69**
NIFTY50							
NIFTY50	100.00	17.14	5835.00	100000.00	11623.90	199209.94	**99.21**

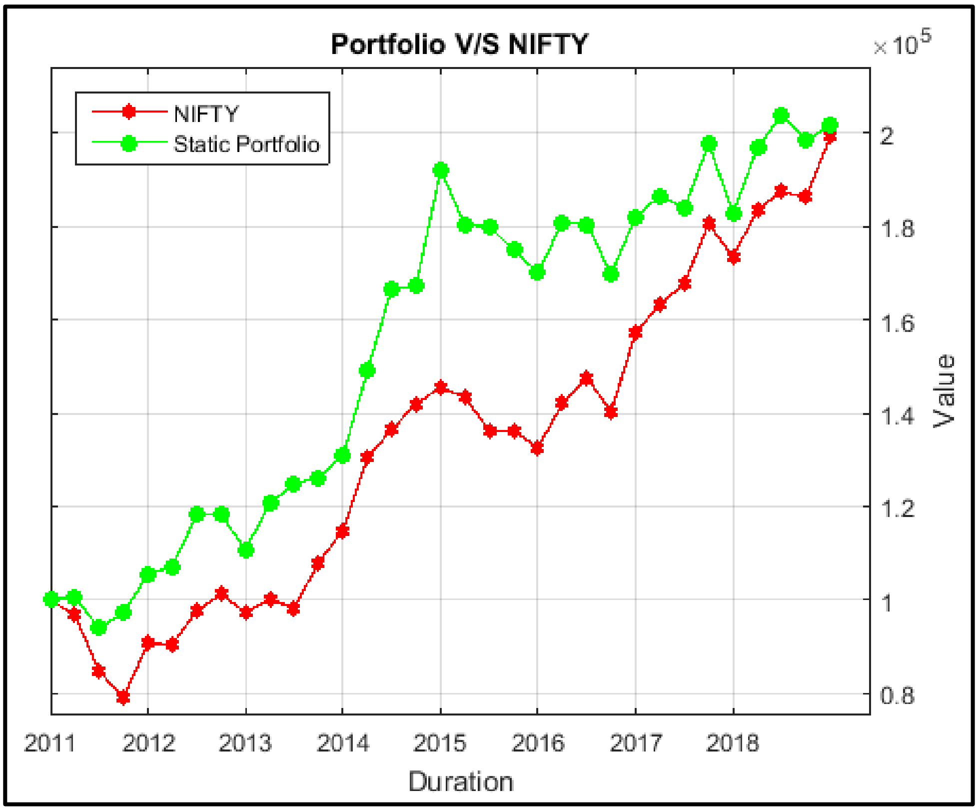


Figure 6.1 Static Conservative Portfolio v/s NIFTY50

Table 6.4 Static Moderate Portfolio v/s NIFTY50

Stock	Weight (%)	Units	Initial Price	Initial Total	Final Price	Final Total	Return (%)
			Portfolio				
ACC	7.96	7.44	1070.00	7958.53	1666.30	12393.73	55.73
BAJAJ-AUTO	14.66	9.92	1478.00	14655.80	2911.10	28866.37	96.96
BPCL	5.00	49.88	100.25	5000.00	397.55	19827.93	296.56
DRREDDY	13.24	8.05	1644.00	13241.44	2780.25	22393.26	69.11
GAIL	6.85	26.35	259.86	6848.26	347.65	9161.85	33.78
HINDUNILVR	12.52	43.79	286.00	12524.88	1706.80	74746.38	496.78
ITC	11.31	93.17	121.37	11308.27	297.25	27695.35	144.91
SIEMENS	9.57	10.87	881.00	9572.72	1128.80	12265.26	28.13
SUNPHARMA	6.76	30.55	221.15	6755.07	478.85	14626.57	116.53
TCS	12.14	20.48	592.50	12135.02	2001.65	40995.90	237.83
Total:	100.00			100000.00		262972.60	**162.97**
			NIFTY50				
NIFTY50	100.00	17.14	5835.00	100000.00	11623.90	199209.94	**99.21**

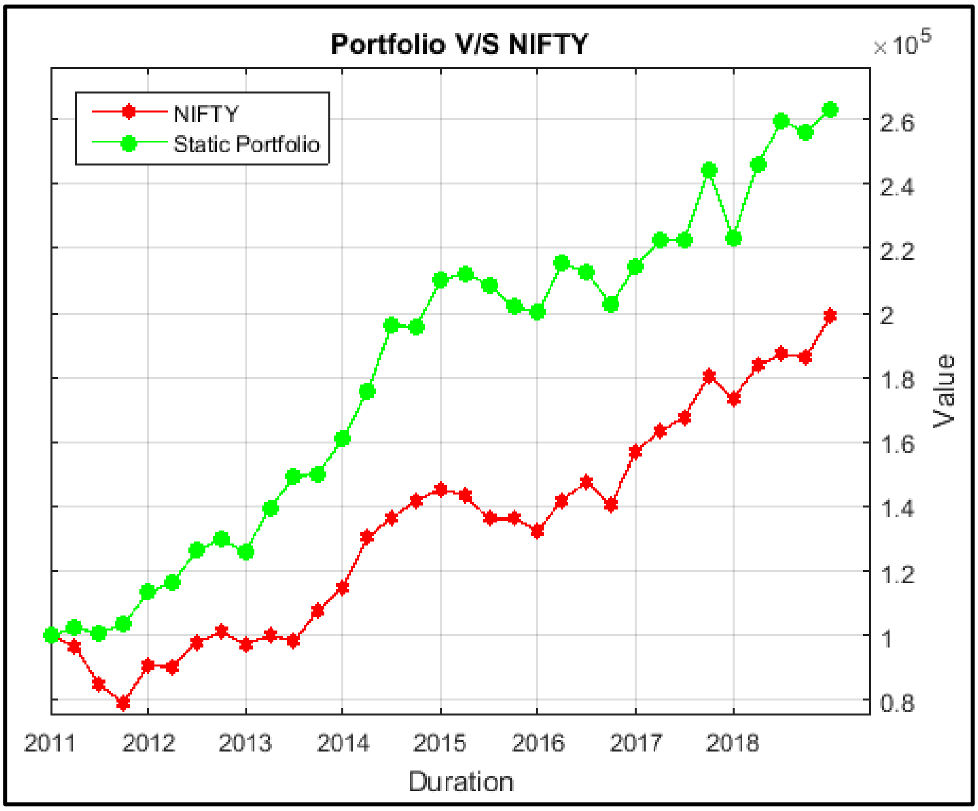


Figure 6.2 Static Moderate Portfolio v/s NIFTY50

Table 6.5 Static Aggressive Portfolio v/s NIFTY50

Stock	Weight (%)	Units	Initial Price	Initial Total	Final Price	Final Total	Return (%)
			Portfolio				
AMBUJACEM	15.00	101.35	148.00	15000.00	235.30	23847.97	58.99
BAJAJ-AUTO	15.00	10.15	1478.00	15000.00	2911.10	29544.32	96.96
DRREDDY	5.00	3.04	1644.00	5002.44	2780.25	8459.87	69.11
HCLTECH	5.00	21.00	238.23	5002.44	1087.45	22834.66	356.47
HDFC	5.00	7.16	699.00	5002.44	1968.25	14085.91	181.58
ITC	5.81	47.83	121.37	5805.56	297.25	14218.52	144.91
M&M	5.00	14.28	350.25	5002.44	673.90	9624.96	92.41
SBIN	14.18	51.30	276.53	14184.69	320.75	16452.97	15.99
TATAMOTORS	15.00	60.44	248.20	15000.00	174.25	10530.82	-29.79
TCS	15.00	25.32	592.50	15000.00	2001.65	50674.68	237.83
Total:	100.00			100000.00		200274.68	**100.27**
			NIFTY50				
NIFTY50	100.00	17.14	5835.00	100000.00	11623.90	199209.94	**99.21**

