



Università
Ca' Foscari
Venezia

Master's Degree programme – Second Cycle
(*D.M. 270/2004*)
in Economia e Finanza – Economics and
Finance

Final Thesis

—

Ca' Foscari
Dorsoduro 3246
30123 Venezia

**PSO and ABC: comparisons for
two-sided risk measures in
portfolio selection problem**

Supervisor

Ch. Prof. Marco Corazza

Graduand

Gurjeet Kaur
824537

Academic Year

2014 / 2015

*Failure
is simply the opportunity to begin again,
this time more intelligently.
H.F.*

*I would like to dedicate this dissertation to my parents, especially to my
mother for all the support and encouragement during these years.*

Contents

Contents	iii
1 Portfolio selection problem	5
1.1 Markowitz model	5
Basic assumptions of Markowitz model	5
Markowitz portfolio selection model	6
Critics to Markowitz Model	11
Improvements of Markowitz Model	12
Concept of coherent risk measure	16
A new class of coherent risk measure: two-sided	20
2 Particle Swarm Optimization and Artificial Bee Colony	23
2.1 Bio-inspired metaheuristics	23
Heuristics and Metaheuristics	23
Swarm Intelligence	26
2.2 PSO - Particle Swarm Optimization	28
Parameter Selection for Particle Swarm	31
Selecting acceleration parameters	31
Selection of maximum velocity	31
Modifications of PSO	32
Inertia Weight approach (IWA)	32
Constriction factor approach (CFA)	33
Fully informed Particle Swarm	34
Population topology	35
2.3 ABC - Artificial Bee Colony	37
Honey Bees in nature	37
Food sources	37
Employed bees	37
Unemployed bees	37
Foraging behaviour	38
Dance	38

ABC - Basic approach	39
Initialization of the population	40
Employed Bees Phase	40
Onlooker Bees Phase	41
Scout Bees Phase	42
ABC selection processes	43
Control parameteres of ABC	43
ABC modifications	44
Modified Artificial Bee Colony	45
Frequency of the perturbation	45
Probability calculation	46
Scout production period (SPP)	47
Chaotic Initialization	48
Modification in Onlooker Bee Phase	48
Other algorithms related to the bee SI	49
2.4 PSO and ABC comparison	51
2.5 Field of Application	52
3 Two-sided risk measure application	55
3.1 Two-sided risk measures and its properties	55
3.2 A realistic portfolio selection model under two-sided risk measure	58
Budget and return constraints	59
Cardinality constraints	60
Portfolio selection model	61
Constraints management	62
Penality function	62
Reformulation of portfolio selection problem	64
4 Applications and discussion for Eurostoxx 50	67
4.1 Preliminary information	67
Problem specific parameters setting	70
PSO - parameters setting	70
ABC parameters setting	71
Application and discussion	72
4.2 Comparison and discussion between PSO and ABC	76
A Appendix	87
Bibliography	95

Acknowledgement

I would like to thanks all the people who helped me during this work. I am especially thankful to my advisor Prof. Marco Corazza, for his guidance, encouragement and advices during the whole work. I could not have imagined having a better advisor.

Besides my advisor I would like to thanks Andrea and my brother for supporting me throughout this time.

Introduction

Portfolio management is one of the most important topics in finance. The portfolio selection problem is concerned with managing the portfolio of assets that minimizes the objective risk subjected to several constraints for guaranteeing a given level of returns. The portfolio selection problem was introduced by Henry Markowitz in 1952 known also as *Modern Portfolio Theory*. His theory revolutionized the investment practice giving it a widespread acceptance as practical tool for portfolio optimization. Even though, the model proposed by Markowitz presents some limitations when dealing with real-world requirements. In fact his model is based on a series of hypothesis and make use of a set of tools that have been revised over the years in order to reduce the simplicity of the model and identify an appropriate measure of risk, suitable with the preferences of investors. The Markowitz portfolio selection and its limitations and the implements for an appropriate measure of risk are discussed in Chapter 1.

Making a real portfolio selection is not an easy task for mainly three reasons.

First of all we need to identify an appropriate measure of risk that satisfies some formal properties, known as coherence, that are shown in chapter 1. In addition it is required that this measure of risk is suitable in identifying investors' attitude towards risk. This means that we need to find a coherent risk measures that fit a non Gaussian world[1]. So making use of variance, which is a measure of dispersion of a random variable, as a measure of risk is not suitable, since it does not satisfies properties of coherence.

Second, we need to take into account several practices and rules of the financial markets that can affect the portfolio selection problem. Generally these are related to a set of constraints such as: the minimum and maximum number of trading stocks. Dealing with these type of constraints leads to problems known as *NP – hard*.

Third, the portfolio selection problem that arises by considering coherent risk measures, paractices and rules of financial markets are generally non-linear, nondifferentiable, nonconvex and mixed-integer, which lead to

NP – hard problems [1]. Due to the complexity of the problem researchers and practitioners have to resort to meta-heuristic techniques that are able to find high-quality solutions in reasonable amount of time [2]. In Chapter 2 we provide an explanation of what meta-heuristic techniques are and mainly we focused on the population-based one: Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC), used to solve the proposed model.

By following what proposed in [1] we tried to replicate the problem. As a measure of risk of returns, a new coherent measure of risk is used, known as two-sided risk measure [3] and this is discussed in Chapter 3. The model proposed by Markowitz make use of the variance of returns, but it is well known that if returns are not normally distributed this could lead to innaccurate investment decisions. This new coherent risk measure deal with this issue and it shows to be able to effectively manage non Gaussian distributions of returns and appropriately reflect investors' attitude towards risk. It allows to take into account both, the risk and gains contained in the left side of returns distribution (“bad tail”) and right side of the same distribution (“good tail”). Apart from [3], one of the first applications of this risk measure to large and complex portfolio can be found in [1]. The problem that we are going to deal with is large and complex since it non-linear mixed-integer constraints are included and this make the considered problem *NP – Hard*.

The issues regarding professional practices and rules of financial markets are dealt through the use of bounds related to the minimum and maximum number of assets to trade. In addition, the model also includes the minimum and the maximum percentage of capital to invest in each asset. All of these rules are formalized in terms of constraints, that make the problem *NP – hard*.

In addition, the considered problem takes into account also standard constraints: minimum return and budget constraint. All these elements make the considered problem nonlinear, nondifferentiable and mixed-integer.

In Chapter 4 in order to investigate the numerical complexity of the problem and provide a reliable solution an exact penalty method is combined with the two population-based algorithms: PSO and ABC. In this last chapter the two algorithms are applied to historical data and the results of the two are compared and there performances are discussed.

Chapter 1

Portfolio selection problem

Portfolio selection is one of the relevant problem in economics and finance. In this chapter we will give a small description of the classical Mean-Variance portfolio selection model proposed by Harry Markowitz, in 1952[4]. This model has been widely recongnized as one of the cornerstone of modern portfolio theory, although in recent years the basic assumptions of the model have been widely challenged. We will give a breif description of the models that try to overcome the limits of the model proposed by Markowitz.

1.1 Markowitz model

More than fifty years ago Harry Markowitz developed a theory that became a foundation of financial economics and revolutionized investment practice, known as *Modern Portfolio Theory*. His work earned him also the Nobel Prize in Economics in 1990. Markowitz proposed his model in 1952, known as Mean-Variance. The model, based on the expected returns (mean) and the standard deviation (variance) of different portfolios, aims at selecting a group of assets which have collectively lower risk than any single asset on its own [5].

Basic assumptions of Markowitz model

Evene though it remains very important in financial economics, the mean-variance analysis proposed by Markowitz rely on some strong assumptions:

- Investors seek to maximize the rate of return that they can achieve from their investments.

- All investors have the same expected single period investment horizon. At the beginning of the period t , the investor allocates his/her wealth among different asset classes, assigning a nonnegative weight to each asset. During the holding period $t + \Delta t$ each asset generates random rate of return, so at the end of the period investor's wealth has been changed by the weighted average of the returns.
- Investors make their investment decisions by taking into account expected returns and standard deviation (that measures risk) of returns from possible investments.
- Investors are generally rational and risk averse¹. They are completely aware of all the risk contained in an investment and take positions based on the risk, demanding a higher return for accepting higher risk. If risks are low, returns are also expected to be low.
- Financial markets are frictionless (e.g. no taxes, no transaction costs etc. . .).
- Investors can reduce risk by adding additional investment assets to their portfolio.
- Investors' assets are infinitely divisible, meaning that they can buy or sell a fraction of a share if they want.
- Individuals make decisions under uncertainty by maximizing the expected value of an increasing utility function of consumption. This is assumed to be increasing and concave.

Markowitz portfolio selection model

The portfolio selection process proposed by Markowitz can be divided into three stages.

1. The first stage consists in identifying an appropriate measure for risk and return.

An investment instrument that can be bought and sold is often called asset. Most of the time the amount of money to be obtained when selling an asset is uncertain at the time of purchase. In that case the

¹Risk-averse investor is the one that, given two identical portfolios, he/she will choose the one with lower standard deviation.

return is random, because we do not know the future price ex-ante. The randomness associated to return is captured by some statistical tools:

- the mean, R_p , is the measure of future return, $\mu_p = E[R_p]$;
 - the variance of R_p is the measure of risk associated to same return, $\sigma_p^2 = E[R_p - \mu_p]^2$;
 - the correlation between return of each pair of risky assets: $\rho_{X,Y}^2$.
2. The second stage consists in separating efficient portfolios from inefficient portfolios based on the mean-variance criterion: by looking at the expected return and variance of the portfolio, investors attempt to make more efficient investment choices—seeking the lowest variance for a given expected return, or seeking the highest expected return for a given level of variance [5]

Mean-Variance criterion

Given two variables X and Y , with mean μ_X and μ_Y and variance σ_X^2 and σ_Y^2 , respectively, it says X dominates Y if and only if the following conditions hold together:

- a) $\mu_X \geq \mu_Y$;
- b) $\sigma_X^2 \leq \sigma_Y^2$;
- c) at least one of the two inequality holds in narrower sense.

It is obvious that the mean-variance criterion not allows to detect all efficient portfolios: it could happen, in case of two efficient portfolios, that we are not able to say which of the two portfolios dominate the other.

All the efficient portfolios form the *efficient frontier*. It describes the relationship between the return that can be expected from a portfolio

²Given two random variables X and Y , the correlation coefficient measures the linear dependency between the two variables and it is defined by $\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$ and $\rho_{X,Y} \in [-1, 1]$. If $\rho = +1$ means that returns of the two variables move in the same direction and are perfectly positively correlated. If $\rho = -1$ means the returns always move in the opposite direction and are negatively correlated. If $\rho = 0$ the two assets are uncorrelated and have no relationship to each other.

This coefficient is used since the variance it is not a linear operator and thus, for the variance of the portfolio, it is necessary to take into account linear dependency existent between pair of assets. Infact, $\sigma_{X+Y}^2 = \sigma_X^2 + \sigma_Y^2 + 2\rho_{X,Y}\sigma_X\sigma_Y$.

and the riskiness (volatility) of the portfolio. The efficient frontier gives the best return that can be expected for a given level of risk or the lowest level of risk needed to achieve a given expected rate of return.

In case on N assets, Markowitz formulation for portfolio selection problem can be stated as:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \mathbf{x}^T \mathbf{V} \mathbf{x} \\ \text{subject to} \quad & \mathbf{x}^T \bar{r} = \pi \\ & \mathbf{x}^T \mathbf{e} = 1 \\ & \mathbf{x} \geq 0 \end{aligned}$$

where:

- \mathbf{x} is an N – *column* vector whose components x_1, \dots, x_n denote the weight or proportion of the wealth allocated to the i – *th* asset in the portfolio with $i = 1, 2, \dots, n$;
- \mathbf{V} is $N \times N$ matrix of covariance³ between assets;
- \bar{r} is an N – *column* vector of means returns r_1, \dots, r_N of n assets, where it is assumed that not all elements of r are equal⁴;
- \mathbf{e} is an N – *column* vector of ones;
- π is the expected rate of return that investor wishes the portfolio to select realizes;
- $\mathbf{x} \geq 0$ this last constraints requires that portions of wealth invested in each asset are non-negative, in order to avoid short-selling⁵.

This is an example of what is called *quadratic program*, an optimization problem with a quadratic utility function and linear constraints.

Markowitz problem minimizes the portfolio variance $\mathbf{x}^T \mathbf{V} \mathbf{x}$ (σ_p^2 in algebraic terms) subject to two constraints:

³We assume that \mathbf{V} is nonsingular. This essentially requires that none of the asset returns is perfectly correlated with the return of a portfolio, made up of the remaining assets; and that none of the assets is riskless.

⁴If this assumption does not hold we can not apply the mean-variance criterion, because the entire wealth would be invested in the asset with lower variance.

⁵Short-selling refers to sale of security that is not owned by the seller, or that the seller has borrowed. Short selling is motivated by the fact that security's price will decline, enabling it to be bought back at a lower price to make profit [6].

- a) $\mathbf{x}^T \mathbf{e}$ ($\sum_{i=1}^N x_i = 1$ in algebraic terms), the portfolio weights must sum unity, which means that all the wealth is invested;
- b) the portfolio must earn an expected rate of return equal to π .

The problem minimize a convex function subject to linear constraints. Note that $\mathbf{x}^T \mathbf{V} \mathbf{x}$ is convex because \mathbf{V} is *positive definite*⁶ and also that the two linear constraints define a convex set. Therefore, the problem has a unique solution and it is given by the following expression:

$$\mathbf{x} = \frac{(\gamma \mathbf{V}^{-1} \mathbf{r} - \beta \mathbf{V}^{-1} \mathbf{e})\pi + (\alpha \mathbf{V}^{-1} \mathbf{e} - \beta \mathbf{V}^{-1} \mathbf{r})}{\alpha\gamma - \beta^2}$$

where:

$$\begin{aligned}\alpha &= \mathbf{r}^T \mathbf{V}^{-1} \mathbf{r} \\ \beta &= \mathbf{r}^T \mathbf{V}^{-1} \mathbf{e} = \mathbf{e}^T \mathbf{V}^{-1} \mathbf{r} \\ \gamma &= \mathbf{e}^T \mathbf{V}^{-1} \mathbf{e}\end{aligned}$$

The analytical expression of the efficient frontier changes according to number of assets:

- For $N > 2$ the expression for the efficient frontier is still a parabola in the mean-variance space, but the vertex changes in case of a risk free asset in the portfolio;
- For $N = 2$ in addition to the presence of a risk free asset, the expression is also particularly affected by the *correlation coefficient* among the two assets.

In a portfolio of N risky assets the efficient frontier is function of $N \times (N - 1)/2$ correlation coefficients, N variances and N expected returns for each asset.

3. Third and last step consists in selecting a proper portfolio for the investor, according to his/her risk aversion, by maximizing the expected utility $E(u(.))$ ⁷ of the same investor [7].

⁶We say that an $N \times N$ matrix \mathbf{V} is positive definite, if for any nonzero N - vector \mathbf{x} , it follows that $\mathbf{x}^T \mathbf{V} \mathbf{x} > 0$.

⁷An utility function is a function U defined on the real numbers (representing possible wealth levels) and giving real value. Once the utility function is defined, all alternative random wealth levels are ranked by evaluating their expected utility values. Specifically, you compare two outcome random wealth variables x and y by comparing values $E[U(x)]$ and $E[U(y)]$; the larger value is preferred.

The utility function used by Markowitz in his model is the *Quadratic utility function*, represented by the following expression:

$$U(R_p) = R_p - \frac{a}{2}R_p^2$$

where:

- R_p is a random variable of the return of the portfolio;
- a is a strictly positive coefficient that reflects the risk aversion of investors⁸; the greater the a the higher the risk aversion.

The mean-variance criterion used in Markowitz portfolio problem can be reconciled with the expected utility approach in two ways[8]:

- a) using a quadratic utility function;
- b) making the assumption that the random variables that characterizes returns are normal (Gaussian) random variables.

It is required that the efficient frontier is compatible with maximization of the expected utility. By taking into mind variety of forms that can affect the efficient frontier, such as all the possible values that the correlation coefficient can take, the optimal portfolio for an investor can be determine by the following maximization problem:

$$\begin{aligned} & \underset{x}{\text{maximize}} && E[U(R_p)] \\ & \text{subject to} && \sigma_p^2 = f(R_p) \end{aligned}$$

The optimal portfolio is the one that maximizes the expected utility of an investor and takes into account his/her personal risk aversion. However, the selected portfolio belongs to the efficient frontier.

