Università
Ca'Foscari
Venezia
Facoltà
di Economia

Corso di Laurea Specialistica
in Economia, curriculum Models and Methods of Quantitative Economics per la Gestione dell'Impresa

Prova finale di Laurea

Particle Swarm Intelligence: an alternative approach in Portfolio Optimization

Relatore
Prof. Marco Corazza

Laureando
Nozim Komilov
Matricola 822556

Anno Accademico
2008-2009

ACKNOWLEDGMENTS

It is a pleasure to express my deep gratitude to my supervisor, prof. Marco Corazza, for the invaluable guidance he gave during the study. He is most responsible for helping me complete the writing of this thesis as well as the challenging research that lies behind it.

Special thanks are due to prof. Riccardo Gusso for providing important and essential input at crucial points. His recommendations and suggestions have been invaluable for the development of the thesis.

I also wish to express my gratitude to my Dad, my Mom, Zulfira, Kaleab and Hanjarivo for their immeasurable support, encouragement and advise during my study.

ABSTRACT

The performance of the Particle Swarm Optimization (PSO) method in coping with Portfolio Selection problems is investigated in this work. We introduce extended PSO algorithm as an alternative PSO approach to solve constrained portfolio selection problems in which the objective function is two-sided coherent risk measure. In PSO, the potential solutions, called particles, surf through the search space by learning from the global best position and its own memory. By doing so, each particle benefits not only from its discoveries, but from what the other particles have found as well, and therefore, it heads towards the best region in the search space discovered by the group. The obtained results are reported and compared with those obtained through Guaranteed Convergence PSO (GCPSO). In the end we will provide the efficient frontier of portfolios with respect to the two-sided coherent risk measure. Conclusions are derived and directions of future research are proposed.

Keywords: Metaheuristics, Evolutionary Computation, Particle Swarm Optimization (PSO), Guaranteed Convergent PSO, Genetic Algorithm, Portfolio Selection, Risk Measure, Markowitz Model, VaR, Coherent Risk Measures, Two-sided Risk Measure.

TABLE OF CONTENTS

"What is it that governs here? What is it that issues orders, foresees the future, elaborates plans, and preserves equilibrium?"

Maeterlinck, M.
The Life of the White Ant., London., 1927

INTRODUCTION

We will apply one of the successful stochastic optimization algorithms, Particle Swarm Optimization (PSO) in portfolio selection problems. PSO algorithm, one of the Evolutionary Computation methods, has come to existence from the interests to replicate the collective foraging behavior of a group of simple agents, like a school of fish, a flock of birds, and similar.

In the first chapter we will describe and discuss about PSO, starting with the main issues, its original formulation, operational mechanisms and some extensions to the original PSO, and then analyze its convergent property which is the main reason to act as an optimization tool.

In second chapter we will give fundamental information related to the portfolio selection problem. We will talk about the possible candidates for the risk measure as an objective function in the minimization problem. The early versions of risk measures, such as variance and VaR are discussed; their shortcomings in terms of properly measuring risk are analyzed. Then we describe coherent risk measures as possible alternatives to traditional risk measures, their properties with some examples. New two-sided coherent risk measure is presented, and a portfolio selection problem with this risk measure is introduced.

In the third chapter we will introduce new extended PSO algorithm and provide the necessary experiments and results from the application of the extended PSO in the portfolio selection problem. We will briefly describe each parameter, possible values it can take, and conduct tests on them with MATLAB. In the end, by using extended PSO we will compute and plot efficient-frontier of portfolios with risky assets according to the new two-sided risk measure.

CHAPTER 1 Particle Swarm Optimization (PSO)

"Ants aren't smart. Ant colonies are."

Deborah M. Gordon
Biologist at Stanford University

This chapter is dedicated to the description and discussion of one of the successful computational models, Particle Swarm Optimization (PSO), inspired by the collective foraging behavior of a group of simple agents, like a school of fish, a flock of birds, and etc. We will tackle with the main issues in PSO, its original formulation, operational mechanisms and some extensions to the original PSO, and analyze their convergent properties which, in turn, act as the stepping-stone in the application to optimization problems.

1.1. Introduction

The purpose behind designing artificial intelligence and artificial life comes from the very beginnings of the information era. The earliest computer scientists were interested not only in electronics, but also in both biology and psychology as they were inspired by the ideas of imbuing computer programs with intelligence, and with the characteristics of bio-system – like capability of self−reproduce, and with the adaptive ability to learn and to control their environments.

Moreover, at a time when the world is becoming so obscure that no single individual can comprehend it, when information (and not the lack of it) is threatening our lives, when software systems become so difficult to deal with that they can no longer be controlled, the

Nature offers an alternative way of designing "intelligent" systems, in which autonomy, emergence, and distributed functioning replace control, preprogramming, and centralization[1].

On Earth, indeed, we can find numerous examples of how various kinds of creatures or species use “smart” techniques to cope with difficulties, such as adaptation to changing environmental conditions, collective work, military strategy, advanced communications network, an astute and rational hierarchy, discipline, technology, and perfect city planning[2]… These are fields where human beings may not always be successful enough, but where the certain species such as ants, bees and termites always are.

An appealing aspect of the biological system is that it, as a whole, rests on an organized behavior aimed at achieving a particular goal through countless interaction and self-organization with simple rules of thumb (one ant merely follows the trail left by another). In case of ants, for example, a colony can solve problems unthinkable for individual ants, such as finding the shortest path to the best food source among huge number of possible paths, allocating workers to different tasks, or defending a territory from neighbors. As individuals, ants might be tiny dummies or may have meager intelligence, but as colonies they are able to respond to their environment quickly and effectively.

Therefore, these natural systems became best candidates as guiding representatives to the early pioneers of computer science to achieve their visions. Since then, the number of its successful applications has been rapidly grown in combinatorial optimization, communications networks, and robotics. The use of these biologically motivated computing algorithms has increased over the years, and the first has grown into the field of neural networks, the second into machine learning, and the third into what is now called "evolutionary computation," of which swarm intelligence is one of the examples.

1.2. Swarm Intelligence

In computer science evolutionary computation is a subclass of artificial intelligence that comprises combinatorial optimization problems. It uses iterative evolution, i.e., growth or

[1] Bonabeau E., Dorigo M., Theraulaz G., 1999, “Swarm intelligence: from natural to artificial intelligence”, Oxford University Press.
2 Yahya H., 2000, “The Miracle in the Ant”, Vural Yayincilik Press, Istanbul, Turkey.

development in a population which is then selected in a coordinated random search using parallel processing to achieve the desired final answer. Evolutionary computation is a group of algorithms, such as:

- swarm intelligence (particle swarm optimization and ant colony optimization);
- evolutionary algorithms (evolutionary programming, genetic algorithms, genetic programming, evolution strategy and learning classifier systems);

Since the first introduction by G. Beni and J. Wang in 1989[3], there has been a growing interest towards a subfield of Evolutionary Computation called "Swarm Intelligence". Swarm Intelligence is an innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates[4]. Swarm Intelligence systems are typically made up of a population of simple agents interacting with one another and with their environment. The solution to an optimization problem is derived through this social interaction of agents, each of which does not have individual capability to address the problem.

The swarm constitutes the following three characteristics:

- flexibility (the swarm is able to react and adapt to a changing environment in which it operates);
- robustness (the swarm can accomplish its goals even if some particles in it fail to perform their duties);
- self-organization[5] (activities are not centrally controlled, agents operate under no supervision);

The analysis of the swarm with respect to these three characteristics has led to the development of so-called Swarm Intelligence algorithms, which comprises a system of established procedures aimed at solving optimization problems, data analysis and information extraction, and etc. through the use of basic rules.

3 Beni, G., Wang, J. (1989) Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy.

4 Nedjah N., Mourelle L.M. (2006) Swarm Intelligent Systems. Springer series. p 1.

5 "[…] Through self-organization, the behavior of the group emerges from the collective interaction of all the individuals…And, to a larger extent, flexibility and robustness result from self-organization", Bonabeau E., Meyer C. (2001) Harvard Business Review.

The way insects distribute tasks also contains important flexibility features. In a honeybee colony, for instance, a worker bee can perform all tasks including making the waxen combs in the hive, nursing, gathering food, producing royal jelly, regulating the temperature in the hive, cleaning it of debris and defending it. Yet the allocation of work is very flexible, thus whenever there is a need for workers in another sector, they can be redistributed in order to fill the gap (when food is scarce, nurse bees will help by foraging).

Probably, the most appealing and compelling characteristic of swarm intelligence is that various complex collective behaviors can be attained from particles following different (simple) rules. Bonabeau E. and Meyer C. (2001) try to explain this phenomenon in the following way:

> "Indeed, predicting the behavior that will emerge from even a couple of simple instructions can be surprisingly difficult. Imagine, for instance, that you are at a cocktail party, mingling among a hundred people or so. At random, you silently pick two other people (call them A and B) and then follow this simple rule: always position yourself so that A is between B and you. If everyone else were to do the same, what would happen? Now, change the rule slightly: always position yourself so that you are between A and B. again, if everyone else were to do likewise, what kind of group behavior would emerge?
>
> When they follow the first rule, people will move around for hours, as they continuously try to keep themselves in the right position. But when they follow the second rule, the result will be markedly different: within seconds, everybody will find themselves clumped into a single, almost stationary cluster. A seemingly minor change in the rules can radically alter group behavior[6]. "

This shows that the behavior of the swarm is unpredictable, and liable to change if slight alteration is made to the simple rules. Although predicting the collective behavior of a group is beyond the scope of comprehension of humankind, scholars have shown that it can often be done through simulation. They have also proved that it is easier and more effective to work with many small and autonomous agents than designing complex and centralized systems. Working together within the system, many agents, where each agent is set to a global strategy,

6 Bonabeau E., Meyer C. (2001) Swarm Intelligence: A whole new way to think about business. Harvard Business Review.

can solve very complex problems with a minimal design effort. Particle Swarm Optimization is one of the Swarm Intelligence algorithms to be analyzed in the following paragraphs.

1.3. Particle Swarm Optimization

Particle swarm optimization (PSO) algorithm is a population-based optimization method which imitates the behaviors observed in a swarm of insects, a flock of birds, or a school of fish, and even human social behavior through the conventional notation—a "swarm of particles". PSO had originally been created in order to graphically simulate the choreography of flocks of birds before it was shown that it could be a valuable tool in Mathematical Programming problems.

PSO as an optimization tool was first developed by Kennedy and Eberhart in 1995[7]. The aim of their research was the analysis of rules that underpin the ability of flocks of birds to fly in a synchronized way, change direction abruptly, dissolve and regroup. According to them, each agent in the swarm uses not only the information about its velocity or position, but also the information on the swarm, i.e. the average direction and velocity taken by the other. Birds in the flock try to maintain an optimal distance between them and those nearby. Furthermore, it is not difficult to discover the very same qualities can be attributed to the schools of fish or to other types of swarms.

The behavior of the population is supposed to be aimed at optimizing some usual activities like searching food and defense. For example, if one of the agents discovers a good path to food the rest of swarm will be instantly able to follow even if they are remotely dispersed in the swarm. Therefore, it is reasonable to assume that if we are able to extrapolate rules and terms of computational behavior of the swarms, we will be able get tools that can solve some complex optimization problems.

The studies have shown that an agent tries to remain within the flock and avoid any kind of collision with the other agents of the flock in neighborhood during the flight. This has very strong implication that each agent behaves taking into account the behaviors of other members

[7] Kennedy J., and Eberhart R., Particle swarm optimization, in Proc. of the IEEE Int. Conf. on Neural Networks, Piscataway, NJ, pp. 1942–1948, 1995.

in its neighborhood. By following these simple rules, we can describe the collective motion of a whole swarm. An agent is faced with two alternatives when it discovers a food source: it leaves the group to get to the food or remains in the group. If more individuals are directed towards the food, other nearby members will change their direction in order to remain within neighborhood, and therefore, exploit the same source of nourishment. Eventually, the whole swarm has changed towards those areas perceived to be more promising, i.e. the information has gradually spread to everyone in the group. Thus, search for food activities reveal an important social aspect in swarms: any individual benefits not only from its discoveries, but from what the other have found as well, therefore, heading towards the best region discovered by the group. This feature is the core of the PSO algorithm: the search strategy can be expressed to keep balance between exploration and exploitation; the first is linked to the individual behavior seeking solution, and the second is related to benefiting from the successes of other individuals.

Particle Swarm behavior is displayed by particles in multidimensional space that have two characteristics: a *position* and a *velocity*. Particles are randomly initialized in the search space. It is desirable that the particles are initially well-distributed in order to ensure necessary exploration capabilities, and to avoid risk of being trapped in local optima. The population consists of many particles (agents), and each of them represents a point which is viewed as a possible solution candidate. Also, each particle has its so-called fitness value and an index of "quality" (i.e. how good is the solution for the particle) measured by a fitness function. These particles wander around the hyperspace or search space and remember the best position (with highest “quality”) that they have discovered. They communicate good positions to each other and adjust their own position and velocity based on these good positions.

The velocity vector determines the direction and distance of motion of each particle. It represents the behavior of each particle at a “time” or in every iteration, and is added to the previous position of the particle to get a new position.

PSO can be implemented and applied easily to solve various functional optimization problems, or the problems that can be transformed to functional optimization problems. For

applying algorithm successfully, one of the key issues is finding how to map the problem solution into the PSO particle, which directly affects its feasibility and performance[8]. Thus, in the following paragraphs we will talk about properties of PSO, its main theorems and application in quantitative context. Before continuing with the analysis of PSO, it is useful to remind some basics of Optimization and to provide fundamental concepts of heuristic and metaheuristic algorithms to highlight the main reasons for the widespread use of PSO.

1.3.1. Optimization

In this section we introduce the main definitions of a basic optimization problem, in order to make the reader easier understand the concepts that will be used through the rest of this work.

Consider the optimization problem

$$\begin{aligned} &\textit{minimize} && f(x) \\ &\textit{s.t.} && x \in \Omega \end{aligned}$$

The function $f: \mathbb{R}^n \to \mathbb{R}$ that we want to minimize is a real-valued function, and is called the *objective function*. The vector x is an n -vector of independent variables, i.e., $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$. The variables $x_1, x_2, \dots, x_n$ are often referred to as *decision variables.* The set Ω is a subset of $\mathbb{R}^n$, called the *constraint set* or *feasible set*.

The optimization problem above can be seen as a decision problem that requires finding the vector x^* of the decision variables over all possible vectors in Ω that results in the smallest value of the objective function. This vector is called the minimizer of f over Ω. It is possible that there may be many minimizers. In this case, finding any of them will be sufficient.

There are also optimization problems that require maximization of the objective function. These problems, however, can be represented in the above form because maximizing f is equivalent to minimizing $-f$. Therefore, we can restrict our attention to minimization problems without loss of generality. The above problem is a general form of a constrained optimization problem, because the decision variables are constrained to be in the constraint set Ω. In case of $\Omega = \mathbb{R}^n$, we refer to the problem as an unconstrained optimization problem.

[8] Nedjah N., Mourelle L.M. (2006) Swarm Intelligent Systems. Springer series.

In considering the general optimization problem above, we distinguish between two kinds of minimizers, as specified by the following definitions.

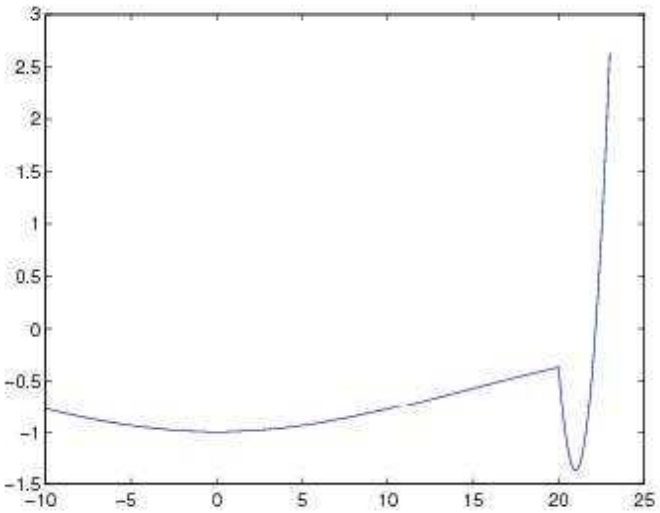
Definition 1.1 (Local minimizer). Suppose that $f: \mathbb{R}^n \to \mathbb{R}$ is a real-valued function defined on some set $\Omega \subset \mathbb{R}^n$. A point $x^* \in \Omega$ is a *local minimizer* of f over Ω if there exists $\varepsilon > 0$ such that $f(x) \geq f(x^*)$ for all $x \in \Omega \setminus \{x^*\}$ and $\|x - x^*\| \leq \varepsilon$.

Definition 1.2 (Global minimizer). Suppose that $f: \mathbb{R}^n \to \mathbb{R}$ is a real-valued function defined on some set $\Omega \subset \mathbb{R}^n$. A point $x^* \in \Omega$ is a *global minimizer* of f over Ω if $f(x) \geq f(x^*)$ for all $x \in \Omega \setminus \{x^*\}$.

As an example, consider the following function

$$f(x) = \begin{cases} -\exp\left(-\left(\frac{x}{20}\right)^2\right) & \text{for} \quad x \leq 20 \\ -\exp(-1) + (x-20)(x-22) & \text{for} \quad x > 20 \end{cases}$$

The following figure shows the plot of the function.



The function has two local minima, one at $x = 0$, where the function value is -1, and the other at $x = 21$, where the function value is $-1 - 1/e$. Since the latter value is smaller, the global minimum occurs at $x = 21$.

Rigorously speaking, an optimization problem is said to be solved only when a global minimizer is found. However, it is, in general, difficult to find global minimizers. Therefore, in practice, we often have to be satisfied with finding local minimizers.

1.3.2. Heuristic and Metaheuristic algorithms

Despite enormous advances in science and development in information technologies, there are still many optimization problems for which either no closed-form solutions have been found or suggested numerical techniques require too much time to get to the final answer. The extent of the drawback can be felt especially if one needs the outcome immediately or in a reasonable period of time. In some circumstances, to find and use the appropriate algorithm which is capable of discovering the optimal solution of a specific problem can be very expensive and cumbersome.

Two fundamental goals in computer science are finding algorithms with provably good run times and with provably good or optimal solution quality[9]. For the reasons explained above, these have stimulated the development of several new techniques, so-called heuristics, which can provide "good" eligible solutions that can be achieved in "reasonable" time. With heuristic algorithms (from Greek *heuriskein* "to find") we can build solving methods whose outcome may not necessarily be optimal, but still be consistent with the optimal solution.

The use of heuristics has become very attractive in real world applications. For many problems in practice, a heuristic algorithm may be the unique way to obtain good solutions in reasonably short period of time. Metaheuristics, a subfield of general heuristic strategies, includes stochastic components to utilize randomized search abilities.

[9] Source: http://www.en.wikipedia.org

In real world applications, the widespread use of heuristic algorithms, as an approximation for the optimal solutions, instead of ordinary mathematical tools can be mainly explained due to a number of factors, such as:

- the optimization problem does not have a closed-form or analytical solution;
- too costly (in terms of time) and cumbersome to compute;

Given an input variable x, a heuristic algorithm A offers a solution value $z^A(x)$ such that $z^A(x) \geq z^*(x)$ in a minimization problem, where $z^*(x)$ is the optimal value of the variable. It implies that the algorithm provides an *upper bound* for the value of the optimal solution. In the case of problem maximization problem, $z^A(x) \leq z^*(x)$ will hold, i.e. the algorithm A provides a *lower bound* for the value of the optimal solution[10].

Metaheuristics, in their original definition, are solution methods that coordinate an interaction between local improvement procedures and higher level strategies to generate a process able to escape from local optima and carry out a robust search of a solution space. Since first introduced[11], these optimization tools have also come to include any procedures that utilize strategies for avoiding the trap of local minima in complex solution spaces.

A number of the tools and mechanisms[12] that have emerged from the creation of metaheuristic methods have proved to be remarkably effective, so much so that metaheuristics have moved into the spotlight in recent years as the preferred line of attack for solving many types of complex problems, particularly those of a combinatorial nature. While metaheuristics is not able to guarantee the optimality of the solutions they find, exact procedures (which empirically can provide such a certification, if allowed to run long enough) have often proved unable to find solutions whose quality is close to that obtained by the leading metaheuristics—particularly for real world problems, which often attain notably high levels of complexity. In addition, some of the more successful applications of exact methods have come about by

[10] Source: http://www.dei.unipd.it/~monaci/euristici_rev2

[11] First works on stochastics optimization methods started in 1952, although first mention of the term "meta-heuristic" was during the conception of Tabu Search by Fred Glover in 1986.

[12] PSO, Genetic Algorithms, Iterated Local Search, Ant Colony Optimization, Genetic Programming, Multi-Start Methods, Scatter Search, Tabu Search, Memetic Algorithms, Variable Neighborhood Search, Guided Local Search, GRASP, Simulated Annealing, Constraint Programming, Constraint Satisfaction, Neural Network Methods for Optimization, Hyper-Heuristics, Parallel Strategies for Metaheuristics, Metaheuristic Class Libraries, and A-Teams.

incorporating metaheuristic strategies within them[13]. These outcomes have motivated additional research and application of new and improved metaheuristic methodologies, of which the Particle Swarm Optimization is a particular case.

1.3.3. Particle Swarm Optimization

Original Particle Swarm Optimization is a population-based optimization method introduced by Kennedy and Eberhart in 1995. Attractive features of using PSO are triviality in implementing and no gradient information requirement. This allows PSO to be utilized on functions where the gradient is either unavailable or difficult to compute[14]. The algorithm utilizes a population of particles, where each particle can be viewed as a potential solution to an optimization problem.

As we stated in the above, Particle Swarm behavior is displayed by particles in multidimensional space (let's say N-dimensional) that have two characteristics: a *position* and a *velocity*. We denote n as the number of particles in the swarm. Each particle i has the following characteristics:

$\boldsymbol{x_i}$: The *current position* of the particle;

$\boldsymbol{y_i}$: The personal *best position* of the particle;

$\boldsymbol{v_i}$: The *current velocity* of the particle.

The personal best position of particle i represents the position where the highest fitness value of that particle has been achieved. For minimization problems, the smaller is the function value the higher is the fitness. Let's also denote f to be the objective function to be minimized. Then the update equation for the personal best position is given in the following equation, where t stands for time.

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{if } f\big(x_i(t+1)\big) < f(y_i(t)) \end{cases} \qquad (\mathbf{1.1})$$

[13] Globel F., Kochenberger G., (2003) Handbook of Metaheuristics, Kluwer Academic Publishers.

14 "[…] From the works of Denault (2001) and Tasche (2000) it is known that differentiability of risk measures is crucial for risk capital allocation in portfolios. The reason is that in the case of differentiable positively homogeneous risk measures the gradient due to asset weights has figured out to be the unique reasonable per-unit allocation principle." Fischer T. (2003) Risk capital allocation by coherent risk measures based on one-sided moments. Journal of Insurance: Mathematics and Economics.

The velocity update equation[15] is then

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_{1,j}(t)\left[y_{i,j}(t) - x_{i,j}(t)\right] + c_2 r_{2,j}(t)[\hat{y}_j(t) - x_{i,j}(t)] \quad (\mathbf{1.2})$$

Two independent random sources, $r_1 \sim U(0,1)$ and $r_2 \sim U(0,1)$, scaled by constants $0 < c_1, c_2 \leq 2$, are used to effect the stochastic nature of the algorithm. These constants are called the *acceleration coefficients*, and they affect the maximum size of the step that a particle can take in a single iteration. Index *j* represents the j^{th} dimension of a vector associated with the i^{th} particle.