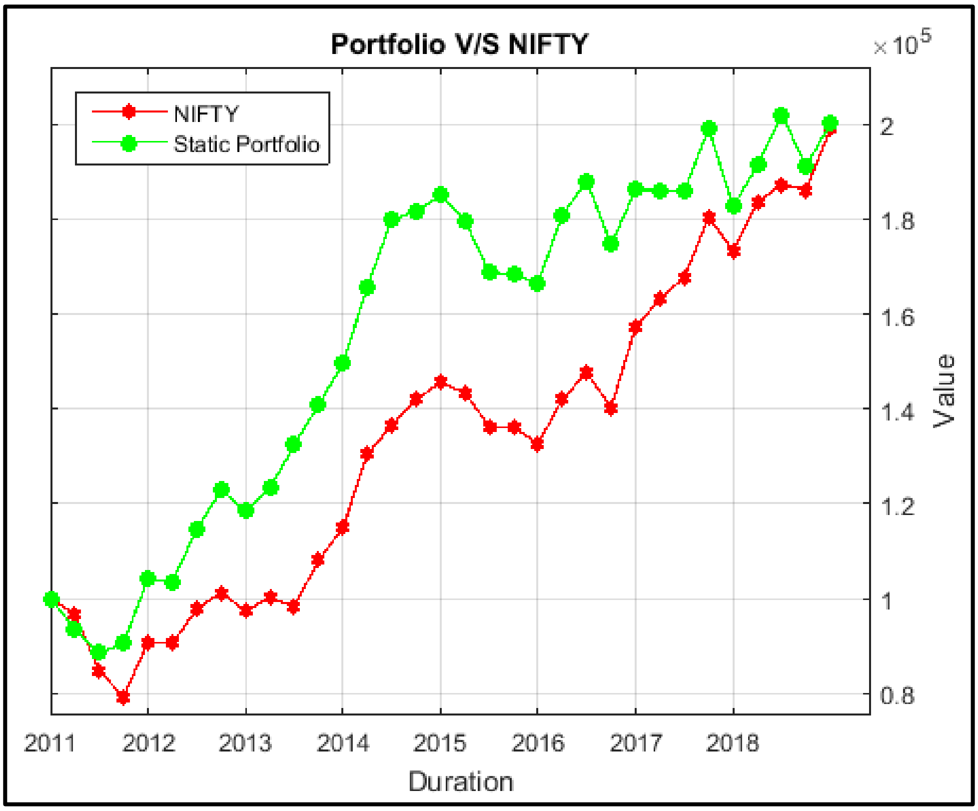


Figure 6.3 Static Aggressive Portfolio v/s NIFTY50

- **Observations**

Study of results presented in Table 6.3 to Table 6.5 suggests that moderate portfolio gives strong returns and outperforms the overall market handsomely. Conservative and aggressive portfolios yield almost similar returns as that of the NIFTY50. In contrast to these statistics, the study of graphs plotted in Figure 6.1 to Figure 6.3 provides a better idea of the performance of the portfolios. These graphs show that all of the three portfolios were successful in outperforming the overall market initially. But, as time passes, their performance deteriorates. All of the three portfolios lose momentum after the year 2015 and could not gain significant returns in the next Bull Run starting from the year 2016. This analysis indicates that however good portfolio today is, it cannot remain the same forever.

The reason behind this is the dynamic nature of the stock market. A number of factors affect the stock price and so the performance of the stock. Today's better-looking stocks may lose relevance with time and may give poor returns in the future. The best portfolio of today may become ordinary tomorrow. So, the portfolio must be adjusted or say optimized to reflect changes over a period of time. The next section discusses portfolio optimization.

6.3 Dynamic Portfolios

The dynamic portfolio does not follow a buy-and-hold strategy. Once it is constructed, it will go through a number of changes either in selected stocks or in their weights over a period of time. This section discusses dynamic portfolios. This section attempts to emphasize what kind of changes in the performance of a portfolio can be achieved by rebalancing it over a period of time.

6.3.1 Pseudo Code for Portfolio Optimization

As discussed earlier, the portfolio must be adjusted to reflect changes occurring in the dynamic financial markets. So, once an initial portfolio is constructed, *Portfolio Optimization* continuously rebalances this portfolio by buying new stocks in a portfolio and/or selling existing stocks of the portfolio. Figure 6.4 provides the pseudo code for portfolio optimization.

```
Construct initial portfolio
While (Portfolio exists)
{
   At specific intervals,
      Update dataset
      Evaluate Portfolio
      Rebalance portfolio
        Construct a new portfolio
        Compare the new portfolio with an existing portfolio
        Update stocks and/or weights as per differences
}
```

Figure 6.4 Pseudo-code for Portfolio Optimization

Portfolio management is a long-lasting process spanning over months or years. Once an initial portfolio is constructed, it becomes necessary to track the performance of this portfolio and optimize it. The entire process begins with the construction of the portfolio to make initial investments. The process remains continuous iteratively for the given dynamic iterations, or in the real world until investor decides to stop investment.

Existing portfolio is updated for the purpose of optimization at specific time intervals. The selection of the time interval is crucial. It can be either predetermined fixed intervals or intervals based on dynamic events occurring in the financial world. If predetermined fixed intervals are used to update the portfolio, the time duration of the interval becomes essential. If it is too short, it will involve frequent changes in a portfolio and will increase the number of transactions. This will result in higher transaction costs and other taxation and will reduce effective returns. If the time interval is too long, the portfolio will get late to reflect changes in the underlying stocks affecting the performance of the portfolio negatively. In this discussion, a fixed interval of the three months, or a single financial quarter, has been used as a time interval to update portfolio.

Dataset is updated by collecting the latest stock data, and, the existing portfolio is evaluated. A new portfolio is constructed by using the WA-PC algorithm and compared with the existing portfolio to obtain the differences either in stocks or in weights. Based on these differences, stocks and their weights are updated to rebalance the existing portfolio. Rebalancing is achieved by selling stocks of the existing portfolio and buying new stocks as per the requirement of the new portfolio. Once a portfolio is rebalanced, its performance will be tracked for the next three months until there is a need to rebalance it again.

6.3.2 Experimental Results

Similar to the previous section, three different kinds of portfolios – conservative, moderate and aggressive – have been used in this section. Each of these portfolios is constructed on the date of 1st April 2011 which is the first day of the FY 2011-12. The historical stock data of the past one year have been used to determine the expected return and risk of a portfolio. The portfolio risk is considered in terms of standard deviation rather than variance. The initial amount to be invested is considered Rs. 100000. These portfolios are evaluated and rebalanced at a fixed interval of three months – at the end of each quarter of the financial year.

For example, the initial portfolio is constructed on 01-APR-2011. For this purpose, historical stock data for the past one year beginning from 01-APR-2010 to 31-MAR-2011 are used to determine expected return and risk of this portfolio. This portfolio is evaluated at the end of the first quarter of FY 2011-12 (on the date of 30-JUN-2011).

The new portfolio is constructed on 01-JUL-2011 based on the historical stock data of past one year beginning from 01-JUL-2010 to 30-JUN-2011. This newly constructed portfolio is compared with the existing portfolio. Based on the differences found in these two portfolios, the existing portfolio is rebalanced by selling existing stocks and buying new stocks. Note that the whole portfolio value is reinvested again in the rebalanced portfolio each time. This means that the last value of the previous portfolio and first value of the rebalanced portfolio will remain the same. This process will be repeated at the end of each quarter up to the fourth quarter of the FY 2018-19 ending at 31-MAR-2019.

Performance of the portfolios has been tracked against the performance of NIFTY50 as well as that of static portfolios. The performance is tracked for the time duration beginning from 1st April 2011 to 31st March 2019 – up to the end of the FY 2018-19. This test duration involves a total of 32 financial quarters and requires the portfolio to be rebalanced 31 times once it is constructed initially.

Similar to static portfolios, transaction costs and dividend returns have been ignored in this discussion too for the sake of simplicity. Also, stock units are allowed to be real values to get the exact difference in the performance of the portfolio.

- **Conservative Portfolio**

Conservative portfolio (λ = 0.0) attempts to minimize risk irrespective of the return. Table 6.6 presents the quarterly returns and values for the NIFTY50 and conservative portfolios – static as well as dynamic. The static portfolio remains as it is once it is constructed, while the dynamic portfolio is updated at the end of each quarter. Quarterly values of the NIFTY50, static portfolio and dynamic portfolio have been plotted against the time duration in Figure 6.5 (a).