Portfolio with only few assets may be subjected to high level of risk, represented by the variance. The basic idea behind Markowitz model is *diversification* [9]: the author highlight that the variance of returns of portfolio decreases as number of assets N increases. As a general rule, the variance of the return of a portfolio can be reduced by including additional assets in the portfolio. This is done by selecting assets that are negatively correlated, if exists, or by adding assets with a positive correlation as close as possible to zero[10].

⁸The risk aversion of an investor represents his/her risk tolerance. The coefficient a changes with the wealth level. For many individuals risk aversion decreases as their wealth increases, reflecting the fact that they are willing to take more risk when they are financially secure [8].

Critics to Markowitz Model

Markowitz model has been widely accepted in financial theory. However, it has been criticized as well because of its strong assumptions about asset-holding behaviour of individuals and financial markets. The criticism of the mean-variance analysis is that its assumptions are unrealistic.

- It assumes that returns are normally distributed which is not always the case. In general the distributions of returns of financial instruments present skewness/asymmetry and “fat tails”⁹.
- For the quadratic utility function it assumes that investors prefers portfolio with the minimum variance for a given expected retrun [11]. The quadratic utility function is a downward parabola the present a maximum, above this the non-satiety of investors is not respected. This implies that investors prefer less than more.
- The use of the variance as a measure of risk has been criticized as well, since it is a symmetric risk measure it put equal weights on positive and negative returns (or upside potential and downside risk), but several empirical studies have shown [12] that investors treat them in different ways. The use of variance as a measure of risk is suitable only in case returns present a symmetric distribution. In addition the variance is contained in $[0; +\infty]$.
- Diversification: adding assets in a portfolio reduce portfolio’s risk. But this effect tends to decrease as additional assets are included and portfolio problem becomes complicated to solve since the size of the covariance matrix increases exponentially.
- Practical limits of Markowitz model are given by the fact that is based on assumptions that are unrealistic such as [13]:
 - the non-existence of frictional aspect like the transaction costs and taxes;
 - the non-existence of specific impositions arising from the environment in which one selects the portfolio;

⁹Skewness is a measure of the asymmetry of a probability distribution of a real-valued random variable around its mean. The skewness can be positive or negative. A negative skew means that the tail on the left side of a probability density function is longer or fatter than the right side. A positive skew means that the tail on the right side is longer or fatter than the left side.

- the non-existence of a constraints that establish that an asset can be bought or sold in a finite number.

Even though the model remain the most important contribution to modern portfolio theory, its use in practice has often been questioned [14], [15], [16]. This has led to some improvements of Markowitz model, that will be discussed in the next paragraph.

Improvements of Markowitz Model

In order to improve and make the portfolio selection problem more realistic, it is possible to follow two paths. The first consist in changing the constraints in order to make the problem more realistic, for example by adding constraints that take into account market frictions such as transaction costs and taxes or the minimum number of assets to be hold in a portfolio. These represent mixed-integer constraints. The second consist in working on the objective function, though on the measure of risk.

The portfolio selection problem that will be discussed in this work, we will mainly focus on the objective function. However, it is still useful to give an explanation of mixed-integer constraints, that makes the portfolio selection problem more realistic. These can be subdivided in three categories:

- constraints related to negotiation of minimum asset lots;
- constraints related to negotiation of maximum positive integer number of different assets;
- constraints related to negotiation of minimum positive integer number of minimum lots of a given asset.

By taking into account these mixed-integer constraints the computational investigation of the programming problem increases but at the same time the complexity in finding solution increases as well. Such mathematical programming problems are known as *NP-complete*¹⁰ and the solution of these problems is a *NP-hard*¹¹ problem.

For the purpose for the our work, we will follow the second path that consists in working on the objective function, in particular on the measure of risk.

¹⁰NP-complete are problems that are very difficult to solve with respect to time required to solve them.

¹¹NP-hard are problems that are difficult as much as *NP-complete*

As said in the previous paragraph the use of the variance as a measure of risk of a portfolio of assets is , since positive and negative deviations from the mean are treated differently from the investors. However, Markowitz himself [9] in 1959 proposed its substitution in favour of the semivariance:

$$Semi - Var(R_p) = \frac{1}{N} \sum_{i=1; R_j < \mu}^N (R_j - \mu)^2 \quad (1.1)$$

By adopting this last solution, only potential losses are considered risky, or actual returns below their expected return, and this known as *downside risk*. Any rational investor would never be averse at all to have a portfolio with a yield above expectations, therefore the notion of risk incorporated in the variance it is not suitable. When a portfolio yield above expectation we can talk about an *opportunity* and for sure not about a risk.

In [17] it is highlighted that investors perception of risk is non-linear, proof of the inadequacy of the variance: its use leads to perceive as more risky a small probability of a large loss than a high probability of small loss.

Variance is very sensitive to outliers, since it has been conceived as measure of dispersion, how much a variable deviates from its mean, without providing any information about the sign of the deviation and it is unable to grasp the connotation of asymmetry that characterized risk [17]. It put equal weights on losses and potential gains, providing a concept of risk that is strongly misleading, since only negative deviations from the mean should be taken into account. This lead to a distinction between *downside risk* and *upside potential* since the risk is conceived as something strictly negative and not as simple measure of dispersion.

The use of variance as a measure of risk is useful only in case in which returns are symmetrically distributed. Otherwise, minimizing variance in presence of positive asymmetry implies minimizing opportunity of gains as well [18].

In the literature, there are numerous efforts that aim to identify a characterization of returns as close as possible to the empirical evidence, resulting from simple time series analysis of prices of various financial assets. As pointed out by Peirò [19], Normal probability density function in general, poorly describes the yields of financial assets in the short term. The causes are mainly to be found in the leptocurtosis and in the asymmetry/skewness of the distribution. In recent years studies have been focused on the latter because it is seen that investors are willing to sacrifice part of their expected return for a more positive asymmetry, for the same variance of the return, revealing however a consistency with the assumption of risk aversion.

Although, it is possible to identify some desirable characteristic that a measure of risk should have, the concept of risk is rather subjective. It is not possible to identify a unique measure that is able to satisfy the problem of maximization of expected utility, common to each investor, once efficient frontier is determined. In addition, it is possible to determine some desirable characteristic that a measure of risk should have, with respect to a particular class of investors: rational agents.

Different paths have been followed in order to find a measure that satisfy properly investors preferences. According to [18], [20], [17] it is possible to define the measure of risk as follows:

“A measure of risk is a function ρ that assign a non-negative numeric value to a random variable X that can be interpreted as future return, $\rho : X \rightarrow R$.”

It is possible to identify basic properties that a measure of risk should have such the ρ is defined, even if they are not the only to take into account: the concept of coherent risk measure [21] benefits of a high consideration nowadays and it will be discussed later.

Desirable characteristics that a measure of risk should have are the following:

- Positivity: a measure of risk associate to a random variable a strictly positive value, at most null in case of no randomness.
- Linearity: the computational complexity of the problem tends to increase with its size, this can be reduced by linearly associating the measure of risk to its future returns: infact the “success” of many measures of risk is to find in their computational tractibility that comes from a “linearized” optimization problem, where the risk and returns are connected in a simple way.
- Covexity: a measure of risk is convex if given two random returns R_X and R_Y and a parameter $\theta \in [0, 1]$, the following relation hold:

$$\rho(\theta R_X + (1 - \theta)R_Y) \leq \theta \rho(R_X) + (1 - \theta)\rho(R_Y) \quad (1.2)$$

This property highlight the importance of diversification, since it allows to reduce total risk of the portfolio so that the invested capital is exposed to a lower level of risk. In addition it is important in optimization processes, since it guarantees that the local minimum is the global minimum as well[22].

This property can be satisfied indirectly, or in the moment in which the measure of risk satisfies the following two properties:

- subadditivity: $\rho(R_X + R_Y) \leq \rho(R_X) + \rho(R_Y)$;
- positive homogeneity: $\rho(\alpha R_X) = \alpha \rho(R_X)$, $\alpha \in R^+$

According to [18], [20], [17], fixed a certain level of desired return, the minimization process of a measure of risk aims to limit the uncertainty related to the future value of the invested capital rather than increase the latter. By doing so, the future value of the invested capital can be estimated a priori, with a degree of uncertainty that depends on the variance of returns. Accordingly, the optimal portfolio, resulting from this optimization process, can be considered as such only by a risk-averse investor and not by investor with connotations of non-satiety.

Another important measure of *downside risk*, that has been widely used in practice is the *Value at Risk (VaR)*. It has been conceived within the American investment bank J.P. Morgan and the notion of VaR is basically simple.

Definition *Value at Risk (VaR)*. Given a confidence level $\alpha \in [0, 1]$, fixed a holding period, it denotes the maximum potential loss associated to a portfolio in $\alpha\%$ of cases during the holding period.

Originally it was conceived as an index of market risk¹², and it is very popular in the credit and savings industries. Although it has the advantage of being designed by an important investment bank, and expresses the risk in the same unit measure of the capital (in monetary terms) regardless the underlying instrument, it presents several limitations when underlying losses are not expressed through Normal distributions.

From computational point of view, VaR is not particularly suitable for constrained optimization problems since it generally leads to a difficult stochastic programming problem, which is difficult to solve [22]. In addition, it is difficult to obtain a specific measure of VaR for a portfolio, except the case in which probably distributions of underlying positions are known.

From a probabilistic point of view VaR with a confidence level α is that value that satisfies the following equality: $P(L > VaR_\alpha) = 1 - \alpha$, where L is a generic distribution of losses. By following this interpretation, it emerges that VaR is a threshold measure¹³, it expresses the maximum potential loss

¹²Market risk is the risk of losses in positions that arises from movements in market prices.

¹³VaR belongs to the class of measure of risk known as *quantile-based*.

with a certain probability, and it does not provide any information on the extent of losses that exceed that threshold, thus on the nature of the left tail of profits and losses distributions exceeding VaR. Being a measure that is unconcerned about the real amount of losses above the threshold, it is possible to state that VaR is not suitable: it is not difficult to identify portfolios with the same VaR_α but with level of losses in the worst $(1 - \alpha)\%$ of dramatically different cases. A good solution to this situation has been widely discussed in the literature, that will be discussed later, and this moves the attention on which is the expected loss in $(1 - \alpha)\%$ of worst cases rather than on the threshold value.

Another drawback of VaR is that it does not behave in suitable way in reference to the aggregation of multiple sources of risk. As underlined by [21] the measure does not encourage and sometimes even forbids diversification, as it does not take into account potential economic consequences of events. In other words, VaR does not satisfy the axiom of subadditivity, because applying this measure of risk, the total risk of a financial portfolio could exceed the sum of single sources of risk of each asset. It is worth underlining how the lack of subadditivity, thus convexity, in addition to produce an inconsistency with the principle of diversification, creates problems from numerical tractability point of view. In fact, VaR can be criticized also for the incapacity to quantify the so called *tail risk*, or for its poor sensitivity towards extreme events [23].

Concept of coherent risk measure

As said in the previous paragraph in addition to desirable characteristics that a measure of risk should have there is also a growing interest for the concept of coherency of a measure of risk. These measures of risk try also to overcome the problems inherent in VaR.

Definition Coherence. A risk measure satisfying four axioms of translation invariance, subadditivity, positive homogeneity and monotonicity is called coherent [21].

- *Translation invariance:* it insures that by investing a part of available capital in a risk free asset, the total risk associated to the portfolio shrinks proportionally to α , allocated in the risk free asset:

$$\rho(X + \alpha) = \rho(X) - \alpha, \quad \forall \text{ r.v. } X, \alpha \in R$$

This implies that $\rho(X + \rho(X)) = 0$. By adding a risk free quantity equal to $\rho(X)$ to a risky position X a risk free entity is obtained, which is coherent with the interpretation of ρ as minimum positive quantity to add to the initial position in order to make this instrument acceptable (risk free).

- *Subadditivity*: represent how a measure of risk should behave in case in which it have to deal with a combination of assets: index of portfolio risk should never be greater than the sum of single indices associated to different assets of the portfolio. Subaditivity is strictly related to the concept of diversification, because it can be assumed that diversification leads to a contraction of total risk only in case in which the risk is inherent a certain position is measured through:

$$\rho(X + Y) \leq \rho(X) + \rho(Y), \quad \forall \text{ r.v. } X, Y.$$

- *Positive homogeneity*: if positions size directly influences risk (e.g., if positions are large enough that the time required to liquidate them depends on their sizes) then consequences of lack of liquidity should be considered when computing the future net worth of a position:

$$\rho(\lambda X) = \lambda \rho(X), \quad \forall \text{ r.v. } X, Y.$$

- *Monotonicity*: it highlights the preference for an asset that provides systematically return greater than another:

$$\rho(X) \leq \rho(Y), \quad \forall X, Y \text{ with, } X > Y$$

The axioms proposed in [21] are not restrictive in specifying a unique risk measure. Instead, they characterize a large class of risk measures. The choice of precisely which measure to use (within the class) should be made on the basis of additional economic considerations.

Artzner et al. in [21] provides the guidelines of concrete risk measures that satisfies the axioms of coherence. In particular they focus mainly on two measures of risk, known as *Tail Conditional Expectation (TCE)* and *Worst Conditional Expectation (WCE)*, for these they demonstrate that $TCE_\alpha \leq WCE_\alpha$.

Definition. *Tail Conditional Expectation (known also as “TailVar”)* Given a base probability measure P on Ω and a level α , the tail conditional expectation is the coherent measure of risk defined by:

$$TCE_\alpha(X) \equiv -E_P[X|X \leq -VaR_\alpha(X)].$$

Definition. *Worst Conditional Expectation.* Given a base probability measure P on Ω and a level α , the worst conditional expectation is the coherent measure of risk defined by:

$$WCE_\alpha = -inf[E_P[X|A]|P[A] \geq \alpha].$$

Financially speaking, TCE and WCE take into account “how bad is bad” because they focus on the left tail of the distribution of returns - where we find losses - and they take the average mean conditionally to the fact that losses are greater than a certain value. Infact, VaR as a measure of risk does not provide any indication on the nature of losses above a certain value represented by the measure itself. After the introduction of axioms of coherence, all the measures of risk that come later do not look at the maximum potential loss in $\alpha\%$ of cases, but which is the worst loss in $(1 - \alpha)\%$ of cases. They do not concentrate on a specific bound value, but on the whole distribution of losses above a specific bound value and they summarize the characteristics through their average mean.

Concepts of TCE and WCE represent the first proposal of coherent risk measures. TCE and WCE could seem very similar but they are not, WCE satisfy the axioms of coherence, but it is used only from a theoretical point of view, since it requires the knowledge of the entire probability space¹⁴, on the other hand TCE is definitely much easier to use in the application field, even if it does not fully satisfy axioms of coherence, since it not always subadditive¹⁵. The aim to build a coherent risk measure that could conugate positive aspects of TCE and WCE was reached through an alternative and more adequate measure of risk, known as *Expected Shortfall (ES)*. It represents the average loss in $(1 - \alpha)\%$ of worst cases.

¹⁴Quaranta and Zaffaroni (2008), highlight how WCE requires an implementation of a very difficult algorithm that is able to identify all the scenarios with high or equal probability with a significant level α .

¹⁵The coherency of TCE is guaranteed only by restricting analysis space of continuous probability functions, while it is not guaranteed in a more general case. This is the main reason behind the introduction of WCE [21].

Definition *Expected Shortfall (ES)*. Given a distribution of returns and losses X , defined the holding period and the significant level $\alpha \in \{0, 1\}$, expected shortfall is defined by:

$$ES_\alpha = -\frac{1}{\alpha}[E[X1_{X \leq x^\alpha}] - x^\alpha[P[X \leq x^\alpha] - \alpha]], \text{ where } x^\alpha = VaR$$

The second algebraic sum in the parenthesis can be interpreted as quantity to subtract to the average mean when $X \leq x^\alpha$ has a probability greater than $(1 - \alpha)$. When $P[X \leq x^\alpha] = 1 - \alpha$, in case of continuous distribution functions, the value given by the formulation of ES_α is equal to the value of TCE_α .