The scale factor *w* is called *inertia weight* which controls how much of the previous velocity should be preserved from the previous iteration. Shi and Eberhart[16] show that *w* values close to 1.0 are preferable. Another possible way of specifying *inertia weight* is to set *w* linearly decrease[17] from 1.0 to, say, 0.5. This setting allows the PSO algorithm to explore large area in the earlier iterations, and to exploit the search space by using smaller inertia weight.

The term $c_1 r_{1,j}(t)\left[y_{i,j}(t) - x_{i,j}(t)\right]$ is called *cognition* effect, since it only takes into account the particle's own experiences. The third term in the velocity update equation, $c_2 r_{2,j}(t)[\hat{y}_j(t) - x_{i,j}(t)]$, represents the *social* interaction between the particles.

There are two possible ways to define the term $\hat{y}$: *gbest* and *lbest* models. The difference between the two algorithms can be explained by the set of particles with which the given particle will interact directly, where the symbol $\hat{y}$ is used to characterize this interaction. The details of the two models will be discussed in full below.

The position of each particle is updated using the new velocity vector for that particle, so that

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (\mathbf{1.3})$$

Here is how the PSO algorithm works:

[15] This version of PSO is called PSO with inertia weight w, while the original PSO does not have coefficient w.

[16] Shi Y., Eberhart R.C. A Modified Particle Swarm Optimizer. In IEEE International Conference of Evolutiorary Computation, Anchorage, Alaska, May 1998.

[17] One of the possible ways to define $\boldsymbol{w}$ can be expressed as $w = w_{max} - ((w_{max} - w_{min})/iteration_{max}) * iteration.$

1) Initial positions of the particles throughout the search space are defined by initializing each coordinate $x_{i,j}$ to a value drawn from the uniform distribution on the interval $[-x_{max}, x_{min}]$, for all $i \in 1,2, \dots n$ and $j \in 1,2, \dots N$.
2) The velocities of the particles are initialized to vector 0.
3) Set $y_i = x_i$[18], $\forall i \in 1,2, \dots n$.

At first, a particle is found to be the best particle in a neighborhood of particles based on its fitness value. Then, according to velocity update equation, all particles move not only in the direction of this particle, but also in the direction of their own best solutions that they have discovered before. During the course of subsequent iterations all particles might discover a better particle, in which case the other particles will start to head towards the new "best" particle. By approaching the current best solution from different directions in the search space, there is high probability that the neighboring solutions will be discovered by some particles.

Gbest Model

Gbest differs from ***lbest*** with respect to the set of particles with which a particle will interact directly, and the symbol $\hat{y}$, the best position discovered by any of the particles so far, represents this interaction. The update equation for $\hat{y}$ is the following:

$$\hat{y}(t) \in \{y_0(t), y_1(t), \dots, y_s(t)\} | f\big(\hat{y}(t)\big) = \min\big\{f\big(y_0(t)\big), f\big(y_1(t)\big), \dots, f\big(y_s(t)\big)\big\} \quad (\mathbf{1.4})$$

In ***gbest*** model all particles converge towards the particle with the highest fitness value at a faster rate at the expense of less exploration. This model results only a single "best solution", called the *global best particle*, across all the particles in the swarm. This particle attracts all other particles in the swarm, so if *global best particle* is not updated regularly, the swarm may converge prematurely.

[18] We can also initialize the best positions of particles as positive infinite without loss of generality.

Lbest Model

Later, ***lbest*** model was introduced in order to prevent premature convergence by maintaining multiple *local best particles.* A subset of particles is defined for each particle from which the *local best particle* $\hat{y}_i$ is then selected. The *lbest* velocity update equation for a neighborhood of size M is as follows:

$$B_i = \{y_{i-M}(t), y_{i-M+1}(t), \dots, y_{i-1}(t), y_i(t), y_{i+1}(t), \dots, y_{i+M-1}(t), y_{i+M}(t)\} \qquad (\mathbf{1.5})$$

$$\hat{y}_i(t+1) \in N_i | f\left(\hat{y}_i(t+1)\right) = \min\{f(a)\}, \forall a \in B_i \qquad (\mathbf{1.6})$$

Neighborhood selection is purely based on each particle's index number. This has two advantages:

- No partitioning or clustering of search space has to be carried out;
- It maintains the spread of information with respect to the best position, regardless of their current location in search space.

1.4. Convergence behavior of Particle Swarm

The first mathematical analysis of the trajectory of PSO is performed by Ozcan and Mohan[19]. To make the problem more tractable, they used original PSO[20] proposed by Kennedy et al. without inertia weight, and held the stochastic components of the update equations as well as the personal best position of the particle, y_i, as constant. Thus, $\phi_1 = r_1(t)c_1$ and $\phi_2 = r_2(t)c_2$ are only particular instances of the stochastic component. From the simplified model, Ozcan et. al provided a recursive form of update equation:

[19] E. Ozcan and C. K. Mohan. Analysis of a Simple Particle Swarm Optimization System. In Intelligent Engineering SYstems Through Artificial Neural Networks volume 8, pages 253-258, 1998.

[20] In Original PSO, the velocity update equation has the form $v_{i,j}(t+1) = v_{i,j}(t) + c_1 r_{1,j}(t)\left[y_{i,j}(t) - x_{i,j}(t)\right] + c_2 r_{2,j}(t)[\hat{y}_j(t) - x_{i,j}(t)]$, where there is no inertia weight.

$$x_i(t) - (2 - \phi_1 - \phi_2)x_i(t-1) + x_i(t-2) = \phi_1 y_i + \phi_2 \hat{y} \qquad (\mathbf{1.7})$$

which can be solved to obtain the following closed form equation:

$$x_i(t) = A * sin(\theta_i t) + B * cos(\theta_i t) + k_i \qquad (\mathbf{1.8})$$

They have discovered that, in the absence of stochastic influences, a particle does not jump in the search space, but rather traces out a sinusoidal waveform.

Yet, another and more robust approach is found in Bergh[21]. He assumed the velocity update equation for the PSO with inertia weight. Since there is no interaction between the different dimensions of the particles in the PSO, for sake of simplicity in notation, he restricted the particle to a single dimension. The trajectory of a single particle in isolation was analyzed, "freezing" other particles in the swarm for a short period of time. By substituting velocity update equation into position update one he obtained the following non-homogeneous recurrence relation:

$$x_{t+1} = (1 + w - \phi_1 - \phi_2)x_t - wx_{t-1} + \phi_1 y + \phi_2 \hat{y} \qquad (\mathbf{1.9})$$

Again, as in Ozcan et. al, ϕ_1 and ϕ_2 are particular instances of $r_1(t)c_1$ and $r_2(t)c_2$, respectively. So, ϕ_1, ϕ_2 and w are assumed to be constant. To solve to obtain the closed-from solution of the recurrence relation, initial conditions $x(0) = x_0$ and $x(1) = x_1$ are specified. Bergh found closed form equation as follows:

$$x_t = k_1 + k_2\alpha^t + k_3\beta^t \qquad (\mathbf{1.10})$$

Where

$$k_1 = \frac{\phi_1 y + \phi_2 \hat{y}}{\phi_1 + \phi_2} \qquad (\mathbf{1.11})$$

[21] Van den Bergh F., (2001) An analysis of particle swarm optimizers, PhD Dissertation, Dept. of Computer Science, University of Pretoria, South Africa.

$$\gamma = \sqrt{(1 + w - \phi_1 - \phi_2)^2 - 4w} \tag{1.12}$$

$$\alpha = \frac{1 + w - \phi_1 - \phi_2 + \gamma}{2} \tag{1.13}$$

$$\beta = \frac{1 + w - \phi_1 - \phi_2 - \gamma}{2} \tag{1.14}$$

$$x_2 = (1 + w - \phi_1 - \phi_2)x_1 - wx_0 + \phi_1 y + \phi_2 \hat{y} \tag{1.15}$$

$$k_2 = \frac{\beta(x_0 - x_1) - x_1 + x_2}{\gamma(\alpha - 1)} \tag{1.16}$$

$$k_3 = \frac{\alpha(x_1 - x_0) + x_1 - x_2}{\gamma(\beta - 1)} \tag{1.17}$$

Note that actual PSO algorithm allows y and $\hat{y}$ to change through the velocity and position update equations while t changes, but the analytic solution is found assuming y and $\hat{y}$ as constant. Thus, the equation above continues to be valid until the new best personal and best particles positions are discovered, after which new values of k_1, k_2 and k_3 are recomputed by implementing above given equations. Therefore, the closed form equation for recurrence relation represents the "long-term" behavior of a particle for a short period of time which is subject to change with respect to t.

To extrapolate the behavior of the particle we continue to assume short period of time, where y and $\hat{y}$ are constants, so that we are able to manipulate the convergent/divergent characteristics of the swarm.

Bergh (2001) performed the analysis of the convergence of a particle's trajectory assuming that ϕ_1 and ϕ_2 are constant. That made the problem easy to tackle, and by using the

largest values that ϕ_1 and ϕ_2 can take (the values of c_1 and c_2 can be considered an upper bound), he studied the worst-case behavior of a particle in terms of convergence.

He found that the convergence of the sequence $\{x_t\}_{t=0}^{+\infty}$ could be determined by the magnitude of the values of α and β using the L_2 norm for vectors. In particular, the trajectory of a particle, $\{x_t\}_{t=0}^{+\infty}$, will converge whenever $\max(\| \alpha \|, \| \beta \|) < 1$, otherwise the limit does not exist.

$$\lim_{t\to+\infty} x_t = \lim_{t\to+\infty} k_1 + k_2\,\alpha^t + k_3\beta^t = k_1 = \frac{\phi_1 y + \phi_2 \hat{y}}{\phi_1 + \phi_2} \qquad (\mathbf{1.18})$$

The average behavior of the system can be observed by considering the expected values of ϕ_1 and ϕ_2 $(i = 1, 2)$:

$$E[\phi_i] = c_i \int_0^1 \frac{r_i}{1-0}\, dr_i = c_i \left.\frac{r_i^2}{2}\right|_0^1 = \frac{c_i}{2} \qquad (\mathbf{1.19})$$

If the expected values of ϕ_1 and ϕ_2 are substituted in equation above, the following results:

$$\lim_{t\to+\infty} x_t = \frac{c_1 y + c_2 \hat{y}}{c_1 + c_2} = \frac{c_1}{c_1 + c_2} y + \frac{c_2}{c_1 + c_2} \hat{y} = (1 - a)y + a\hat{y} \qquad (\mathbf{1.20})$$

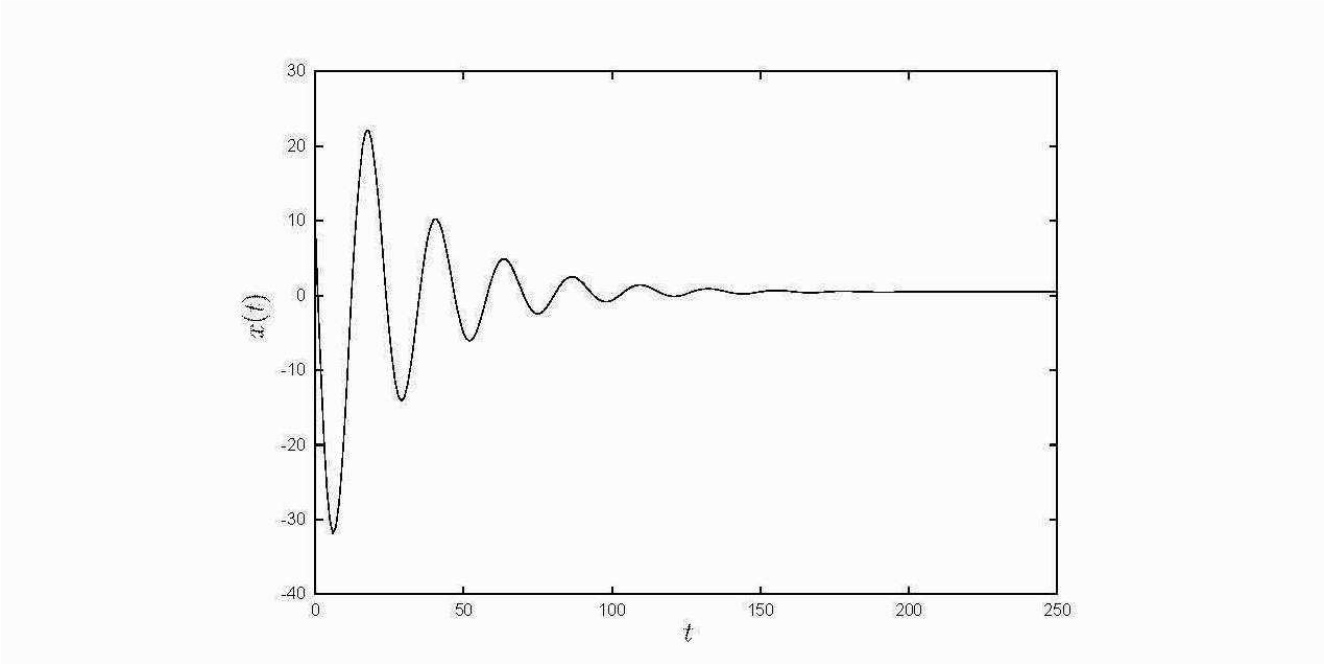
where $a = c_2/(c_1 + c_2)$, therefore $\alpha \in [0,1]$. This outcome suggests that the particle will seek for better solutions in the area between its personal best and the global best position.

The trajectory of a particle can be guaranteed to converge if the parameters of ϕ_1, ϕ_2 and w are chosen so that $\max(\| \alpha \|, \| \beta \|) < 1$. Taking into account that $\phi_1 \in [0, c_1]$ and $\phi_2 \in [0, c_2]$, all w values must satisfy the inequality

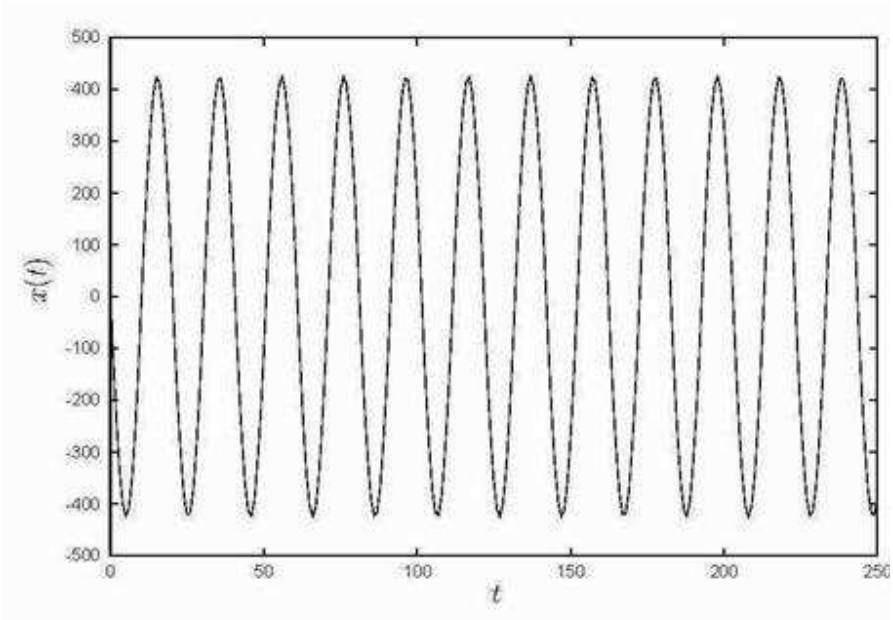
$$w > \tfrac{1}{2}(c_1 + c_2) - 1 \qquad (\mathbf{1.21})$$

An infinite number of parameter scenarios exist that guarantee a convergent behavior. To ensure convergence, the value for w should thus be chosen so that it satisfies

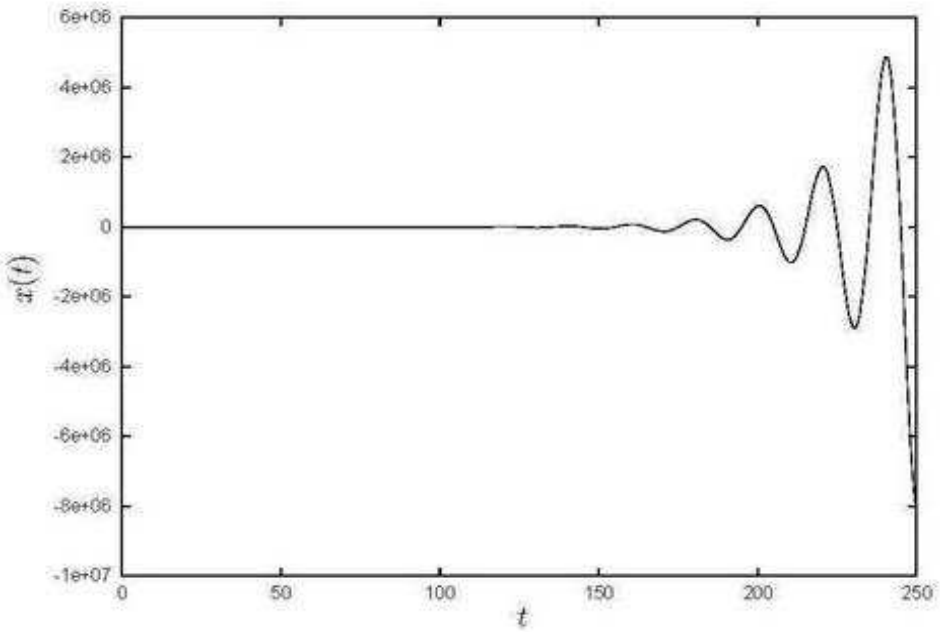
$\mathrm{w} > \frac{1}{2}(\mathrm{c}_1 + \mathrm{c}_2) - 1.$



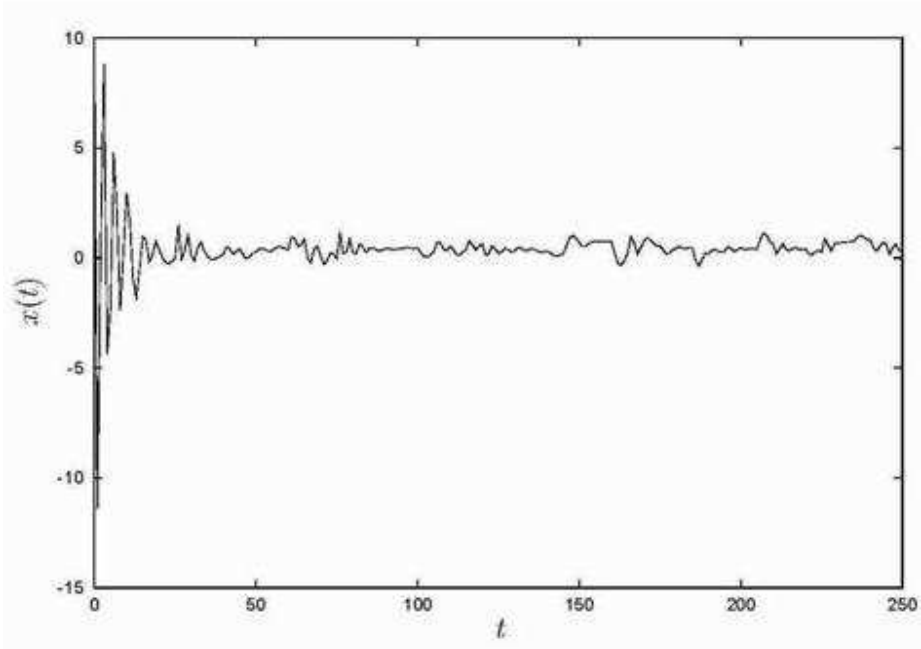
Graph 1.1 plots the particle position over time. A convergent behavior of the particle is obtained with the parameter settings $y = 1.0$, $\hat{y} = 0$, $x(0) = 10$, $x(1) = 10 - 9\phi_1 - 10\phi_2$, $w = 0.5$ and $\phi_1 = \phi_2 = 1.4$. *Source:* Van den Bergh F., (2001) An analysis of particle swarm optimizers, PhD Dissertation, Dept. of Computer Science, University of Pretoria, South Africa.



Graph 1.2 plots the particle position over time. A divergent behavior of the particle is obtained with the parameter settings $y = 1.0$, $\hat{y} = 0$, $x(0) = 10$, $x(1) = 10 - 9\phi_1 - 10\phi_2$, $w = 1.0$ and $\phi_1 = \phi_2 = 1.999$. *Source:* Van den Bergh F., (2001) An analysis of particle swarm optimizers, PhD Dissertation, Dept. of Computer Science, University of Pretoria, South Africa.



Graph 1.3 plots the particle position over time. A divergent particle trajectory is obtained with the parameter settings $y = 1.0, \hat{y} = 0, x(0) = 10, x(1) = 10 - 9\phi_1 - 10\phi_2,\ w = 0.7$ and $\phi_1 = \phi_2 = 1.9$. *Source:* Bergh F., (2001) An analysis of particle swarm optimizers, PhD Dissertation, Dept. of Computer Science, University of Pretoria, South Africa.



Graph 1.4 plots the stochastic particle position over time. A convergent stochastic particle trajectory is obtained with the parameter settings $y = 1.0, \hat{y} = 0, x(0) = 10, x(1) = 10 - 9\phi_1 - 10\phi_2,\ w = 0.7$ and $c_1 = c_2 = 1.4$. *Source:* Van den Bergh F., (2001) An analysis of particle swarm optimizers, PhD Dissertation, Dept. of Computer Science, University of Pretoria, South Africa.

This relation can also be rearranged to compute the values of c_1 and c_2 once an appropriate w has been selected. In the original PSO parameters with $c_1 = c_2 = 2$ and $w = 1$ do not satisfy the inequality, which imply that original particle swarm has divergent trajectories. Ozcan et al. as well as Bergh have concluded that considering the stochastic components is still consistent with the outcome.

Although some characteristics of the trajectory can be applied to whole ranges of ϕ values, it is still not clear what the influence of randomness will be on the trajectory. Bergh has discovered that it is not strictly necessary to choose the values of c_1 and c_2 so that relation $w > \frac{1}{2}(c_1 + c_2) - 1$ is satisfied for a given w value. According to him, as long as the ratio

$$\phi_{ratio} = \frac{\phi_{crit}}{c_1 + c_2} \qquad (\mathbf{1.22})$$

is close to 1.0, the trajectory will converge without too many disruptions, where ϕ_{crit} is defined as

$$\phi_{crit} = \sup \phi \,|0.5\phi - 1 < w, \;\; \phi \in (0, c_1 + c_2] \qquad (\mathbf{1.23})$$

Holland[22] argues that the algorithms must preserve the balance between *exploration* and *exploitation.* Exploration ability is the algorithm's propensity to explore more regions of the search space, while exploitation is the tendency to search a smaller region more thoroughly. Smaller w values result in faster rate of convergence, and it is offset by divergent steps as measured by the value ϕ_{ratio}, which is influenced by c_1 and c_2. The rule of thumb is, once a proper w value has been selected, to choose slightly larger c_1 and c_2, thus keeping $\phi_{ratio} < 1$, in order to motivate the algorithm to have more "exploration" behavior. After letting the particle swarm explore for some time, we can gradually decrease the values of c_1 and c_2 until $\phi_{ratio} = 1$ has been attained.