Note that the dynamic portfolio has given 228.76% return in contrast to 101.69% return of static portfolio and 99.21% return of the NIFTY50. The dynamic portfolio has outperformed the overall market in 23 quarters out of the entire 32 test quarters while static portfolio could outperform the market in only 13 quarters. The dynamic portfolio has also outperformed the static portfolio in 24 quarters. These results show the ability of the conservative portfolio to keep risk as minimum as possible while achieving higher returns.

- **Moderate Portfolio**

Moderate portfolio (λ = 0.7) attempts to achieve a balance between risk and return of a portfolio instead of only minimizing risk or only maximizing return. Table 6.7 presents quarterly returns and values for the NIFTY50 and moderate portfolios. The same results are graphically depicted in Figure 6.5 (b). Observe that dynamic portfolio has given 301.23% return outperforming 162.97% return of static portfolio and 99.21% return of NIFTY50. The moderate portfolio also outperforms static portfolio. Quarter-wise, the moderate portfolio has outperformed the overall market 19 times and static portfolio 22 times.

- **Aggressive Portfolio**

An aggressive portfolio (λ = 1.0) attempts to maximize return irrespective of risk values. Table 6.8 presents quarterly returns and values for an aggressive portfolio. The same results are graphically depicted in Figure 6.5 (c). The dynamic portfolio has given 156.06% return compared to 100.27% return of static portfolio and 99.21% return of NIFTY50. An aggressive portfolio has outperformed the market and static portfolio in 18 and 19 quarters respectively. Though dynamic portfolio could outperform a static portfolio and NIFTY50, it performed poorly compared to conservative and moderate portfolios.

Table 6.6 Dynamic Conservative Portfolio v/s NIFTY50

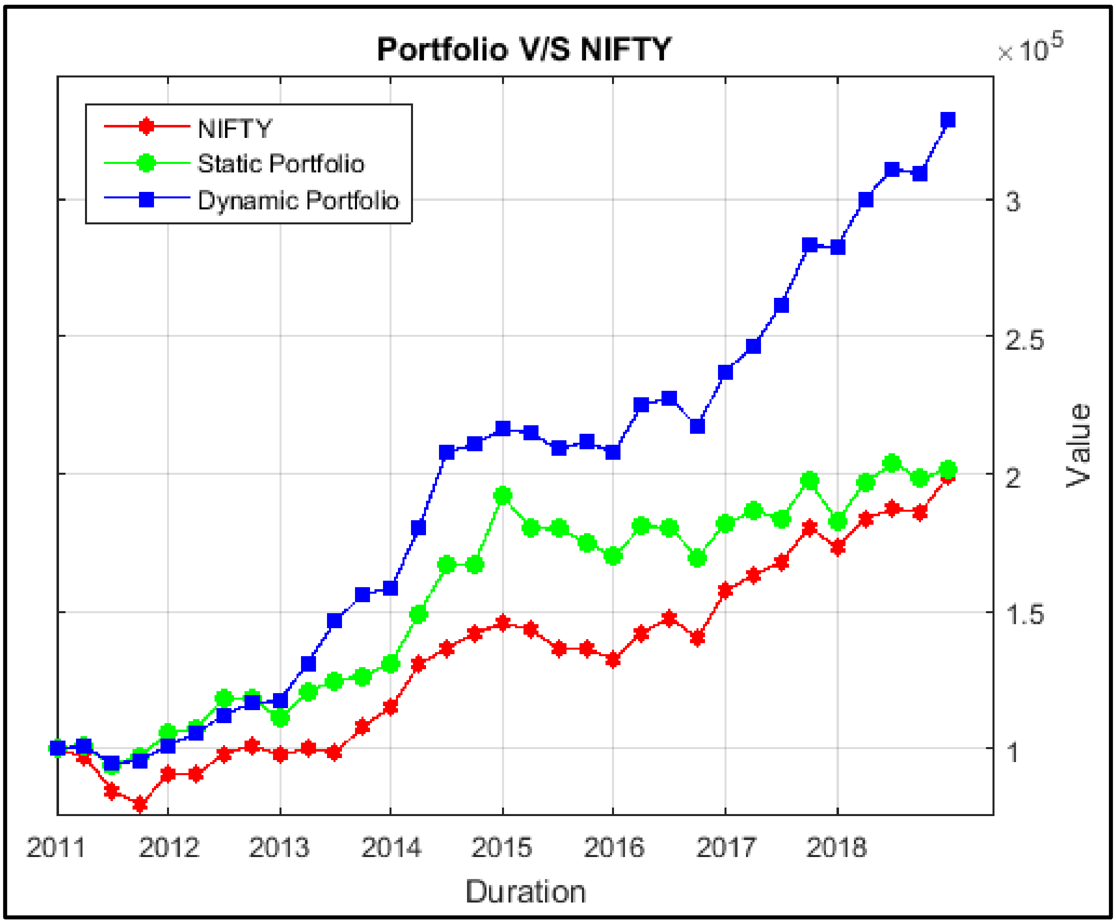
Quarter	NIFTY50		Static Portfolio		Dynamic Portfolio	
	Return (%)	PF Value	Return (%)	Value	Return (%)	Value
30-06-2011	-3.22	96784.92	0.73	100732.37	0.73	100732.37
30-09-2011	-12.47	84717.22	-6.77	93914.17	-6.38	94304.31
31-12-2011	-6.45	79251.07	3.60	97296.03	1.26	95492.77
31-03-2012	14.52	90754.93	8.44	105503.01	5.83	101063.27
30-06-2012	-0.31	90469.58	1.56	107144.01	4.31	105420.42
30-09-2012	8.04	97742.93	10.43	118317.41	6.42	112185.86
31-12-2012	3.54	101201.37	-0.08	118226.25	3.73	116372.53
31-03-2013	-3.77	97387.32	-6.27	110817.54	0.86	117374.05
30-06-2013	2.81	100123.39	8.89	120672.57	11.57	130954.19
30-09-2013	-1.83	98291.35	3.42	124795.19	11.71	146286.13
31-12-2013	9.92	108037.70	1.16	126245.31	6.70	156091.61
31-03-2014	6.35	114896.32	3.72	130942.47	1.49	158415.34
30-06-2014	13.53	130443.02	13.83	149050.94	13.79	180266.66
30-09-2014	4.64	136500.43	11.81	166656.67	15.38	207986.64
31-12-2014	3.99	141948.59	0.36	167260.87	1.31	210709.83
31-03-2015	2.51	145518.42	14.76	191956.64	2.72	216450.86
30-06-2015	-1.44	143419.02	-6.11	180230.54	-0.80	214728.67
30-09-2015	-5.01	136227.93	-0.15	179953.40	-2.59	209162.35
31-12-2015	-0.03	136184.23	-2.86	174799.99	1.15	211570.11
31-03-2016	-2.62	132620.39	-2.67	170124.18	-1.76	207843.17
30-06-2016	7.10	142035.13	6.29	180827.28	8.36	225221.07
30-09-2016	3.90	147577.55	-0.23	180409.92	0.98	227429.80
31-12-2016	-4.94	140287.92	-5.93	169716.49	-4.48	217234.04
31-03-2017	12.07	157219.37	7.20	181943.25	9.26	237345.75
30-06-2017	3.78	163168.81	2.49	186481.83	3.88	246555.10
30-09-2017	2.81	167756.64	-1.44	183803.27	6.10	261600.75
31-12-2017	7.58	180474.72	7.52	197621.09	8.22	283100.70
31-03-2018	-3.96	173328.19	-7.53	182732.22	-0.13	282733.75
30-06-2018	5.94	183621.25	7.69	196783.28	6.16	300139.94
30-09-2018	2.02	187325.62	3.58	203822.00	3.67	311159.47
31-12-2018	-0.62	186161.95	-2.72	198269.99	-0.65	309138.52
31-03-2019	7.01	199209.94	1.72	201685.09	6.35	328758.62
Overall	**99.21**	**199209.94**	**101.69**	**201685.09**	**228.76**	**328758.62**