ES is a universal measure of risk, in the sense that it can be applied to any financial instrument and to any source of risk. The measure has properties of simplicity and completeness, since it leads to a unique value also in case of portfolios exposed to different level of risk, and robustness because it guarantees the convergence in results even if the confidence level varies of few basis points, with respect to other measure of risk. This last aspect it is not guaranteed in case of VaR, TCE or WCE.

An alternative expression for ES is the one coined by Rockafellar and Uryasev [24], denominated *Conditional Value-at-Risk (CVaR)*¹⁶.

Definition *Conditional Value at Risk*. Given a significant level $\alpha \in [0, 1]$, $CVaR_\alpha$ of the loss associated with a decision x is the value $\theta_\alpha = \text{mean}$ of the α -tail distribution of $z = f(x, y)$, where the distribution function is defined by:

$$\Psi_\alpha(x, \zeta) = \begin{cases} 0 & \text{for } \zeta < \zeta_\alpha(x), \\ [(x, \zeta) - \alpha]/(1 - \alpha) & \text{for } \zeta \geq \zeta_\alpha(x). \end{cases}$$

where $\zeta_\alpha(x)$ is equal to VaR_α associated to portfolio x .

Rockafellar and Uryasev[24], in addition to underline the coherency of CVaR¹⁷, highlight another interesting aspect: solving a simple convex optimization problem it is possible to separate $CVaR_\alpha$ and VaR_α associated to the portfolio x . It is an important result since it allows to compute CVaR

¹⁶Definition of CVaR is unique only in case in which distribution of losses is continuous; otherwise, authors make distinction between $CVaR^+$ and $CVaR^-$ (they are non-coherent), for these two the relation $CVaR^- \leq CVaR \leq CVaR^+$ can be written in an equality form only in case in which there is no discontinuity in the neighbourhood of VaR of losses distribution

¹⁷Acerbi and Tasche [25] makes us note how ES and CVaR substantially are just different labels that identify the same object, that is the expected loss in $(1 - \alpha)$ of cases.

of a position without necessarily knowing the corresponding VaR. The simultaneous computation of two measures of risk it is possible through the following formula:

$$F_\alpha(x, \zeta) = \zeta + \frac{1}{(1-\alpha)} E\{[f(x, y) - \zeta]^+\}$$

where $[f(x, \zeta) - \zeta]^+ = \max\{f(x, y) - \zeta, 0\}$.

Through a theorem they provide an important result where it is possible to determine VaR_α through two steps: first it requires the determination of set of values of ζ that minimize $F_\alpha(x, \zeta)$ and then identify its extreme left if it contains more than one point. This provide that minimizing $CVaR_\alpha$ associated to a portfolio x is equivalent to minimize $F_\alpha(x, \zeta)$ on the entire domine: $\min_{x \in X} CVaR_\alpha(x) = \min_{(x, \zeta) \in X \times R} F_\alpha(x, \zeta)$.

The latter it is an important result since it allows to work directly on a simple and convex expression with respect to the variable ζ in $F_\alpha(x, \zeta)$, instead of working on an expression that requires to know *ex-ante* the value for VaR. Empirical analysis conducted in [22] ensures how this process is implicitly possible also for the minimization of VaR_α , being $CVaR_\alpha \geq VaR_\alpha$.

A new class of coherent risk measure: two-sided

Coherent risk measures discussed in the previous paragraph are widely used and recognized, but a new class of coherent measures of risk have been introduced in recent years. This new class of risk measures are different from the exisiting one since they take into account both positive and negative deviations from the expected return.

In order to have a well describing measure of risk, Chen and Wang[3] proposed a new class of *two-sided risk measures*, that take into account higher orders moments and are built combining positive and negative deviations from the mean, or combining downside and upside of random returns.

In all the coherent risk measures seen the previous paragraph the attention is put on either the demand side or the offer side. In other words, only one side of the distribution is considered in the associated risk measure. However, from the game theory¹⁸ point of view, investors should take into account action of the seller of a stock when he/she wants to buy that stock.

¹⁸Game theory is the study of mathematical model of conflict and cooperation between intelligent ration decision-makers[26]. It is a zero-sum game: mathematical representation in which each individual gain or loss of utility are exactly balanced by losses or gains of the other individual.

This implies that attitudes towards potential losses is taken into account from both, demand and offer side. In other words, to suitably measure investment risk and thereafter to make robust investment decision, a more general “two-sided” risk measures should be adopted[3].

Main limitations of existing measures of risk, as pointed out by Chen and Wang [3] are to find in their symmetry - in the sense that semivariances are used for measuring both below and above deviations from the mean - and non-coherency for example *VaR*.

Compared with the existing risk measures, this new class of risk measures present the following advantages [3]:

- The whole domain loss distribution information is used, which makes the measure superior for finding robust (with respect to both tradingsides) and stable (with respect to the estimation error) investment decisions;
- By suitably selecting the convex combination coefficient and the order of the norm of the downside random loss, it is easy for this risk measure to reflect investor’s risk aversion and control the asymmetry and fat-tails of the loss distribution.

Another important aspect of this measure of risk is that its value is easy to compute and to apply in order to find the optimal portfolio. For the purpose of our work this new class of two-sided risk measures are used and will be discussed deeply in Chapter 3.

Chapter 2

Particle Swarm Optimization and Artificial Bee Colony

In the following chapter Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) are discussed. Since both, PSO and ABC, are bio-inspired, evolutionary computation (metaheuristics), some basic knowledge of metaheuristics and swarm intelligence are provided in order to give a better understanding to the reader.

In the second part analytical formulation, the main variants, the differences and the field of application of the two algorithms are provided.

2.1 Bio-inspired metaheuristics

Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) algorithms belongs to the so called bio-inspired metaheuristics. Bio-inspired metaheuristic can be based on swarm intelligence, biological systems, physical and chemical systems. PSO and ABC belong to the first category, so for the purpose of our work we will give an explanation of swarm intelligence.

Heuristics and Metaheuristics

In many operational research problems, from mathematical programming to combinatorial optimization, exist a variety of *exact methods* to find optimal solution and they represent an important tool to solve a large number of real-world problems.

In some cases, problem characteristics, such as its size, or realworld requirements, a very limited computation time allowed, make exact methods not suitable for tackling large instances of a problem. To overcome such prob-

lems researchers and practitioners adopt approximate methods, known as *heuristics* and *metaheuristics* [27].

Heuristic means “to find” or “to discover” by trial and error. *Heuristic* methods give a good solution to a tough optimization problem in a reasonable time, although there is no guarantee that optimal solutions are reached. This is good when we do not necessarily want the best solutions but rather good solutions which are easily reachable [28]. It is often an iterative algorithm, in which at each iteration the algorithm search for the new best of the best solutions found so far. When an appropriate stopping criterion is satisfied the algorithm stops and provides an optimal solution, the best solution determined in all the iterations performed. A drawback of heuristic methods is that they are usually problem-specific as they exploit the property of the problem [29]. It means that each method is developed to solve a specific problem rather than a general of problems. To overcome this problem in the last years *metaheuristic*¹ methods have been developed.

In reality, *Metaheuristics* are further development of heuristic methods, they are high level solving techniques based on which approximate algorithms for combinatorial optimization problems can be designed and implemented [31]. In fact, metaheuristics are problem-independent techniques that provide a set of guidelines or strategies to develop heuristic methods that can be adapted to fit the needs of most real-world optimization problems. Heuristics or metaheuristics are therefore developed specifically to find a solution that is “good enough” in a time that is “small enough” [32]. The ability to find good solutions where other methods fail has made the metaheuristics the method of choice to solve majority of large real-life optimization problems, both in academic research and practical applications. There are different ways to classify and describe metaheuristic algorithms. Depending on the characteristics chosen to differentiate them, different classifications are possible, each of them being a result of a specific viewpoint [27]. The two main components that determine the behaviour of any metaheuristic algorithm are: intensification or diversification, and exploitation or exploration². A metaheuristic will be successful on a given

¹The term metaheuristic was coined by Glover in 1986, meta- means “beyond” or “higher level” and heuristic means “to find” or “to discover” by trial and error. Metaheuristics can be considered as a “master strategy that guides other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality” [30].

²Two major components of any metaheuristic algorithms are: intensification and diversification, or exploitation and exploration [27]. Diversification means to generate diverse solutions so as to explore the search space on a global scale, while intensification

optimization problem if it can provide a balance between exploitation and exploration of the search space to identify regions with high quality solutions in a problem specific, near optimal way. Concerning the exploration and exploitation of the search space metaheuristic can be classified into two categories [33]:

- **Trajectory-based** metaheuristics: techniques that start with a single initial solution and, at each step of the search, the current solution is replaced by another solution, often the best, with respect to the one found in its neighborhood so far. They allow to find locally optimal solution, so they are *exploitation-oriented* methods promoting intensification in the search space [27].

The most well know metehuristic families that belong to this category are simulated annealing (SA) [34], tabu search (TA) [30] and variable neighborhood search (VNS) [35].

- **Population-based** metaheuristics: techniques that make use of a population of solutions. The initial population is generated randomly, and then enhanced through an iterative process. At each generation of the process, the whole population, or a part of it, is replaced by newly generated individuals, often the best ones. These techniques are called *exploration-oriented* methods, since their main ability resides in the diversification in the search space [27].

The most well known metaheuristic families based on the manipulation of population of solutions are evolutionary algorithms (EAs) [36], ant colony optimization(ACO) [37], particle swarm optimization (PSO) [38], artificial bee colony (ABC) [39] and differential evolution (DE) [40].

Fundamental properties of metaheuristics can be summarized as follows [27]:

- Metaheuristics are strategies that guide the search process;
- They are not problem-specific;

means to focus the search in a local region knowing that a current good solution is found in this region. A good balance between intensification and diversification should be found during the selection of the best solutions to improve the rate of algorithm convergence. The selection of the best ensures that solutions will converge to the optimum, while diversification via randomization allows the search to escape from local optimum and, at the same time, increases the diversity of solutions. A good combination of these two major components will usually ensure that global optimality is achievable [32].

- The aim of metaheuristics is to explore the search space in order to find near-optimal solutions;
- They are usually non-deterministic and approximate methods;
- They provide mechanisms to avoid getting trapped in confined search space area;
- Metaheuristics provide basic concepts that permit an abstract level description;
- Today’s advanced metaheuristics are based on search experience (memory) that guide the search.

“A metaheuristic is a set of concepts that can be used to define heuristic methods that can be applied to a wide set of different problems. In other words, a metaheuristic can be seen as a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem [41].”

Swarm Intelligence

Swarm Intelligence is an important concept of Artificial Intelligence (AI) and computer intelligence (CI)³.

The term swarm is used for an aggregation of animals like birds, fishes, insects such as ants, termites and bees performing collective behaviour [43]. A swarm is a large number of homogenous, unsophisticated agents that interact locally among themselves and their environment, without any central control to yield a global behaviour to emerge [44]. Although a single ant or a bee is not a smart individual, their colonies are smart. When it comes to deciding what to do next, ants or bees do not have any idea. However, as a swarm they can find the shortest way to the best food source, allocate workers to different tasks and defend the territory from invaders without having the need of a control or manager. As individual they are tiny dummies, but they answer quickly and effectively to their environment as colonies.

³There exists many definitions in the literature of AI, the most widely used belongs to John McCarthy, who defined it as “the science and engineering of making intelligent machines” [42]. Computational science, is a new research area and commonly referred to as AI as well. It is defined as the study of the design of intelligent agents where an intelligent agent is a system that perceives its environment and then takes actions to maximize its chance of success. CI techniques are considered as AI techniques, but there is a difference between them. CI includes a set of nature-inspired computational methodologies and approaches to address complex problems of the real world applications.

The term “Swarm Intelligence” was first introduced in 1988 by Beni, Hackwood and Wang [45] as a set of algorithms for controlling robotic swarm.

Swarm intelligence [46] is defined as a collective behaviour of decentralized and self-organized swarms. Well known examples of swarm are bird flocks and the colony of social insects such as ants and bees. The intelligence of the swarm lies in the networks of interactions among these simple agents, and between agents and their environment.

In the world there are so many kind of swarms. It is not possible to call all of them intelligent or their intelligence level could be vary from swarm to swarm. The key concept of the swarm system is self-organization which results from the collective behavior resulting from interactions among simple agents. Bonabeau [47] interpreted the self-organization in swarm through four characteristics:

- *Positive feedback*: promoting the creation of convenient structures.
- *Negative feedback*: counterbalancing positive feedback and helping stabilize the collective pattern.
- *Fluctuations*: random walks, errors, random task switching among swarm individuals which are vital for creativity. Randomness is often significant for emergent structures since it enables the discovery of new solutions.
- *Multiple interactions*: agents in the swarm use the information coming from the other agents so that the information spreads throughout the network.

In addition to these characteristics, performing tasks simultaneously by specialized agents, called *division of labour* is also an important feature of a swarm as well as self-organization for the occurrence of the intelligence [47]. In order to call a swarm intelligent, according to Mark Millonas [48], it must satisfy the following principles:

- *Quality principle*: a swarm should be able to respond to different factors in the environment that is.
- *Principle of diverse response*: a swarm should not commit its activity along excessively narrow channels.
- *Principle of stability*: a swarm should not change behaviour upon every fluctuations the environment, but only when it is worth.

- *Principle of adaptability*: a swarm must be able to change its behaviour when needed.

SI is becoming increasingly important research area for computer scientists, engineers, economists, bioinformaticians, operational researchers, and many other disciplines since the problems that the natural intelligent swarms can solve (finding food, dividing labour, building nests, etc.) have important counterparts in several engineering areas of real world [43]. The two main SI approaches that we are going to discuss and apply for our work are Particle Swarm Optimization (PSO) based on the social behaviour of bird flocking, and Artificial Bee Colony (ABC) based on bee swarm intelligence.

2.2 PSO - Particle Swarm Optimization

Particle Swarm Optimization is a population-based global optimization technique developed by Eberhart and Kennedy in 1995. It belongs to Swarm Intelligence (SI) and it is inspired by the social behavior of bird flocking. It has been applied in many fields due to its unique searching mechanism, simple concept, computational efficiency and easy implementation [49].

In PSO a fixed number of autonomous entities, defined as *particles*, are randomly generated in the search space. Each particle is a candidate solution to the problem, and it is associated with a velocity and a location in the search space. In addition, each particle explores a specific area of the search space, keeping track of the best position reached so far in a memory, and shares this information with a part or the whole swarm. A swarm consists of N particles (number of possible solution of a given problem) flying around in a $D - dimensional$ (or number of variables of a given problem) search space. The quality of particle position, considered as a candidate solution of optimization problem, is measured by the *fitness function* (or objective function). The key concept of the PSO lies in the experience-sharing behavior in which the experience of each particle is continuously communicated to part or the whole swarm, leading the overall swarm towards the most promising areas and the global best position detected so far in the search space. Interaction and exchange of information among particles of a swarm depend on how they are connected to their neighbourhood: population topology. Since the success of the PSO depends also this aspect further, different type of population topologies will be illustrated.

In the initialization phase of PSO, the positions and velocities of individuals are randomly initialized.

At $k - th$ step of the algorithm each particle is characterized with three $D - dimensional$ vectors:

- x_i^k , represents the current position of the $i - th$ particle at step k ;
- v_i^k , represents the current velocity of the $i - th$ particle at step k ;
- p_i , represents the personal best position ($pbest$) reached so far by the $i - th$ particle;
- p_g , represents the global best position ($gbest$) reached so far by any particle of the swarm.

In addition, $pbest_i = f(p_i)$ is the value of the fitness function evaluated in the personal best position visited by the $i - th$ particle, and $gbest = f(p_g)$ is the corresponding fitness value of p_g . At each iteration, the current position x_i^k of each particle is considered a possible solution of an optimization problem. If that position is better than any that has been found so far by the same particle, this is stored in the vector p_i and the value of the fitness function associated to this is known as $pbest_i$. The aim is to find new, better locations that improve the value of the fitness function. In the case of a minimization problem we should have something like as follows:

$$p_i^{k+1} = \begin{cases} p_i^k & \text{if } f(x_i^{k+1}) \geq f(p_i^k) \\ x_i^{k+1} & \text{if } f(x_i^{k+1}) < f(p_i^k) \end{cases}$$

and we want the function value to be as small as possible, viceversa in the case of a maximization problem.