[22] Holland J.. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI, 1975.

Convergence Criteria

An extensive analysis of convergence of stochastic search algorithms can be found in Solis and Wets[23]. They have mostly studied pure random search algorithms and provided criteria under which algorithms can be considered to be global search or local search algorithms. Most of the definitions and theorems about convergence characteristics can be found in their work, and readers may also refer to Bergh (2001) for their proofs.

Proposition 1.1. Given a function f from $\mathbb{R}^n$ to $\mathbb{R}$ and Ω a subset of $\mathbb{R}^n$. We seek a point z in Ω which minimizes f on Ω or at least which yields an acceptable approximation of the infimum of f on Ω.

In the k[th] iteration, the algorithm is defined on a probability space $(\mathbb{R}^n, \mathfrak{B}, \mu_k)$.

Definition 1.3. Initialize z_o in Ω and generate a vector ξ_k from sample space $(\mathbb{R}^n, \mathfrak{B}, \mu_k)$. D is a function that constructs a solution to the problem and defined as $z_{k+1} = D(z_k, \xi_k)$.

H1 $f\big(D(z,\xi)\big) \leq f(z)$ and if $\xi \in S$, then $f\big(D(z,\xi)\big) \leq f(\xi)$;

Let's denote z^* in S where f is at its infimum and define optimality region as

$$R_\epsilon = \{z \in \Omega | f(z) < z^* + \epsilon\} \qquad (\mathbf{1.24})$$

where $\epsilon > 0$. If the algorithm finds a point in the optimality region, then it has found an acceptable approximation to the global minimum of the function.

[23] F. Solis and R. Wets. Minimization by Random Search Techniques. Mathematics of Operations Research, 6:19-30, 1981.

H2 For any (Borel) subset A of Ω with $\mathrm{v}[\mathrm{A}] > 0$, we have that

$$\prod_{k=0}^{\infty}(1 - \mu_k[A]) = 0 \qquad (\mathbf{1.25})$$

where $v[A]$ is the Lebesgue measure on the set A and $\mu_k[A]$ is the probability of A being generated by μ_k.

Above condition implies that for $\mathrm{R}_\epsilon \subset \Omega$ the probability of sampling a point in the optimality region must be nonzero. A true global search algorithm must satisfy **H2** assumption.

H3 To any $z_0 \in S$, where corresponds a $\gamma > 0$ and an $0 < \eta \leq 1$ such that:

$\mu_k[(dist(D(z,\xi), R_\epsilon) < dist(z, R_\epsilon) - \gamma)$ or $(D(z,\xi) \in R_\epsilon)] \geq \eta$ $\qquad (\mathbf{1.26})$

For all k and all z in the compact set $L_0 = \{z \in \Omega | f(z) \leq f(z_0)\}$

$dist(\boldsymbol{z}, A)$ denotes the distance between a point $\boldsymbol{z}$ and a set A, being defined as

$$dist(\boldsymbol{z}, A) = inf_{b \in A} dist(\boldsymbol{z}, \boldsymbol{b}) \qquad (\mathbf{1.27})$$

Theorem 1.1 (Global Search) Suppose that f is a measureable function, Ω is a measurable subset of $\mathbb{R}^n$ and ***(H1)*** and ***(H2)*** are satisfied. Let $\{z_k\}_{k=0}^{+\infty}$ be a sequence generated by the algorithm. Then

$$\lim_{k \to +\infty} P[z_k \in R_\epsilon] = 1 \qquad (\mathbf{1.28})$$

where $P[z_k \in R_\epsilon]$ is the probability that at step k, the point z_k generated by the algorithm is in R_ϵ.

Global Search Theorem implies that to be able to find the global minima an algorithm must satisfy **H1** and **H2.**

Theorem 1.2 (Local Search) Suppose that f is a measureable function, Ω is a measurable subset of $\mathbb{R}^n$ and ***(H1)*** and ***(H3)*** are satisfied. Let $\{z_k\}_{k=0}^{+\infty}$ be a sequence generated by the algorithm. Then

$$\lim_{k\to+\infty} P[z_k \in R_\epsilon] = 1 \qquad (\mathbf{1.29})$$

where $P[z_k \in R_\epsilon]$ is the probability that at step k, the point z_k generated by the algorithm is in R_ϵ.

Bergh (2001) has showed that original PSO is neither a local nor global stochastic search algorithm. For some initial states in which the original PSO algorithm will start can lead to a premature convergence thus violating (H3). By using large number of initial particles we can dramatically decrease the probability of PSO being trapped in the global best particle's position. Yet, another approach to handle this problem is to modify the velocity update equation for global best particle as presented below. This algorithm is called Guaranteed Convergence PSO or GCPSO, this algorithm satisfies both (H1) and (H3) according to Bergh (2001). But both original PSO and GCPSO algorithms do not satisfy (H2) which indicates that GCPSO is merely local stochastic search algorithm.

1.5. Modified Particle Swarm Optimization (GCPSO)

If the trajectory of a particle converges, then it will do so towards the value derived from the line between its personal best position and the global best particle's position. Due to velocity update equation, the personal best position of the particle gradually moves toward the global best particle's position, so that once this position is reached all particles will stop moving. However, there is no guarantee that the outcome is the solution for the optimization problem – it merely means that all particles have converged to the best position discovered so far by the particle swarm. If $x_i = y_i = \hat{y}$, then the velocity update equation will only depend on wv_i. So, unless the initial velocity and the value w are significantly greater than zero, all particles will

stop moving once they "arrive" at global best particle's position, which creates premature convergence problem. In order to address this question a new modified PSO algorithm was introduced by Bergh (2001).

Let τ be the index of the global best particle, so that $y_\tau = \hat{y}$. Then, a new velocity update equation for the global best particle (*i.e.* particle τ) is

$$v_{\tau,j}(t+1) = -x_{\tau,j}(t) + \hat{y}_j(t) + wv_{\tau,j}(t) + \rho(t)\left(1 - 2r_{\tau,j}(t)\right) \qquad (\mathbf{1.30})$$

where $\rho(t)\left(1 - 2r_{\tau,j}(t)\right)$ term generates a random sample from a sample space with side lengths $2\rho(t)$. Parameter ρ causes the PSO to perform a random search in an area surrounding the global best position $\hat{y}$. After each time step the value of $\rho(t)$ is adapted, using $(0 < \alpha < 1)$:

$$\rho(t+1) = \begin{cases} \rho(t)/\alpha & \text{if } \#successes > s_c \\ \alpha\rho(t) & \text{if } \#failures > f_c \\ \rho(t) & \text{otherwise} \end{cases} \qquad (\mathbf{1.31})$$

where the terms *#failures* and *#successes* denote the number of consecutive failures or successes with a failure defined as $f\left(\hat{y}(t)\right) = f\left(\hat{y}(t-1)\right)$. A default initial value of $\rho(0) = 1.0$ was empirically found by Bergh to produce acceptable results. It is also recommended to choose f_c and s_c such that $f_c < s_c$, which means that the algorithm is quicker to punish a poor ρ setting than it is to reward a successful ρ value. The following additional rules are defined to ensure the μ adaptation equation is well-defined:

$$\#successes(t+1) > \#successes(t) \Longrightarrow \#failures(t+1) = 0 \qquad (\mathbf{1.32})$$

and

$$\#failures(t+1) > \#failures(t) \Longrightarrow \#successes(t+1) = 0 \qquad (\mathbf{1.33})$$

Thus, whenever there is success event the failure count is reset to zero, and likewise for the success count. The PSO algorithm using updating equation for its global best particle's position is called the *Guaranteed Convergence Particle Swarm Optimizer* (GCPSO).

1.6. Conclusion

In this chapter we described and discussed one of the successful stochastic computational models, Particle Swarm Optimization, inspired by the collective foraging behavior of a group of simple agents, like a school of fish, a flock of birds, and etc. An appealing aspect of the biological system is that it, as a whole, rests on an organized behavior aimed at achieving a particular goal through countless interaction and self-organization with simple rules of thumb.

In Section 1.2, we presented some fundamental issues related to the Swarm Intelligence. Swarm Intelligence is an innovative distributed intelligent paradigm for solving optimization problems and constitutes the following three characteristics: *flexibility, robustness,* and *self-organization.* By flexibility, we mean that the swarm is able to react and adapt to a changing environment in which it operates. The swarm is robust because it can accomplish its goals even if some particles in it fail to perform their duties. And the last but not least, activities are not centrally controlled, agents operate under no supervision.

Particle swarm optimization (PSO) algorithm is a population-based optimization method which imitates the behaviors observed in a swarm of insects, a flock of birds, or a school of fish, and even human social behavior through the conventional notation—a "swarm of particles" (Section 1.3). Particle Swarm behavior is displayed by particles in multidimensional space that have two characteristics: a *position* and a *velocity*. The behavior of the population is supposed to be aimed at optimizing some usual activities like searching food. Each particle in the swarm uses not only the information about its velocity or position, but also the information on the swarm, i.e. the average direction and velocity taken by the other. By doing so, any individual benefits not only from its discoveries, but from what the other have found as well, therefore, heading towards the best region discovered by the group.

In the very same section we discussed about heuristics techniques (which include PSO) in optimization. With heuristic algorithms (from Greek *heuriskein* "to find") we can build solving methods whose outcome may not necessarily be optimal, but still be consistent with the optimal solution. Metaheuristics, a subfield of general heuristic strategies, includes stochastic components to utilize randomized search abilities. In real world applications, the widespread use of heuristic algorithms, as an approximation for the optimal solutions, instead of ordinary mathematical tools can be mainly explained due to a number of factors, such as, 1) the optimization problem does not have a closed-form or analytical solution; 2) too costly (in terms of time) and cumbersome to compute.

Later, we tackled with the main issues in PSO, its original formulation, operational mechanisms and some extensions to the original PSO, and analyzed their convergent properties which, in turn, act as the stepping-stone in the application to optimization problems. We concluded the chapter by providing basic idea of Guaranteed Convergence PSO which will be used in the 3rd chapter.

CHAPTER 2 Portfolio Selection Problem

In this chapter we will provide fundamental information related to the portfolio selection problem. The early versions of risk measures, such as variance and VaR are discussed; their shortcomings in terms of properly measuring risk are analyzed. Later we will talk about coherent risk measures as possible alternatives to traditional risk measures, their properties with some examples. New two-sided risk measure is presented, and a portfolio selection problem with this risk measure as objective function is introduced.

2.1. Introduction

All large fund management companies in the world are engaged in the investment of several billions of money. This wealth is invested on behalf of pension funds, mutual funds, insurance companies, other financial and nonfinancial institutions, and many affluent people. Thus, selection of an optimal portfolio of assets in which to invest is a crucial task of the fund management. "Whether the portfolio is designed to fund a firm's pension plan, an insurer's loss reserve, or a corporation's financial planning needs, the selection process should incorporate the complexities inherent in financial markets and the decision maker's risk preferences and portfolio requirements"[24].

One of the major developments in the portfolio theory during the past few decades has been the recognition that the formation of an optimum investment portfolio is not basically an issue of pooling many individual assets that have desirable risk-return characteristics. In particular, the decision to hold an asset should not be made simply by comparing its expected return and variance to those of the other assets. Early scholars[25] have shown that one must consider the relationship among the assets if one is planning to create an optimal portfolio that

[24] Puelz A. V., A Stochastic Convergence Model for Portfolio Selection. Operations Research, Vol. 50, No. 3 (May - Jun., 2002), pp. 462-476

[25] Markowitz, Miller, Tobin, Sharpe, Lintner and so on.

will meet one's investment objectives. They argue that the decision to hold any asset would depend on what other assets the investor would like to hold.

2.2. Markowitz model

The portfolio selection problem is concerned with the selection of a portfolio of assets that provides the investor a minimum risk for a pre-specified expected return such that his/her (expected) utility is maximized. One of the first major contributions in this problem was set forth by Markowitz[26] and was described in detail later in his book in 1959[27]. He introduced the so-called mean-variance model, which considers the variance of the portfolio as the measure of investor's risk under a reasonable set of assumptions. According to him, the portfolio selection problem can be expressed as an optimization problem over real-valued variables with a quadratic objective function subject to some linear constraints.

Following Markowitz, the investment horizon includes a single period whereas the investment strategy is to select an optimal portfolio at the beginning of the period which will be held unchanged to the terminal date. Joint distribution of one-period security returns is assumed to be multivariate normal from which it then follows that the distribution of the portfolio return is normal as well. We are given a set of N assets $S = (S_1, S_2, \dots, S_N)$ where each asset S_i has a rate of return represented by a random variable R_i with a real-valued expected return r_i, and each pair of assets (S_i, S_j) has a real-valued covariance σ_{ij}. The variance-covariance matrix $\sigma_{n\times n}$ is symmetric and the diagonal elements σ_{ii} represent the variance of assets S_i. A positive value π stands for the investor desired expected return. Generally, the values r_i and σ_{ij} are estimated from past data and are fixed during the period of investment.

A portfolio is a real valued vector $X = (x_1, x_2, \dots, x_N)$ such that each of the decision variables x_i characterizes the fraction invested in the asset S_i. The value $\sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \sigma_{ij}$ represents the variance of the portfolio, and is considered as the measure of the risk associated

[26] Markowitz H., "Portfolio Selection," Journal of Finance 7, no. 1 (March 1952): 77–91.

[27] Markowitz H., Portfolio Selection—Efficient Diversification of Investments (New York: John Wiley & Sons, 1959)

with the portfolio. Then, using the standard mean-variance approach we have the constrained portfolio optimization problem:

Minimize

$$\sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \sigma_{ij} \qquad (\mathbf{2.1})$$

s.t.

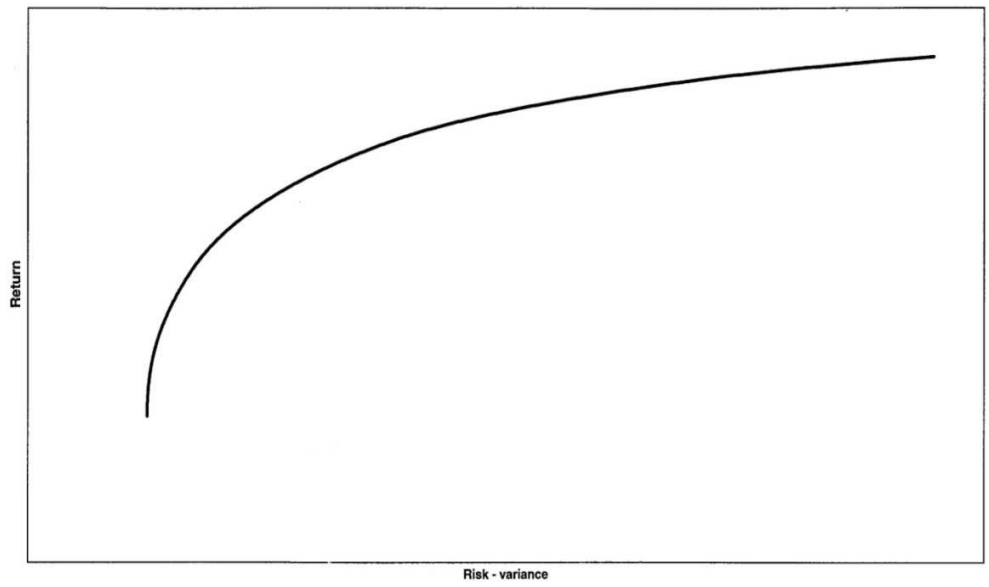
$$\sum_{i=1}^{N} x_i r_i = \pi \qquad (\mathbf{2.1a})$$

$$\sum_{i=1}^{N} x_i = 1 \qquad (\mathbf{2.1b})$$

$$x_i \geq 0, \quad i \in \{1, 2, \dots, N\} \qquad (\mathbf{2.1c})$$

This is a quadratic programming (QP) problem, and nowadays many efficient algorithms can be used to solve it. So, in practice, there is little difficulty in calculating the optimal solution for a particular data set. The objective function minimizes the total variance (risk) associated with the portfolio whilst equational constraints ensure that the portfolio has an expected return of π and the proportions sum up to one. The non-negativity constraint simply means that no short-selling is allowed in this problem.

As we said in the above, Markowitz was the first to set up a quantitative framework for the selection of a portfolio. In this framework it is presumed that investor has quadratic utility function or asset returns follow a multivariate normal distribution which, in the latter, implies that the rate of return of a portfolio of assets can be completely explained by expected return and variance. For a particular space of assets, the set of portfolios of assets that provide the minimum risk for a specified level of return produce the so-called efficient frontier. By solving the above QP for varying values of rate of return we can draw the efficient frontier, a smooth non-decreasing curve that represents the best possible trade-off of risk against return, i.e. the curve is the set of Pareto-optimal portfolios.



Graph 2.1. An example of Efficient Frontier without a risk-free asset. Each point on the line represents a portfolio of assets which is considered to be efficient (i.e., no other portfolio of assets provides higher expected return with the same (or lower) risk, or lower risk with the same (or higher) expected return.)

Unfortunately, the underlying assumptions of the M-V model are subject to serious criticisms:

- "A measure of dispersion can be adopted as a measure of risk only if the relevant distribution is symmetric".[28]
- The multivariate normality assumption is not maintained in empirical tests. The distribution of individual asset returns has a tendency to show a higher probability of being fat-tailed than what is consistent with normality. This departure from multivariate normality means that distribution moments higher than the first two moments need to be taken into account to fully express portfolio behavior.
- "In the case of non-normal, albeit symmetric, distributions the utility functions must be quadratic."[29]

[28] G. Szegö, Measures of risk. European Journal of Operational Research 163 (2005) 5–19.
[29] The same article.

We should note that mean-variance (M-V) criteria must be applied with caution since portfolios of assets follow non-normal return distributions in real world situations[30]. M-V approach can be appropriate only if the investor's utility function is quadratic with nonpositive second derivatives (i.e. the investor is risk averse) or if the asset's return distribution can be fully described by the first two moments, that is, the probabilistic distribution functions are from some two-parameter family (i.e. a family of elliptic distributions[31]). Either assumption is sufficient to make the M-V approach consistent with Von Neumann-Morgenstern utility maximization[32]. Nevertheless, several researchers have indicated that the quadratic utility function implies that beyond some return level, marginal utility of the decision maker for wealth becomes negative[33,34].

2.3. Stochastic Dominance

One of the essential results of these analyses of the M-V approach has been the appearance of the stochastic dominance portfolio choice criteria. Stochastic dominance (SD) is a portfolio selection tool that employs the entire probability distribution of returns rather than a finite number of moments. Stochastic dominance criteria, when implemented to a set of different portfolios, partitions the set into efficient and inefficient subsets such that each portfolio in the inefficient subset is dominated by at least one portfolio in the efficient subset and no portfolio of the efficient subset is dominated by any other portfolio.

Although the conceptual superiority of SD over mean-variance is clear, its practical use requires more rigorous theoretical instruments based on some robust assumptions. First, the investor must explicitly take into account the probability distribution of future values rather than first two moments. Second, all stochastic dominance criteria require some restrictions on

[30] Billio M., Simulation Based Methods for Financial Time Series., Working Paper., GRETA (March 2002).

[31] "(**Elliptical distribution**) Let X be a d-dimensional random vector. X is said to be 'elliptically distributed' (or simply 'elliptical') if and only if there exist a vector $\mu \in R^d$, a positive semi definite matrix $\Sigma \in R^{d\times d}$, and a function $\phi : R_+ \to R$ such that the characteristic function $t \to \varphi_{X-\mu}(t)$ of $X - \mu$ corresponds to $t \to \phi(t'\Sigma t), t \in R^d$."

[32] J. R. Booth, H. Tehranian, G. L. Trennepohl. Efficiency Analysis and Option Portfolio Selection. The Journal of Financial and Quantitative Analysis, Vol. 20, No. 4 (Dec., 1985), pp. 435-450.

[33] W. Breen, "Specific Versus General Models of Portfolio Selection." Oxford Economic Papers, vol. 20 (November 1968), pp. 361-368.

[34] Feldstein, M. S. "Mean-Variance Analysis in the Theory of Liquidity Preference and Portfolio Selection." Review of Economic Studies, vol. 36 (January 1969), pp. 5-12.

utility function. For example, First Stochastic Dominance (FSD) necessitates that the first derivative of the utility function be positive throughout; therefore, it allows risk preference, risk indifference, or risk aversion. Second Stochastic Dominance (SSD) eliminates risk preference by adding the restriction that the second derivative be everywhere nonpositive[35].

2.4. More on risk measurement

How risk should correctly be measured is a fundamental and still debated question which is continuously being discussed since the Markowitz era. Without defining the risk measure and what properties it should have in quantitative context, it may be pointless or have little value to deduce that we prefer a portfolio to the other. Markowitz suggested to replace the variance by an alternative risk measure, so-called "down-side risk measure" which only takes into account the part of loss distribution below a prior specified threshold[36].

Another research that deals with the extension of Markowitz model is due to Konno and Yamazaki[37]. They proposed that the mean absolute deviation (MAD) of portfolio returns to be taken as the risk measure. This has allowed the portfolio selection problem to be formulated and solved via linear programming methods. Like the original Markowitz model, however, MAD risk measure is subject to many criticisms. Simaan, for instance, has analyzed that the economy of time in computation from the use of MAD as the objective function is outweighed by the information loss from not using the variance-covariance matrix[38].

2.4.1. Value-at-Risk

Until 1990s, proposals for risk measures for portfolio theory were merely applications of measures found in the statistics literature. Though, some of these proposed measures do not

[35] Porter R. B. and Gaumnitz J. E., Stochastic Dominance vs. Mean-Variance Portfolio Analysis: An Empirical Evaluation., The American Economic Review, Vol. 62, No. 3 (Jun., 1972), pp. 438-446.

[36] Markowitz H. M., Portfolio Selection: Efficient Diversification of Investments, John Wiley, New York, 1959.

[37] Konno H, Yamazaki H. Mean-absolute deviation portfolio optimization model and its applications to Tokyo Stock Market. Management Science 1991;37:519-31.

[38] Simaan Y. Estimation risk in portfolio selection: the mean variance model versus the mean absolute deviation model. Management Science 1997;43:1437-46.

always take into account the range of investor attitudes towards risk. Attempts to quantify risk correctly have led to the development of risk measure to as an independent subject. A risk measure is a functional that assigns a numerical value to a random variable which is seen as a loss. We must admit that the modern theory of risk started to advance rapidly since the introduction of Value-at-Risk (VaR) by JPMorgan in 1994. VaR provides a single number that summarizes the total risk inherent in a portfolio of financial assets with the precise task of answering to the questions all financial institutions face every day: how much one can expect to lose during a specified time interval with a given probability? What percent of the value of the investment is exposed to risk? We provide theoretical background of VaR according to Rockafellar et al.[39]

For each x, the loss $f(x;y)$ is a random variable having a distribution in $\mathbb{R}$ induced by the y. We assume y is a random vector defined on a probability measure P on Υ (a Borel measure) that is independent of x. For each x, we denote by $\Psi(x;\cdot)$ on $\mathbb{R}$ the resulting distribution function for the loss $z = f(x;\ y)$, i.e.,

$$\Psi(x;\zeta) = P\{y|f(x;y) \leq \zeta\} \qquad (\mathbf{2.2})$$

where $f(x;\ y)$ is continuous in x and measurable in y, and that $E\{|f(x;y|\} < \infty$ for each $x \in X$.