Table 6.7 Dynamic Moderate Portfolio v/s NIFTY50

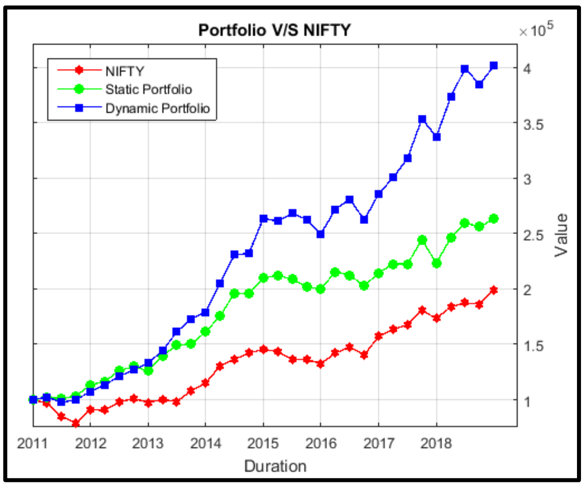
Quarter	NIFTY50		Static Portfolio		Dynamic Portfolio	
	Return (%)	PF Value	Return (%)	Value	Return (%)	Value
30-06-2011	-3.22	96784.92	2.36	102364.12	2.36	102364.12
30-09-2011	-12.47	84717.22	-1.51	100822.15	-4.58	97675.11
31-12-2011	-6.45	79251.07	2.57	103411.99	2.12	99741.72
31-03-2012	14.52	90754.93	9.61	113349.04	7.81	107530.12
30-06-2012	-0.31	90469.58	2.74	116458.51	5.42	113359.84
30-09-2012	8.04	97742.93	8.39	126229.85	6.76	121025.67
31-12-2012	3.54	101201.37	2.90	129896.07	4.98	127056.87
31-03-2013	-3.77	97387.32	-3.02	125967.89	4.73	133070.73
30-06-2013	2.81	100123.39	10.78	139551.32	8.23	144026.46
30-09-2013	-1.83	98291.35	6.91	149198.86	11.95	161230.69
31-12-2013	9.92	108037.70	0.69	150231.87	7.38	173125.09
31-03-2014	6.35	114896.32	7.28	161173.05	3.41	179029.82
30-06-2014	13.53	130443.02	8.87	175472.34	14.58	205130.48
30-09-2014	4.64	136500.43	11.79	196161.14	12.46	230690.52
31-12-2014	3.99	141948.59	-0.33	195504.90	0.57	232012.19
31-03-2015	2.51	145518.42	7.44	210049.95	13.64	263653.49
30-06-2015	-1.44	143419.02	1.06	212268.37	-0.95	261155.21
30-09-2015	-5.01	136227.93	-1.71	208633.34	2.64	268046.17
31-12-2015	-0.03	136184.23	-3.25	201856.38	-1.99	262700.30
31-03-2016	-2.62	132620.39	-0.83	200172.40	-4.93	249740.04
30-06-2016	7.10	142035.13	7.60	215384.21	8.87	271896.70
30-09-2016	3.90	147577.55	-1.37	212437.66	3.34	280979.75
31-12-2016	-4.94	140287.92	-4.65	202552.07	-6.70	262152.28
31-03-2017	12.07	157219.37	5.87	214440.38	8.99	285716.43
30-06-2017	3.78	163168.81	3.77	222522.34	5.10	300290.91
30-09-2017	2.81	167756.64	-0.05	222412.78	5.77	317630.05
31-12-2017	7.58	180474.72	9.83	244266.14	11.30	353515.22
31-03-2018	-3.96	173328.19	-8.58	223318.58	-4.60	337252.75
30-06-2018	5.94	183621.25	10.07	245817.44	10.59	372976.41
30-09-2018	2.02	187325.62	5.54	259437.20	6.95	398904.16
31-12-2018	-0.62	186161.95	-1.29	256078.86	-3.62	384467.65
31-03-2019	7.01	199209.94	2.69	262972.60	4.36	401229.32
Overall	**99.21**	**199209.94**	**162.97**	**262972.60**	**301.23**	**401229.32**

Table 6.8 Dynamic Aggressive Portfolio v/s NIFTY50

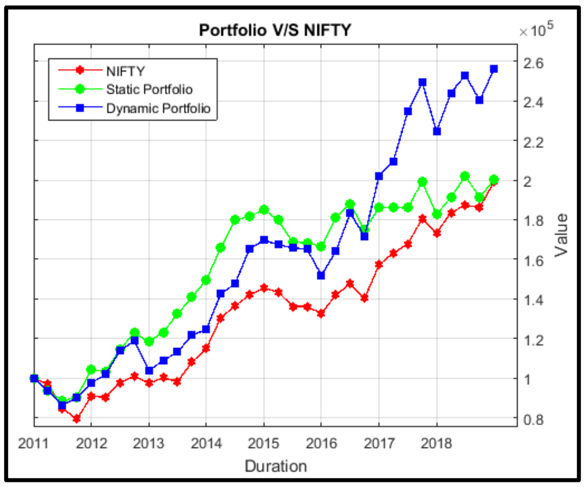
Quarter	NIFTY50		Static Portfolio		Dynamic Portfolio	
	Return (%)	PF Value	Return (%)	Value	Return (%)	Value
30-06-2011	-3.22	96784.92	-6.44	93562.74	-6.44	93562.74
30-09-2011	-12.47	84717.22	-5.41	88500.16	-7.62	86428.98
31-12-2011	-6.45	79251.07	2.28	90522.01	4.20	90062.73
31-03-2012	14.52	90754.93	15.08	104172.13	8.26	97502.29
30-06-2012	-0.31	90469.58	-0.65	103499.31	4.49	101876.75
30-09-2012	8.04	97742.93	10.71	114586.78	11.79	113892.64
31-12-2012	3.54	101201.37	7.18	122812.89	4.32	118812.53
31-03-2013	-3.77	97387.32	-3.58	118417.66	-12.64	103797.03
30-06-2013	2.81	100123.39	4.01	123167.36	4.93	108916.88
30-09-2013	-1.83	98291.35	7.45	132346.45	3.85	113109.02
31-12-2013	9.92	108037.70	6.44	140869.89	7.61	121713.69
31-03-2014	6.35	114896.32	6.17	149554.52	2.31	124529.98
30-06-2014	13.53	130443.02	10.87	165807.50	14.43	142493.76
30-09-2014	4.64	136500.43	8.54	179963.94	3.62	147654.69
31-12-2014	3.99	141948.59	0.88	181554.96	11.96	165319.78
31-03-2015	2.51	145518.42	1.96	185111.25	2.58	169581.86
30-06-2015	-1.44	143419.02	-2.90	179734.62	-1.32	167345.41
30-09-2015	-5.01	136227.93	-6.10	168772.38	-0.93	165793.22
31-12-2015	-0.03	136184.23	-0.29	168282.19	-0.40	165124.65
31-03-2016	-2.62	132620.39	-1.02	166573.76	-8.00	151908.13
30-06-2016	7.10	142035.13	8.65	180979.50	7.93	163961.39
30-09-2016	3.90	147577.55	3.88	188000.69	11.77	183257.64
31-12-2016	-4.94	140287.92	-7.09	174678.84	-6.38	171563.28
31-03-2017	12.07	157219.37	6.58	186168.74	17.73	201988.92
30-06-2017	3.78	163168.81	-0.03	186119.67	3.75	209560.91
30-09-2017	2.81	167756.64	-0.11	185918.43	11.93	234566.83
31-12-2017	7.58	180474.72	7.14	199197.98	6.27	249278.61
31-03-2018	-3.96	173328.19	-8.33	182612.20	-9.92	224544.99
30-06-2018	5.94	183621.25	4.79	191364.50	8.67	244022.64
30-09-2018	2.02	187325.62	5.55	201990.92	3.66	252960.52
31-12-2018	-0.62	186161.95	-5.34	191212.60	-5.04	240220.27
31-03-2019	7.01	199209.94	4.74	200274.68	6.60	256062.92
Overall	**99.21**	**199209.94**	**100.27**	**200274.68**	**156.06**	**256062.92**



(a) Conservative Portfolio



(b) Moderate Portfolio



(c) Aggressive Portfolio

Figure 6.5 Dynamic Portfolios v/s NIFTY50

6.4 Results and Discussion

This section discusses the obtained experimental results and comparative analysis of the static and dynamic portfolios. The winnowing algorithm is applied to the real-world stock market to assess the performance of the generated portfolios over a period of time. For this purpose, the dataset has been prepared based on the data collected from the NSE of India. Performance of the portfolio has been evaluated for static and dynamic portfolios. For each of these two approaches, three different kinds of portfolios have been considered in this chapter – conservative, moderate and aggressive.