At each iteration, a particle i adjust its position x_i^k and velocity v_i^k , based on the best position p_i^k it has visited so far in its flight and the global best position p_g found by any another particle in its neighborhood [50].

A general structure of the original PSO can be explained in few steps:

1. Initialize a population of particles with random positions and velocities on D dimensions in the search space;
2. Begin loop (stopping criterion is not satisfied);
 - For each particle i evaluate the desired fitness function $f(x_i^k)$ in the current position;

- Compare the value of the fitness function with its $pbest_i$: if it current value is better than $pbest_i$, then set $pbest_i$ equal to the current value, and p_i equal to the current position x_i^k ;
- Identify the particle that has the best fitness value. The value of its fitness function is identified as $gbest$ and its position as p_g ;
- Update the velocity and the position of all particles by the following equations:

$$\begin{cases} v_i^{k+1} = v_i^k + U(0, \phi_1) \otimes (p_i^k - x_i^k) + U(0, \phi_2) \otimes (p_g - x_i^k) \\ x_i^{k+1} = x_i^k + v_i^{k+1} \end{cases}$$

where:

- $U(0, \phi_1)$ and $U(0, \phi_2)$ are two vectors with random numbers uniformly distributed in $[0, \phi_1]$ and $[0, \phi_2]$ and randomly generated at each iteration and for each particle;
- p_g is global best position of the swarm;
- \otimes is component-wise multiplication;
- If the criterion is met (usually a sufficiently good fitness or a maximum number of iterations) exit loop.

3. End loop.

The first term v_i^k is the *current velocity*, responsible for keeping the particle moving in the same direction it was originally heading.

The second term, $U(0, \phi_1) \otimes (p_i^k - x_i^k)$, is called the *cognitive component*. It works as the particle's memory, causing to tend to return to the regions of the search space in which it has experienced high individual fitness. $U(0, \phi_1)$ is a random number with uniform distribution in the range of $[0, 1]$.

The third and the last term, $U(0, \phi_2) \otimes (p_g - x_i^k)$, called *social component*, cause the particle to move to the best region the swarm has found so far. $U(0, \phi_2)$ is a random number with uniform distribution in the range of $[0, 1]$.

The original version of PSO has a very few number of parameters that need to be fixed. The first one is the number of particles (the size of the population). This is often set empirically, depending on the size and the complexity of the problem, but values in the range 20 – 50 are usually common [51].

Other parameters that need to be fixed in order to facilitate the convergence of the algorithm appear in the update velocity equation and they are illustrated in the following section.

Parameter Selection for Particle Swarm

When implementing the particle swarm algorithm, several considerations must be taken into account in order to facilitate the convergence and prevent an “explosion” of the swarm. These considerations include selecting acceleration parameters and limiting the maximum velocity.

Selecting acceleration parameters

The behavior of a PSO changes radically depending on the values assigned to the two acceleration parameters, ϕ_1 and ϕ_2 . These two parameters respectively, known as *cognitive coefficient* and *social coefficient*, control the movement of each particle towards its personal best p_i^k and global best position p_g , respectively. By changing ϕ_1 and ϕ_2 one can make the PSO more or less responsive and even unstable, with particle speed increasing without control. Small values limit the movement of the particles, while large numbers may cause the particles to diverge. In general the maximum value for this constant should be $\phi = 4.0$, meaning $\phi_1 + \phi_2 = 4$. A good starting point has been proposed [46], [52] to be $\phi_1 = \phi_2 = 2$. It is important to note that ϕ_1 and ϕ_2 should not necessarily be equal since the “weights” for individual and group experience can vary according to the characteristics of the problem [53]. Since the value of ϕ_1 and ϕ_2 influence particles velocity, this need to be controlled as well.

Selection of maximum velocity

At each iteration step, the algorithm proceeds by adjusting the distance (velocity) that each particle moves in every dimension of the problem hyper-space. The velocity of the particle is a stochastic variable and is, therefore, subject to creating an uncontrolled trajectory, making the particle follow wider cycles in the problem space [54][55]. In order to avoid these explosion, upper and lower limits can be defined for the velocity v_i [46]:

- if $v_{id} > V_{max}$ then $v_{id} = V_{max}$
- else if $v_{id} < -V_{max}$ then $v_{id} = -V_{max}$

The usual method for preventing explosion is simply to define a parameter V_{max} and prevent the velocity from exceeding it on each dimension d for individual i . Most of the time, the value for V_{max} is selected empirically according to the characteristics of the problem. It is important to note that

if the value of this parameter is too high, then the particles may go beyond a good solution; on the other hand, if V_{max} is too small, then particle's movement is limited and the optimal solution may not be reached.

This allow particles to oscillate within bounds, although with no tendency for convergence or collapse of the swarm toward a point. Even without converging, the swarm's oscillations do find improved points in the optimal regions [46]. Even this choice seems not exempt from criticism, infact the parameter V_{max} required some care since it appeared to influence the balance between exploration and exploitation.

One approach to controlling the search process is the implementation of "inertia weight". Another method that will be discussed later.

Modifications of PSO

Although PSO is one the most promising techniques to find a good solution much faster than the other algorithms, it suffers from the premature convergence. PSO may fail to find global optimum when the problem is complicated and complex. Convergence property may cause the swarm to be trapped into local optimum and not be able to explore other promising area. In order to solve the problem of premature convergence and to improve the performance of the algorithm, the original scheme has to be modified [56]. In this section we will illustrate the most important modifications of the PSO used so far.

Inertia Weight approach (IWA)

Motivated by the desire to better control the scope of search and reduce the importance of V_{max} Eberhart and Shi in 1998 [57] proposed a new modification of the original algorithm by introducing a new parameter ω , called *inertia weight* [51]. It was introduced to control the exploration and exploitation abilities of the swarm, and to converge the swarm more accurately and efficiently compared to the traditional PSO velocity update equations with:

$$\begin{cases} v_i^{k+1} = \omega v_i^k + U(0, \phi_1) \otimes (p_i^k - x_i^k) + U(0, \phi_2) \otimes (p_g - x_i^k) \\ x_i^{k+1} = x_i^k + v_i^{k+1} \end{cases}$$

The inertia weight can be either implemented as a fixed value or dynamically changing. If $\omega \geq 1$, then the velocities increase over time and particles hardly change their direction to move back again towards optimum and the swarm diverges. If $\omega \leq 1$, then quick changes of direction are to set in process. If $\omega = 0$, particles velocity vanishes and all particles

move without knowledge of the previous velocity in each step [58]. For fixed values of ω PSO algorithm performs well, although, changing inertia values is used because this parameter controls the exploration and exploitation of the search space.

Commonly, a linearly decreasing inertia weight has produced good results in many optimization problems [59], [60]. To control the balance between global and local exploration, to obtain quick convergence, and to reach an optimum, the inertia weight whose value decreases linearly with time is set according to the following expression:

$$w = w_{max} - \frac{w_{max} - w_{min}}{K} \times k$$

where:

- w_{max} and w_{min} are desired maximum and minimum values of the inertia weight;
- K is the maximum allowed number of iterations (decided by the user).
- k is the current iteration number;

Generally, ω decreases linearly from 0.9 to 0.4 over the entire process. Van den Bergh and Engelbrecht, Trelea have defined a condition that $\omega > \frac{1}{2}(\phi_1 + \phi_2) - 1$, guarantees convergence [61]. If this condition is not satisfied divergent or cyclic behaviour can occur in the process.

The inertia weight method is very useful to ensure convergence. However, it still has a drawback, that once the inertia weight is decreased, it cannot increase if the swarm needs to search new areas, so the method is not able to recover its exploration mode [53].

Naturally, other strategies can be adopted to adjust the inertia weight. For example, in [51] the adaptation of ω using a fuzzy system was reported to significantly improve PSO performance. Another effective strategy is to use the inertia weight with random component, rather than time-decreasing. For example [51] successfully used $\omega = U(0.5, 1)$. There are also studies [51], in which an increasing inertia weight was used to obtaining good results.

Constriction factor approach (CFA)

Constriction factor model is another variant of PSO and it was proposed by Clerc in 1999 [62]. It incorporates a new parameter χ in the update velocity equation v_i^k , called *constriction factor*. The new about the parameter is that it influences, not only v_i^{k+1} but also $pbest$ and $gbest$. The update velocity equation proposed by Clerc is the following:

$$\begin{cases} v_i^{k+1} = \chi[v_i^k + U(0, \phi_1) \otimes (p_i^k - x_i^k) + U(0, \phi_2) \otimes (p_g - x_i^k)] \\ x_i^{k+1} = x_i^k + v_i^{k+1} \end{cases}$$

where the constriction factor is given by the following expression:

$$\chi = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \text{ and } \phi_1 + \phi_2 = \phi > 4.0$$

the new position is updated as in the original PSO algorithm.

Given the update velocity equation of the constriction factor approach, the algorithm works only if the parameter ϕ is greater than 4. Typically, when this approach is used, ϕ is set equal to 4.1 and the constant χ is thus 0.7289. This implies the the previous velocity v_i^k is being multiplied by 0.7289 and the two $(p - x)$ terms are being multiplied by 1.49445 ($= 0.7295 \times 2.05$).

For instance if $\phi = 4.1$ then $\chi = 0.7298$, if $\phi = 5.0$, then $k = 0.38$. Since k is smaller than unity if $\phi > 4.0$, convergence of the swarm always occurs [63].

PSO with constriction factor is equivalent to PSO with inertia, $w^t = \chi$, with $\phi_1 + \phi_2 = \phi$ and $\phi \geq 4$. When Clerc's constriction method is used, ϕ is commonly set to 4.1, $\phi_1 = \phi_2$, and the constant multiplier χ is approximately 0.7298. This is equivalent to the algorithm the inertia weight $w \approx 0.7298$ and $\phi_1 = \phi_2 = 1.49445$

The mathematical analysis in constriction factor approach (CFA) is unlike other evolutionary computations, and it is known that CFA obtains better results than Inertia weight approach (IWA) when neither the global best position p_g nor the individual best position p_i changes [63].

Fully informed Particle Swarm

In the original version of PSO, the sources of influence of the particle are mainly two: personal and global best. Information that come from the remaining neighbours are not used. Mendes has revised the way particles interact with their neighbours [64] by introducing a new variant of the PSO, known as *Fully Informed Particle Swarm (FIPS)* [65]. In the classical algorithm each particle is influenced by its own previous performance and the single best performance of its neighborhood, in fully informed particle swarm, the particle is affected by all its neighbours, sometimes without any influence from its own previous experience. Following this logic *FIPS* can be formulated as follows [65]:

$$v_i^{t+1} = \chi[v_i^t + \frac{1}{K} \sum U^t \otimes (p_{nbr_k^t} - x_i^t)]$$

where:

- K_i is the number of neighbours for particle i ;
- nbr_k is the i 's k - th neighbour.

The new position is updated as in the traditional version of the PSO:

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

Fully informed particle swarm appears to find better solutions in fewer iterations than the traditional PSO, but the main drawback is, it is much more dependent on topology of the population.

Population topology

The PSO algorithm is based on social interaction among particles of the entire swarm, they communicate with each other by exchanging information about the success of each particle in the swarm. When a particle in the whole swarm finds a better position, all particles move towards that one. This performance of the particles is determined by the particle's neighbourhood[61]. Since the neighbourhood determines the extent of social interaction within the swarm and influences a particular particle's movement, it must be defined for each particle [61].

When the neighbourhoods in the swarm are small, less interaction occurs [61], in this case the convergence will be slower but it may improve the quality of solutions. For large neighbourhood, the convergence will be faster but the risk is that convergence may occur earlier [66]. Researchers have worked on developing this performance by designing different types of neighbourhood structures [58]. In PSO, particles have been studied in two general type of neighbourhoods [46]:

- *global best (gbest)*: particles are attracted to the best solution found by any member of the swarm. This represents a fully connected network in which each particle has access to the information of all other members in the swarm.
- *local best (lbest)*: each particle has access to the information corresponding to its immediate neighbours, according to some swarm topology.

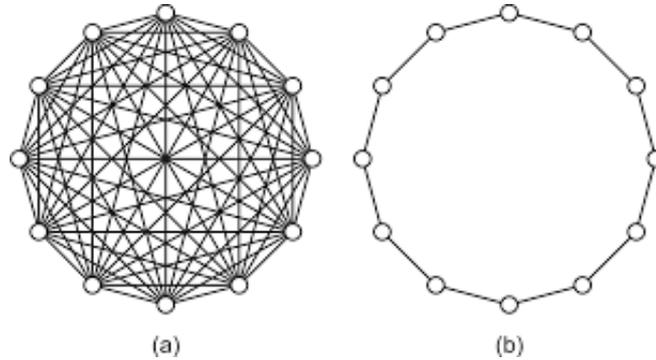


Figure 2.1: (a) Global best (b) Local best

The most common topologies are the following:

- *Star topology*: consists of only one central particle that is connected to the others, and all information are communicated through this. This particle compares the best performance of all particles in the swarm, and adjust its position towards the best performance particle. Then the new position of the central particle is shared with all the particles.
- *Ring topology*: each particle is connected to exactly two others particles in its neighbours, forming a single pathway - a ring. In this process a particle that find the better solution passes the information to its immediate neighbours, and these two neighbours share this information with their immediate neighbours, until it reaches the last particle. In this way the best solution found is spread very slowly around the ring by all particles. Convergence is slower, but larger parts of the search space are covered than with the star topology.

Kennedy [67] suggests that the *gbest*, converges fast but maybe be trapped in a local optimum, while the *lbest* network has more chances to find an optimal solution, although with slower convergence.

There are more different topologies such as Von Neumann topology, pyramid topology and so on. Kennedy and Mendes [68] suggested that the Von Neumann configuration may perform better than other topologies including the *gbest* version. Nevertheless, selecting the most efficient neighbourhood depends on the type of problem.

2.3 ABC - Artificial Bee Colony

The first comprehensive survey on the algorithms related to the bee SI and their applications was proposed by Karaboga and Akay[69] in 2009. Depending on the different intelligent behaviours of honey bee swarms many algorithms have been proposed in the last decade[43]. Like in any other population-based optimization algorithm, ABC consists of a population of potential solutions. In the ABC algorithm, the positions of food sources represent potential solutions in search space. The amount of nectar for a food source indicates the quality or the fitness of the associated solution. Before introducing the ABC algorithm we give an explanation about the behaviour of honey bees in nature.

Honey Bees in nature

ABC algorithm is inspired from the foraging behaviour of honey bees. Honey bee swarm follows collective intelligent manner, while searching the food source. It has many qualities like how to communicate the information, memorize the environment, store and share the information and how to take decisions based on that. By looking at the environment, the swarm updates itself. The behaviour of real honey bees can be summarized in the heads namely, food sources, employed bees, unemployed bees, foraging behaviour and dances.

Food sources

While searching the food, bee selects a particular flower (called food source) for itself. From this food source it collects information about the amount of nectar that it contains, how easily the nectar can be extracted, how far and in which direction it is from the nest. Bee stores all these facts in its memory as a vector.

Employed bees

The available food sources are exploited by a particular group of bees, called employed bees. They find the food source, then evaluate its profitability (fitness) and share this information with bees waiting in the hive.

Unemployed bees

Group of bees that employed bees share their information with are called unemployed bees. These are responsible to summarize the information they

get from employed bees and select a food source to exploit. Unemployed bees are further divide into two groups: onlooker bees and scout bees.

- Onlooker bees collect information from the employed bees in the hive and after analyzing the data, they select the food source to exploit for themselves.
- Scout bees are responsible for finding the new food sources around the hive.

When some of existing food sources exhaust then these bees start searching the environment around the hive and find new food sources randomly. In a honey bee swarm, usually on average half of the bees are employed and the other half are unemployed bees and 5% to 10% of the total bees are scouts.

Foraging behaviour

The most important characteristic of the real honey bee swarm is *foraging behaviour*. In this process, bee leaves the hive and start searching for the food source. When it gets a food source, it extracts the nectar from it and stores it in its stomach. It extracts nectar till 30-120 minutes according to conditions like richness, distance of food source from hive. Then with the secretion of enzymes in her stomach, honey making process starts and she unloads nectar in empty cells after reaching to the hive. Finally she shares her information with other bees in the hive through various types of dance.