Definition 2.1 (VaR). Given confidence level $\alpha \in (0,1)$, the *α-VaR* of the loss associated with a decision x is the value

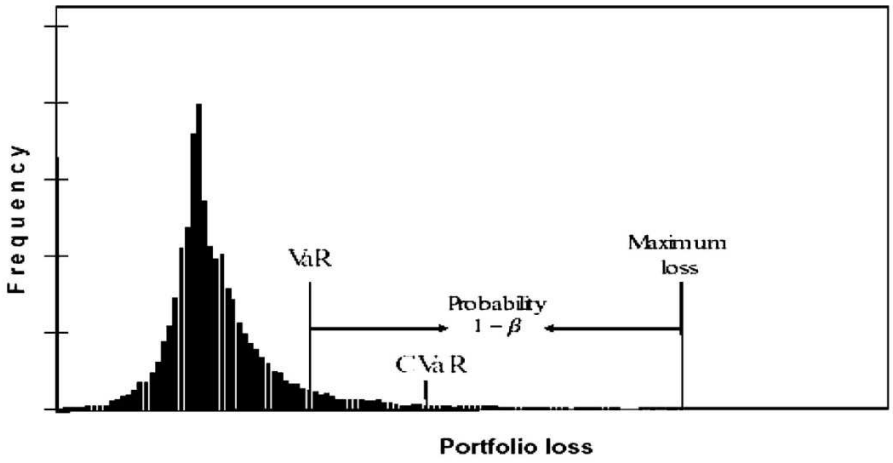
$$\zeta_\alpha = min\ \{\zeta|\Psi(x;\zeta) \geq \alpha\} \qquad (\mathbf{2.3})$$

The minimum is attained because $\Psi(x;\zeta)$ is nondecreasing and right-continuous in ζ. When $\Psi(x;\zeta)$ is continuous and strictly increasing, $\zeta_\alpha(x)$ is simply the unique ζ satisfying $\Psi(x;\zeta) = 0$. Otherwise, this equation can have no solution or a whole range of solutions.

"VaR has become broadly used by corporate treasurers and fund managers as well as by financial institutions. Central bank regulators also use VaR in determining the capital a bank is

[39] T. Rockafellar, S. Uryasev, Conditional Value-at-Risk for general loss distributions. Research report #2001-5.

required to keep to reflect the market risks it is bearing"[40]. With the encouragement of regulators, VaR has become ''solution in search of a problem'' and was incorrectly adopted[41] as a risk measure[42]. Basel Committee for Banking Supervision has failed to address many important issues about global financial supervision system through its Basel II proposals.



Graph 2.2. Graphical representation of VaR and CVaR at the confidence level $\beta \in (0,1)$. Source: G. Szegö, Measures of Risk. European Journal of Operational Research 163 (2005) 5–19.

Danielsson et al.[43] in their paper "An Academic Response to Basel II" present the following arguments versus VaR, we quote:

- "The proposed regulations fail to consider the fact that risk is endogenous. Value-at-Risk can destabilize an economy and induce crashes when they would not otherwise occur.
- Statistical models used for forecasting risk have been proven to give inconsistent and biased forecasts, notably under-estimating the joint downside risk of different assets. The Basel Committee has chosen poor quality measures of risk when better risk measures are available.
- Heavy reliance on credit rating agencies for the standard approach to credit risk is misguided as they have been shown to provide conflicting and inconsistent forecasts of

[40] J. C. Hull, Options, futures, and other derivatives, 6th edition. Prentice Hall, 2006.

[41] For example, VaR does not satisfy the diversification principle (i.e.,the aggregation of portfolios has always the effect of reducing or at most leaving unchanged the overall risk).

[42] G. Szegö, Measures of Risk. European Journal of Operational Research 163 (2005) 5–19.

[43] J. Danielsson, P. Embrechts, Ch. Goodhart, C. Keating, F. Muennich, O. Renault, H.S. Shin. An Academic Response to Basel II. Special Paper Series, LSE Financial Markets Group and ESRC Research center, May 2001.

individual clients' creditworthiness. They are unregulated and the quality of their risk estimates is largely unobservable.

- Operational risk modeling is not possible given current databases and technology even if a meaningful definition of this risk were to be provided by Basel. No convincing argument for the need of regulation in this area has yet been made.
- Financial regulation is inherently procyclical. Our view is that this set of proposals will, overall, exacerbate this tendency significantly. In so far as the purpose of financial regulation is to reduce the likelihood of systemic crisis, these proposals will actually tend to negate, not promote this useful purpose".

Indeed VaR cannot be adequate risk measure if it is implemented to non-elliptical return distributions because it does not consider any losses exceeding VaR, it may provide contradictory or paradoxical outcomes if different confidence levels are applied, and most importantly it is very difficult to apply in optimization problems since VaR is non-convex and has many local extremes leading to unstable VaR ranking[44].

2.4.2. Coherent Risk Measures

After the appearance of *Thinking Coherently*[45] by Artzner et al. followed by *Coherent Measures of Risk*[46] by the same authors, risk management has become a science with its own axioms, rules and theorems thoroughly defined in a deductive framework. The risk measure is defined through a set of axioms that involve most of the realistic behaviors of investors. And VaR, the risk measure widely adopted as best practice by mainly all banks and regulators, was shown to fail being coherent. It is not a coherent risk measure as it doesn't satisfy one of the axioms of coherence which we will see after defining what coherent measure is.

[44] G. Szegö, Measures of Risk. European Journal of Operational Research 163 (2005) 5–19.

[45] Artzner, P., Delbaen, F., Eber, J.-M., Heath, D. (1997) Thinking coherently. RISK 10 (11).

[46] Artzner, P., Delbaen, F., Eber, J.-M., Heath, D. (1999) Coherent measures of risk. Mathematical Finance. 9 (3), 203—228.

The source of risk comes from the random payoff X, where $X \geq 0$ gives random profit, defined on probability space $(\Omega, \mathcal{F}, P)$, of some portfolio of assets. We assume that X belongs to the space $L^2(\Omega, \mathcal{F}, P)$.

Definition 2.2 (Coherent risk measures). By a coherent risk measure will be meant any functional $\rho: L^2(\Omega) \to (-\infty, \infty]$ satisfying

(1) Translation invariance: $\rho(X + C) = \rho(X) - C$, for all X and constants C.

(2) Positive homogeneity: $\rho(0) = 0$, and $\rho(aX) = a\rho(X)$, for all X and all $a > o$.

(3) Subadditivity: $\rho(X + Y) \leq \rho(X) + \rho(Y)$, for all X and Y.

(4) Monotonicity: $\rho(X) \leq \rho(Y)$ when $X \geq Y$.

Property (1) can be interpreted as whenever there is a certain payoff, say C, then the risk must be reduced by the same amount. The combination of Properties (2) and (3) is sub-linearity, from where we can derive the property of convexity:

$$\begin{aligned} \rho(\alpha X + (1-\alpha)Y) &\leq \rho(\alpha X) + \rho((1-\alpha)Y) \\ &\leq \alpha\rho(X) + (1-\alpha)\rho(Y) \\ \Longrightarrow\ \rho(\alpha X + (1-\alpha)Y) &\leq \alpha\rho(X) + (1-\alpha)\rho(Y) \qquad (convexity) \end{aligned}$$

■

The last axiom simply indicates that we prefer asset X to asset Y if the relative risk of the asset X is smaller than that of Y.

Property (3) implies subadditivity which means that a merger should not generate extra risk. In other words the risk measure ρ of a portfolio consisting of assets X and Y will always be less than or equal to the sum of the risk measure of portfolio X with the risk measure of portfolio Y. VaR is not coherent because it does not fulfill this diversification principle.

Clearly, there is a need for practical and intuitive coherent risk measures. In the last ten years, there has been a great moment in research on finding several risk measures that can formulate the behaviors of investors, i.e. satisfy the coherency conditions in the sense of

Arztner et al[47]. The basic example is that in the place of VaR new coherent risk measure known as Expected Tail Loss (ETL) or Expected Shortfall (ES) has been proposed[48]. It is simple to understand in the setting of a historical-type VaR calculation, let us say α-VaR. It would entail instead of taking the $(1-\alpha)^{\text{th}}$ percentile of the loss distribution to yield a VaR number, take the average of the distribution's up to the $(1-\alpha)^{\text{th}}$ percentile to yield an ES number. We describe the definition of Expected Shortfall in mathematical settings as follows:

Definition 2.3 (Expected Shortfall) Let $\alpha \in (0,1)$ be fixed and $f(x,y)$ be a real random variable (i.e. loss distribution as we denoted above) on a probability space $(\Omega, \mathcal{F}, P)$ with $E[max\{0, f(x,y)\}] < \infty$. Define ζ_α as in Definition 2.1. We then call Expected Shortfall *(ES)* at level α as:

$$ES_\alpha(x) = -(1-\alpha)^{-1}\left(E\left[f(x;y)\mathbb{I}_{\{f(x;y)>\zeta_\alpha\}}\right] + \zeta_\alpha\{\alpha - P[f(x;y) < \zeta_\alpha]\}\right) \qquad (\mathbf{2.4})$$

This contribution was preceded by the introduction of a similar measure, Conditional Value at Risk, CVaR due to Uryasev[49]. In spite of many theoretical similarities with Expected Shortfall, the main contribution of Rockafellar et al.[50] has been to suggest a simple linear programming algorithm. It can substitute VaR by providing the same results where VaR computations are easy to tract, for example, in case, loss distribution follows normal distributions[51]. Most importantly for applications, however, CVaR has a clear advantage over VaR since it can be expressed by a minimization formula.

[47] "[...]We are of course prepared to give up this definition as soon as a new different set of axioms is proposed which is more suitable to a mathematical formulation of the concept of risk measure. What we are not prepared to do anymore, after we learned the lesson of Arztner et al., is discussing of risk measures without even defining what "risk measure" means." Acerbi, C., Tasche, D., 2002. On the coherence of expected shortfall. In: Giorgio Szegö (Ed.), Special Issue on ''Beyond VaR, 26 July. Journal of Banking and Finance. Available from <www.gloriamundi.org /var/wps.html>.

[48] Acerbi, C., Tasche, D., 2002. On the coherence of expected shortfall. In: Giorgio Szegö (Ed.), Special Issue on ''Beyond VaR, 26 July. Journal of Banking and Finance. Available from <www.gloriamundi.org/var/wps.html>.

[49] Uryasev, S., 2000. Conditional value-at-risk: optimization algorithms and applications, Financial Engineering News, 14 (Feb), 1–5.

[50] Rockafellar, R.T., Uryasev, S., 2001. Optimization of conditional value-at-risk. Presented at the Conference on Statistical and Computational Problems in Risk Management, University of Rome, La Sapienza, June14–16.

[51] Rockafellar R. T., Uryasev S., Optimization of conditional value-at-risk, Journal of Risk 2 (2000), 21-41.

The standard deviation is not a coherent risk measure[52] and considers both positive and negative variations from the mean as a potential risk, thus penalizing outperformance relative to the mean just as much as underperformance. Though that may be quite reasonable in some conditions, such as if we are trying to capture the ‘‘stability” around a ‘‘central tendency”. It is misleading if we are interested in keeping under control the overperformance and/or the underperformance. This problem could get worse if we deal with skewed and fat tailed returns[53]. Thus, in addition to the coherency properties in the sense of Arztner et al., it is also obvious that risk should be an asymmetric concept related to downside outcomes, and any pragmatic way of measuring risk should not consider upside and downside potential outcomes equally[54].

2.4.3. Two-sided Coherent Risk Measure

In this paper we consider the two-sided coherent risk measure introduced by Chen and Wang[55]. Unlike the existing coherent risk measures, both positive and negative variations from the mean are taken into account at the same time with different levels of degree. By using a different derivation scheme, they constructed a new class of two-sided coherent risk measures that can control the asymmetry and fat-tail characteristics of the profit/loss distribution and describe the investor’s risk attitude appropriately.

Below we will extensively use definitions due to Chen et al. in the study of two-sided coherent risk measure. We consider one-period model $(0, T)$ and there is no trade between time 0 and T. Thus, the source of risk comes from the random payoff X, where $X \geq 0$ gives random profit, defined on probability space $(\Omega, \mathcal{F}, P)$, of some portfolio of assets at time T. We assume that X is p-intergrable, that is, X belongs to the space $L^p(\Omega, \mathcal{F}, P)$ for $1 \leq p \leq \infty$.

[52] The property of translation invariance does not hold in standard deviation.

[53] Farinelli S., Ferreira M., Rossello D., Thoeny M., Tibiletti L. Beyond Sharpe ratio: Optimal asset allocation using different performance ratios Journal of Banking & Finance 32 (2008) 2057–2063.

[54] Rachev S., S. Ortobelli, Stoyanov S., Fabozzi F.J, Biglova A. Desirable properties of an ideal risk measure in portfolio theory. International Journal of Theoretical and Applied Finance Vol. 11, No. 1 (2008) 19–54

[55] Chen Z. Wang Y., Two-sided coherent risk measures and their application in realistic portfolio optimization. Journal of Banking and Finance (2008) 2667-2673.

Definition 2.4 Given $p \in [1, \infty)$, $0 \leq a \leq 1$, the two-sided risk measure $\rho_{a,p}: L^p(Q) \to \mathbb{R}$ is determined by

$$\rho_{a,p}(X) = a \left\| \left(X - E_Q(X)\right)^+ \right\|_1 + (1-a) \left\| \left(X - E_Q(X)\right)^- \right\|_p - E_Q(X) \qquad (\mathbf{2.5})$$

where as usual we define $\|X\|_p = \left(E_Q|X|^p\right)^{1/p}$, $X^- = \max(-X, 0)$ and $X^+ = \max(X, 0)$.

We see that $\rho_{a,p}(X)$ is generated by a convex combination of "upside" random variable $\left(X - E_Q(X)\right)^+$ and "downside" random variable $\left(X - E_Q(X)\right)^-$. Since the norm of a random variable is non-decreasing function with respect to the degree of norm, the higher norm of the random variable $\left(X - E_Q(X)\right)^-$ indicates that investors are more concerned with the expected losses below a threshold. Expected value with negative sign ensures the coherency property of the risk measure.

Proposition 2.1 Two sided risk measure $\rho_{a,p}(R)$ defined as below is coherent in the sense of Artzner et al.

$$\rho_{a,p}(R) = a\|(R - E[X])^+\|_1 + (1-a)\|(R - E[X])^-\|_p - E[X]$$

Proof:

A risk is called coherent if it satisfies the following properties:

(1) Translation invariance: $\rho(X + C) = \rho(X) - C$, for all X and constants C.

(2) Subadditivity: $\rho(X + Y) \leq \rho(X) + \rho(Y)$, for all X and Y.

(3) Positive homogeneity: $\rho(0) = 0$, and $\rho(\lambda X) = \lambda\rho(X)$, for all X and all $\lambda > 0$.

(4) Monotonicity: $\rho(X) \leq \rho(Y)$ when $X \geq Y$.

Using the relationship $\left(X - E(X)\right)^+ = \left(X - E(X)\right)^-$

$$\begin{aligned}\rho_{a,p}(R) &= a\|(R - E[X])^+\|_1 + (1-a)\|(R - E[X])^-\|_p - E[X] \\ &= a(\|(R - E[X])^+\|_1 - E[X]) + (1-a)(\|(R - E[X])^-\|_p - E[X])\end{aligned}$$

$$= a(\|(R - E[X])^-\|_1 - E[X]) + (1 - a)(\|(R - E[X])^-\|_p - E[X])$$

Since convex combination of coherent risk measures is also coherent, it is sufficient to show that $\|(R - E[X])^-\|_p - E[X]$ is coherent risk measure for $p \in [1, \infty)$.

(1) Translation invariance

$$\rho(X + C) = \|(X + C - E[X + C])^-\|_p - E[X + C]$$
$$= \|(X - E[X])^-\|_p - E[X] - C$$

■

(2) Subadditivity

We use the fact that $a^- = \frac{1}{2}(|a| - a)$ and $|a + b| \leq |a| + |b|$:

$$(a + b)^- = \frac{1}{2}\big(|a + b| - (a + b)\big) \leq \frac{1}{2}(|a| + |b| - a - b) = a^- + b^-$$

$$\Longrightarrow \quad \rho(X + Y) = \|(X + Y - E[X + Y])^-\|_p - E[X + Y]$$
$$= \big\|\big((X - E[X]) + (Y - E[Y])\big)^-\big\|_p - E[X] - E[Y]$$
$$\leq \|(X - E[X])^-\|_p + \|(Y - E[Y])^-\|_p - E[X] - E[Y]$$
$$= \rho(X) + \rho(Y)$$

■

(3) Positive homogeneity

(i) $\rho(0) = 0$ is trivial.

(ii) $$\rho(\lambda X) = \big\|\big((\lambda X - E[\lambda X])\big)^-\big\|_p - E[\lambda X]$$
$$= \|\lambda(X - E[X])^-\|_p - \lambda E[X]$$
$$= \lambda\|(X - E[X])^-\|_p - \lambda E[X]$$
$$= \lambda\rho(X)$$

■

(4) Monotonicity

$$X \geq Y \implies \exists\, Z \geq 0 \text{ such that } X = Y + Z$$

$$\implies \rho(X) = \rho(Y + Z) \leq \rho(Y) + \rho(Z) \qquad (*)$$

where $\rho(X) = \|(Z - E[Z])^{-}\|_p - E[Z]$

$$= \begin{cases} \|-Z + E[Z]\|_p - E[Z] & \text{if } Z \leq E[Z] \\ -E[Z] & \text{else} \end{cases}$$

We know that $0 \leq E[Z] - Z \leq E[Z]$

$$\implies \|E[Z] - Z\|_p \leq \|E[Z]\|_p = E[Z]$$

So, $\rho(Z) \leq 0$ for all $Z \geq 0$

and $(*) \implies \rho(X) \leq \rho(Y) + \rho(Z) \leq \rho(Y)$

■

Q.E.D.

The two-sided risk measure comprises two parameters p and a, which can be adjusted to describe various attitudes of investors towards risk. "a is a 'global' factor linearly adjusting the balance between good volatility and bad volatility"[56]. On the contrary, p is a "local" factor through which we can nonlinearly control the level of risk-aversion, and skewness or leptokurtosis of the loss distribution can be handled by choosing suitable p. Thus, for every investor there may be different p and a depending on his/her attitude towards risk.

In the same paper Chen et al. investigated the new risk measure applying it in real portfolio selection problems. They used the risk measures $\rho_{0.5,1}$, $\rho_{0.5,2}$, $\rho_{0.5,5}$ and $CVaR_{0.05}$ as the

[56] Chen Z. Wang Y., Two-sided coherent risk measures and their application in realistic portfolio optimization. Journal of Banking and Finance (2008) 2667-2673.

objective function, and compared the performance measures[57] of each others. The analyses showed that the performance of the optimal portfolio obtained under the two-sided risk measure is better than that under CVaR. They have also discovered that the larger is the p, the more significant is the performance improvement. "There are two reasons for this result: one is that, while keeping the advantages of CVaR, our two-sided risk measure considers both below-mean losses and above mean earnings simultaneously but differently; the other is that the parameter p can be used to control the fat-tail phenomenon and to flexibly reflect the degree of the investor's risk aversion"[58].

The two-sided risk measure has several advantages over other existing risk measures. First, it is coherent in the sense of Arztner et al, i.e. it can improve problems inherent in VaR and other earlier risk measures. Second, it considers different orders of moments of non-normally distributed losses in order to reflect it comprehensively while ES or CVaR risk measures are computed merely as the linear probability weighted combination of losses beyond VaR[59]. Third, while being coherent, it also belongs to the family of deviation measures in the sense of Rockafellar et al[60], i.e. we are able to measure the level of uncertainty inherent in the losses. Fourth, unlike other coherent risk measures, two-sided risk measure can fully exhibit risk attitudes of both protection sellers and protection buyers simultaneously[61].

2.5. Portfolio Selection Problem with two-sided risk measure

Consider the following model for portfolio selection. Suppose we have N assets to choose from, and, for $i = 1, \dots, N$, let $x_i \in \mathbb{R}$ be the weight of asset i in the portfolio. Moreover, for $i = 1, \dots, N$ let r_i be a real valued random variable that represents the return of asset i, and $\hat{r}_i$ denotes its expected value, that is $\hat{r}_i = E[x_i]$. Then the random variable that

[57] Performance measure is the ratio of rate of return over level of risk (return/risk).

[58] Chen Z. Wang Y., Two-sided coherent risk measures and their application in realistic portfolio optimization. Journal of Banking and Finance (2008) 2667-2673.

[59] "[…] it should be better to consider the probability weighted combination of higher order losses below some critical value." Chen Z. Wang Y., Two-sided coherent risk measures and their application in realistic portfolio optimization. Journal of Banking and Finance (2008) 2667-2673.

[60] Rockafellar R.T., Uryasev S, Zabarankin M. , Generalized deviations in risk analysis Finance and Stochastics. 10, 51–74 (2006)

[61] Chen Z. Wang Y., Two-sided coherent risk measures and their application in realistic portfolio optimization. Journal of Banking and Finance (2008) 2667-2673.

represents the return of the portfolio can be expressed as $R = \sum_{i=1}^{N} x_i r_i$ and its expected value is $\hat{R} = \sum_{i=1}^{N} x_i \hat{r}_i$

As a measure of risk we consider two-sided risk measure:

$$\rho_{a,p}(R) = a \left\| \left(R - \hat{R}\right)^{+} \right\|_1 + (1-a) \left\| \left(R - \hat{R}\right)^{-} \right\|_p - \hat{R} \qquad (\mathbf{2.6})$$

Portfolio selection problem is

$$\min_{x_i} \rho_{a,p}(R) \qquad (\mathbf{2.6a})$$

s.t.

$$\hat{R} = \pi \qquad (\mathbf{2.6b})$$

$$\sum_{i=1}^{N} x_i = 1 \qquad (\mathbf{2.6c})$$

$$d_i \leq x_i \leq u_i \qquad (\mathbf{2.6d})$$

We have to select the weights for each asset such that the expected return of our portfolio should be equal to some threshold π, which is, in Economics term, the opportunity cost of capital from not investing in analogous businesses. d_i and u_i are lower and upper values of the boundary constraints, respectively. So, we may interpret as the weight of each asset in the portfolio should be within the boundaries in order to reduce transaction costs and obtain a well-diversified portfolio. Provided that d_i is positive, we also assume that no short selling is possible. And, obviously, budget constraint must also be satisfied since, besides being rational, we cannot form a portfolio with the initial value more than what we can afford today (neither short-selling nor borrowing is allowed).

2.6. Conclusion

In this chapter we provided fundamental information related to the portfolio selection problem. We discussed about significant assumptions underlying Markowitz theory and presented a portfolio selection problem with respect to the variance risk measure. In the Markowitz framework it is presumed that investor has quadratic utility function or asset returns follow a multivariate normal distribution. We argued that mean-variance criteria must be applied with caution since this approach can be appropriate only if the investor's utility function is quadratic with nonpositive second derivatives or if the asset's return distribution can be fully described by the first two moments.