For each of these portfolios, quarter-wise returns have been presented in the previous section. Table 6.9 summarizes annual returns and gives a broader idea about the performance of each of these portfolios.

Table 6.9 Annual Returns of NIFTY50, Static Portfolios and Dynamic Portfolios

FY	NIFTY50	Static Portfolio			Dynamic Portfolio		
		Conservative	Moderate	Aggressive	Conservative	Moderate	Aggressive
2011-12	-9.25	**5.50**	**13.35**	**4.17**	**1.06**	**7.53**	**-2.50**
2012-13	7.31	**5.04**	**11.13**	**13.67**	**16.14**	**23.75**	6.46
2013-14	17.98	**18.16**	**27.95**	**26.29**	**34.97**	**34.54**	**19.97**
2014-15	26.65	**46.60**	**30.33**	23.78	**36.64**	**47.27**	**36.18**
2015-16	-8.86	-11.37	**-4.70**	-10.01	**-3.98**	**-5.28**	-10.42
2016-17	18.55	6.95	7.13	11.76	14.19	14.41	**32.97**
2017-18	10.25	0.43	4.14	-1.91	**19.12**	**18.04**	**11.17**
2018-19	14.93	10.37	**17.76**	9.67	**16.28**	**18.97**	14.04
Overall	99.21	**101.69**	**162.97**	**100.27**	**228.76**	**301.23**	**156.06**

In a broader view, portfolios generated by the winnowing algorithm, either static or dynamic, yield higher return and outperform the overall market. Bold values in this table indicate that that value has outperformed the corresponding NIFTY50 value.

Static portfolios have outperformed the market in initial years. But their performances falter with progress in financial years. Stocks selected in the year 2011 to construct these portfolios lose relevance over a longer time duration, and their performance deteriorates. Among static portfolios, the moderate portfolio provides a right blend of risk and return, and, outperforms the conservative and aggressive portfolios along with the overall market.

Compared to static portfolios, dynamic portfolios have remained successful in outperforming the overall market throughout all the years. Among dynamic portfolios, the conservative portfolio has given return more than twice the return of NIFTY50 and moderate portfolio has given return more than thrice the return of NIFTY50. The conservative portfolio has remained successful in maximizing returns when the market is positive and in minimizing risk when the market is negative. Even when it could not provide better returns compared to the market, it has provided returns that are very near to that of the market. The moderate portfolio has perfectly balanced between the risk and return. When the market is negative, it could not minimize the risk as that of the conservative portfolio, but it has remained successful in outperforming the market. To compensate this, it has provided much higher returns when the market is positive.

Aggressive portfolios have outperformed the NIFTY50, but they cannot outperform conservative and moderate counterparts. There are mainly two reasons behind this: First, this discussion considers the mean value of historical returns as an expected return of the stocks. It has been expected here that if a stock has performed well in the past, it will continue to do so. But, in reality, this assumption does not hold the ground always. Second, aggressive portfolios ignore the risk entirely and only attempt to maximize the return. In quest of this, it performs poorly when overall market falls, yielding lower returns compared to other two portfolios.

And at the end, study and comparative analysis of these results further prove the efficiency and reliability of the winnowing algorithm to manage the stock market portfolio successfully.

CHAPTER 7

Conclusion and Future Work

7.1 Conclusion

In this research work, the novel natural computing algorithm, Winnowing Algorithm whose design is inspired by the real-world winnowing process, has been proposed and implemented to manage the stock market portfolio. The cardinality constrained mean-variance model, which is an extension to the classical mean-variance model, has been used as the mathematical model to represent the stock market portfolio.

This research work has been divided into three different phases. The first phase has applied the winnowing algorithm (WA-PWO) to optimize the weight of stocks to be included in a portfolio. The performance of WA-PWO has been tested using a small dataset given in the literature. The second phase applies winnowing algorithm (WA-PC) to construct a portfolio by selecting stocks and their weights from a pool of available stocks. The performance of WA-PC has been assessed using five different datasets given in OR-Library, representing the stock markets of five different countries. Obtained experimental results prove the accuracy and consistency of this algorithm to optimize weights of stocks as well as to construct optimal portfolios. The winnowing algorithm has also been found reliable to generate an efficient frontier of the optimal portfolios with different risk-return profiles. Performance of the winnowing algorithm has been compared and analyzed with other state-of-the-art algorithms. The comparative analysis establishes the superiority of this algorithm in the context of portfolio management.

The third phase focuses on the performance of the portfolio rather than the performance of the algorithm itself. Here, winnowing algorithm has been applied to optimize constructed portfolio over a period of time by updating stocks and their weights included in a portfolio.

The real-world stock market data collected from the NSE of India have been used in this phase. The performance of the portfolio is compared to the performance of the benchmark index NIFTY50. This performance has been evaluated for two different approaches – static and dynamic. For each of these two approaches, three different kinds of portfolios – conservative, moderate and aggressive – have been considered and performance has been tracked for 32 fiscal quarters beginning from April-2011 to March-2019. The NIFTY50 gave 99.21% returns during this time span. The conservative, moderate and aggressive portfolios gave 101.69%, 162.97%, and 100.27% returns respectively with the static approach. These portfolios gave 228.76%, 301.23%, and 156.06% returns respectively with a dynamic approach. They have successfully outperformed the overall market. Performance of the static portfolios found to be deteriorated with progress in financial years. Dynamic portfolios found to be able to get adjusted to dynamic changes in the stock market and continue to outperform the overall market.

Obtained results and comparative analysis concludes that natural computing algorithm can be applied to manage the stock market portfolio, so that the return is maximized for a given risk, or the risk is minimized for a given return. Also, appropriately selected stocks and their weights can outperform the overall stock market.

7.2 Future Scope

This research work provides two different fronts where there are further scopes to work as a researcher. The first one is winnowing algorithm and the second one is the portfolio management model.

- **The Winnowing Algorithm**

  - The winnowing algorithm extends the simplicity of the winnowing process and requires very few parameters to be tuned in. Yet it has been found useful in obtaining optimal results within a reasonable time duration. This algorithm uses derivation-free mechanism and can avoid local optima successfully. These attributes of the winnowing algorithm show that it possesses the potential to be applicable in other fields also. So, the applicability of this algorithm can be explored in areas where optimization is the necessary component of the application.

- The winnowing algorithm proposed in this research work applies a straightforward approach as a local search such as either exchanging weight of two stocks or replacing portfolio member stock with other stock. Instead of using such a simple approach, other enhanced techniques can be applied as a local search to improve the performance of the algorithm.

- It has been found that selection of Δw_{max} (maximum fraction of weight used to update portfolio) plays a crucial role in determining the convergence time of the algorithm and quality of the obtained optimal solution. The WA-PC used regressive values for Δw_{max} to provide high exploration in initial iterations and fine (small) exploration in late iterations of the algorithm. Other innovative methods can be explored to vary the values of Δw_{max} dynamically to further reduce the convergence time of the algorithm without affecting the quality of the obtained optimal solution.

- **The Portfolio Management Model**

  - The work presented here considers the mean value of the historical returns as an expected return of the stock. This assumption has remained the core factor behind the relatively poor performance of the aggressive portfolios. The performance of the portfolio can be improved by incorporating the prediction component in determining the expected return of a stock. Even fundamental and/or technical analysis can be explored to determine the pool of available stocks to construct a portfolio.

  - This work used variance and standard deviation as a portfolio risk measure. Study of literature has found many other risk measures which can be explored to check their effects on the performance of portfolios.

  - Static values have been used in this work for cardinality constraint, floor-ceiling constraint, and time duration for historical stock data as well as portfolio rebalancing. A flexible way can be attempted with dynamic values for these parameters to improve the performance of the portfolio further.

  - Also, other portfolio management models can be explored and can be tested rather than the cardinality constrained mean-variance model.