Dance

In order to tell the other bee in the hive how good, far and in which direction her food source is, employed bee performs a dance on different parts of hive area. Based on the profitability of food source, employed bee performs one of the following dances:

- Round dance: it is performed when the food source is near to the hive.
- Waggle dance: it is performed to inform other bees about the direction of the food source with respect to some conditions.
- Tremble dance: if a bee takes a longer time to unload the nectar she starts trembling and it indicates that she does not know about the profitability of the food source.

ABC - Basic approach

ABC algorithm is a new swarm intelligent optimization technique inspired by the behavior of honey bees and it was initially proposed by Karaboga [39] to solve unconstrained problems. ABC algorithm later was extended for solving constrained optimization problems.

In an artificial bee colony, bees try to maximize the nectar amount unloaded to the food stores in the hive by division of labour and self-organization, these represents two essential elements of swarm intelligence. Compared with other swarm intelligence algorithms, the swarm in the ABC consists of three groups of bees: *employed bees* (constitute the first half of the swarm,), *onlooker bees* and *scout bees* (they constitute the second half of the swarm)[70]. Each of them plays different role in the process:

- Employed bees, they exploit the food sources explored and share information with other bees in the hive about the quality (fitness) of the food source they are exploiting;
- Onlooker bees wait in the hive and decide food source to exploit depending on the information shared by employed bees;
- Scout bees are responsible for finding new food source depending on an internal motivation or external clues or randomly[71].

In the ABC algorithm first half of the colony consists of the employed artificial bees. Employed bees, whose food source is exhausted, becomes scout bee and searches for new food source randomly. Each food source is exploited by only one employed bee. It means that the number of employed bees is equal to the number of food sources around the hive. Employed bees exploit the food sources in their neighborhood and share the information of food source with the onlooker bees. By watching the dance of employed bees, onlookers easily determine the amount of nectar of food sources provided by the employed bees. It means that employed bees are associated with food sources while onlooker bees are those bees that stay in the hive and use the information gathered from employed bees to decide the food source to choose. Once employed bee food source is exhausted it becomes a scout.

In every successful algorithm, the exploitation and exploration process are very important. In the ABC algorithm, the exploitation process is carried out by the employed bees and onlookers, it means that they exploit the food source until it is exhausted, while the scout bees execute the exploration process of new food sources once the food sources found so far

are exhausted or can not be exploited. ABC process is organized in four phases: initialization phase, employed bees phase, onlooker bees phase and scout bee phase, each of which is explained below.

Initialization of the population

ABC algorithm, as iterative algorithm, generates a randomly distributed initial population $P(G = 0)$ of SN solutions (that represent food source positions), where SN denotes the size of the population. Each solution x_i ($i = 1, \dots, SN$) is a D -dimensional vector, where D is the number of optimization parameters (variables) and x_i represents the i -th food source in the population. At the initialization stage, a set of food sources is randomly selected by the bees through the following expression:

$$x_{i,j} = lb_j + rand(0, 1)(ub_j - lb_j), \text{ for } i = 1, \dots, SN \text{ and } j = 1, \dots, D$$

where $x_{i,j}$ is the j -th parameter of the i -th solution in the population, $rand(0, 1)$ is a random real number between 0 and 1, and ub_j and lb_j are upper and lower bound of the j -th parameter respectively [72].

Employed Bees Phase

In the initialization phase, to each randomly selected food source an employed bee is associated. In the Employed bee phase, bee updates its position (solution) in its memory depending on the individual experiences and the quality (nectar amount) of the new food source (new solution).

Employed bees search for new food sources position $v_{i,j}$, having more nectar within the neighbourhood of the food source $x_{i,j}$ in the memory. Once they found a new food source in the neighbour they evaluate its profitability (fitness).

Update of food source position

In employed bee phase, the update of new food source position for j -th parameter of i -th candidate solution is given by the following expression:

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j})$$

where:

- $x_{i,j}$ is the j -th parameter of the old solution i ;
- $x_{k,j}$ is the j -th parameter of the neighbour solution k ;

- $\phi_{i,j}$ is a random number between $[-1, 1]$, it controls the production of neighbour food sources around $x_{i,j}$ and represents the comparison of two food positions visually by a bee;
- $k = 1, \dots, SN$, with $k \neq i$ and $j = 1, \dots, D$, are randomly chosen indexes. k must be different from i so the step size $\phi_{i,j}(x_{i,j} - x_{k,j})$ that represents the random perturbation, has some significant contribution, since it avoids getting stuck into the local minimum.

As we can see from the expression, x_i is subtracted from x_k of the same $j - th$ parameter, this difference is multiplied by a random number $\phi_{i,j} \in [-1, 1]$. Finally this quantity is added to the $j - th$ parameter of x_i to get $j - th$ parameter of new food position $v_{i,j}$. This $v_{i,j}$ is a vector of the same dimension of x_i . As the difference between x_i and x_k decreases, the perturbation on the position x_i decreases as well. Therefore, as the search approaches to the optimum solution the difference between x_i and x_k adaptively reduces.

If the nectar amount of the new food source $v_{i,j}$ is greater than the old food source $x_{i,j}$, the bee memorizes the new position and forgets the old one. Otherwise she keeps the position of the old one in her memory. After the search process is completed by all the employed bees, they share the nectar information of the food sources and their positions with onlookers waiting on the dance area.

Onlooker Bees Phase

After completion of the employed bees phase, the onlooker bees phase is started. In the selection phase, onlooker bee evaluates the nectar information provided by employed bees and chooses the food source depending on the probability, p_i , associated to its fitness. The probability p_i , is estimated using the following expression:

$$p_i = \frac{Fit_i}{\sum_{i=1}^{SN} Fit_i}$$

where Fit_i with $i = 1, 2, \dots, N$ is the fitness (profitability) value of the $i - th$ solution in the swarm, thus fitness is proportional to the nectar amount of the food source in the position i . It is calculated by the following equation:

$$Fit_i = \begin{cases} \frac{1}{1+f_i} & \text{if } f_i \geq 0 \\ 1 + |f_i| & \text{if } f_i < 0 \end{cases}$$

where f_i is the objective function value for $i = 1, 2, \dots, N$, which is specific for the optimization problem.

Update of food source position

As in the case of employed bee, onlooker bee updates its position based on the information obtained from the employed bees and checks the fitness of the candidate source. If the new food source position has a fitness value greater or equal than the old one, then the old food source position is replaced by the new one. Otherwise the old one is retained.

Scout Bees Phase

If the position of a food source is not updated for a predetermined number of cycles, then the food source is assumed to be abandoned and scout bee phase is started.

Avoidance of Suboptimal Solutions

In this phase the bee associated with the abandoned food source becomes scout bee and the food source is replaced by the randomly chosen food source within the search space.

In ABC, the value of predetermined number of cycles is an important control parameter and it is called “limit” for abandonment. If we assume that the abandoned source is x_i then the scout bee starts for discovering a new food source to be replaced with x_i as follows:

$$x_{i,j} = lb_j + rand(0, 1)(ub_j - lb_j), \text{ with } i = 1, \dots, SN \text{ and } j = 1, \dots, D$$

where ub_j and lb_j are upper and lower bound of the j – *th* parameter respectively. After $v_{i,j}$ is calculated, each candidate source position is produced and computed by the ABC algorithm, position’s performance is compared with its old one. If the new food source has an equal or greater nectar than the old source, it is replaced with the old in the memory. Otherwise, the old is retained in the memory.

The framework of the standard ABC algorithm is the following:

1. Initialize the population

2. Set $cycle = 1$
3. **While** $cycle \leq MCN$ **do**
4. Update food source solutions by employed bees
5. Select feasible solutions by onlooker bees
6. Update food source solutions by onlooker bees
7. Avoid suboptimal solutions by scout bees
8. Set $cycle = cycle + 1$
9. **end while**

ABC selection processes

ABC algorithm employs four different selection processes:

1. a *global selection process*, which is used by the onlooker bees for discovering promising areas through p_i ;
2. a *local selection process*, which is carried out in the region by employed bees and the onlooker bees depending on local information, for determining a neighbour food source around the source in the memory as defined by $v_{i,j}$;
3. another *local selection process* called *greedy selection process*, which is carried out by all bees. If the nectar amount of the candidate source is better than that of the present one, the bee forgets the present one and memorizes the candidate source, otherwise the bee keeps the present one in the memory;
4. a *random selection* process carried out by scouts.

Control parameteres of ABC

Now that we gave an explanation of the algorithm, it is clear that there are mainly three parameters of the ABC that need to set: number of food sources, *limit* and $\phi_{i,j}$. As stated Karaboga[39] and Akay [73], ABC performance is very sensitive to the choice of the parameter $\phi_{i,j}$ and *limit*. The setting of these control parameters is suggested by Karaboga[39][70]:

- $\phi_{i,j}$ represents the weight to the difference between current food source and randomly selected food source, and it is responsible to maintain the proper diversity mechanism⁴ of the search procedure in ABC, usually is a uniformly distributed random number in the range $[-1, 1]$.
- $limit = \frac{SN}{2}D$ is responsible to explore new areas of search space. After employed and onlooker bees phases are checked whether there is any food source which is exhausted. If there exists an exhausted food source then it is abandoned and replaced with a new food source discovered by the scout. In ABC, at each iteration only one food source can be exhausted, in other words only one bee can be a scout bee.
- SN represents the colony size. The colony of ABC consists of employed bees, onlooker bees and scout bees, in which the number of onlooker bees is equal to the number of employed bees and scout bee is one of the employed bee whose food source is exhausted. From the literature it can be concluded that as SN increases, the algorithm leads to better results. However, after a certain value for SN any increment in its value does not improve the performance of the ABC significantly. Karaboga and Bastruk [74] have proved through experimental results for different test problems and suggested that SN between 50 and 100 can provide an acceptable convergence.

Another parameter that plays an important role in ABC algorithm, is p_i generated for probabilistic selection. This probabilistic selection is a function of fitness values of the current population members.

ABC modifications

Many researches have been carried out in order to make ABC efficient and widely applicable in different types of problems. Since its inception researchers have improved the algorithm in many ways:

1. introducing new strategies⁵ in ABC and fine tuning of existing control parameters
2. introducing new control parameters in ABC;

⁴Are mechanisms used to handle the constraints of an optimization problem.

⁵For example by changing the initialization scheme [75] for the employed bees, or changing selection mechanism for the onlooker bees, in order to maximize the exploitation capacity [76] and improve convergence ability of the algorithm.

3. hybridisation of ABC with other population-based probabilistic or deterministic algorithms
4. miscellaneous⁶.

Modified Artificial Bee Colony

ABC was originally used to solve unconstrained problems. Karaboga and Bastruk have compared the performance of the ABC algorithm with those of other modern metaheuristic algorithms on unconstrained problems in [78]. Later it has been modified to solve constrained optimization problems and its performance has been investigated on constrained problems [78]. When working with constrained problems, the convergence rate of the algorithms is poorer [69].

Frequency of the perturbation

One of the modifications in the ABC algorithm is controlling the frequency of perturbation. In the basic version of ABC this frequency is fixed. In order to overcome the slow convergence rate of the basic version of the ABC this issue Akay and Karaboga [73] proposed the modification of the algorithm by introducing a new control parameter, modification rate MR . For each $x_{i,j}$ a uniformly distributed random number, $0 \leq R_{i,j} \leq 1$, is produced and if the random number is less than MR , the parameter $x_{i,j}$ is modified through the following equation:

$$v_{i,j} = \begin{cases} x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}), & \text{if } R_{i,j} < MR, \\ x_{i,j}, & \text{otherwise} \end{cases}$$

where $k = 1, \dots, SN$ is randomly chosen index. k is determined randomly and it has to be different from i ; $\phi_{i,j}$ is a uniformly distributed random real number in the range $[-1,1]$; R_j is uniformly distributed random real number in the range $[0,1]$ and $j = 1, \dots, D$; MR (modification rate) is a control parameter of ABC algorithm in the range $[0,1]$. A lower value of MR may cause solutions to improve slowly, on the other hand a higher value

⁶ABC is hard to apply to optimization problems where the variables can have binary values only Kashan [77] proposed DisABC for binary optimization. In real world optimization problems, we have to optimize more than one objective function Zou [2] proposed multiobjective artificial bee colony(MOABC) to solve multiobjective optimization problems.

may cause higher diversity in a solution and hence in the population[70]. Usually it is predefined by the user in the beginning.

In the basic ABC the random perturbation $\phi_{i,j}(x_{i,j} - x_{k,j})$ is added to the current solution $x_{i,j}$ in order to get the new solution $v_{i,j}$, while in the modified ABC version $\phi_{i,j}$ can be replaced by a new control parameter called scaling factor, SF that varies within $[-SF, SF]$. Hence the magnitude of the perturbation is controlled by SF . A lower value of SF allows the search to fine tune the process in small steps while causing slow convergence. A larger value of SF speeds up the search, but it reduces the exploitation capability of the perturbation process[70].

After generating a new solution, instead of greedy selection, in order to understand the impact of MR in ABC, the problem constraints are handled through Deb's rule[79]. Deb's method consists of very simple three heuristic rules, it uses a tournament selection operator, where two solutions are compared at a time by applying the following criteria:

- Any feasible solution is preferred to any infeasible solution;
- Among two feasible solutions, the one having better objective function value is preferred;
- Among two infeasible solutions, the one having smaller constraint violation is preferred.

Since starting with a feasible initial population is a very time consuming process, ABC algorithm does not consider the initial population to be feasible.

Probability calculation

Karaboga and Akay[80] proposed a second modification regarded the probability calculation method for onlooker bees. After all employed bees complete the search process, they share the nectar information of the food sources and their position information with the onlookers on the dance area by calculating probability values. Since infeasible solutions are allowed to populate the colony, a modification is needed here to assign probability value for infeasible solutions as well as for feasible solutions. They used the following assignment of calculation probability for onlooker bees:

$$p_i = \begin{cases} 0.5 + \frac{fitness_i}{\sum_1^{SN} fitness_i} \times 0.5, & \text{if solution is feasible} \\ 1 - \frac{violation_i}{\sum_1^{SN} violation_i} \times 0.5, & \text{if solution is infeasible.} \end{cases}$$

where $violation_i$ is a penalty value of the solution x_i when we have a constraint violation of solution; $fitness_i$ is the fitness value associated to the solution x_i , which is proportional to the nectar amount of the food source. The fitness value is defined as:

$$fitness_i = \begin{cases} \frac{1}{1+f_i} & f_i \geq 0 \\ 1 + |f_i| & f_i < 0 \end{cases}$$

Probability values of infeasible solutions are between 0 and 0.5 while those of feasible ones are between 0.5 and 1. By a selection mechanism, solutions are selected probabilistically proportional to their fitness values in the case of feasible solutions and inversely proportional to their violation values in case of infeasible solutions.

An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount. As employed bees, onlookers update their position according to $v_{i,j}$.

After all onlookers are distributed, food sources that are not worth to exploit anymore are determined. If a solution can not be improved through a predetermined number of cycles, "limit", it is abandoned. The food source abandoned by the bees is replaced with a new food source discovered by the scouts. This is still done by generating solutions randomly.

Scout production period (SPP)

Another difference between unconstrained and constrained version of the ABC is that artificial scouts are produced at a predetermined period of cycles for discovering new food source randomly. This period is another control parameter called scout production period (*SPP*). At each *SPP* cycle, it is checked if there is an abandoned food source exceeding *limit* or not. If so, the scout production process is carried out. *SPP* provides a diversity mechanism that allows new and probably infeasible individuals to be in the population.

In overall, the modified ABC algorithm employs two more control parameters in order to improve the coverage capability of ABC algorithm. These two parameters are *MR* and *SPP*. Another modification is to use Deb's selection instead of greedy selection mechanism.