Attempts to quantify risk correctly have led to the development of risk measure to as an independent subject. We talked about various other risk measures, such as Value-at-Risk and the family of coherent risk measures. In particular, we emphasized on the shortcomings of VaR in terms of properly measuring risk. VaR cannot be adequate risk measure if it is implemented to non-elliptical return distributions because it does not consider any losses exceeding VaR, it may provide contradictory or paradoxical outcomes if different confidence levels are applied, and most importantly it is very difficult to apply in optimization problems since VaR is non-convex and has many local extremes leading to unstable VaR ranking[62].

Therefore, we later told about coherent risk measures as possible alternatives to traditional risk measures, and their properties with some examples. We presented new two-sided risk measure introduced by Chen et al. which, unlike the existing coherent risk measures, takes into account both positive and negative variations from the mean at the same time with different levels of degree. In the end, a constrained portfolio selection problem with this risk measure as objective function is introduced for the application of PSO.

[62] G. Szegö, Measures of Risk. European Journal of Operational Research 163 (2005) 5–19.

CHAPTER 3
Application of Particle Swarm Optimization in Portfolio Selection Problems

In this chapter we will provide the necessary experiments and results concerning the application of the Particle Swarm Optimization in the portfolio selection problem described in the previous chapter. We will briefly describe fitness function, an equivalent unconstrained version of the optimization problem, all parameters used in both PSO and fitness function and their possible values, and conduct tests on them with MATLAB-developed software. In the end, by using PSO we will compute and plot efficient-frontier of portfolios with risky assets according to the portfolio selection problem based on the risk measure introduced by Chen et al.

3.1. Introduction

Portfolio selection is frequently seen problem in finance and economics. Although basic formulation of it can be solved through mathematic programming such as linear and/or quadratic programming, more practical and realistic versions, which may involve many different kinds of constraints, have in most occasions to be tackled by approximate algorithms.

In Chapter 1 we have presented and described the Particle Swarm algorithm, highlighting its nature and biological origin inspired by flocks of birds and schools of fish. We have tackled with the main issues in PSO, its original formulation, operational mechanisms and some extensions to the original PSO, and analyzed their convergent properties.

In chapter 2 we have provided fundamental information related to the theory of portfolio management. Further, we have seen and analyzed the early versions of risk measures, such as variance and VaR, their shortcomings in terms of properly measuring the risk. We have made our emphasis on the family of so-called coherent risk measures as possible alternatives to traditional risk measures, and discussed about their properties with some examples including

the new two-sided risk measure presented by Chen et al. In the end we have introduced a portfolio minimization problem with this risk measure which will be used in the application of PSO in this chapter.

Particle swarm model, ones being only a curiosity, now has attracted the interests of many researchers around the globe. "Particle swarm optimization has undergone many changes since its introduction in 1995. As researchers have learned about the technique, they have derived new versions, developed new applications, and published theoretical studies of the effects of the various parameters and aspects of the algorithm."[63] PSO can be and has been implemented in a wide range of applications. PSOs have exhibited particular promise in the areas that involve multimodal problems and problems for which there is no solutions method available or all solution methods give unsatisfactory results[64].

One of the systematic analyses in the application of PSO is due to Bergh (2001), who investigated the performance of various PSO-based algorithms using several benchmark functions. The following functions[65] were used to test the different versions of PSO algorithm (in particular, original PSO and Guaranteed Convergent PSO (GCPSO) algorithms) with the various parameter values, thoroughly explained in Chapter 1:

(i) **Spherical:**

$$f(x) = \sum_{i=1}^{n} x_i^2 \qquad (\mathbf{3.1})$$

(ii) **Rosenbrock:**

$$f(x) = \sum_{i=1}^{n/2} (100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2) \qquad (\mathbf{3.2})$$

(iii) **Quadratic:**

$$f(x) = \sum_{i=0}^{n} \left(\sum_{j=0}^{i} x_j \right)^2 \qquad (\mathbf{3.3})$$

[63] Poli R., Kennedy J., Balckwell T., Particle swarm optimization—An overview., Swarm Intelligence (2007) 1:33-57.

[64] The same article.

[65] There were actually seven functions to be optimized: Spherical, Rosenbrock, Ackley, Rastrigin, Griewank, Schwefel, and Quadratic. Since other functions involved trigonometric values, we have decided to omit them.

He argued that even though these functions might not necessarily provide an accurate indication of the performance of an algorithm on real-world problems, they could be employed to discover particular characteristics of the algorithms under consideration. The particles were initialized so that they were distributed throughout $(-x_{max}, x_{max})$ using a uniform random number generator. The acceleration coefficients were set $c_1 = c_2 = 1.49$, and the inertia weight w of the PSO was set, so that the inequality $w > \frac{1}{2}(c_1 + c_2) - 1$ holds[66]. These values, in theory, should lead to convergent trajectories. For each parameter settings the PSO algorithms were run 50 times and the results were shown by the mean values with its corresponding standard deviation. Bergh showed that generally the performance of the GCPSO was more consistent than the performance of the original PSO, indicating that it was less sensitive to the choice of parameters. Throughout the chapter we will employ these parameter settings of the PSO to our Risk Measure function according to Bergh.

We can see similar works in Parsopoulos et al[67]. and Hu et al[68]., though unlike Bergh, they solved Constrained Optimization problems by using PSO. But, all share the same inefficiency when real-world problems are considered. They had to run the algorithm several times in order to get possible results, and the solution set was given by a confidence-like interval—mean plus and minus standard deviation. Usually, the final solution was said to be the best solution, say, in 50 experiments in terms of minimum fitness value. Obviously, some questions will appear: *how many times should experiments be carried out?* Or: *is it possible to get a unique solution instead of 50 solutions which is at least as good as the best solution from the experiments?*

Trying to find a possible answer for these questions, we introduce an extended version of PSO or shortly the extended PSO—an algorithm that works not only with particles surfing the search space, but also with the solution vector found in each experiment. In all the works

[66] In chapter 1 it was shown that particular choices of c_1, c_2 and w ensures the swarm to have a convergent trajectory. Recall that the trajectory is convergent if $\max(\| \alpha \|, \| \beta \|) < 1$.

[67] Parsopoulos K.E, Vrahatis M.N., Particle Swarm Optimization Method for Constrained Optimization Problems. Department of Mathematics, University of Patras Artificial Intelligence Research Center (UPAIRC), Greece.

[68] Hu X., Eberhart R., Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization. Working paper. Department of Electrical and Computer Engineering, Indiana University Purdue University at Indianapolis, IN, USA.

mentioned above, the results from the experiments were used to construct the confidence-like interval of possible solutions for the optimization problem and were partially disregarded in the sense that they can be used as particles in other experiments of PSO to find even better solutions. Indeed, since a solution vector for the optimization problem found in each experiment gives the least fitness value in those subsets of the search space where the particles have explored during the experiment, it contains more information than any particle initialized all over again in other PSO algorithm runs.

In this extended PSO, we claim that only "best" particles must be used to generate new global best position in the search space. Here, by saying the word "best" we mean the particle whose personal best position has less fitness value than those of many possible particles in the search space. The outcome from each experiment (i.e. a single run of ordinary PSO to find a minimum in the optimization problem) is suitable or can be seen as "best" particle in our extended PSO, because each of the outcomes, at least, can be assumed as local minimum , if not the global minimum, for the optimization problem.

The algorithm for the extended PSO is simple and, in this work, we restrain ourselves by giving only informal presentation of it. In the first stage, we run ordinary PSO K times ($K > N$), where N is the number of particles in the PSO, in order to get K solutions which Bergh and others[69] call the experiment results. Then we select N of them that give less fitness values with respect to the objective function. In the second stage, we run the PSO N times with these N particles to generate again other N "best" particles, and so on. We have discovered that all of N "best" particles found in the following stages converge to a single particle (normally 6-7 stages). This single particle is the solution of our extended PSO, and by the construction of the algorithm it is trivial to detect that it produces a result that is better than results which can be found by using the method according to Bergh (2001) and others[70].

This feature reminds us the selection operator in Genetic Algorithm[71]—only fitted parents have right to produce an offspring. Therefore, we may assume that this extended PSO

[69] Parsopoulos et al. and Hu et al.

[70] Parsopoulos et al. and Hu et al.

[71] "[…] A **genetic algorithm (GA)** is a search technique used in computing to find exact or approximate solutions to optimization and search problems. GAs are a particular class of evolutionary algorithms (EA) that use techniques inspired by evolutionary biology such as *inheritance*, *mutation*, *selection*, and *crossover*. GAs are implemented in a computer simulation in which a population of abstract representations (called

is a sort of hybridized PSO—a PSO algorithm with the characteristic of Genetic Algorithm. The pseudo-MATLAB code merging all processes into one run is provided in the following sections.

3.2. The PSO parameter settings

As known, very little research has been done concerning the application of PSO to the constrained optimization, particularly, in portfolio selection problems. Kendal et al.[72] who were one of the first in implementing the Particle Swarm Solver in the construction of optimal risky portfolios. They used Markowitz model with various constraints, however, they have proposed neither thorough analysis nor reference regarding the equality constraints in their work. A thorough analysis is due to Barro[73], who observed and described the behavior of PSO algorithm in Markowitz portfolio selection problems. Unlike Kendal et al., Barro has carried out a systematic analysis of handling constraints, which is quite similar with the methods we use in later sections.

The PSO algorithm will be implemented in MATLAB-R2007b software, using personal computer HP Pavilion dv6000 (Intel (R) Core (TM) 2 Duo CPU T7300@2.00GHz, 2.00 GB RAM).

The search space in the consideration is D-dimensional where D is the number of assets in the portfolio. Obviously, if we consider the institutional constraints in the portfolio selection problem, then the search space we are interested in can be reduced to

---

chromosomes or the *genotype* of the *genome*) of candidate solutions(called *individuals*, *creatures*, or *phenotypes*) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. During each successive generation, a proportion of the existing population is selected to breed a new generation… Individual solutions are selected through a *fitness-based* process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions." Source: www.wikipedia.org .

[72] Kendal G., Su Y., 2005, "A particle swarm optimization approach in the construction of optimal risky portfolios", School of Computer Science and IT, University of Nottingham, UK.

[73] Barro M., Tecniche Particle Swarm Optimization per la selezione di portafoglio., Master Thesis, Department of Economics., University of Ca Foscari, Italy,2007- 2008.

$$(d_1, u_1) \otimes (d_2, u_2) \otimes \dots \otimes (d_D, u_D) = \prod_{i=1}^{D} (d_i, u_i) \qquad (\mathbf{3.4.})$$

where, by $\otimes$ stands for Cartesian product of sets.

This space is not "restricted" for particles in the PSO, that is , a particle can pass across the boundaries, but, for doing so, it will be penalized, which is thoroughly explained in the following sections. The points in the search space represent possible portfolio choices, and each point is called a "*particle*" in the context of PSO. Each element of a particle corresponds to the *weight* of an asset in the portfolio.

The subspace defined as the intersection of the search space and the hyperplanes obtained from the usual equality constraints of the portfolio selection problem is called the *space of feasible portfolios*. So, our goal is to find a feasible portfolio with the minimum risk through the application of PSO in the optimization problem.

In this thesis we would like to apply gbest model, and the update equations for the particles are:

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_{1,j}(t)\left[y_{i,j}(t) - x_{i,j}(t)\right] + c_2 r_{2,j}(t)\left[\hat{y}_j(t) - x_{i,j}(t)\right] \qquad (\mathbf{3.5}\boldsymbol{a})$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \qquad (\mathbf{3.5}\boldsymbol{b})$$

where $y_{i,j}(t)$ is the j^{th} element of the personal best position of particle i, and $\hat{\boldsymbol{y}}$ is the best position discovered by any of the particles so far.

Hence, each particle moves in the search space of possible solutions with a dynamically changing velocity which is continuously updated according to the personal best position achieved by the particle and the global best position achieved so far by all particles at each iteration. It receives information about the gbest and transmits information about its position if it finds new global best position in the search space. Each particle, therefore, persistently communicates and compares its performance with those of the others, and moves towards the

average direction of its pbest and the gbest (i.e. the position that gives the minimum fitness value found so far).

We have not found any literature providing the exact values of the inertia weight w and the acceleration coefficients c_1 and c_2. However, Blackwell et al.[74] suggests that the inertia weight, which controls the impact of the particles' velocities, should take values below one. Higher values (for example, $w = 0.9$) of the inertia weight permits the swarm to explore in the search space, and, conversely, lower values (for example, w = 0.4) enables the swarm to exploit the local area assumed to involve the solution to the optimization problem. Moreover, in the same article we find that acceleration coefficients c_1 and c_2, as a result of empirical findings, have shown good performance if $(c_1 + c_2) \in [0,\ 4]$[75] .

Taken into account the above givens, it appears to be appropriate and plausible to attribute to the inertia weight w, and acceleration coefficients c_1 and c_2 the following values:

(i) We set $c_1 = c_2 = 1.49$ according to Bergh (2001);

(ii) inertia weight $\boldsymbol{w}$ linearly decreases from 0.9 to 0.5 according to the following formula:

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{iteration_{max}}\right) * iteration \qquad (\mathbf{3.6})$$

Moreover, each particle's position is randomly initialized in the interval (d, u) using uniform random number generator, and following Barro (2008), the maximum number of iteration is set to 8000[76]. However, the major distinction in our analysis is the stopping criterion. According to Bergh, the stopping criterion was when the maximum iteration had reached. In our case, we stop the algorithm if the number of consecutive events defined as the

[74] Blackwell T., Kennedy J., Poli R., 2007, "Particle swarm optimization", Springer Science, Business Media, 33-57.

[75] Recall that Bergh (2001) has suggested the relationship between the inertia weight and the acceleration coefficients should satisfy the inequality $w > \frac{1}{2}(c_1 + c_2) - 1$.

[76] Different values for the number of iterations in the PSO are used in the literature ranging from 2,000 to 10,000. For example, Saravanan R., in his book "Manufacturing optimization through intelligent techniques", set the number of iterations to 2,000. Bergh (2001), on the contrary, set the number of iterations to 10,000. This issue is problem dependent, and we see below in our experiments that setting 8,000 due to Barro is appropriate choice in this thesis.

difference between two successive minimum fitness values is below some tolerance value (i.e. very close to zero), reaches some predefined number or, otherwise, if the maximum iteration has reached. We will denote this event as “exit_flag” in our MATLAB code, so if the number of consecutive “exit_flags” reaches some predefined number, computer stops evaluating current assignment and passes to the next one. Given this criterion we can rid of excess evaluation and, therefore, significantly reduce the time needed for computation.

Another important issue is how many particles should be used in the PSO. In the literature, there is no detailed information about the exact number of particles that ensures, with certainty, the convergence of PSO in the optimal point. It is problem dependent and, therefore, a number of particles which can assure location of the minimum point in one problem may not be sufficient in the other.

Hu et al[77]., however, argued that, depending on the complexity and size of the optimization problem, compromise between "reasonable" time for computation and "optimality" should be achieved. They also noted that the number of particles should be in the range between 20 and 50, above which the performance of the PSO algorithm did not seem to depend on the number of particles. Obviously, it seems reasonable to assert that a high number of particles should ensure a greater exploration in the search space or, at least, in the early stages. And, for particular stopping criterion, it may cause more rapid convergence of the PSO algorithm. In the following experiments we use various numbers of particles and consider the cases where the swarm population is composed of 20, 25 or 30 particles.

3.3. Fitness Function

Since deterministic optimization methods require strong assumptions on the continuity and differentiability of the objective function, there have been many discussions about alternative way of solving problems. Surprisingly, the stochastic solving tools are being considered to be good approximate solution tools even for constrained optimization problems,

[77] Hu S., Yu H., Zangh L., 2005, “Optimal choice of parameters for particle swarm optimization”, Journal of Zhejiang University, Dip. Chemical Engineering, Zhejiang University, China, 528-534.

although Evolutionary Computation methods (including PSO) were initially developed to deal with unconstrained optimization[78].

In the previous chapter we introduced the following model for portfolio selection. We will make a few changes in the "names" of parameters in this section. Number of securities will be denoted by D rather than N. D stands for the dimension of the particles in the search space, where each dimension indicate the weight of an asset. Letter N will be used to indicate the number of particles in the swarm. Thus, we have D assets to choose from, and, for $i = 1, \dots, D$, let $x_i \in \mathbb{R}$ be the weight of asset i in the portfolio. Moreover, for $i = 1, \dots, D$ let r_i be a real valued random variable that represents the return of asset i, and $\hat{r}_i$ denotes its expected value, that is $\hat{r}_i = E[x_i]$. Then the random variable that represents the return of the portfolio can be expressed as $R = \sum_{i=1}^{N} x_i r_i$ and its expected value is $\hat{R} = \sum_{i=1}^{N} x_i \hat{r}_i$

Using historical data (r_{it} , $t = 1, 2, \dots, T$) without any assumption on the distribution of asset returns, we compute then the two-sided coherent measure as

$$\rho_{a,p}(R) = a\left(\frac{1}{T}\sum_{t=1}^{T}\left(\sum_{i=1}^{D} x_i(r_{it} - \hat{r}_i)\right)^{+}\right) + (1-a)\left(\frac{1}{T}\sum_{t=1}^{T}\left[\left(\sum_{i=1}^{D} x_i(r_{it} - \hat{r}_i)\right)^{-}\right]^{p}\right)^{1/p} - \hat{R} \qquad (\mathbf{3.7})$$

As we discussed in the 2nd Chapter, choosing a particular values for a and p is investor-dependent in the sense that any investor can set his own values that characterize his/her attitude towards risk. In this work we set $a = 0.5$ and $p = 3$, so that we are able to characterize behavior of risk-averse investor. It was shown by Chen et al.[79] that this risk measure outperforms $CVaR_{0.05}$.

[78] Parsopoulos K.E., Vrahatis M.N., Particle Swarm Optimization Method for Constrained Optimization Problems., Department of Mathematics, University of Patras Artificial Intelligence Research Center (UPAIRC), GR-26110 Patras, Greece.

[79] Chen Z. Wang Y., Two-sided coherent risk measures and their application in realistic portfolio optimization. Journal of Banking and Finance (2008) 2667-2673.

We consider the following minimization problems:

$$\min_{x_i} \rho_{a,p}(R) \qquad (\mathbf{3.7}\boldsymbol{a})$$

s.t.

$$\hat{R} = \pi \qquad (\mathbf{3.7}\boldsymbol{b})$$

$$\sum_{i=1}^{N} x_i = 1 \qquad (\mathbf{3.7}\boldsymbol{c})$$

$$d \leq x_i \leq u \qquad (\mathbf{3.7}\boldsymbol{d})$$

We have to select the weights for each asset such that the expected return of our portfolio should be equal to some threshold π, which, for instance, can be seen as the opportunity cost of capital from not investing in analogous businesses. d and u are lower and upper values of the boundary constraints, respectively. So, we may interpret as the weight of each asset in the portfolio should be within the boundaries in order to reduce transaction costs and obtain a well-diversified portfolio. Provided that d is positive, we also assume that no short selling is possible. And, obviously, budget constraint must also be satisfied since we cannot form a portfolio with the initial value more than what we can afford today (neither short-selling nor borrowing is allowed).

The search space in any constrained optimization problem contains feasible and unfeasible sets, where any element in the feasible set satisfies all the constraints, while those of the unfeasible sets violate at least one of them. One of the most frequent approaches to solve constrained optimization problems is the use of a penalty function. Using such an approach the constrained problems can be transformed to the unconstrained one by constructing a single objective function and penalizing its addenda for any dissatisfaction of the constraints. So, the starting constrained problem is solved by minimizing an unconstrained optimization algorithm[80].

[80] Coello C. A. C., Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art., Computer Methods in Applied Mechanics and Engineering., 191 (2002) 1245–1287.

Our portfolio selection problem can then be transformed to the following unconstrained one:

$$\min_{x_i} f(x) = \min_{x_i} \rho_{a,p}(R) + \frac{1}{\varepsilon} * \left|\hat{R} - \pi\right| + \frac{1}{\varepsilon} * \left|\sum_{i=1}^{N} x_i - 1\right| + \frac{1}{\varepsilon} * \left(\sum_{i=1}^{N} \max\{0, d - x_i\} + \sum_{i=1}^{N} \max\{0, x_i - u\}\right) \quad (\mathbf{3.8})$$

where $\varepsilon > 0$ is the penalty parameter, and 1/ε is called penalty value. The penalty parameter should be small so that whenever variable x violates any of the constraints by one unit, the fitness value will be penalized by 1/ε . While using a penalty parameter, one should take into account what changes it will produce in the behaviors of particles which, in turn, are solution candidates. For high penalty values, minimization algorithms get usually trapped in local minima while low penalty values often result in unfeasible solutions[81]. We will examine the parameter settings in the penalty value in the following sections.

Proposition 3.1. Show that, for a particular choice of $\varepsilon > 0$, the solution to the unconstrained minimization problem in Equation 3.8 coincides with the solution to the constrained minimization problem in Equations $3.7a - 3.7d$.

Proof:

Let's denote the constrained optimization problem as $P_1$and the unconstrained one as P_2. Let's also denote the set of feasible portfolios (i.e. the set of points satisfying the Equations $3.7b - 3.7d$) as $\mathcal{F}$. Assume that x' is the solution to the problem P_1, then $x' \in \mathcal{F}$.

For a given value for ε, let x^* be the solution to the problem P_2, then for all $x \neq x^*$, the inequality $f(x) \geq f(x^*)$ holds.

(a) if $x^* \in \mathcal{F}$, the proof is trivial.

(b) if $x^* \in \mathcal{F}^C$, then $f(x') \geq f(x^*)$

[81]"[...] If the penalty is too high or too low, then the problem might become very difficult for an EA... If the penalty is too high and the optimum lies at the boundary of the feasible region, the EA will be pushed inside the feasible region very quickly, and will not be able to move back towards the boundary with the infeasible region. A large penalty discourages the exploration of the infeasible region since the very beginning of the search process. On the other hand, if the penalty is too low, a lot of the search time will be spent exploring the infeasible region because the penalty will be negligible with respect to the objective function. These issues are very important in EAs, because many of the problems in which they are used have their optimum lying on the boundary of the feasible region." Coello C. A. C., Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art., Computer Methods in Applied Mechanics and Engineering., 191 (2002) 1245–1287.

$$\Longrightarrow \rho\big(R(x')\big) \geq \rho\big(R(x^*)\big) + \text{"positive number"} > \rho\big(R(x^*)\big)$$

Let $m^* = \max\{|\widehat{R} - \pi|, |\sum_{i=1}^{N} x_i^* - 1|, \sum_{i=1}^{N} max\{0, d - x_i^*\}, \sum_{i=1}^{N} max\{0, x_i^* - u\}\}$ and

$k^* = \rho\big(R(x')\big) - \rho\big(R(x^*)\big)$

Since, $x^* \in \mathcal{F}^C$ we easily detect that $m^* > 0$ and $k^* > 0$. Then, for $\varepsilon < m^*/k^*$

$$\begin{aligned} f(x^*) &= \rho\big(R(x^*)\big) + \rho\big(R(x')\big) - \rho\big(R(x^*)\big) + \text{"positive number/ zero"} \\ &> \rho\big(R(x')\big) \\ &= f(x') \\ &\Longrightarrow f(x') \geq f(x^*) > f(x') \end{aligned}$$

Hence, x^* is the solution to the P_2 for penalty parameter $\varepsilon \in (0, \frac{m^*}{k^*})$, if and only if $x^* \in \mathcal{F}$. From (a) we get that $x^* = x'$.