References

[1] “What impact does inflation have on the time value of money?” [Online]. Available: https://www.investopedia.com/ask/answers/042415/what-impact-does-inflation-have-time-value-money.asp. [Accessed: 16-Feb-2019].

[2] “Inflation erodes purchasing power - Military.” [Online]. Available: https://handsonbanking.org/military/building-wealth/investing-basics/inflation-erodes-purchasing-power/. [Accessed: 16-Feb-2019].

[3] A. E. Yilmaz and G.-W. Weber, “Why You Should Consider Nature-Inspired Optimization Methods in Financial Mathematics,” in *Nonlinear and Complex Dynamics*, Springer New York, 2011, pp. 241–255.

[4] F. J. Fabozzi and H. M. Markowitz, *The Theory and Practice of Investment Management: Asset Allocation, Valuation, Portfolio Construction, and Strategies*. John Wiley & Sons, 2011.

[5] J. Dang, A. Brabazon, D. Edelman, and M. O’Neill, “An Introduction to Natural Computing in Finance,” in *Applications of Evolutionary Computing*, M. Giacobini, A. Brabazon, S. Cagnoni, G. A. D. Caro, A. Ekárt, A. I. Esparcia-Alcázar, M. Farooq, A. Fink, and P. Machado, Eds. Springer Berlin Heidelberg, 2009, pp. 182–192.

[6] X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.

[7] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.

[8] “NSE - National Stock Exchange of India Ltd.” [Online]. Available: https://www.nseindia.com/products/content/equities/equities/archieve_eq.htm. [Accessed: 15-Mar-2019].

[9] R. Jeurissen and J. van den Berg, “Optimized index tracking using a hybrid genetic algorithm,” in *IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*, 2008, pp. 2327–2334.

[10] N. K. Lioudis, “Passive vs Active Portfolio Management,” *Investopedia*. [Online]. Available: https://www.investopedia.com/ask/answers/040315/what-difference-between-passive-and-active-portfolio-management.asp. [Accessed: 17-Feb-2019].

[11] “Active management,” *Wikipedia*. 22-Aug-2018.

[12] P. Grant, “Developing risk management strategies for stock market investment portfolio management,” Thesis, 2007.

[13] R. W. Colby, *The Encyclopedia Of Technical Market Indicators, Second Edition*, 2 edition. New York: McGraw-Hill Professional, 2004.

[14] A. Gujral, *How to Make Money Trading with Charts*. New Delhi, India: Vision Books, 2008.

[15] “Passive management,” *Wikipedia*. 18-Dec-2018.

[16] H. Markowitz, “Portfolio Selection*,” *J. Finance*, vol. 7, no. 1, pp. 77–91, Mar. 1952.

[17] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investments*. New Haven, Conn.: Yale University Press, 1971.

[18] “Systematic Risk - Learn How to Identify and Calculate Systematic Risk,” *Corporate Finance Institute*. [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/finance/systematic-risk/. [Accessed: 17-Feb-2019].

[19] “Difference Between Systematic and Unsystematic Risk (with Comparison Chart),” *Key Differences*, 11-Jan-2016. .

[20] "Idiosyncratic Risk - Overview & Example of Idiosyncratic Risk," *Corporate Finance Institute*. [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/other/idiosyncratic-risk/. [Accessed: 17-Feb-2019].

[21] T.-J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha, "Heuristics for cardinality constrained portfolio optimisation," *Comput. Oper. Res.*, vol. 27, no. 13, pp. 1271–1302, 2000.

[22] S. Wang and Y. Xia, "A Model for Portfolio Selection with Order of Expected Returns," in *Portfolio Selection and Asset Pricing*, Springer Berlin Heidelberg, 2002, pp. 23–38.

[23] D. Maringer and H. Kellerer, "Optimization of cardinality constrained portfolios with a hybrid local search algorithm," *Spectr.*, vol. 25, no. 4, pp. 481–495, 2003.

[24] T. Cura, "Particle swarm optimization approach to portfolio optimization," *Nonlinear Anal. Real World Appl.*, vol. 10, no. 4, pp. 2396–2406, Aug. 2009.

[25] A. L. J. Antonin Ponsich, "A Survey on Multiobjective Evolutionary Algorithms for the Solution of the Portfolio Optimization Problem and Other Finance and Economics Applications," *Evol. Comput. IEEE Trans. On*, vol. 17, no. 3, pp. 321–344, 2013.

[26] T. Fu, C. Ng, K. Wong, and F.-L. Chung, "Models for portfolio management on enhancing periodic consideration and portfolio selection," in *2011 Seventh International Conference on Natural Computation (ICNC)*, 2011, vol. 1, pp. 176–180.

[27] G. Di Tollo and A. Roli, "Metaheuristics for the portfolio selection problem," *Int. J. Oper. Res.*, vol. 5, no. 1, pp. 13–35, 2008.

[28] "Efficient-market hypothesis," *Wikipedia*. 20-Feb-2019.

[29] F. Streichert, H. Ulmer, and A. Zell, "Comparing Discrete and Continuous Genotypes on the Constrained Portfolio Selection Problem," in *Genetic and Evolutionary Computation – GECCO 2004*, K. Deb, Ed. Springer Berlin Heidelberg, 2004, pp. 1239–1250.

[30] T. Krink, S. Paterlini, and others, "Differential evolution for multiobjective portfolio optimization," *Cent. Econ. Res. RECent*, vol. 21, 2008.

[31] T. M. G. A. Vijayalakshmi Pai, "Evolutionary Optimization of Constrained -Means Clustered Assets for Diversification in Small Portfolios," *Evol. Comput. IEEE Trans. On*, vol. 13, no. 5, pp. 1030–1053, 2009.

[32] F. Black, "Capital Market Equilibrium with Restricted Borrowing," *J. Bus.*, vol. 45, no. 3, pp. 444–55, 1972.

[33] R. Mansini and M. G. Speranza, "Heuristic algorithms for the portfolio selection problem with minimum transaction lots," *Eur. J. Oper. Res.*, vol. 114, no. 2, pp. 219–233, 1999.

[34] M. A. Gomez, C. X. Flores, and M. A. Osorio, "Hybrid Search for Cardinality Constrained Portfolio Optimization," in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, New York, NY, USA, 2006, pp. 1865–1866.

[35] W. R. It, "Portfolio Selection: efficient diversification of investments," 1959.

[36] P. Jorion, *Value at Risk: The New Benchmark for Managing Financial Risk, 3rd Edition*, 3rd edition. New York: McGraw-Hill, 2006.

[37] N.-A. Bradshaw, C. Walshaw, C. Ierotheou, A. K. Parrott, and others, "A multi-objective evolutionary algorithm for portfolio optimisation," *Proc. AISB09*, 2009.

[38] "Value at Risk," *Varsity by Zerodha*. .

[39] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent Measures of Risk," *Math. Finance*, vol. 9, no. 3, pp. 203–228, Jul. 1999.

[40] R. T. Rockafellar and S. Uryasev, "Optimization of Conditional Value-at-Risk," *J. Risk*, vol. 2, pp. 21–41, 2000.

[41] R. T. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *J. Bank. Finance*, pp. 1443–1471, 2002.

[42] Sergey Sarykalin, Gaia Serraino, and Stan Uryasev, “Value-at-Risk vs. Conditional Value-at-Risk in Risk Management and Optimization,” in *State-of-the-Art Decision-Making Tools in the Information-Intensive Age*, 0 vols., INFORMS, 2008, pp. 270–294.

[43] W. F. Sharpe, “Mutual Fund Performance,” *J. Bus.*, vol. 39, no. 1, pp. 119–138, Jan. 1966.

[44] W. F. Sharpe, “The sharpe ratio,” *J. Portf. Manag.*, vol. 21, no. 1, pp. 49–58, 1994.

[45] H. Y. Hiroshi Konno, “Mean-Absolute Deviation Portfolio Optimization Model and Its Applications to Tokyo Stock Market,” *Manag. Sci.*, vol. 37, no. 5, pp. 519–531, 1991.

[46] L. Chiodi, R. Mansini, and M. G. Speranza, “Semi-Absolute Deviation Rule for Mutual Funds Portfolio Selection,” *Ann. Oper. Res.*, vol. 124, no. 1–4, pp. 245–265, Nov. 2003.