Chaotic Initialization

Generally speaking, if no information about the solution is available, then random initialization is the most commonly used method to initialize the population. However, this may affect the algorithm performance on the convergence speed and the quality of the final solution. Recently chaotic initialization method is proposed to increase the population diversity and improve the global convergence, and some good results have been shown in many applications [81]. Chaotic initialization methods ensures better distribution in search space. Chaos is apparently random and unpredictable but it also presents regularity, and it is sensitive to initial conditions and parameters. Mathematically speaking, chaotic maps can be considered as source of randomness. Because of these randomness and sensitivity dependence on the initial conditions of chaotic maps, it has been considered as an initialization method for the heuristic algorithms to improve the global convergence by escaping form local optimum [82][83]

A simplest chaotic map is logistic map, whose equation is the following:

$$C_i^j = \mu C_{i-1}^j (1 - C_{i-1}^j)$$

where μ is a control parameter and C_i is called chaotic number, where i is the number of the colony size and j is the dimension. $c_i^j = (0, 1)$ under the conditions that the initial $C_0 = (0, 1)$. In particular, C_i^j behaves as chaotic dynamics when $\mu = 4$ and $C_0 = (0, 0.25, 0.5, 0.75)$.

Using chaotic initialization, random initialization equation of the basic ABC is replaced by the following expression:

$$x_i^j = x_{min}^j + (x_{max}^j - x_{min}^j) * C_i^j$$

All the other steps of the algorithm remains the same as the basic ABC.

Modification in Onlooker Bee Phase

The basic algorithm updates the food source according to $x_{i,j}$. Because of the neighbour food source is randomly chosen, the fitness of the neighbour food source may be higher. In addition, the updating equation does not change as the iteration processes. Thus, the original updating mechanism cannot help bee to find the optimum solutions [84].

As we know, onlookers and employed bees find feasible solution according to $x_{i,j}$, but in some cases the convergence performance of the algorithm is not good. To overcome this drawback, Wei Chan proposed a new search method to produce new solutions, in which two new factors are considered: *forgetting factor* and *neighborhood factor*.

This lead to the following expression for:

$$v_{i,j} = \varphi x_{i,j} + \psi \phi_{i,j} (x_{i,j} - x_{k,j}),$$

where:

- $\varphi = \omega_{max} - \frac{iteration}{MCN} (\omega_{max} - \omega_{min})$
- $\psi = \omega_{min} + \frac{iteration}{MCN} (\omega_{max} - \omega_{min})$

φ is the forgetting factor and it expresses the memory strength for current food source and it decreased gradually. $x_{k,j}$, the quality of the neighborhood source, affects the convergence speed and quality of the final solution. ψ factor is introduced to accelerate the convergence speed by adjusting the radius of the search for new candidates. ω_{max} and ω_{min} represent the maximum and minimum percentage of the position adjustment for the employed bees or onlookers. With these values, the value of φ will linearly decrease and the of ψ will linearly increase.

In the scout bee phase, if a food source is abandoned, scouts produce a new food source by applying $x_{i,j}$. However, at some extent it has the defects of prematurity and stagnation, this is why the chaotic search method is used to get out of local optima.

Although ABC has great potentials, it is clear that some modifications to the original structure are still necessary in order to significantly improve its performance. In order to improve the performance of ABC in terms of convergence, new neighbour production mechanisms can be proposed.

Like all other evolutionary optimization approached, ABC also has some drawbacks. For example since it does not uses an operator like crossover as employed in GA or DE, the distribution of a good information between solutions is not at a required level. This causes the convergence performance of ABC for local minimum to be slow. This topic can be searched and its convergence performance can be improved [43].

Other algorithms related to the bee SI

The first comprehensive on the algorithms related to the bee SI and their applications was prepared by Karaboga and Akay [69]in 2009. They survey

shows that many algorithms have been developed by researchers depending on different intelligent behaviours of honey bee swarms. Some known algorithms based on bee SI are the following [43]:

- *Virtual Bee algorithm (VBA)*, developed by Yang [85] in 2005, to solve numerical function optimizations. In this model a swarm of virtual bees are generated and they are allowed to move randomly in the search space and these bees interact when they find some target nectar. Nectar sources correspond to the encoded values of the function and solution is given by the intensity of exploration through bee interactions.
- *BeeAdHoc (BAH)* algorithm was defined by Wedde and Farooq [86] (2005), it is a routing algorithm⁷ for energy efficient routing in mobile ad-hoc networks. In the algorithm honey-bees starts with a solitary colony, represented by a single queen without a family.
- *BeeHive algorithm (BHA)*, was proposed by Webbe [87] in 2004 and it has been inspired by the communication in the hive of honey bees and applied to the routing in networks.
- *Bee system (BS)*, introduced by Lucic and Teodorovic [88] in 2001 for solving difficult combinatorial optimization problems.
- *Bee colony optimization (BCO)*, it was described by Teodorovic and Dell'orco [89] in 2005 for the ride-matching problem⁸, for the routing and wavelength assignment (RWA)⁹ in all-optical networks¹⁰.

⁷Routing is a process that aim to select best paths in a network. Routing can be performed on many kind of networks, such as telephone network, electronic data networks and transportation networks.

⁸Ride matching problems deal with ridesharing systems, drivers and riders that decide to share their trips with each other in order to share costs etc. This problem consists in matching a drivers' offers and riders' requests based on their sources, desinations and timing.

⁹Routing and wavelength assignment (RWA) problem is optical networking problem (a communication at distance using light to carry information) that aim to maximize the number of optical connections.

¹⁰An optical communication network uses transmitter, which encodes the message into an optical signal, a channel, which carries the signal to its destination, and a receiver, which reproduces the messages from received optical signal [90]

2.4 PSO and ABC comparison

In this section we will illustrate some difference between the two algorithms, PSO and ABC, that are subjects of our work. PSO and ABC presents some *common parameters* and some *specific parameters* that are related to the algorithm itself.

Common parameters

Both PSO and ABC share two common parameters that are:

- Population size;
- Maximum number of cycle (MCN), in the case of ABC, or maximum number of iterations, in the case of PSO.

Specific parameters

In addition to these common parameters, PSO and ABC presents their own parameters that need to be set at beginning of the searching process.

- PSO parameters:
 - Cognitive and social parameters (ϕ_1 and ϕ_2) used to change the weighting between personal and population experience, respectively.
 - Inertia weight¹¹, which determines how the previous velocity of the particle influences the velocity in the next iteration.
- Basic ABC parameters:
 - *limit*, which is a control parameter. The expression for determining the value of *limit* depends on the population (colony size) and dimension of the problem D .

The basic ABC algorithm has only one control parameter (*limit*) apart from the colony size (SN) and Maximum cycle number (MCN). Consequently, ABC is as simple as flexible as PSO, also employ less control parameters.

¹¹This in case of PSO based on Inertia weight approach. In the case of different approaches metioned previously for the PSO we will have different parameters.

There can be found also some comparative studies on the performance of PSO and ABC. In [91] PSO, ABC and BFO¹² algorithms are applied to economic load dispatch problem (ELD) and the results show that ABC and BFO converge to optimal fuel cost with reduced computational time when compared to PSO.

In [93] ABC, PSO and GA¹³, another population-based algorithm, are applied to solve “Nuclear Power plant” problem. The ABC algorithm obtained results compatible with GA and PSO.

2.5 Field of Application

PSO can be and has been used across wide range of applications. Areas where PSOs have shown particular promise include multimodal problems and problems for which there is no specialized method available or all specialized methods give unsatisfactory results. Many research on PSO can be found, to improve its performance, to modify it for various objectives and to apply it to various problems. Over 1600 articles can be found in IEEE publications alone, over 450 articles are found in CiNii¹⁴ database which reflects research activity in Japan (<http://ci.nii.ac.jp/en>), and more than 177.000 web pages in relation with PSO exist throughout the world according to Google. These numbers reflect the ever-increasing interest of researchers and engineers in PSO.

On the other hand Artificial Bee Colony is just 6 years old. Despite the fact that it is younger compared to PSO the growth of this field has exceeded the expectations. It has been used in many applications in several different fields. One of the most interesting application area is training neural networks [96]. ABC was also used by some researchers to solve optimization problem of electrical, mechanical and civil engineering [97]. The number of publications in the literature have increased exponentially. Both PSO and ABC are applied many fields such as:

¹²The Bacterial Foraging Optimization (BFO) algorithm belongs to the field of Bacteria Optimization algorithms and Swarm Optimization. It is inspired by the social foraging behaviour of bacteria, in particular *Escherichia coli*. Bacteria search for nutrients in a manner to maximize energy obtained per unit time. Individual bacterium communicates with others by sending signals[92].

¹³Genetic Algorithm (GA) is a search heuristic that mimics the process of natural selection. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generates solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover [94]

¹⁴CiNii is a bibliographic database service for material in Japanese academic libraries, especially focusing on Japanese works in English works published in Japan[95].

- image and video analysis applications;
- design and restructuring of electricity networks and load dispatching;
- control applications;
- application in electronics and electromagnetics;
- antenna design;
- power generation and power system;
- finance and economics;
- music generation and games.

In order to make the two algorithms even more powerful, both have been combined with some traditional and evolutionary optimization algorithms as well. Shi [98] proposed a novel hybrid swarm intelligent algorithm based on PSO and ABC. In the literature we can find many papers related to this topic.

Chapter 3

Two-sided risk measure application

In this chapter two-sided risk measure implementation for a portfolio selection problem is illustrated. The two-sided risk measure proposed by Chen and Wang [3] different from the existing coherent risk measures discussed in Chapter 1, since it considers both positive and negative deviations from the expected return. This innovation makes it easy to reasonably describe and control the asymmetry and fat-tail characteristics of the loss distribution and to properly reflect the investor's risk attitude. In this work a realistic portfolio selection model proposed by Corazza, Fasano and Gusso [1] is considered.

3.1 Two-sided risk measures and its properties

As said in chapter 1, in the classical model proposed by Markowitz, the risk is measured by the variance of retruns. But if returns are not normally distributed, this may lead to a not good investment decision, since the variance puts equal weights on losses and potential gains, providing a concept of risk that is strongly misleading. Since the risk is conceived as something strictly negative and not as simple measure of dispersion, this need distinction between *downside risk* (risk contained in the “bad tail”, the left side of portfolio returns) and *upside potential* (the gains contained in the “good tail”, the right side of same portfolio returns). In order to deal with this issue many risk measures have been proposed over the years. In this work we will deal with a new measure of risk, two-sided risk measure, that can be regarded as improvement of the preliminary coherent risk measures

proposed in [99], [100] and as a variant of deviation in [101].

The two-sided risk measure [3] is constructed by properly combining the downside and upside of the random payoff. The risk is defined by a random payoff of Y , which is a real valued random variable on a probability space (Ω, F, P) of some asset or portfolio. If $Y \geq 0$ we have a random profit and if $Y < 0$ we have a random loss. Measuring risk is then equivalent to establishing a correspondence ρ between the space of Y and the set of real numbers. Y is assumed to be a p -integrable¹. Thus, Y can be treated as element of the space $L^p(\Omega, F, P)$ ² for $1 \leq p \leq \infty$. Then follows that $\rho(Y)$ is considered to be the minimum extra cash added to Y that makes the position acceptable for the holder [21]. This implies that:

- for a positive $\rho(Y)$ investor has to add $\rho(Y)$ amount of extra cash to ensure the acceptance of his future position;
- for a negative $\rho(Y)$ investor can withdraw $-\rho(Y)$ amount fo money without affecting the acceptance of his future position.

Let us define $\|Y\|_p = (E[|Y|^p])^{1/p}$, where $E[\cdot]$ represent the expected value of a random variable with $p \in [1, +\infty]$. Let Y^- deonote $\max(-Y, 0)$ and Y^+ denote $(-Y)^-$. Lastly define, $\sigma_p^\pm(Y) = \|(Y - E[Y])^\pm\|_p$.

As we can see, in the new two-sided risk measure both side of the returns distribution are considered. Relative to the expected value $E[Y]$, the random variable $(Y - E[Y])^-$ represent the “downside” of Y and the random variable $(X - E[Y])^+$ is the corresponding “upside” of Y . $\rho_{a,p}(Y)$ is an improvement of one-sided risk measures in [102], [100] and [103] since it considers both, lower and upper partial moments of Y , respectively $\sigma_p^-(Y)$ (bad volatility) and $\sigma_p^+(Y)$ (good volatility).

Then the measure of risk introduced in [3] in given by the following defintion.

Definition 3.1. Given $p \in [1, +\infty)$, $0 \leq a \leq 1$, the new risk measure $\rho_{a,p}: L^p \rightarrow R$ is determined by

$$\begin{aligned} \rho_{a,p}(Y) &= a\sigma_1^+(Y) + (1-a)\sigma_p^-(Y) - E[Y] \\ &= \frac{a\|(Y - E[Y])^+\|_1 + \|(1-a)\|(Y - E[Y])^-\|_p - E[Y]}{1} \end{aligned} \quad (3.1)$$

¹A real-valued function $f(x)$ of a real variable x on an the interval $[a, b]$ is defined as: $\int_a^b f(x)dx$. If the function has an integral, it is said to be integrable.

²The L^p spaces are function spaces defined using the generalization of p -norm for finite dimension vector spaces. Let $p \geq 1$ be a real valued number, the p -norm is defined as follows: $\|x\|_p := (\sum_{i=1}^n |x_i|^p)^{1/p}$. As p approaches ∞ the p -norm approaches the infinity norm or maximum norm ($\|x\|_\infty$).

where:

- $a \in [0, 1]$ is a *global risk factor* that reflects the desired balance between good volatility and bad volatility of an investor;
- p is a *local risk factor* that increase proportionally with the risk aversion of the investor. In addition it includes information regarding the distribution of returns, as asymmetry and kurtosis.

This new measure of risk $\rho_{a,p}(Y)$ is generated by first taking the 1–norm of the positive deviation and the p –norm of the negative deviation, respectively, and then taking the convex combination of these two norms. Two-sided risk measure (3.1) is different from the other coherent risk measures [101] as $\|(Y - E[Y])^+\|_1$ and $-E[Y]$ are simultaneously included in (3.1). The expression $-E[Y]$ in (3.1) ensures that $\rho_{a,p}(Y)$ has good properties, such as coherency. From the characteristics of the parameters, two important properties are defined for $\rho_{a,p}(X)$. It is non-decreasing with respect to p and non-increasing with respect to a . The non-decreasing property of $\rho_{a,p}$ with respect to p means that, the greater the p , the larger the $\rho_{a,p}$, that is the investor adopting this risk measure will treat Y riskier than the investor adopting $\rho_{a,p}$ with a smaller p . In addition, for large values of p we would have a strongly-risk averse investor. On the other hand, the non-increasing property of $\rho_{a,p}$ with respect to a means that for large value of a a less risk-averse investor would be connected. In addition when $a = 1$, $\rho_{a,p}$ collapse to absolute deviation measure.

Moreover, since $E((Y - E[Y])^+) = E((Y - E[Y])^-)$, when the investor minimizes $\|(Y - E[Y])^+\|_1 = E((Y - E[Y])^+)$, he/she actually minimized his/her dispersion from the mean. This is the main difference between one-sided coherent risk measures and two-sided risk measures in (3.1). The advantage of $\rho_{a,p}(Y)$ is the fact that it can be used to find realistic and robust optimal portfolios with respect others coherent risk measures.

Since $\|(Y - E[Y])^+\|_1 = \|(Y - E[Y])^-\|_1$, $\rho_{a,p}(Y)$ can be treated as a convex combination of coherent risk measures $\|(Y - E[Y])^+\|_1 - E[Y]$ and $\|(Y - E[Y])^-\|_p - E[Y]$ mentioned in [104] and [101], because convex combination of coherent risk measures is still a coherent risk measure [3].

3.2 A realistic portfolio selection model under two-sided risk measure

Once an appropriate measure of risk is defined, it is important to introduce a series of constraints in order to formulate a realistic portfolio selection model.

It is important to notice that the searching process of portfolio that minimizes any measure of risk does not lead to a solution, that can be used for practical purposes, if not accompanied with constraints to be followed. A generic rational investor has to make a choice by pursuing conflicting goals, such as maximizing returns and minimizing risks, in the overall search to maximize its expected utility.

Constrained risk minimization in terms of expected returns that takes into account only the budget constraint³, leads to an optimal solution that is given by the minimum risk portfolio⁴. Without going into the logics and reasoning of expected utility theory, it is clear that this efficient solution is one of the many possible solutions and it does not take into account risk aversion of each investor and conveyed through a specific utility function.

The model that takes into account just the budget constraint it is too simplistic with respect to the financial market reality. For this reason, in the proposed model real aspects of financial markets are taken into account. These are market frictions such as transactions costs, or costs that are charged by the intermediary agent to an investor when a transaction is ended, and taxes. These costs will be tracked indirectly by introducing cardinality constraints and it important to include them since they represent one of the major categories of constraints that a manager is subjected to in daily practice [1].