For penalty parameters $\varepsilon \in (0, \frac{m^*}{k^*})$, let $\{x^j\}_{j \in \Im}$ be the solution set to the P_2 such that for $\forall j \in \Im$, $x^j \in \mathcal{F}^C$. For the penalty parameter $\varepsilon = \inf_{j \in \Im} \frac{m^j}{k^j}$, the solution to the P_2 coincides with the solution to the P_1.

■

3.4. Assets in the portfolio selection problem

The monthly returns[82] used in this study are evaluated from contemporaneous stock-price data for five publicly traded individual stocks from S&P500, namely, Genesco, Hormel Foods, Humana, Intel, and Koss, for the period December 31st, 1997, through December 31st, 2007 (i.e. we have data of 5 assets for 120 months). The monthly rates of return are calculated

[82] Monthly return is not preferred statistics for us, and instead we could also use daily, weekly or yearly returns equivalently.

by using logarithmic based transformation, and the monthly expected return on each asset is computed by averaging historical monthly rates of return for this particular asset. We see monthly expected rates of return for these five assets in Table 3.1.

Asset	E[R]
GENESCO	0.0091
HORMEL FOODS	0.0090
HUMANA	0.0107
INTEL	0.0041
KOSS	0.0118

Table 3.1. Monthly expected rate of returns for assets in the portfolio for period December 31st, 1997, through December 31st, 2007.

3.5. Overview of PSO Algorithm MATLAB Code

The MATLAB code for the procedures in the PSO can be described as follows:

I. Set the parameters.

II. For each particle:

1) Initialize each coordinate $x_{i,j}$ to a value drawn from the uniform random distribution on the interval $[d, u]$, for all $i \in 1, 2, \dots, N$ and $j \in 1, 2, \dots, D$. This distributes the initial positions of the particles throughout the search space;

2) Initialize each $v_{i,j}$ to a value drawn from the uniform random distribution on the interval $[kd, ku]$[83], for all $i \in 1, 2, \dots, N$ and $j \in 1, 2, \dots, D$;

3) Initialize the best fitness values found so far by a particle (pbest) and by the swarm (gbest) to "$-\infty$";

[83] "[…] If the search space is defined by the bounds $[-x_{max}, x_{max}]$, then the value of v_{max} is typically set so that $v_{max} = kx_{max}$, where $0.1 \leq k \leq 1.0$". Van den Bergh F., (2001) An analysis of particle swarm optimizers, PhD Dissertation, Dept. of Computer Science, University of Pretoria, South Africa.

III. Perform:

a. For each particle:

1) Calculate its fitness value;

2) If the fitness value is better than the best fitness value of the particle (pbest), set current value as the new pbest;

b. For each particle:

1) Find in the particle neighborhood, the particle with the best fitness value (gbest);

2) Calculate particle velocity according to the following procedure:

i. If the current velocity of a particle with the best fitness value is smaller than a specified infinitesimal number, then use velocity update equation of GCPSO for this particle and standard velocity update equation for other particles.

- Increase "rho" if number of consecutive successes (i.e., better gbest has been found) becomes greater than some value s_c;
- Decrease "rho" if the number of consecutives failures (i.e., better gbest hasn't been found) becomes greater than some value f_c.

ii. Else, use standard velocity update equation for all particles;

3) Update particle position according to the position equation;

IV. Continue (III) until maximum iterations or minimum error criteria is not attained;

V. Perform (II)-(IV) K times ($K > N$) in order to get K "solution-candidate" particles for our optimization problem;

VI. Choose N of the "solution-candidate" particles with smallest fitness values;

VII. Repeat (II)-(V) procedures S times, with the only difference in (V) such that we perform N times instead of K.

3.6. Experiments and other parameter settings

In this section we will summarize the experiments we have performed on the penalty parameter and the population of the swarm in the application of PSO. We note that the results from the experiments will only be appropriate for the particular portfolio selection problem, and may not be appropriate for other types of optimization problems.

(i) *Experiment on penalty parameter ε*: we seek to establish the most appropriate value of the parameter ε by running the PSO algorithm with various penalty parameters, with respect to the portfolio selection problem given a particular number of particles and stopping criterion.

(ii) *Experiment on the population of the swarm*: by varying the number of particles, we analyze the convergence of the extended PSO algorithm.

We assume the weight of each asset in the desired portfolio should not be less than 0.05 and should not be more than 0.35. Moreover, as will be seen in the following graphs, we tentatively set the $\text{exit\_flag} = 1000$, which implies that computer will stop evaluating current assignment and pass to the other if there is no development in finding minimum fitness value in 1000 consecutive iterations.

The parameter values for “rho”, “Sc”, “Fc” and “alpha” are consistent with the suggestion by Bergh (2001) in his Guaranteed Convergent Particle Swarm Optimization (GCPSO) described in chapter 1.

3.6.1. Experiment on penalty parameter ε

The first experiment is to find a proper value for the parameter ε which ensures that all constraints are satisfied, and our particle swarm finds the best possible optima in the search space. The value, then, will be used in further experiments on other parameters and in plotting the efficient-frontier of risk-return with respect to the risk measure chosen in this work. We

start with experimenting on penalty parameter, because we will give more weight on constraints in the sense that our final solution, whether we reach minimum risk or not, must be at least feasible. We will see that setting any small penalty parameter ε almost ensures full respect for the constraints.

The summary of other parameter values is given in the Table 3.2. Since the number of iterations is sufficiently large, we will also take into account the need to reduce the time for computation. Therefore, we consider four possible values for the parameter ε, i.e. 1, 0.1, 0.01, and 0.001 for the experiments.

Table 3.2. PSO parameters.

itermax=8000;	number of iteration
N=30;	number of particles in each iteration
wmax=0.9;	inertia weight--upper bound
wmin=0.5;	inertia weight--lower bound
c1=1.49445;	acceleration coefficient for pbest
c2=1.49445;	acceleration coefficient for gbest
D=5;	number of securities in portfolio
T=120;	120 monthly data considered
LB=0.05;	lower constraint
UB=0.35;	upper constraint
a=0.5;	risk measure factors
p=3;	
S=7;	
K=100;	number of "best candidates" for swarm particles
exit_flag = 1000;	exit loop if objective function is constant in consecutive 1000 iterations
rho=1;	value suggested number by Bergh (2001)
Sc=7;	rho=(1/alpha)*rho if consecutive number of successes is greater than Sc
Fc=5;	rho=(alpha)*rho if consecutive number of failures is greater than Fc
alpha=0.95;	

Given the expected monthly return on each of the five assets and the institutional constraints, the expected return of the portfolio varies between 0.0075 and 0.0103. For the

experiment we would like to consider these two boundary cases[84], $\pi = 0.0075$ and $\pi = 0.0103$.

For $\pi = 0.0075$, we summarize the results of each asset with its corresponding share in the portfolio, the expected return and the risk level of the portfolio, and time needed for computation at all values of ε (Table 3.3.).

pi=0.0075

epsilon =1

	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.0500	0.3500	0.0539	0.1961	0.3500	1.00
Expected Return	0.0091					
Risk level	0.0279					
Time (sec)	554.22					

epsilon =0.1

	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.0522	0.3497	0.0598	0.3468	0.1915	1.00
Expected Return	0.0079					
Risk level	0.0346					
Time (sec)	1155.85					

epsilon =0.01

	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.1993	0.3500	0.0504	0.3500	0.0504	1.00
Expected Return	0.0075					
Risk level	0.0443					
Time (sec)	1357.96					

epsilon =0.001

	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.2494	0.3002	0.0502	0.3500	0.0502	1.00
Expected Return	0.0075					
Risk level	0.0482					
Time (sec)	1405.06					

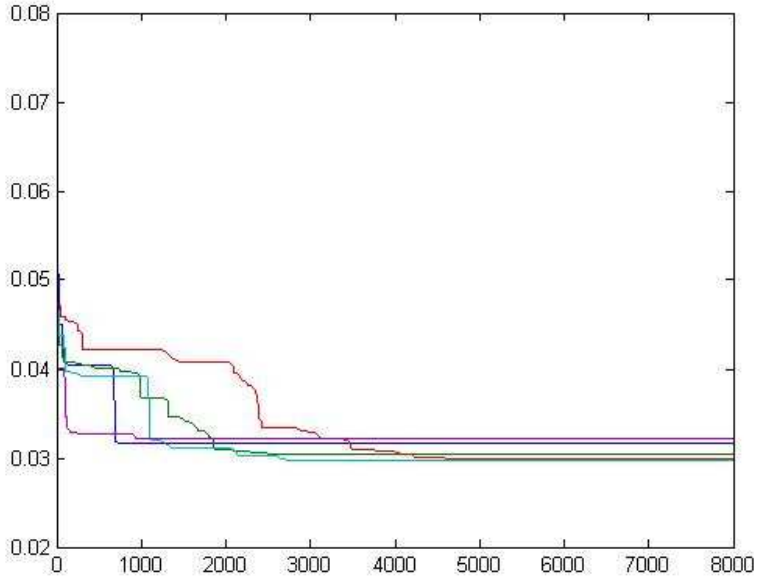
Table 3.3. PSO results in portfolio selection at various penalty parameter values ($\pi = 0.0075$).

We see that the share of any asset in the portfolio belong to the interval $(0.05, 0.35)$, and the sum of shares of all assets is equal to one at each ε which implies that using any of the penalty values the institutional constraint and the budget constraint are satisfied.

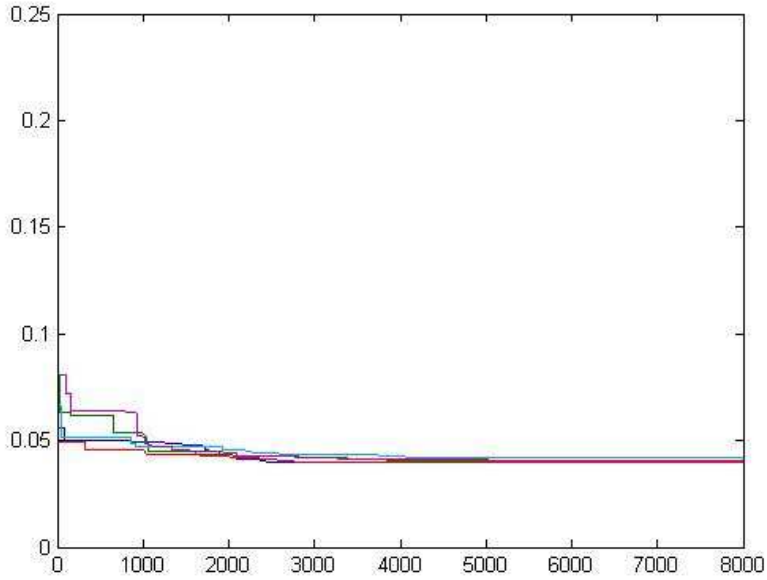
[84] The weights of assets in the portfolio are 0.35, 0.35, 0.20, 0.05 and 0.05.

Parameter values $\varepsilon = 1$ and $\varepsilon = 0.1$ are not small enough to penalize the swarm for deviation from the target expected return constraint. In the experiment of $\varepsilon = 1$, we see that the expected return is found to be 0.0091 when we restricted our portfolio only to have 0.0075. The same situation happens in the experimenting penalty parameter $\varepsilon = 0.1$, where expected return comes out to be 0.0079. Thus, for penalty parameters $\varepsilon = 1$ and $\varepsilon = 0.1$, the constraint given by Equation 3.7b is not satisfied. For these penalty parameter values Equation 3.7b does not hold, while for $\varepsilon = 0.01$ and $\varepsilon = 0.001$ we can get the required target return.

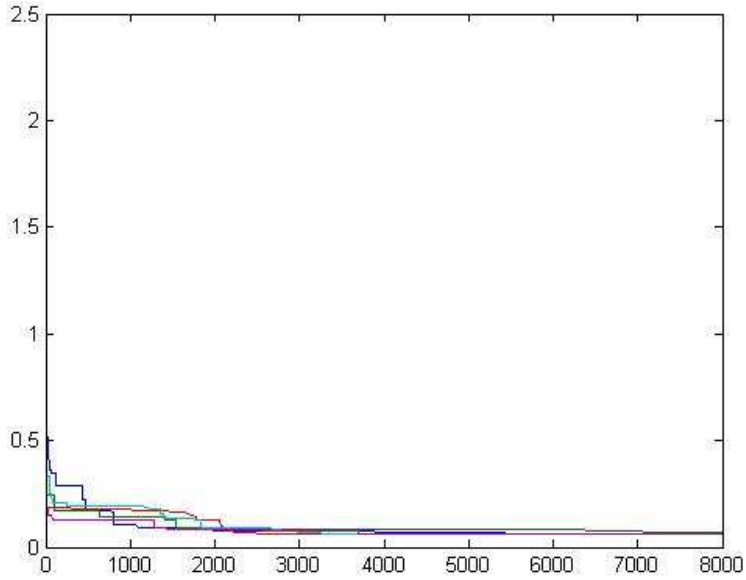
Although for penalty parameter values $\varepsilon = 0.01$ and $\varepsilon = 0.001$, all constraints in the portfolio selection problem are satisfied, we would prefer $\varepsilon = 0.01$ than $\varepsilon = 0.001$. Because the former gives us less fitness value than does the latter in a shorter period of time.



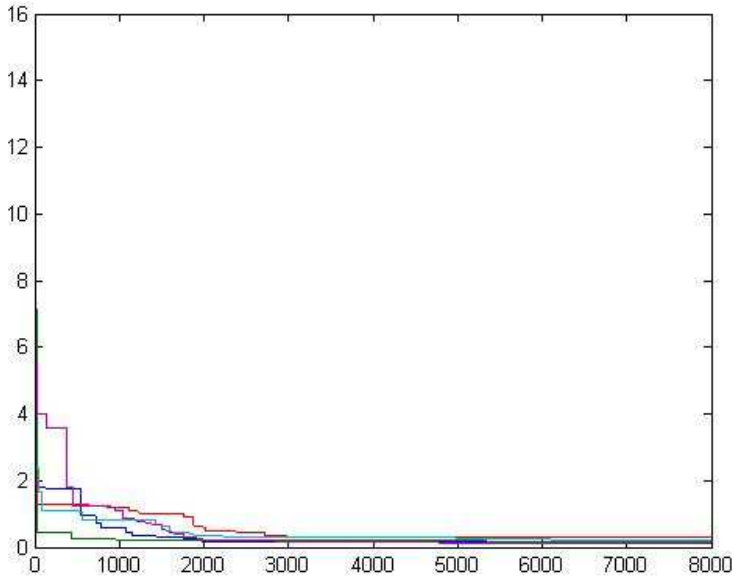
Graph 3.1. Evolutions of "best" fitness values of five various swarms in one run of the PSO ($\pi = 0.0075, \varepsilon = 1.0$).



Graph 3.2. Evolutions of "best" fitness values of five various swarms in one run of the PSO ($\pi = 0.0075, \varepsilon = 0.1$).

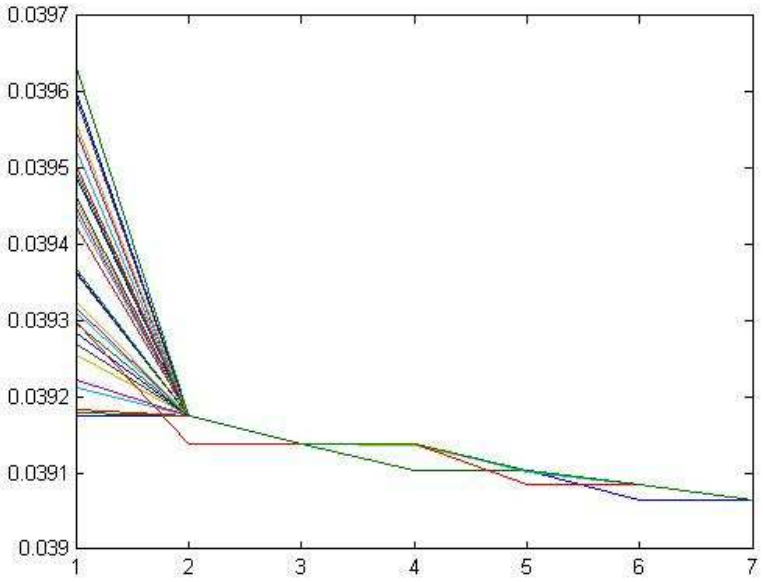


Graph 3.3. Evolutions of "best" fitness values of five various swarms in one run of the PSO ($\pi = 0.0075, \varepsilon = 0.01$).

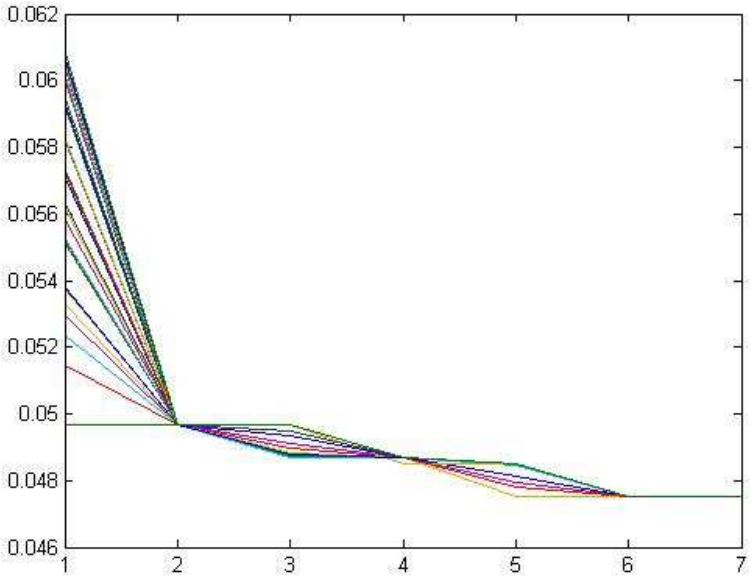


Graph 3.4. Evolutions of "best" fitness values of five various swarms in one run of the PSO ($\pi = 0.0075, \varepsilon = 0.001$).

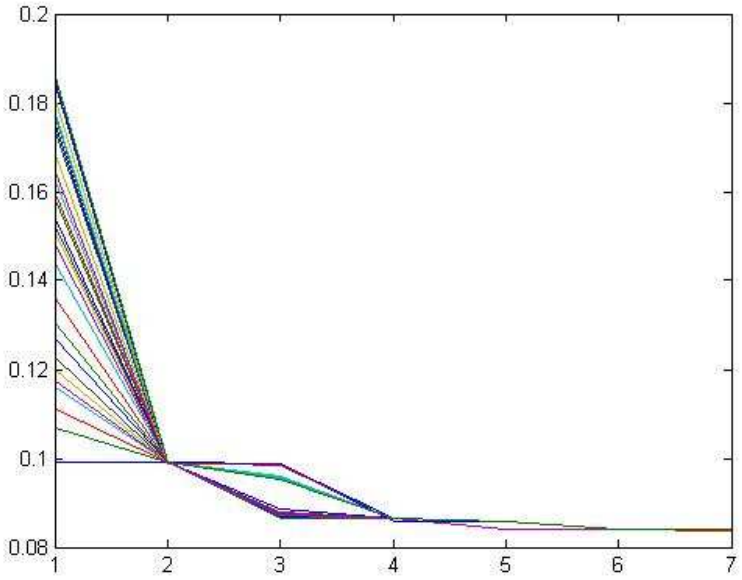
We see what improvements could be achieved by using our extended PSO, the algorithm that mobilizes the results found by running GCPSO several times to produce even better results, in the Graphs 3.5-3.7 and 3.12-3.14. Recall that, in the extended PSO, at first we generate K solutions. In the second stage, we select N of them with less fitness values and run PSO N times to generate other N solutions, and we continue in the same way in following stages. In each stage we save the best fitness values found so far by the swarm, and plot them in the graphs (i.e., Graphs 3.5-3.7 and 3.12-3.14). For a particular penalty parameter value and other parameter settings, values in the ordinate corresponding to the axis value one represent the solutions of GCPSO according to Bergh (2001). The fitness values found by various simultaneously used swarms have converged to a single number in the last stage, which is our solution to the portfolio selection problem. Within these stages, the minimum fitness values found by swarms persistently decrease.



Graph 3.5. Our extended PSO algorithm. The local minima are used in the PSO to find a global minimum ($\varepsilon = 0.1$).



Graph 3.6. Our extended PSO algorithm. The local minima are used in the PSO to find a global minimum ($\pi = 0.0075, \varepsilon = 0.01$).



Graph 3.7. Our extended PSO algorithm. The local minima are used in the PSO to find a global minimum ($\pi = 0.0075, \varepsilon = 0.001$)

For $\pi = 0.0103$, we summarize the results of each asset with its corresponding share in the portfolio, the expected return and the risk level of the portfolio, and time needed for computation at all values of ε (Table 3.4.). We again see that the share of any asset in the portfolio belong to the interval $(0.05, 0.35)$, and the sum of shares of all assets is equal to one at each ε which implies that using any of the penalty values the Equations 3.7c and 3.7d are satisfied.

Again, penalty parameter values $\varepsilon = 1$ and $\varepsilon = 0.1$ are not small enough to penalize the swarm for deviation from the target expected return constraint. In the experiment of $\varepsilon = 1$, we see that the expected return is found to be 0.0092 when we restricted our portfolio to have 0.0103 rate of return. The same situation happens in the experimenting penalty parameter $\varepsilon = 0.1$, where expected return comes out to be 0.0101. For these penalty parameter values Equation 3.7b does not hold, while for $\varepsilon = 0.01$ and $\varepsilon = 0.001$ we can get the required target return.

pi=0.0103

epsilon =1

	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.0500	0.3500	0.0702	0.1798	0.3500	1.00
Expected Return	0.0092					
Risk level	0.0279					
Time (sec)	816.34					

epsilon =0.1

	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.0505	0.3500	0.1989	0.0507	0.3500	1.00
Expected Return	0.0101					
Risk level	0.0326					
Time (sec)	1212.10					

epsilon =0.01

	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.0562	0.2170	0.3269	0.0501	0.3499	1.00
Expected Return	0.0103					
Risk level	0.0415					
Time (sec)	1344.47					

epsilon =0.001

	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.0890	0.1799	0.3343	0.0504	0.3464	1.00
Expected Return	0.0103					
Risk level	0.0438					
Time (sec)	1378.87					

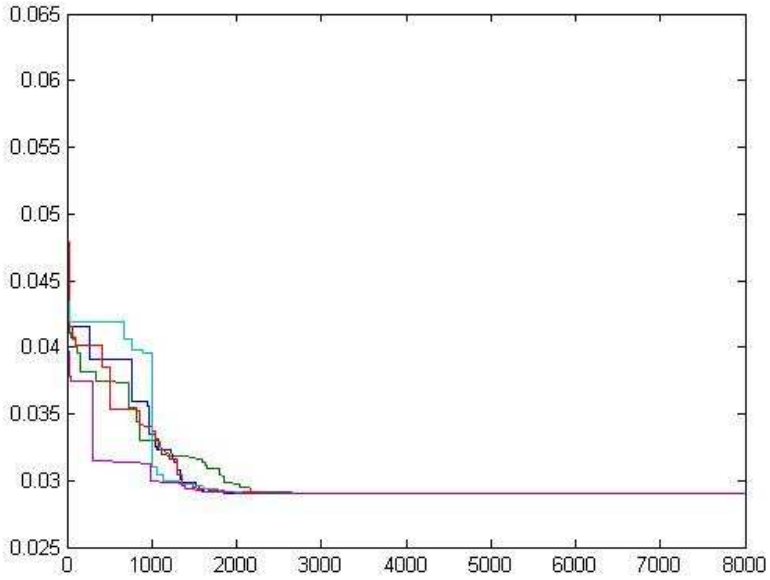
Table 3.4. PSO results in portfolio selection at various penalty parameter values ($\pi = 0.0103$).