[47] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *Evol. Comput. IEEE Trans. On*, vol. 1, no. 1, pp. 67–82, 1997.

[48] J. H. Holland, “Genetic algorithms and the optimal allocation of trials,” *SIAM J. Comput.*, vol. 2, no. 2, pp. 88–105, 1973.

[49] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.

[50] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA, USA: MIT Press, 1992.

[51] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *SCIENCE*, vol. 220, no. 4598, pp. 671–680, 1983.

[52] D. Bookstaber, *Simulated Annealing for Traveling Salesman Problem*. Spring, 1997.

[53] P. Moscato, *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts - Towards Memetic Algorithms*. 1989.

[54] A. Colorni, M. Dorigo, V. Maniezzo, and others, “Distributed optimization by ant colonies,” in *Proceedings of the first European conference on artificial life*, 1991, vol. 142, pp. 134–142.

[55] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[56] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *, IEEE International Conference on Neural Networks, 1995. Proceedings*, 1995, vol. 4, pp. 1942–1948 vol.4.

[57] R. Storn and K. Price, “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces,” *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[58] N. E. Nawa and T. Furuhashi, “Fuzzy system parameters discovery by bacterial evolutionary algorithm,” *Fuzzy Syst. IEEE Trans. On*, vol. 7, no. 5, pp. 608–616, 1999.

[59] D. DasGupta, *Artificial immune systems and their applications*. Springer Publishing Company, Incorporated, 2014.

[60] D. Dasgupta, Z. Ji, F. A. González, and others, “Artificial immune system (AIS) research in the last five years.,” in *IEEE Congress on Evolutionary Computation (1)*, 2003, pp. 123–130.

[61] H.-G. Beyer and H.-P. Schwefel, “Evolution strategies–A comprehensive introduction,” *Nat. Comput.*, vol. 1, no. 1, pp. 3–52, 2002.

[62] K. M. Passino, “Biomimicry of bacterial foraging for distributed optimization and control,” *IEEE Control Syst.*, vol. 22, no. 3, pp. 52–67, Jun. 2002.

[63] X. Li and J. Qian, “Studies on Artificial Fish Swarm Optimization Algorithm based on Decomposition and Coordination Techniques [J],” *J. Circuits Syst.*, vol. 1, pp. 1–6, 2003.

[64] M. M. Eusuff and K. E. Lansey, “Optimization of water distribution network design using the shuffled frog leaping algorithm,” *J. Water Resour. Plan. Manag.*, vol. 129, no. 3, pp. 210–225, 2003.

[65] M. Eusuff, K. Lansey, and F. Pasha, “Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization,” *Eng. Optim.*, vol. 38, no. 2, pp. 129–154, 2006.

[66] X.-F. Xie and W.-J. Zhang, “Solving engineering design problems by social cognitive optimization,” in *Genetic and Evolutionary Computation–GECCO 2004*, 2004, pp. 261–262.

[67] A. R. Mehrabian and C. Lucas, “A novel numerical optimization algorithm inspired from weed colonization,” *Ecol. Inform.*, vol. 1, no. 4, pp. 355–366, 2006.

[68] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.

[69] D. Karaboga and B. Basturk, “A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm,” *J Glob. Optim.*, vol. 39, no. 3, pp. 459–471, Nov. 2007.

[70] S. He, Q. H. Wu, and J. R. Saunders, “A novel group search optimizer inspired by animal behavioural ecology,” in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, 2006, pp. 1272–1278.

[71] R. A. Formato, “Central force optimization: a new metaheuristic with applications in applied electromagnetics,” *Prog. Electromagn. Res.*, vol. 77, pp. 425–491, 2007.

[72] R. A. Formato, “Central force optimization: a new nature inspired computational framework for multidimensional search and optimization,” in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*, Springer, 2008, pp. 221–238.

[73] P. Rabanal, I. Rodríguez, and F. Rubio, “Using river formation dynamics to design heuristic algorithms,” in *Unconventional Computation*, Springer, 2007, pp. 163–177.

[74] P. Rabanal, I. Rodríguez, and F. Rubio, “Applying river formation dynamics to solve NP-complete problems,” in *Nature-inspired algorithms for optimisation*, Springer, 2009, pp. 333–368.

[75] H. Shah-Hosseini, “Problem solving by intelligent water drops,” in *IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, 2007, pp. 3226–3231.

[76] T. C. Havens, C. J. Spain, N. G. Salmon, and J. M. Keller, “Roach infestation optimization,” in *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*, 2008, pp. 1–7.

[77] R. Zhao and W. Tang, “Monkey algorithm for global numerical optimization,” *J. Uncertain Syst.*, vol. 2, no. 3, pp. 165–176, 2008.

[78] D. Simon, “Biogeography-Based Optimization,” *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008.

[79] A. H. Kashan, “League championship algorithm: a new algorithm for numerical function optimization,” in *2009 International Conference of Soft Computing and Pattern Recognition*, 2009, pp. 43–48.

[80] K. N. Krishnanand and D. Ghose, “Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions,” *Swarm Intell.*, vol. 3, no. 2, pp. 87–124, 2009.

[81] Y. Marinakis, M. Marinaki, and N. Matsatsinis, “A hybrid bumble bees mating optimization-grasp algorithm for clustering,” in *Hybrid Artificial Intelligence Systems*, Springer, 2009, pp. 549–556.

[82] R. Oftadeh and M. J. Mahjoob, “A new meta-heuristic optimization algorithm: Hunting Search,” in *Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, 2009. ICSCCW 2009. Fifth International Conference on*, 2009, pp. 1–5.

[83] X.-S. Yang, “Firefly algorithms for multimodal optimization,” in *Stochastic algorithms: foundations and applications*, Springer, 2009, pp. 169–178.

[84] X.-S. Yang, “Harmony Search as a Metaheuristic Algorithm,” in *Music-Inspired Harmony Search Algorithm*, Z. W. Geem, Ed. Springer Berlin Heidelberg, 2009, pp. 1–14.

[85] U. Premaratne, J. Samarabandu, and T. Sidhu, “A new biologically inspired optimization algorithm,” in *2009 International Conference on Industrial and Information Systems (ICIIS)*, 2009, pp. 279–284.

[86] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, “GSA: a gravitational search algorithm,” *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.

[87] X.-S. Yang and S. Deb, “Cuckoo search via Lévy flights,” in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, 2009, pp. 210–214.

[88] X.-S. Yang and S. Deb, “Engineering optimisation by cuckoo search,” *Int. J. Math. Model. Numer. Optim.*, vol. 1, no. 4, pp. 330–343, 2010.

[89] X.-S. Yang, “A New Metaheuristic Bat-Inspired Algorithm,” in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds. Springer Berlin Heidelberg, 2010, pp. 65–74.

[90] Y. Tan and Y. Zhu, “Fireworks Algorithm for Optimization,” in *Advances in Swarm Intelligence*, Y. Tan, Y. Shi, and K. C. Tan, Eds. Springer Berlin Heidelberg, 2010, pp. 355–364.

[91] A. Salhi and E. S. Fraga, “Nature-inspired optimisation approaches and the new plant propagation algorithm,” 2011.

[92] E. Cuevas, M. González, D. Zaldivar, M. Pérez-Cisneros, and G. García, “An algorithm for global optimization inspired by collective animal behavior,” *Discrete Dyn. Nat. Soc.*, vol. 2012, 2012.

[93] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, “Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems,” *Comput. Struct.*, vol. 110, pp. 151–166, 2012.

[94] A. H. Gandomi and A. H. Alavi, “Krill herd: A new bio-inspired optimization algorithm,” *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, no. 12, pp. 4831–4845, Dec. 2012.

[95] B. Niu and H. Wang, “Bacterial colony optimization,” *Discrete Dyn. Nat. Soc.*, vol. 2012, 2012.

[96] B. R. Rajakumar, “The Lion’s Algorithm: A New Nature-Inspired Search Algorithm,” *Procedia Technol.*, vol. 6, pp. 126–135, 2012.