However, it is important to keep in mind that the computational complexity of the problem increases as the number of constraints increase. The computational complexity is even more bigger if constraints depart from linear and/or continuous forms. The resulting problem is known to be *NP-hard*, and in order to solve it the need of metaheuristics emerges, since search of optimal solution through exact methods could not lead to a solution.

³Risk minimization without any constraint will inevitably lead to a vector of zero rates for investment, since no percentage of wealth will be allocated in an asset with random return.

⁴It is known as the *global minimum variance portfolio* in the mean-variance approach of Markowitz, or a portfolio with the minimum variance of returns

Before illustrating the portfolio selection model that will be used for the purpose of our work, single constraints of the problem are discussed.

Budget and return constraints

The most important constraints of the problem are *budget* and *return*, since they represent its essential part. These two constraints are included in the constrained Markowitz model and are common to many portfolio selection model.

$$\sum_{i=1}^N x_i = 1 \quad (3.2)$$

$$\sum_{i=1}^N r_i x_i \geq \pi \quad (3.3)$$

Budget constraint (3.2) requires that all the available capital must be invested, where x_i represent the percentage of capital invested in the i -th asset. In matrix terms budget constraint can be expressed as $\mathbf{x}'\mathbf{e} = \mathbf{1}$.

Return constraint (3.3) requires that the expected return of the portfolio can not go below the desired return π of the investor. This allows to select the portfolio from all those lying on the efficient frontier that minimizes the measure of risk adopted for the portfolio, in our case two-sided risk measure. In algebraic terms, the mean return of the portfolio is equal to $\mu_p = \sum_i^N x_i r_i$, whereas in terms of matrix it is given by $\mathbf{x}'\mathbf{r}$. In the specific selection problem that is being proposed it is not required that the mean return of portfolio is exactly equally to π , but a broader condition is established in order to avoid that the feasible region results empty. On the other hand, any rational investor would not be disappointed by a financial portfolio that, for a given level of risk, provide a return that exceeds their expectations.

From a conceptual point of view, analogous results are reached in case in which it is decided to maximize the return for a given maximum level of risk, instead of minimizing the risk. However, from the point of view of a risk-averse investor, the choice to set the objective function to optimize with a measure of risk (to minimize) rather than a measure of returns (to maximize) further reflects investor's risk aversion, which tends to focus his/her attention on risk rather than return. An example of this emphasis

towards risk is represented by the so called MIFID ⁵ survey, done by lending institution to test both, the knowledge and experience of each client regard asset and financial instruments and investment objectives. This allows to have a financial profile of client-investor and on the basis of this profile a proper investment choice for the investor is formulated. The questions of survey are mainly related to risk aversion aspects of investors rather than propensity of gains.

Cardinality constraints

Cardinality constraints are those related to the number of asset to include in the portfolio or small subset of available asset. The introduction of this constraint is due to have an indirect control on transaction cost and to meet the needs of fund managers, who in their daily practice have to face the problem to build a portfolio by choosing from several hundreds of assets. When the number of assets selected is too large, many practical problems can occur, such as high dimensionality of the problem, which can increase transaction costs as well. By using the following cardinality constraint, we consider transaction costs in our portfolio selection problem.

$$K_d \leq \sum_{i=1}^N z_i \leq K_u, \text{ where } 1 \leq K_d \leq K_u \leq N \quad (3.4)$$

where $z_i \in \{0, 1\}$ represent a binary variable, $z_i = 1$ implies that the i -th asset is included in the portfolio, if $z_i = 0$ otherwise. K_d and K_u respectively are the minimum and the maximum number of assets included in the portfolio.

Furthermore, it is required that each of the selected asset can not be too small or too large fraction of the portfolio. This consists in establishing a priori minimum and maximum fractions respectively, d_i and u_i to allocate in each asset:

$$z_i d_i \leq x_i \leq z_i u_i, \text{ where } 0 \leq d \leq u \leq 1 \quad (3.5)$$

In order to guarantee the compatibility between (3.4) and (3.5), the parameters d and u must satisfy the following inequalities: $d \leq \frac{1}{K_d}$ and

⁵MIFID (Markets in Financial Instruments Directive) is 2004/39/EU directive, introduced in 2007. It aims to increase investor protection and to guarantee the maximum level of transparency through mandatory information to customer.

$$u \geq \frac{1}{K_u}.$$

Now that we have illustrated all the constraints that should be taken into account in order to have a more realistic portfolio, we can describe the portfolio selection model used for the purpose of our work.

Portfolio selection model

Suppose we have N assets to choose from, and for $i = 1, \dots, N$ let $x_i \in R$ be the weight of the i -th asset in the portfolio, with $X^T = (x_1, \dots, x_n)$. Let $Z^T = (z_1, \dots, z_n) \in \{0, 1\}^N$ be a binary vector. In addition, be r_i a real valued random variable for $i = 1, \dots, N$ that represents the return of the i -th security, and \hat{r}_i its expected value⁶. Then the random variable R that represents return of the whole portfolio can be expressed as:

$$R = \sum_{i=1}^N x_i r_i \tag{3.6}$$

and its expected value as:

$$\hat{R} = \sum_{i=1}^N x_i \hat{r}_i \tag{3.7}$$

Considering the (3.1), our goal is to minimize $\rho(R)$, subjected to the constraints previously discussed. The overall resulting portfolio selection problem is the following:

$$\underset{X, Z}{\text{minimize}} \quad \rho_{a,p}(R)$$

subject to

$$\hat{R} \geq \pi$$

$$\sum_{i=1}^N x_i = 1$$

$$K_d \leq \sum_{i=1}^N z_i \leq K_u$$

$$z_i d_i \leq x_i \leq z_i u_i, \text{ with } i = 1, \dots, N$$

$$z_i \in \{0, 1\}, \text{ with } i = 1, \dots, N$$

⁶ $\hat{r}_i = E(r_i) = \frac{1}{T} \sum_{t=1}^T r_{i,t}$. It is possible to calculate this quantity by its sample estimator

In the problem, in addition to the classical constraints on budget and return, two more constraints are considered: cardinality on the number of assets to include in the portfolio and minimum and maximum percentages of investment for each asset[1].

The resulting portfolio selection problem is a nonlinear and nonconvex mixed-integer problem, which in general admits several local solutions. But we want to possibly find global solutions and not local minimizers. Exact methods could be heavily time consuming this is why metaheuristique techniques, PSO and ABC are applied to our problem.

Constraints management

As said in the previous chapter, PSO and ABC are born as unconstrained metaheuristic techniques, but they can still be applied to constrained optimization problems by appropriately reformulating the constraints. The key issue for constrained optimization problems is related to how to deal with them.

The strategy proposed by Parsopoulos and Vrahatis [105] consist in introducing a penalty function. This method allows to transform a constrained optimization problem into an unconstrained one by penalizing all the violations of the constraints and adding them into a unique fitness function that takes into account the objective function and the initial constraints. The fitness function obtained through this method will be optimized by applying an algorithm designed to solve unconstrained problems, as constraints have been “integrated” to the fitness function.

Penalty function

Even though there are different strategies that can be adopted to deal with constraints, in the wake of [1] it is decided to choose the *exact penalty method*.

In general, this method aims to solve constrained optimization problems through unconstrained problems. The approximation is computed by adding to the objective function a term that put a high cost in case of violation of constraints. A parameter, ϵ measures the seriousness of the penalty, so the fitness of unconstrained problem in approximating the original constrained problem.

According to Luenberger [106], the basic idea of this method consist in substituting a generic constrained minimization problem:

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && h_j(x) = 0, \quad j = 1, \dots, m \\ & && g_i(x) \leq 0, \quad i = 1, \dots, p. \end{aligned}$$

where:

- $f(x)$ is a continuous function R^n ;
- $h(x) = 0$ is a vector of m constraints expressed in equality form;
- $g(x) \leq 0$ is a vector of p constraints expressed in inequality form.

through an equivalent unconstrained minimization problem as:

$$\min f(x) + \epsilon P(x) \tag{3.8}$$

where:

- ϵ is a positive constant;
- $P(x)$ is a function in R^n with the following characteristics:
 - $P(x)$ is continuous;
 - $P(x) \geq 0 \forall x \in R^n$;
 - $P(x)$ if and only if $x \in S$

The author underline the main issues that we have to deal with when using this method. First it is regarded to “how well” the unconstrained optimization problem approximate the constrained one. As ϵ tends to infinite, we have to check if the solution of the unconstrained problem converges towards the constrained problem. In any case, this is not the issue for analysing of our problem, since it is decided to choose a particular family of penalty methods, that ensures the biunivocal correspondence between the solutions of two problems. From a practical point of view, the second important aspect is related to the resolution method to adopt for the objective function of the the unconstrained problem the include the penalty term.

From a theoretical point of view, by taking into account a generic constrained minimization problem shown before, if it is decided to approximate this problem through an unconstrained one by applying the following penalty function:

$$P(x) = \sum_{i=1}^m |h_i(x)| + \sum_{j=1}^p \max(0, g_j(x)) \tag{3.9}$$

then the following theorem guarantees the correspondence between solutions of unconstrained problem, that exploit the penalty function, and the original constrained problem.

Theorem of exact penalty. Suppose that x^* is a point that satisfies sufficient conditions of the second order to be a local minimum point of the constrained problem. Let λ and μ be two vectors containing the associated Lagrange multipliers, with respect to m constraints in $h(x) = 0$ and to p constraints in $g(x) \leq 0$. Then, for $\epsilon > \max\{\lambda_i|\mu_j : i = 1, \dots, m, j = 1, \dots, p\}$, x^* is the local minimum also for the unconstrained problem and that uses the previously indicated type of penalty function.

The demonstration of the theorem can be seen in [106], but it is important to underline how the theorem does not give any information on the value of the penalty parameter ϵ but merely provides sufficient conditions, such that the solutions of the unconstrained problem coincide with the constrained one. The problem related to this aspect is dealt in the following section.

Reformulation of portfolio selection problem

In order to provide a better understanding of exact penalty method, we first provide the original portfolio selection problem and then apply the method, as done by Corazza et al. [1].

The formulation of the original problem is the following:

$$\begin{aligned}
 & \underset{X, Z}{\text{minimize}} && \rho_{a,p}(R) \\
 & \text{subject to} && \\
 & && \hat{R} \geq \pi \\
 & && \sum_{i=1}^N x_i = 1 \\
 & && K_d \leq \sum_{i=1}^N z_i \leq K_u \\
 & && z_i d_i \leq x_i \leq z_i u_i, \text{ with } i = 1, \dots, N \\
 & && z_i \in \{0, 1\}, \text{ with } i = 1, \dots, N
 \end{aligned}$$

We reformulate the original problem by using the exact penalty method. First of all constraints can be reformulated in the following way:

- $\hat{R} \geq \pi \quad \rightarrow \max\{0, \pi - \hat{R}\} = 0$

- $\sum_{i=1}^N x_i = 1 \quad \rightarrow |\sum_{i=1}^N x_i - 1| = 0;$
- $K_d \leq \sum_{i=1}^N z_i \quad \rightarrow \max\{0, K_d - \sum_{i=1}^N z_i\} = 0$
- $\sum_{i=1}^N z_i \leq K_u \quad \rightarrow \max\{0, \sum_{i=1}^N z_i - K_u\} = 0$
- $z_i d_i \leq x_i \quad \rightarrow \sum_{i=1}^N \max\{0, z_i d - x_i\} = 0$
- $x_i \leq z_i u_i \quad \rightarrow \sum_{i=1}^N \max\{0, x_i - z_i u\} = 0$
- $z_i \in \{0, 1\} \quad \rightarrow \sum_{i=1}^N |z_i(1 - z_i)| = 0^7$

and the reformulating problem becomes:

$$\min_{X, Z; \epsilon} P(X, Z; \epsilon)$$

where:

$$\begin{aligned} P(X, Z; \epsilon) = & \rho_{a,p}(R) + \frac{1}{\epsilon} [\max\{0, \pi - \sum_{i=1}^N x_i \hat{r}_i\} \\ & + |\sum_{i=1}^N x_i - 1| + \max\{0, \sum_{i=1}^N z_i - K_u\} \\ & + \sum_{i=1}^N \max\{0, z_i d - x_i\} + \sum_{i=1}^N \max\{0, x_i - z_i u\} \\ & + \sum_{i=1}^N |z_i(1 - z_i)|] \end{aligned} \quad (3.10)$$

ϵ is the penalty parameter, and using the same idea of (3.8) $\rho_{a,p}(R)$, the two-sided risk measure, is defined as follows:

$$\rho_{a,p}(R) = \frac{a}{T} \left[\sum_{t=1}^T \left(\sum_{i=1}^N (r_{i,t} - \hat{r}_i) x_i \right)^+ \right] + (1 - a) \left\{ \frac{1}{T} \sum_{t=1}^T \left[\left(\sum_{i=1}^N (r_{i,t} - \hat{r}_i) x_i \right)^- \right]^p \right\}^{\frac{1}{p}} \quad (3.11)$$

where:

- $r_{i,t} = \frac{p_{i,t+1} - p_{i,t}}{p_{i,t}};$

⁷This last constraint $z_i(1 - z_i) = 0$ with $i = 1, \dots, N$ represents the just the reformulation of the integrality constraint $z_i \in \{0, 1\}$

- $\hat{r}_i = \frac{1}{T} \sum_{t=1}^T r_{i,t}$;
- $[\sum_{i=1}^N (r_{i,t} - \hat{r}_i)x_i]^+ = \max\{0, \sum_{i=1}^N (r_{i,t} - \hat{r}_i)x_i\}$;
- $[\sum_{i=1}^N (r_{i,t} - \hat{r}_i)x_i]^- = \min\{0, \sum_{i=1}^N (r_{i,t} - \hat{r}_i)x_i\}$

It is possible to see that the reformulation of the constrained problem leads to define a unique unconstrained objective function. Since, now we have an unconstrained objective function it is possible to apply the two algorithms discussed so far: PSO and ABC.

Chapter 4

Applications and discussion for Eurostoxx 50

This last chapter is focused on the application of the model introduced in the previous chapter. The application is based on real data, in particular on daily closing prices of assets quoted on Eurostoxx 50 index. The time length of considered historical data goes from August 2005 to December 2012.

4.1 Preliminary information

In order to assess the capability of the portfolio selection model, application to real data is provided. We used the inertia weight approach for the PSO and the basic approach for the ABC [107]. Both algorithms were launched on MATLAB 2013, and the experiments have been performed on Windows 10 with a Intel i5 4460 processor, with 8gb ram. The analysis is conducted on assets quoted on Eurostoxx 50 index, which includes the securities listed at Table 4.1

For the purpose of our analysis historical data of daily closing prices are divided into several subsets: that refers to different time periods between August 2005 and December 2012.

It is decided to consider 4 different time periods, each with the length of 18 months, each period is then divided into two parts: *in-sample* and *out-of-sample* with time length of 12 months and 6 months, respectively. The first set of data is used to estimate the parameters of the model and the second, known also as virtual future, is used to verify the effectiveness of the model on the basis of the estimation coming from first set of observations.

Table 4.1: EuroStoxx 50 stocks

Air Liquide	Allianz	Airbus Group
Anheuser-Busch InBev	Ass. Generali	ASML
AXA	Banco Santander Central Hispano	Banco Bilbao Vizcaya Argentaria
BASF	BMW	Bayer
BNP Paribas	Compagnie de Saint-Gobain	Carrefour
Daimler	Deutsche Post	Deutsche Bank
Deutsche Telekom	Enel	EON
Enel	Essilor International	Eni
Fresenius	Iberdrola	Groupe Danone
Inditex	Intesa San Paolo	ING Group NV
L'Oréal	MünchenerRückversicherungs-Gesellschaft	LVMH
Nokia	Philips Electronics	Orange
SAFRAN	SAP	Sanofi
Schneider Electric	Société Générale	Siemens
Telefonica	Unibail-Rodamco	TOTAL
Unicredit	Vinci	Unilever
Vivendi		Volkswagen

The basic idea behind is that, expected return and portfolio risk for out-of-sample remains approximately the same as those for in-sample. This implies that the percentage of investment suggested by the model using in-sample observations should provide the best portfolio solution also for subsequent instants. In other words, it is assumed to be at the last day of the twelfth month with a view of investing some capital for the next six months by adopting optimal investment percentages suggested by the model. At the end of the 18th month it can be observed if the formed portfolio has produced results in line with expectations at the end of the twelfth month.