Even though for penalty parameter values $\varepsilon = 0.01$ and $\varepsilon = 0.001$, all constraints in the portfolio selection problem are satisfied, we once more would prefer $\varepsilon = 0.01$ than $\varepsilon = 0.001$, since the former gives us less fitness value than does the latter in a shorter period of time.

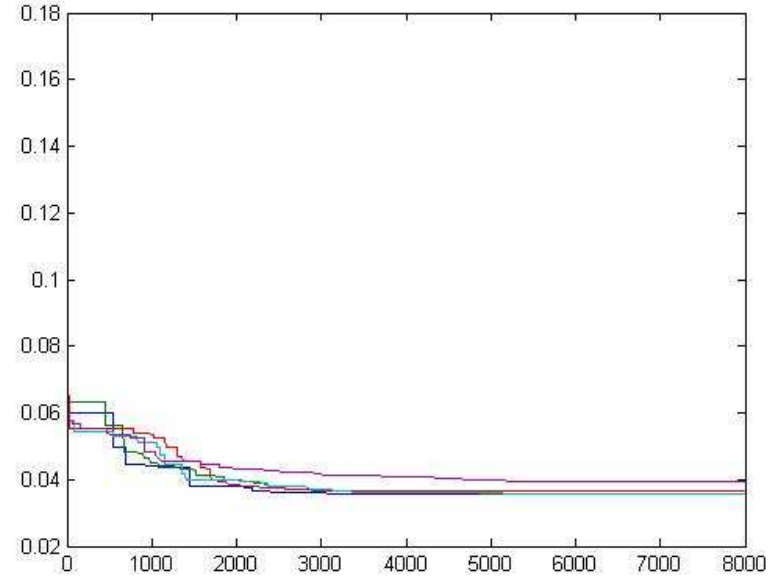
From Tables 3.3-3.4., we see that the portfolio which gives $\pi = 0.0092$ rate of return has the risk value ($\rho(0.0092) = 0.0279$) less than the risk values ($\rho(0.0075) = 0.0443$ and $\rho(0.0103) = 0.0415$) of the portfolios with lower and higher rates of return. This exactly corresponds to the case when efficient-frontier of return-risk has "$\subset$"-shape like the one we learnt in the Markowitz model.

To sum up, with the given parameter settings in Table 3.2 (the number of particles used, the stopping criterion, and so on), we decide to opt penalty parameter value $\varepsilon = 0.01$, in the

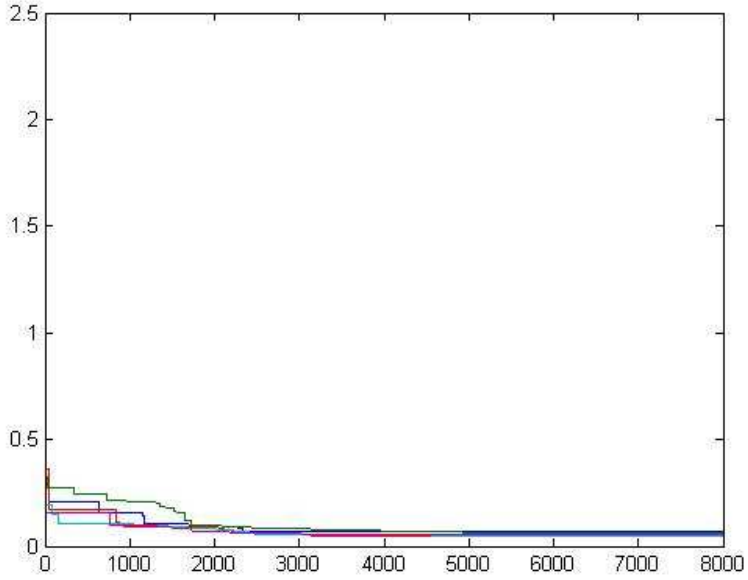
following experiments since the solution found by using this penalty value satisfies all the constraints in the portfolio selection problem, has less risk level as a result of the fact that $\varepsilon = 0.01$ is small enough not to trap the particles at the boundary.



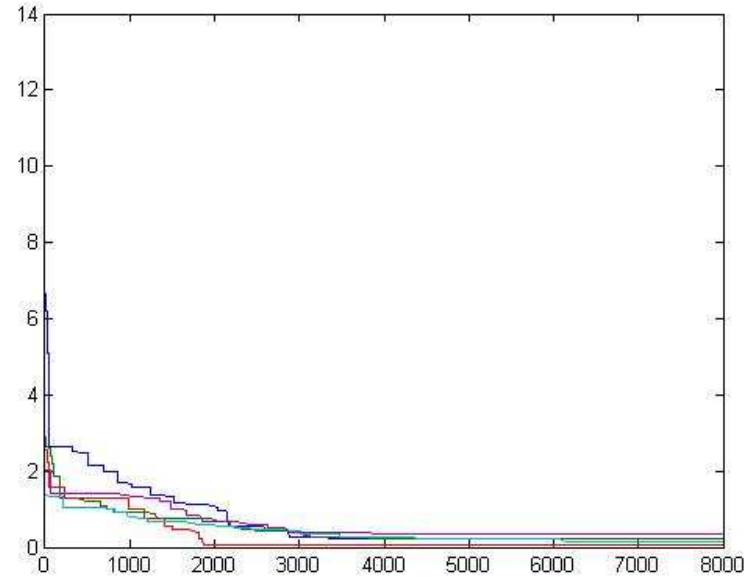
Graph 3.8. Evolutions of "best" fitness values of five various swarms in one run of the PSO ($\pi = 0.0103, \varepsilon = 1$).



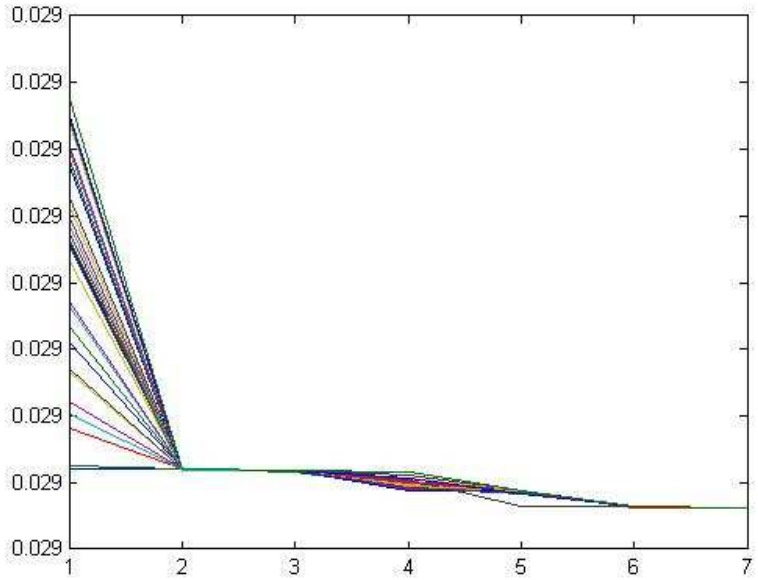
Graph 3.9. Evolutions of "best" fitness values of five various swarms in one run of the PSO ($\pi = 0.0103, \varepsilon = 0.1$).



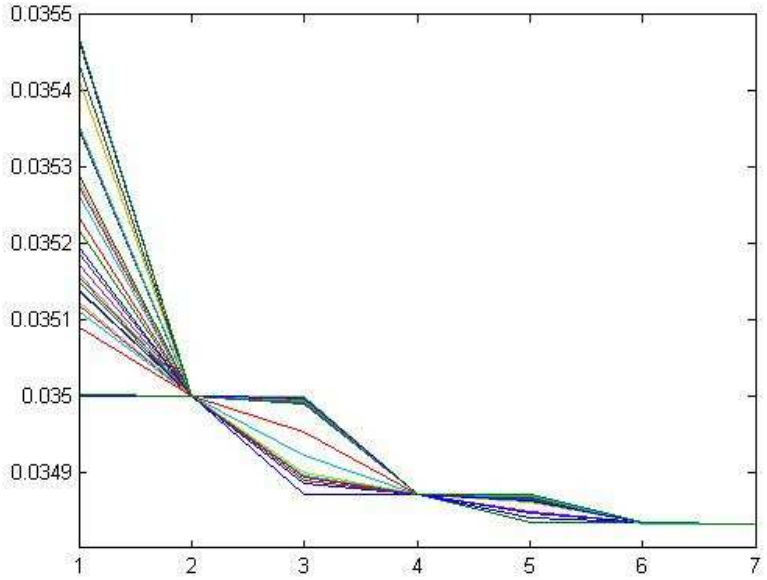
Graph 3.10. Evolutions of "best" fitness values of five various swarms in one run of the PSO ($\pi = 0.0103, \varepsilon = 0.01$).



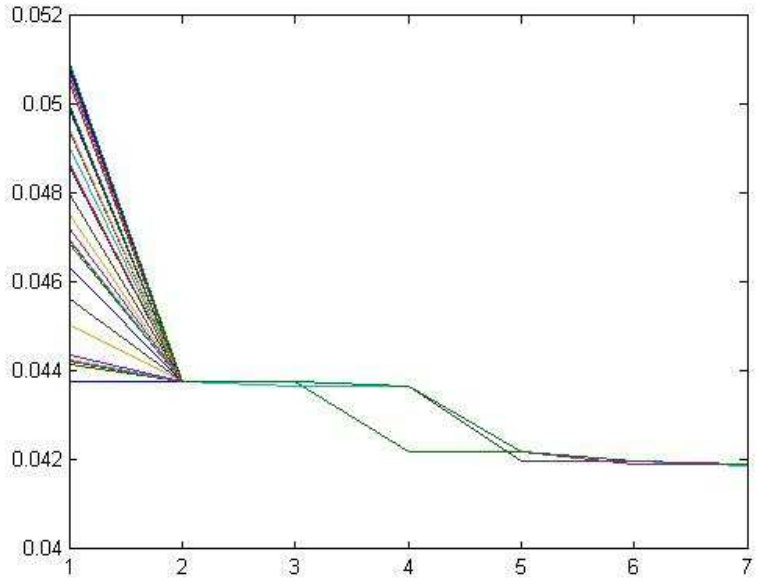
Graph 3.11. Evolutions of "best" fitness values of five various swarms in one run of the PSO ($\pi = 0.0103, \varepsilon = 0.001$).



Graph 3.12. Our extended PSO algorithm. The local minima are used in the PSO to find a global minimum ($\pi = 0.0103, \varepsilon = 1$)



Graph 3.13. Our extended PSO algorithm. The local minima are used in the PSO to find a global minimum ($\pi = 0.0103, \varepsilon = 0.1$)



Graph 3.14. Our extended PSO algorithm. The local minima are used in the PSO to find a global minimum ($\pi = 0.0103, \varepsilon = 0.01$)

3.6.2. Experiment on the number of particles in the swarm

In this section we will experiment on the number of particles in the swarm for different values of expected return. As we discussed above, small number of particles in the swarm enables the swarm to explore more area in the search space while high number of particles makes the swarm converge faster.

itermax=8000;	number of iteration
epsilon=0.01;	penalty parameter value
wmax=0.9;	inertia weight--upper bound
wmin=0.5;	inertia weight--lower bound
c1=1.49445;	acceleration coefficient for pbest
c2=1.49445;	acceleration coefficient for gbest
D=5;	number of securities in portfolio
T=120;	120 monthly data considered
LB=0.05;	lower constraint
UB=0.35;	upper constraint
a=0.5;	risk measure factors
p=3;	
S=5;	

K=100;	number of "best candidates" for swarm particles
exit_flag = 1000;	exit loop if objective function is constant in consecutive 1000 iterations
rho=1;	value suggested number by Bergh (2001)
Sc=7;	rho=(1/alpha)*rho if consecutive number of successes is greater than Sc
Fc=5;	rho=(alpha)*rho if consecutive number of failures is greater than Fc
alpha=0.95;	

Table 3.5. Selection of other parameter values.

We consider three values for the population of the swarm, in particular, $N = 20$, $N = 25$, and $N = 30$ for two randomly selected levels of expected return, $\pi = 0.0090$ and $\pi = 0.00965$. We keep the other parameter values as before (Table 3.5) in which the slight difference is that $S = 5$ instead of $S = 7$. Recall that S is the number of stages in our extended PSO algorithm, and by giving less value for this parameter we would also like to check whether the number of particles we set has tendency to converge faster.

pi=0.0090

N =20

	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.0537	0.3153	0.0731	0.2197	0.3381	1.00
Risk level	0.0294					
Time (sec)	624.96					

N =25

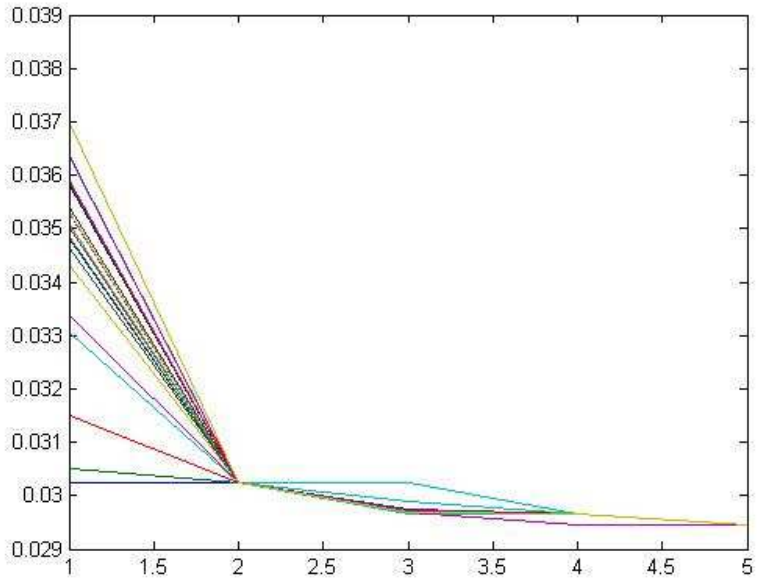
	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.0701	0.3290	0.0627	0.2105	0.3277	1.00
Risk level	0.0292					
Time (sec)	748.08					

N =30

	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.0880	0.3233	0.0543	0.2075	0.3270	1.00
Risk level	0.0296					
Time (sec)	894.84					

Table 3.6. PSO results in portfolio selection at various numbers of particles in the swarm.

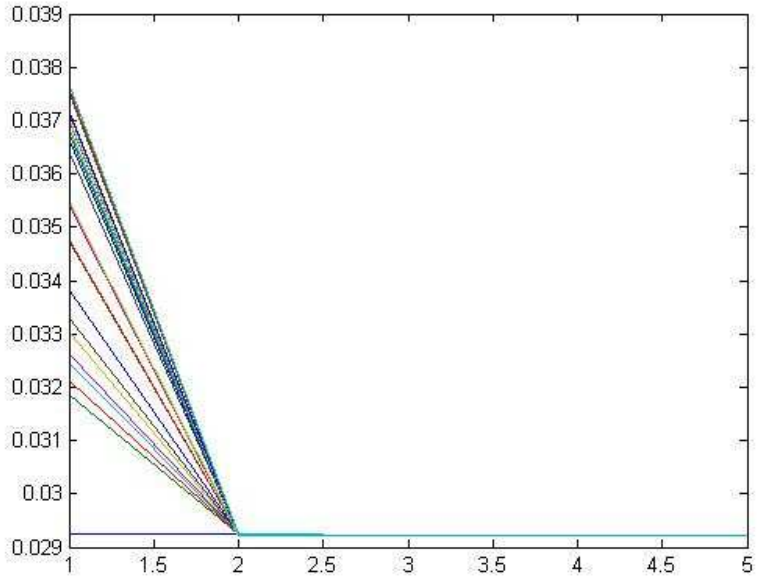
For $\pi = 0.0090$ the solutions for the portfolio selection problem at all levels of N satisfy the constraints given by Equations 3.7b, 3.7c, and 3.7d. Although, for the portfolio weights computed by using 25 particles in the PSO give the minimum risk level, we can conclude that we are almost indifferent among them since the difference in the risk level is insignificant.



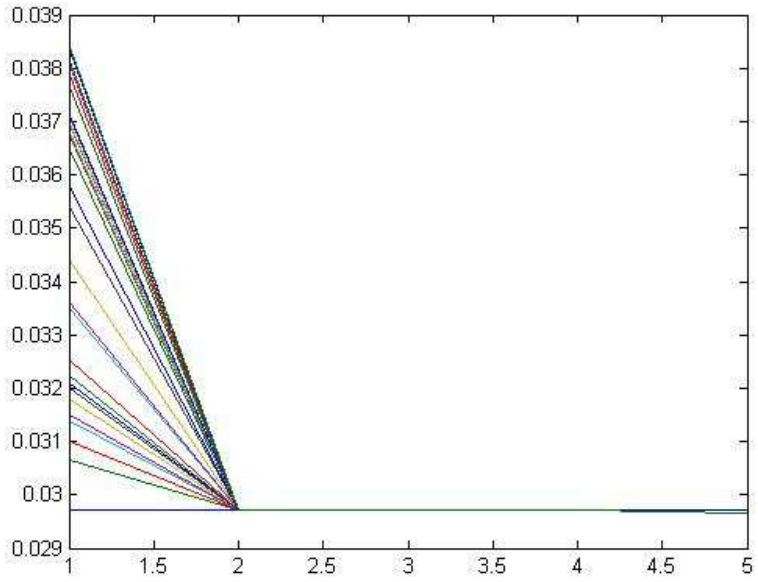
Graph 3.15. Our extended PSO algorithm. The local minima are used in the PSO to find a global minimum ($N = 20, \pi = 0.0090, \varepsilon = 0.01$)

However, from the graphs 3.15-3.17 we see that our extended PSO tends to converge faster with the high number of particles. It took 5 stages 20 particles to converge in the extended PSO, while only 2 stages were needed for $N = 25$ or $N = 30$.

Furthermore, for the level of expected return of the portfolio $\pi = 0.00965$ we again see that any solution from using any number of particles in the PSO satisfies all constraints in the portfolio selection problem (Table 3.7).



Graph 3.16. Our extended PSO algorithm. The local minima are used in the PSO to find a global minimum ($N = 25, \pi = 0.0090, \varepsilon = 0.01$)



Graph 3.17. Our extended PSO algorithm. The local minima are used in the PSO to find a global minimum ($N = 30, \pi = 0.0090, \varepsilon = 0.01$)

pi=0.00965

N =20

	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.1695	0.3275	0.0628	0.0917	0.3484	1.00
Risk level	0.0326					
Time (sec)	620.67					

N =25

	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.0693	0.3373	0.1344	0.1130	0.3460	1.00
Risk level	0.0302					
Time (sec)	772.59					

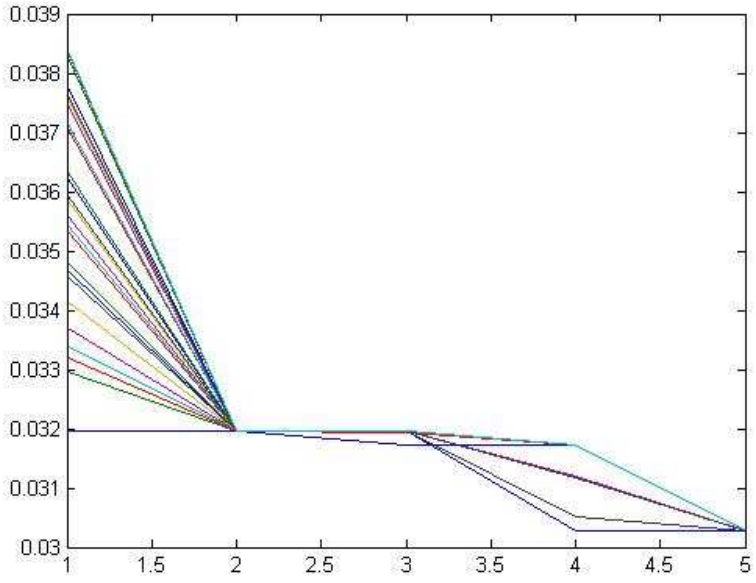
N =30

	GENESCO	HORMEL FOODS	HUMANA	INTEL	KOSS	SUM
gbest	0.0610	0.3499	0.1360	0.1111	0.3419	1.00
Risk level	0.0298					
Time (sec)	972.49					

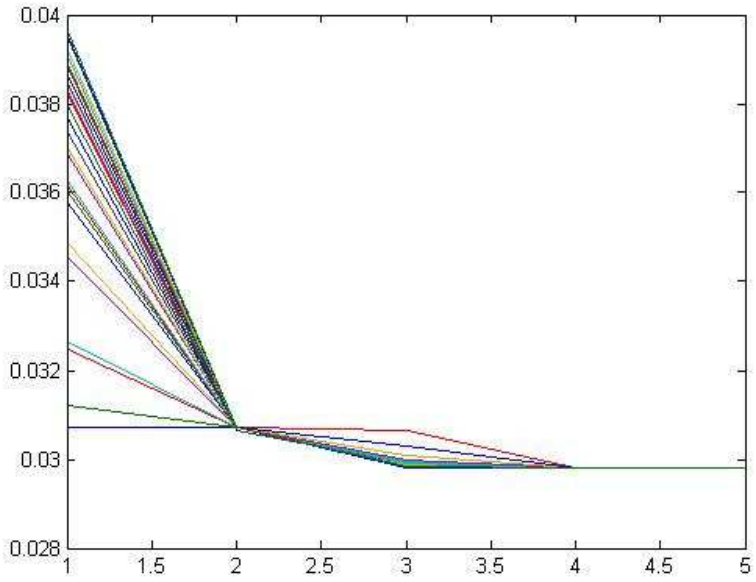
Table 3.7. PSO results in portfolio selection at various numbers of particles in the swarm.

However, the risk level of portfolio found using 20 particles is significantly higher than those of the portfolios found using 25 and 30 particles. The extended PSO converges faster with 30 particles, in contrast to the very same PSO which needs less time using only 25 particles (see graphs 3.18-3.19).

Thus, given the parameter settings in this section we are indifferent between using 25 or 30 particles. One converges faster in terms of stages in the extended PSO, other needs less time to compute. But the final result in terms of the minimum risk level found so far almost the same. So, in the rest of the work we would like to work with 25 particles since the extended PSO needs less time to evaluate the procedures in the algorithm and has more exploring behavior.



Graph 3.18. Our extended PSO algorithm. The local minima are used in the PSO to find a global minimum ($N = 25, \pi = 0.00965, \varepsilon = 0.01$)



Graph 3.19. Our extended PSO algorithm. The local minima are used in the PSO to find a global minimum ($N = 30, \pi = 0.00965, \varepsilon = 0.01$)

3.7. Efficient-Frontier of Portfolios with risky assets

In this chapter we will draw the efficient-frontier of rewards with respect to the two-sided risk measure by using the extended PSO and GCPSO algorithms. Recall that x is a frontier portfolio if its return has the minimum risk among all portfolios that have the same expected payoff.