[97] M. Taherdangkoo, M. Yazdi, and M. H. Bagheri, “Stem Cells Optimization Algorithm,” in *Bio-Inspired Computing and Applications*, D.-S. Huang, Y. Gan, P. Premaratne, and K. Han, Eds. Springer Berlin Heidelberg, 2012, pp. 394–403.

[98] M. T. M. H. Shirzadi and M. H. Bagheri, “A novel meta-heuristic algorithm for numerical function optimization: blind, naked mole-rats (BNMR) algorithm,” *Sci. Res. Essays*, vol. 7, no. 41, pp. 3566–3583, 2012.

[99] X.-S. Yang, “Flower pollination algorithm for global optimization,” in *Unconventional computation and natural computation*, Springer, 2012, pp. 240–249.

[100] X.-S. Yang, M. Karamanoglu, and X. He, “Flower pollination algorithm: a novel approach for multiobjective optimization,” *Eng. Optim.*, vol. 46, no. 9, pp. 1222–1237, 2014.

[101] T. Gerstner, M. Holtz, and R. Korn, “Valuation of performance-dependent options in a Black Scholes framework.”

[102] A. S. Eesa, A. M. Abdulazeez, and Z. Orman, “Cuttlefish algorithm–a novel bio-inspired optimization algorithm,” *Int. J. Sci. Eng. Res.*, vol. 4, no. 9, pp. 1978–1986, 2013.

[103] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, “Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems,” *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2592–2612, May 2013.

[104] E. Cuevas, M. Cienfuegos, D. Zaldívar, and M. Pérez-Cisneros, “A swarm optimization algorithm inspired in the behavior of the social-spider,” *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6374–6384, 2013.

[105] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, “Spider monkey optimization algorithm for numerical optimization,” *Memetic Comput.*, vol. 6, no. 1, pp. 31–47, 2014.

[106] X. Li, J. Zhang, and M. Yin, “Animal migration optimization: an optimization algorithm inspired by animal migration behavior,” *Neural Comput. Appl.*, vol. 24, no. 7–8, pp. 1867–1877, 2014.

[107] A. Askarzadeh, “Bird mating optimizer: An optimization algorithm inspired by bird mating strategies,” *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 4, pp. 1213–1228, Apr. 2014.

[108] M. Ghaemi and M.-R. Feizi-Derakhshi, “Forest optimization algorithm,” *Expert Syst. Appl.*, vol. 41, no. 15, pp. 6676–6687, 2014.

[109] B. Doğan and T. Ölmez, “A new metaheuristic for numerical function optimization: Vortex Search algorithm,” *Inf. Sci.*, vol. 293, pp. 125–145, 2015.

[110] Y.-J. Zheng, “Water wave optimization: A new nature-inspired metaheuristic,” *Comput. Oper. Res.*, vol. 55, pp. 1–11, Mar. 2015.

[111] S. D. Gai-Ge Wang, "Elephant Herding Optimization," 2015.

[112] A. Brabazon, W. Cui, and M. O'Neill, "The raven roosting optimisation algorithm," *Soft Comput.*, vol. 20, no. 2, pp. 525–545, Jan. 2015.

[113] S. Sefiane and M. Benbouziane, "Portfolio selection using genetic algorithm," *J. Appl. Finance Bank.*, vol. 2, no. 4, p. 143, 2012.

[114] M. Tuba, N. Bacanin, and B. Pelevic, "Framework for constrained portfolio selection by the firefly algorithm," *Int. J. Math. Models Methods Appl. Sci.*, vol. 7, no. 10, pp. 888–896, 2013.

[115] X.-S. Yang and X. He, "Firefly algorithm: recent advances and applications," *Int. J. Swarm Intell.*, vol. 1, no. 1, pp. 36–50, 2013.

[116] N. Bacanin, B. Pelevic, and M. Tuba, "Artificial bee colony (ABC) algorithm for portfolio optimization problem."

[117] M. Tuba, N. Bacanin, and B. Pelevic, "Krill herd (KH) algorithm applied to the constrained portfolio selection problem," *Int. J. Math. Comput. Simul.*, vol. 8, pp. 94–102, 2014.

[118] N. Bacanin and M. Tuba, "Fireworks algorithm applied to constrained portfolio optimization problem," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, 2015, pp. 1242–1249.

[119] I. Strumberger, N. Bacanin, and M. Tuba, "Constrained Portfolio Optimization by Hybridized Bat Algorithm," in *Constrained Portfolio Optimization by Hybridized Bat Algorithm*, Bangkok (Thailand), 2016.

[120] M. Mozafari, F. Jolai, and S. Tafazzoli, "A new IPSO-SA approach for cardinality constrained portfolio optimization," *Int. J. Ind. Eng. Comput.*, vol. 2, no. 2, pp. 249–262, 2011.

[121] "Portfolio optimisation." [Online]. Available: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html. [Accessed: 04-Mar-2019].

[122] N. Bacanin and M. Tuba, "Firefly algorithm for cardinality constrained mean-variance portfolio optimization problem with entropy diversity constraint," *Sci. World J.*, vol. 2014, 2014.

[123] M. Salahi, M. Daemi, S. Lotfi, and A. Jamalian, "PSO and harmony search algorithms for cardinality constrained portfolio optimization problem," *AMO–Advanced Model. Optim.*, vol. 16, no. 3, pp. 559–573, 2014.

[124] J. Xu, A. Y. Lam, and V. O. Li, "Stock portfolio selection using chemical reaction optimization," in *Proceedings of the international conference on operations research and financial engineering. Paris, France*, 2011.

[125] S. K. Mishra, G. Panda, B. Majhi, and R. Majhi, "Improved portfolio optimization combining multiobjective evolutionary computing algorithm and prediction strategy," in *Proceedings of the World Congress on Engineering*, 2012, vol. 1.

[126] A. H.-L. Chen, Y.-C. Liang, and C.-C. Liu, "An artificial bee colony algorithm for the cardinality-constrained portfolio optimization problems," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, 2012, pp. 1–8.

[127] T. Cui, S. Cheng, and R. Bai, "A combinatorial algorithm for the cardinality constrained portfolio optimization problem," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, 2014, pp. 491–498.

[128] M. Tuba and N. Bacanin, "Upgraded firefly algorithm for portfolio optimization problem," in *Computer Modelling and Simulation (UKSim), 2014 UKSim-AMSS 16th International Conference on*, 2014, pp. 113–118.

[129] S. K. Mishra, G. Panda, and R. Majhi, "Constrained portfolio asset selection using multiobjective bacteria foraging optimization," *Oper. Res.*, vol. 14, no. 1, pp. 113–145, 2014.

[130] "NSE - National Stock Exchange of India Ltd." [Online]. Available: https://www.nseindia.com/products/content/equities/indices/historical_index_data.htm. [Accessed: 15-Mar-2019].

References

[131] “NSE - National Stock Exchange of India Ltd.” [Online]. Available: https://www.nseindia.com/products/content/derivatives/equities/adjust_in_case_corp.htm. [Accessed: 15-Mar-2019].

List of Publications

- Bharat V. Chawda and Jayeshkumar M. Patel, "Stock Market Portfolio Management: A Walk-through", International Journal of Recent and Innovation Trends in Computing and Communication, Volume-3, Issue-6, pp. 4136-4143, Jun 2015

- Bharat V. Chawda and Jayeshkumar M. Patel, "Natural Computing Algorithms – A Survey", International Journal of Emerging Technology and Advanced Engineering, Volume-6, Issue-6, pp. 221-227, Jun 2016

- Bharat V. Chawda and Jayeshkumar M. Patel, "Investigating Performance of Various Natural Computing Algorithms", International Journal of Intelligent Systems and Applications, Volume-9, Issue-1, pp. 46-59, Jan 2017

- Bharat V. Chawda and Jayeshkumar M. Patel, "Winnowing Algorithm – A Novel Natural Computing Algorithm for Portfolio Weight Optimization", In Proceedings of the International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM-2019), Amity University Rajasthan, Jaipur, Rajasthan, 2019, pp. 1670–1681, February 26 – 28, 2019.

- Bharat V. Chawda and Jayeshkumar M. Patel, "Winnowing Algorithm with Enhanced Exploration to Optimize Portfolio Weights", International Journal of Intelligent Information and Database Systems, Inderscience Publication (Accepted; Yet to be published)