Since our asset choices must vary in the interval $(0.05, 0.35)$ and the expected return on each asset is given by Table 3.1, the minimum possible expected return on our portfolio is 0.0075 and the maximum one is 0.0103[85].We selected 16 possible rates of return in this interval (see Table 3.9) and solved the portfolio selection problems by using our extended PSO and GSPSO algorithms explained above. PSO parameters values are chosen as shown in the Table 3.8.

itermax=8000;	number of iteration
N=25;	Number of particles in the swarm
epsilon=0.01;	penalty parameter value
wmax=0.9;	inertia weight--upper bound
wmin=0.5;	inertia weight--lower bound
c1=1.49445;	acceleration coefficient for pbest
c2=1.49445;	acceleration coefficient for gbest
D=5;	number of securities in portfolio
T=120;	120 monthly data considered
LB=0.05;	lower constraint
UB=0.35;	upper constraint
a=0.5;	risk measure factors
p=3;	
S=7;	
K=100;	number of "best candidates" for swarm particles
exit_flag = 1000;	exit loop if objective function is constant in consecutive 1000 iterations
rho=1;	value suggested number by Bergh (2001)
Sc=7;	rho=(1/alpha)*rho if consecutive number of successes is greater than Sc
Fc=5;	rho=(alpha)*rho if consecutive number of failures is greater than Fc
alpha=0.95;	

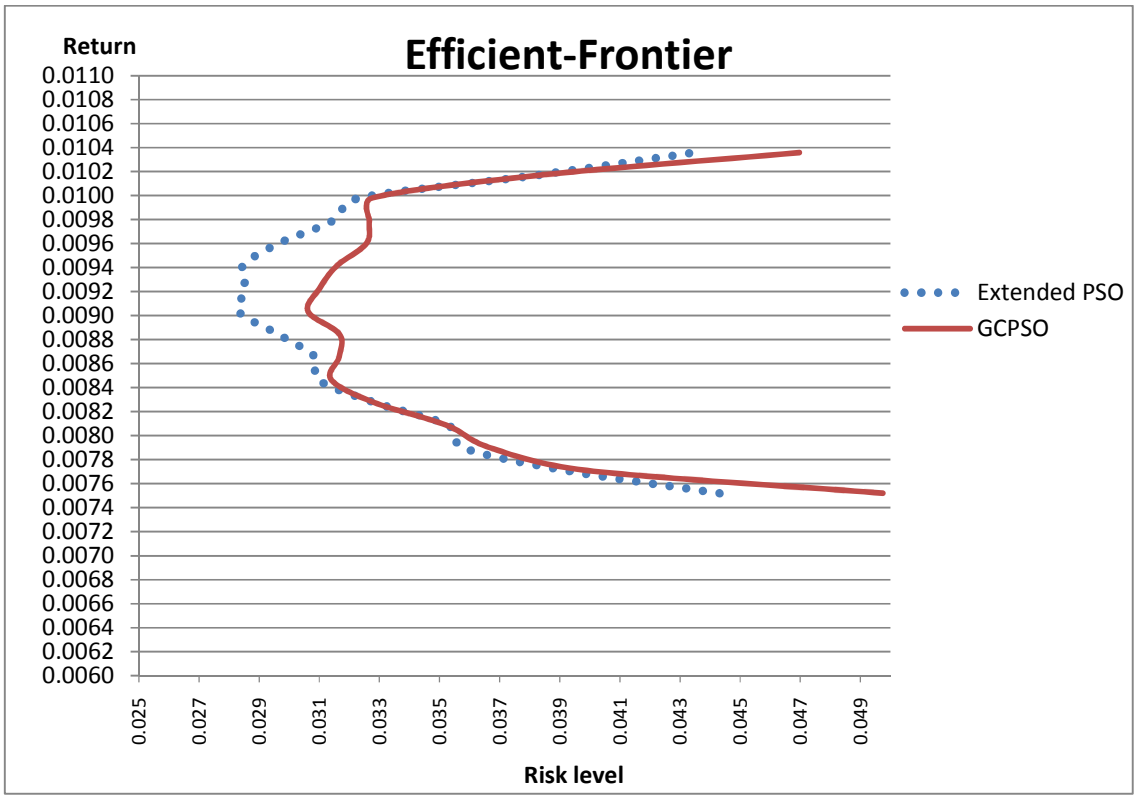
Table 3.8. PSO parameters.

[85] For instance, the case in which portfolio return is 0.0075 can be obtained by choosing asset' weights (0.35, 0.35, 0.20, 0.05, 0.05) in which the assets with less rates of return are chosen with higher weights.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
RETURN	0.00752	0.00771	0.00790	0.00809	0.00828	0.00847	0.00866	0.00884	0.00903	0.00922	0.00941	0.00960	0.00979	0.00998	0.01017	0.01036
RISK (Extended PSO)	0.04431	0.03923	0.03579	0.03529	0.03284	0.03097	0.03087	0.02965	0.02833	0.02856	0.02846	0.02964	0.03145	0.03231	0.03817	0.04343
RISK(GCPSO)	0.04975	0.03991	0.03669	0.03522	0.03281	0.03140	0.03166	0.03168	0.03063	0.03100	0.03158	0.03257	0.03267	0.03276	0.03829	0.04698

Table 3.9. Risk-return values on the efficient-frontier line

The levels of risk for each investor desired rate of return are shown in Table 3.9 and plotted in Graph 3.20. Combinations along this line represent portfolios (involving only the risky assets) for which there is lowest risk for a given level of return. Using extended PSO, the efficient frontier is obtained by running the algorithm only once for each rate of return. On the contrary, the efficient frontier found by GCPSO is plotted using the minimum risk value found in 100 runs. We see that for this particular portfolio selection problem our extended PSO algorithm performs better than the GCPSO algorithm suggested by Bergh to search for global minima.



Graph 3.20. Efficient Frontiers found by Extended PSO and GCPSO.

Another interesting fact is that the efficient frontiers in Graph 3.20 looks, more or less, like a parabola when expected return is plotted against two-sided risk measure. The efficient

frontiers (EF) remind us the Markowitz frontier –this is because the risk-return characteristics of a portfolio generally change in a non-linear fashion as its component weightings are changed. The upper part of the EF curve is the *frontier of efficient portfolios* and the lower part of the EF curve is the *frontier of inefficient portfolios*.

The efficient frontier implies that the risk levels evaluated by solving portfolio selection problems for various levels of investor desired rates of returns using extended PSO are consistent with the interpretation of two-sided risk measure. Recall that our two-sided risk measure, with parameter settings $a = 0.5$ and $\rho = 3$, represents behavior of risk-averse investor, because the investor is more concerned with the downside risk, and therefore is giving more weight on the negative variations. Analogously, in the efficient frontier, we see that the additional risk can only be undertaken if the compensation for it is higher, which coincides with the behavior of risk-averse investor.

CHAPTER 4 Conclusion

In this work we have presented and provided tools of the Particle Swarm Optimization, introduced a new hybridized PSO algorithm – which we call as the extended PSO –, and the performance of the extended PSO in coping with portfolio selection problems with respect to a two-sided risk measure is investigated. The obtained results are reported and compared with those obtained through Guaranteed Convergence Particle Swarm Optimization (GCPSO).

In the first chapter we talked and discussed about Particle Swarm Optimization, which is inspired by the collective foraging behavior of a group of simple agents, like a school of fish, a flock of birds, and similar. Particle Swarm behavior is displayed by particles in multidimensional space that have two characteristics: a *position* and a *velocity*. Each particle in the swarm uses the information about its velocity or position, and the information in the swarm, i.e. the average direction and velocity taken by the other. By doing so, each individual benefits not only from its discoveries, but from what the other particles have found as well, and therefore, it heads towards the best region discovered by the group.

The need for application of PSO in the real-world problems, as an approximation for the optimal solutions, instead of ordinary mathematical tools can be explained due to a number of factors, such as: 1) the optimization problem does not have a closed-form or analytical solution; 2) too costly (in terms of time) and cumbersome to compute.

Later, we described the theoretical background of PSO, its formulation and operational mechanisms, and analyzed its convergent properties. We concluded the chapter by providing fundamental idea behind Guaranteed Convergence PSO introduced by Van den Bergh (2001) as a possible search algorithm for global minima.

In the second chapter we provided fundamental information related to the portfolio selection problem. We discussed about significant assumptions underlying Markowitz theory and presented a portfolio selection problem with respect to the variance risk measure. In this framework it is presumed that investor has quadratic utility function or asset returns follow a multivariate normal distribution. We argued that mean-variance criteria must be applied with

caution since this approach can be appropriate only if the investor's utility function is quadratic with nonpositive second derivatives or if the asset's return distribution can be fully described by the first two moments.

Attempts to quantify risk correctly have led to the development of risk measure to as an independent subject. In the section 2.4, we talked about various other risk measures, such as Value-at-Risk and the family of coherent risk measures. In particular, we emphasized on the shortcomings of VaR in terms of properly measuring risk. VaR cannot be adequate risk measure if it is implemented to non-elliptical return distributions because it does not consider any losses exceeding VaR, it may provide contradictory or paradoxical outcomes if different confidence levels are applied, and most importantly it is very difficult to apply in optimization problems since VaR is non-convex and has many local extremes leading to unstable VaR ranking[86].

Therefore, we later told about coherent risk measures as possible alternatives to traditional risk measures, and their properties with some examples. We presented new two-sided risk measure introduced by Chen et al. which, unlike the existing coherent risk measures, takes into account both positive and negative variations from the mean at the same time with different levels of degree. In the end of Chapter 2, a constrained portfolio selection problem with this risk measure as objective function is introduced for the application of PSO.

Finally, in Chapter 3 we provided the necessary experiments and results concerning the application of the Particle Swarm Optimization in the portfolio selection problem. We started by mentioning some literature about applications of PSO in real-world problems and recognized the inefficiencies in terms of obtaining final result. To be precise, the usual PSO algorithm has to be run several times in order to get possible results, and the solution set was given by a confidence-like interval – mean plus and minus standard deviation. This method and the outcomes generally might not be satisfying, since it is not known how many times one should run the algorithm or we might prefer obtaining single result rather than an interval of possible solutions.

Trying to solve these inefficiency problems we introduced a so-called extended PSO –an algorithm that works not only with particles surfing the search space, but also with the solution

[86] G. Szegö, Measures of Risk. European Journal of Operational Research 163 (2005) 5–19.

vector found in each experiment. In the extended PSO, we claimed that only "best" particles must be used to generate new global best position in the search space. Here, by saying the word "best" we meant the particle whose personal best position has less fitness value than those of many possible particles in the search space. This extended PSO can be treated as a sort of hybridized PSO – a PSO algorithm with the characteristic of Genetic Algorithm –, in which only fitted parents have right to produce an offspring. We further explained the computational procedure of the extended PSO in pseudo-MATLAB code.

We presented the fitness function, an unconstrained version of the optimization problem, and showed that the solution to our unconstrained optimization is equivalent to the original constrained one under proper conditions. We carried out some experiments on the penalty parameter ε and the number of particles in the swarm under specific parameter settings. The monthly returns which are used in these experiments are evaluated from contemporaneous stock-price data for five publicly traded individual stocks from S&P500 for the period December 31st, 1997 through December 31st, 2007

We considered four possible values for the parameter ε, i.e. 1, 0.1, 0.01, and 0.001 for the first experiment. For penalty parameters $\varepsilon = 1$ and $\varepsilon = 0.1$, the constraint in which the expected rate of return for the portfolio must be π, is not satisfied. We preferred $\varepsilon = 0.01$ to $\varepsilon = 0.001$, because the former gives less fitness value than does the latter in a shorter period of time. This happens when particles highly penalized whenever they cross the boundaries and, therefore, are trapped in a local minimum in case of $\varepsilon = 0.001$.

Second, we experimented on the number of particles in the swarm for different values of expected return. Small number of particles in the swarm enables the swarm to explore more area in the search space while high number of particles makes the swarm converge faster. We considered three values for the population of the swarm, $N = 20$, $N = 25$, and $N = 30$. The results from the experiments show that the risk level of portfolio found using 20 particles is significantly higher. The result in terms of the minimum risk level found so far almost the same using 25 or 30 particles. In the latter, particles converge faster in terms of stages in the extended PSO; while in the former the algorithm needs less time to compute. Nevertheless, we

chose 25 particles in the further analysis since the extended PSO needs less time to evaluate the computational procedures, and has more exploring behavior.

In the end, we computed and plotted efficient-frontier of portfolios with risky assets by using both GCPSO and extended PSO algorithms. Better fitness value can be achieved by using our extended PSO than by using GCPSO. This empirical result is consistent with the theoretical background, since in the extended PSO we utilize the results of GCPSO to search better position in the search space. Moreover, our results suggest that the efficient frontier with respect to the two-sided risk measure looks like the one obtained through mean-variance in Markowitz model, and shares similar interpretations such as investor is willing to undertake a higher risk if he/she is compensated with higher rate of return.

The capability of the extended PSO algorithm to tackle with constrained optimization problems was investigated through the performance of numerous experiments. The results obtained in this work imply that extended PSO is good alternative to solve portfolio selection problems and superior to GCPSO introduced by Bergh. However, to be useful in practical applications future work may include investigation of the extended PSO's performance in other benchmark and real-world problems. It might also be interesting to compare, in the context of a portfolio selection problem, the performance of the extended PSO algorithm with another type of optimization algorithm such as Genetic Algorithm or Ant Colony Optimization. These analyses are beyond the scope of this work, and therefore are delayed for further research.

REFERENCES

1. Acerbi, C., Tasche, D., 2002. On the coherence of expected shortfall. In: Giorgio Szegö (Ed.), Special Issue on ‘‘Beyond VaR, 26 July. Journal of Banking and Finance. Available from <www.gloriamundi.org /var/wps.html>.
2. Artzner, P., Delbaen, F., Eber, J.-M., Heath, D. (1997) Thinking coherently. RISK 10 (11).
3. Artzner, P., Delbaen, F., Eber, J.-M., Heath, D. (1999) Coherent measures of risk. Mathematical Finance. 9 (3), 203—228.
4. Barro M., Tecniche Particle Swarm Optimization per la selezione di portafoglio., Master Thesis, Department of Economics., University of Ca Foscari, Italy,2007- 2008.
5. Beni, G., Wang, J. (1989) Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy.
6. Billio M., Simulation Based Methods for Financial Time Series., Working Paper., GRETA (March 2002).
7. Blackwell T., Kennedy J., Poli R., 2007, “Particle swarm optimization”, Springer Science, Business Media, 33-57.
8. Bonabeau E., Dorigo M., Theraulaz G., 1999, “Swarm intelligence: from natural to artificial intelligence”, Oxford University Press.
9. Bonabeau E., Meyer C. (2001) Swarm Intelligence: A whole new way to think about business. Harvard Business Review.
10. Bonabeau E., Meyer C. (2001) Harvard Business Review.
11. Booth J. R., Tehranian H., Trennepohl G. L., Efficiency Analysis and Option Portfolio Selection. The Journal of Financial and Quantitative Analysis, Vol. 20, No. 4 (Dec., 1985), pp. 435-450.
12. Breen W., Specific Versus General Models of Portfolio Selection. Oxford Economic Papers, vol. 20 (November 1968), pp. 361-368.

13. Chen Z. Wang Y., Two-sided coherent risk measures and their application in realistic portfolio optimization. Journal of Banking and Finance (2008) 2667-2673.
14. Coello C. A. C., Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art., Computer Methods in Applied Mechanics and Engineering., 191 (2002) 1245–1287.
15. Danielsson J., Embrechts P., Goodhart C., Keating C., Muennich F., Renault O., Shin H.S., An Academic Response to Basel II. Special Paper Series, LSE Financial Markets Group and ESRC Research center, May 2001.
16. Farinelli S., Ferreira M., Rossello D., Thoeny M., Tibiletti L. Beyond Sharpe ratio: Optimal asset allocation using different performance ratios Journal of Banking & Finance 32 (2008) 2057–2063.
17. Feldstein, M. S. "Mean-Variance Analysis in the Theory of Liquidity Preference and Portfolio Selection." Review of Economic Studies, vol. 36 (January 1969), pp. 5-12.
18. Fischer T. (2003) Risk capital allocation by coherent risk measures based on one-sided moments. Journal of Insurance: Mathematics and Economics.
19. Globel F., Kochenberger G., (2003) Handbook of Metaheuristics, Kluwer Academic Publishers.
20. Holland J., Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI, 1975.
21. Hu X., Eberhart R., Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization. Working paper. Department of Electrical and Computer Engineering, Indiana University Purdue University at Indianapolis, IN, USA.
22. Hu S., Yu H., Zangh L., 2005, “Optimal choice of parameters for particle swarm optimization”, Journal of Zhejiang University, Dip. Chemical Engineering, Zhejiang University, China, 528-534.
23. Hull J. C., Options, futures, and other derivatives, 6th edition. Prentice Hall, 2006.
24. Kendal G., Su Y., 2005, ”A particle swarm optimization approach in the construction of optimal risky portfolios”, School of Computer Science and IT, University of Nottingham, UK.

25. Kennedy J., and Eberhart R.,Particle swarm optimization, in Proc. of the IEEE Int. Conf. on Neural Networks, Piscataway, NJ, pp. 1942–1948, 1995.
26. Konno H, Yamazaki H. Mean-absolute deviation portfolio optimization model and its applications to Tokyo Stock Market. Management Science 1991;37:519-31.
27. Markowitz H., "Portfolio Selection," Journal of Finance 7, no. 1 (March 1952): 77–91.
28. Markowitz H. M., Portfolio Selection: Efficient Diversification of Investments, John Wiley, New York, 1959.
29. Nedjah N., Mourelle L.M. (2006) Swarm Intelligent Systems. Springer series.
30. Ozcan E. and Mohan C. K.. Analysis of a Simple Particle Swarm Optimization System. In Intelligent Engineering Systems Through Artificial Neural Networks volume 8, pages 253-258, 1998.
31. Parsopoulos K.E, Vrahatis M.N., Particle Swarm Optimization Method for Constrained Optimization Problems. Department of Mathematics, University of Patras Artificial Intelligence Research Center (UPAIRC), Greece.
32. Poli R., Kennedy J., Balckwell T., Particle swarm optimization—An overview., Swarm Intelligence (2007) 1:33-57.
33. Porter R. B. and Gaumnitz J. E., Stochastic Dominance vs. Mean-Variance Portfolio Analysis: An Empirical Evaluation., The American Economic Review, Vol. 62, No. 3 (Jun., 1972), pp. 438-446.
34. Puelz A. V., A Stochastic Convergence Model for Portfolio Selection. Operations Research, Vol. 50, No. 3 (May - Jun., 2002), pp. 462-476
35. Rachev S., S. Ortobelli, Stoyanov S., Fabozzi F.J, Biglova A. Desirable properties of an ideal risk measure in portfolio theory. International Journal of Theoretical and Applied Finance Vol. 11, No. 1 (2008) 19–54
36. Rockafellar T., Uryasev S., Conditional Value-at-Risk for general loss distributions. Research report #2001-5.
37. Rockafellar, R.T., Uryasev, S., 2001. Optimization of conditional value-at-risk. Presented at the Conference on Statistical and Computational Problems in Risk Management, University of Rome, La Sapienza, June14–16.

38. Rockafellar R. T., Uryasev S., Optimization of conditional value-at-risk, Journal of Risk 2 (2000), 21-41.
39. Rockafellar R.T., Uryasev S, Zabarankin M. , Generalized deviations in risk analysis Finance and Stochastics. 10, 51–74 (2006)
40. Shi Y., Eberhart R.C. A Modified Particle Swarm Optimizer. In IEEE International Conference of Evolutiorary Computation, Anchorage, Alaska, May 1998.
41. Simaan Y. Estimation risk in portfolio selection: the mean variance model versus the mean absolute deviation model. Management Science 1997;43:1437-46.
42. Solis F. and Wets R., Minimization by Random Search Techniques. Mathematics of Operations Research, 6:19-30, 1981.
43. Szegö G., Measures of risk. European Journal of Operational Research 163 (2005) 5–19.
44. Uryasev, S., 2000. Conditional value-at-risk: optimization algorithms and applications, Financial Engineering News, 14 (Feb), 1–5.
45. Van den Bergh F., (2001) An analysis of particle swarm optimizers, PhD Dissertation, Dept. of Computer Science, University of Pretoria, South Africa.
46. Yahya H., 2000, “The Miracle in the Ant”, Vural Yayincilik Press, Istanbul, Turkey.
47. http://www.en.wikipedia.org.
48. http://www.dei.unipd.it/~monaci/euristici_rev2.
49. http://www.swarmintelligence.org/index.php.
50. http://www.particleswarm.info/.

Università Ca' Foscari - Venezia

ESTRATTO PER RIASSUNTO DELLA TESI DI LAUREA E DICHIARAZIONE DI CONSULTABILITA' (*)

Il sottoscritto/a **NOZIM KOMILOV**

Matricola n. **822556**

Facoltà **ECONOMIA**

iscritto al corso di ○ laurea ◉ laurea magistrale/specialistica in:

ECONOMIA

Titolo della tesi (**): **Particle Swarm Intelligence:an alternative approach in Portf Optimization**

DICHIARA CHE LA SUA TESI E':

◉ Consultabile da subito ○ Consultabile dopo ___ mesi ○ Non consultabile

○ Riproducibile totalmente ○ Non riproducibile ◉ Riproducibile parzialmente

Venezia, _____ Firma dello studente _____

(spazio per la battitura dell'estratto)

The performance of the Particle Swarm Optimization (PSO) method in coping with Portfolio Selection problems is investigated in this work. We introduce extended PSO algorithm as an alternative PSO approach to solve constrained portfolio selection problems in which the objective function is two-sided coherent risk measure. In PSO, the potential solutions, called particles, surf through the search space by learning from the global best solution and its own memory. By doing so, each particle benefits not only from its discoveries, but from what the other particles have found as well, and therefore, it heads towards the best position in the search space discovered by the group. The obtained results are reported and compared with those obtained through Guaranteed Convergence PSO (GCPSO). In the end we will provide the efficient frontier of portfolios with respect to the two-sided coherent risk measure. Conclusions are derived and directions of future research are proposed.

(**) il titolo deve essere quello definitivo uguale a quello che risulta stampato sulla copertina dell'elaborato consegnato al Presidente della Commissione di Laurea (*) Da inserire come ultima pagina della tesi. L'estratto non deve superare le mille battute

Informativa sul trattamento dei dati personali

Ai sensi dell'art. 10 della Legge n. 675/ 96 recante disposizioni a "Tutela delle persone e di altri soggetti rispetto al trattamento dei dati personali" si informa che:

1) I dati personali richiesti o acquisiti, i dati relativi alla carriera universitaria e comunque prodotti dall'Università Ca' Foscari di Venezia nello svolgimento delle proprie funzioni istituzionali, nonchè i dati derivanti dal trattamento automatizzato di entrambi, possono essere raccolti, trattati, comunicati e diffusi dall'Università tramite i propri uffici - sia durante la carriera universitaria dell'interessato che dopo la specializzazione - a soggetti esterni per finalità connesse allo svolgimento delle attività istituzionali dell'Università nonchè per comunicazione e diffusione rivolte esclusivamente ad iniziative di avviamento o orientamento al lavoro (stage e placement) e per attività di formazione post-laurea.
2) Il conferimento dei dati personali nell'ambito descritto al punto 1 è obbligatorio
3) Il trattamento dei dati può essere effettuato attraverso strumenti manuali, informatici e telematici atti a gestire i dati stessi ed avviene in modo da garantire la sicurezza e la riservatezza.