Time periods are divided as follows:

- Period 1: in-sample August 2005 - July 2006; out-of-sample August

2006 - January 2007;

- Period 2: February 2007 - January 2008; out-of-sample February 2008 - July 2008;
- Period 3: in-sample August 2008 - July 2009; out-of-sample August 2009 - January 2010;
- Period 4: in sample February 2010 - January 2011; out-of-sample February 2011 - July 2011.

On the basis of the observed daily closing prices, daily returns for each asset are calculated using the following formula:

$$r_{i,t} = \frac{p_{i,t+1} - p_{i,t}}{p_{i,t}}$$

and \hat{r}_i is estimated using the historical data:

$$\hat{r}_i = \frac{1}{T} \sum_{t=1}^T r_{i,t}$$

According to the portfolio selection model, the minimum level of desired return l in $P(X, Z; \epsilon)$ is set equal to the global average of stock returns: $l = \sum_{i=1}^N \hat{r}_i$. In order to investigate the capability of PSO and ABC and select the number of particles and the colony size to use both algorithms are tested for the following combination of the parameters:

- PSO with 5000 iterations, number of particles and maximum number of asset to hold in the portfolio, with the following values: $P = \{50, 100, 200\}$ and $K_u = \{10, 30\}$.
- ABC with 2000 iterations, colony size and maximum number of asset to hold in the portfolio with the following values: $SN = \{20, 50, 100\}$ and $K_u = \{10, 30\}$.

For the setting of algorithm specific parameteres we conducted some preliminary tests for ABC and PSO. We noticed that ABC was able to reach good solutions as PSO even with a lower population size and a lower number of iterations. This is the reason we decided to set different population size and number of iterations for the two algorithms.

At the end the performances of the two algorithms are compared in terms of fitness and time of convergence.

Problem specific parameters setting

Before the application of the model, it is useful to set a series of parameters related to the constraints of the model common to both algorithms. Regarding the constraints, in each run some parameters remain unchanged, in particular:

- The minimum number of holding assets $K_d = 5$;
- The maximum number of holding assets are set to two different values: $K_u = 10$ and $K_u = 30$;
- The minimum fraction allowed in each asset $d = 3\%$;
- The maximum fraction allowed in each asset $u = 20\%$;
- The values of the parameters a, p in the risk measure are set as: $a = 0.5$ and $p = 2$.

The fact of testing the algorithm for different values of K_u allows to make an evaluation regarding the degree of diversification of the portfolio. The values for d and u are set according to the values for K_u and K_d .

PSO - parameters setting

Regarding the PSO, it is decided to use the PSO with inertia weight and by following what suggested by the literature, parameters that rule the velocity of the particles, within the search space, are set as:

- inertia weight: $\omega = 0.7298$;
- cognitive acceleration coefficient: $\phi_1 = 1.49681$;
- social acceleration coefficient: $\phi_2 = 1.49681$;
- number of iterations: 5000.

The last parameter that need to be set is the penalty parameter ϵ . It is a very important parameter since it assures the correspondency between the original optimization (constrained) and the unconstrained optimization problem. Following the values suggested in the literature, 5 runs for different values of ϵ are performed. Table 4.2 reports the results in terms of averaged best values of the fitness function (normalized to take into account the effects of using different values of ϵ) and its standard deviation. Table 4.2 also suggested to select $\epsilon = 10^{-2}$.

Table 4.2: Results for different values of ϵ - PSO

ϵ	Normalized Fitness	Standard deviation
1	0.474128731	0.481023243
0.1	0.335652293	0.413366971
0.01	0.309641952	0.411157151
0.001	0.478669604	0.398127678
0.0001	0.521252222	0.416949842
0.00001	0.310817422	0.394867892

Table 4.3: Results for different choices of P number of particles in PSO

P	Fitness	Ratio of decrease (F_f/F_i)	Time (seconds)
50	111.9665	0.025768146	19.18393
100	2.712269	0.000724291	88.00986
200	2.655834	0.000668402	173.8534

Regarding the number of particles, we performed 5 runs for each of the period under analysis. Table 4.3 shows the average best fitness value for different values of P and it refers to the case $\epsilon = 10^{-2}$. It can be seen from Table 4.3 how the fitness value decreases as the number of particles P increases. We have noticed that this happens for both, $K_u = 10$ and $K_u = 30$. In Table 4.3 we reports the case in $K_u = 30$.

The number of particles is then quite high, but this concurs with the general evidence that larger populations perform better on higher dimensional problems[1]. According to what suggested by the Table 4.3 the number of particles P is set equal to 200 and all the test from now on for the PSO are based on this value.

ABC parameters setting

Regarding the parameters setting related to the ABC algorithm, we proceeded as follows. For the selection of the colony size we conducted 5 runs on one year time period for $SN = \{20, 50, 100\}$. We noticed that the algorithm performances improve, in terms of fitness, as the colony size increases. The limit was set as: $limit = (Colony\ Size \times D)/2$. Below we report the values assigned to parameters of the ABC:

- Colony Size (number of food source) $SN = 100$;
- Number of parameters of the objective function $D = 50$;
- Control parameter in order to abandon the food source $limit = 2500$;
- Maximum cycle number $MCN = 2000$.

Table 4.4: Results for different choices of Colony Size.

Colony Size	Fitness	Ratio of decrease (F_f/F_i)	Time (seconds)
20	9.320801103	0.007251959	50.98514375
50	2.646298363	0.002023777	251.2205349
100	2.613200848	0.002077802	944.1702421

It is worth noticing that, as the colony size increases ABC finds better solutions, however after a sufficient value for colony size, any increment in the value does not improve the performance of the ABC algorithm significantly, colony size of 50-100 can provide acceptable convergence for search [108].

Application and discussion

After all the preliminary tests for the PSO and ABC we tested the portfolio selection problem with different values for the parameters a and p , that represent attitudes towards risk, of the risk measure $\rho_{a,p}$ and K_u by considering one year data of daily returns of different time periods.

We set the maximum number of iterations to 5000 for the PSO and 2000 for the ABC. We performed 5 runs of both algorithms with different combinations of the parameters and for different time periods, with random initial positions and velocities. We noticed that the standard deviation of the values of the fitness function was not that high and the optimal portfolios in the 5 runs were not significantly different, see Table A.1.

It is interesting notice that the monotonicity properties with respect to a and p expected by Theorem 2.3 ¹ in [3] are fulfilled, see Tables 4.5–4.6 for the PSO and Tables 4.7–4.8 for the ABC.

From $\rho_{0.5,1}$ to $\rho_{0.5,2}$ and to $\rho_{0.5,5}$, the risk value of the corresponding optimal portfolio is monotonically increasing, for each of the period and

¹Theorem 2.3 The coherent risk measure $\rho_{a,p}$ is non-decreasing with respect to p , and non-increasing with respect to a , respectively [3].

Table 4.5: PSO - Monotonicity of $\rho_{a,p}(R)$ for $a = 0.5$ and different values of p and K_u , with one year data from four time periods. $N_{a,p}(K_u)$ represents the number of assets in the optimal portfolio.

Period 1	$p = 1$	$p = 2$	$p = 5$
$\rho_{0.5,p} : K_u = 10$	0.002508	0.003853	0.006576
$N_{a,p}(K_u)$	9	9	9
$\rho_{0.5,p} : K_u = 30$	0.002234	0.003500	0.005881
$N_{a,p}(K_u)$	20	20	21
Period 2	$p = 1$	$p = 2$	$p = 5$
$\rho_{0.5,p} : K_u = 10$	0.003207	0.005045	0.010480
$N_{a,p}(K_u)$	8	8	8
$\rho_{0.5,p} : K_u = 30$	0.003952	0.005869	0.013056
$N_{a,p}(K_u)$	25	25	25
Period 3	$p = 1$	$p = 2$	$p = 5$
$\rho_{0.5,p} : K_u = 10$	0.008116	0.011982	0.020146
$N_{a,p}(K_u)$	9	9	9
$\rho_{0.5,p} : K_u = 30$	0.007981	0.011849	0.019669
$N_{a,p}(K_u)$	25	25	25
Period 4	$p = 1$	$p = 2$	$p = 5$
$\rho_{0.5,p} : K_u = 10$	0.003990	0.006137	0.010386
$N_{a,p}(K_u)$	8	8	8
$\rho_{0.5,p} : K_u = 30$	0.003553	0.005521	0.009296
$N_{a,p}(K_u)$	24	24	24

for different choices of K_u , see Tables 4.5–4.7. We also have that $\rho_{a,2}$ for optimal portfolios is $\rho_{0,2} > \rho_{0.25,2} > \rho_{0.5,2} > \rho_{0.75,2} > \rho_{1,2}$, see Tables 4.6–4.8, for almost all periods.

The non-increasing property of a with respect to $\rho_{a,p}$ means that, $\rho_{a,p}$ with a large a is connected with a less risk-averse investor. The monotonic property of $\rho_{a,p}$ with respect to p and a may be used to reflect investor's risk attitude towards risk as said in the previous chapter. The greater the p , the larger the risk $\rho_{a,p}$. This implies that an investor adopting the measure of risk $\rho_{a,p}$, will treat riskier the investment with a higher p than one with

Table 4.6: PSO - Monotonicity of $\rho_{a,p}(R)$ for $p = 2$ and different values of a and K_u , with one year data from four time periods. $N_{a,p}(K_u)$ represents the number assets in the optimal portfolio.

Period 1	$a = 0$	$a = 0.25$	$a = 0.5$	$a = 0.75$	$a = 1$
$\rho_{a,2} : K_u = 10$	0.006686	0.005804	0.003947	0.003218	0.002490
$N_{a,p}(K_u)$	9	9	10	10	10
$\rho_{a,2} : K_u = 30$	0.004666	0.004118	0.003500	0.002882	0.002234
$N_{a,p}(K_u)$	21	20	20	20	20
Period 2	$a = 0$	$a = 0.25$	$a = 0.5$	$a = 0.75$	$a = 1$
$\rho_{a,2} : K_u = 10$	0.008871	0.007885	0.006738	0.005591	0.004409
$N_{a,p}(K_u)$	8	8	8	8	8
$\rho_{a,2} : K_u = 30$	0.008195	0.007157	0.006151	0.005143	0.004163
$N_{a,p}(K_u)$	20	20	20	20	20
Period 3	$a = 0$	$a = 0.25$	$a = 0.5$	$a = 0.75$	$a = 1$
$\rho_{a,2} : K_u = 10$	0.018581	0.016408	0.014229	0.012049	0.004408
$N_{a,p}(K_u)$	8	8	8	8	8
$\rho_{a,2} : K_u = 30$	0.015849	0.013685	0.011849	0.009774	0.007981
$N_{a,p}(K_u)$	25	25	25	25	25
Period 4	$a = 0$	$a = 0.25$	$a = 0.5$	$a = 0.75$	$a = 1$
$\rho_{a,2} : K_u = 10$	0.008267	0.007195	0.006137	0.005063	0.003786
$N_{a,p}(K_u)$	8	8	8	8	9
$\rho_{a,2} : K_u = 30$	0.007519	0.006508	0.005521	0.004537	0.003553
$N_{a,p}(K_u)$	24	24	24	24	24

a smaller p . In addition, $\rho_{a,p}$ with large p is connected with risk-averse investors. On the other hand the non-increasing property of a with respect to $\rho_{a,p}$ means that, $\rho_{a,p}$ with a large a is connected with a less risk-averse investor.

For the PSO we noticed that $\rho_{a,p}$ slightly decreases with K_u , in particular this is more evident in Period 1, in Period 3 and Period 4 except for $\rho_{0.5,5}$ in Period 3. This is consistent also with the fact that PSO has performed good exploration when $K_u = 10$ which is the subset of $K_u = 30$.

Table 4.7: ABC -Monotonicity of $\rho_{a,p}(R)$ for $a = 0.5$ and different values of p and K_u , with one year data from four time periods. $N_{a,p}(K_u)$ represents the number of assets in the optimal portfolio.

Period 1	$p = 1$	$p = 2$	$p = 5$
$\rho_{0.5,p} : K_u = 10$	0.002924	0.004566	0.006020
$N_{a,p}(K_u)$	6	9	5
$\rho_{0.5,p} : K_u = 30$	0.002704	0.003676	0.008236
$N_{a,p}(K_u)$	12	9	7
Period 2	$p = 1$	$p = 2$	$p = 5$
$\rho_{0.5,p} : K_u = 10$	0.003304	0.005029	0.009518
$N_{a,p}(K_u)$	5	5	5
$\rho_{0.5,p} : K_u = 30$	0.005009	0.006699	0.014575
$N_{a,p}(K_u)$	5	9	5
Period 3	$p = 1$	$p = 2$	$p = 5$
$\rho_{0.5,p} : K_u = 10$	0.009912	0.014814	0.021839
$N_{a,p}(K_u)$	5	5	5
$\rho_{0.5,p} : K_u = 30$	0.011318	0.015126	0.025514
$N_{a,p}(K_u)$	7	9	9
Period 4	$p = 1$	$p = 2$	$p = 5$
$\rho_{0.5,p} : K_u = 10$	0.003724	0.006085	0.010936
$N_{a,p}(K_u)$	6	6	6
$\rho_{0.5,p} : K_u = 30$	0.003673	0.005932	0.011391
$N_{a,p}(K_u)$	6	6	7

Regarding the ABC the monotonicity property of the $\rho_{a,p}$ is fulfilled as in the case of PSO, but interestingly we noticed that the $\rho_{a,p}$ does not decrease with K_u , this may due to the fact that the portfolio selected by ABC is less diversified with respect to the one chosen by PSO, see Table 4.9.

Table 4.8: ABC - Monotonicity of $\rho_{a,p}(R)$ for $p = 2$ and different values of a and K_u , with one year data from four time periods. $N_{a,p}(K_u)$ represents the number assets in the optimal portfolio.

Period 1	$a = 0$	$a = 0.25$	$a = 0.5$	$a = 0.75$	$a = 1$
$\rho_{a,2} : K_u = 10$	0.005303	0.004602	0.003646	0.003488	0.002520
$N_{a,p}(K_u)$	7	7	7	7	8
$\rho_{a,2} : K_u = 30$	0.005403	0.005145	0.005008	0.004256	0.002744
$N_{a,p}(K_u)$	5	9	7	8	8
Period 2	$a = 0$	$a = 0.25$	$a = 0.5$	$a = 0.75$	$a = 1$
$\rho_{a,2} : K_u = 10$	0.009358	0.008262	0.006398	0.005259	0.004304
$N_{a,p}(K_u)$	5	5	5	5	5
$\rho_{a,2} : K_u = 30$	0.009589	0.008340	0.007067	0.005029	0.004346
$N_{a,p}(K_u)$	5	5	9	5	5
Period 3	$a = 0$	$a = 0.25$	$a = 0.5$	$a = 0.75$	$a = 1$
$\rho_{a,2} : K_u = 10$	0.019523	0.016798	0.012967	0.010597	0.009291
$N_{a,p}(K_u)$	6	6	8	7	8
$\rho_{a,2} : K_u = 30$	0.019102	0.013694	0.013138	0.012913	0.008576
$N_{a,p}(K_u)$	8	5	7	7	9
Period 4	$a = 0$	$a = 0.25$	$a = 0.5$	$a = 0.75$	$a = 1$
$\rho_{a,2} : K_u = 10$	0.007016	0.006139	0.008577	0.004227	0.003374
$N_{a,p}(K_u)$	6	6	6	6	6
$\rho_{a,2} : K_u = 30$	0.012592	0.011029	0.006315	0.005281	0.004096
$N_{a,p}(K_u)$	7	7	8	5	5

4.2 Comparison and discussion between PSO and ABC

In this last section comparison between the two algorithms are performed. First of all we checked the diversification of the portfolio for two different approaches, see Table 4.9.

In the case of ABC it is interesting notice that the number of assets $N_{a,p}$ included in the portfolio does not excess 10 even when $K_u = 30$. The

4.2. COMPARISON AND DISCUSSION BETWEEN PSO AND ABC 77

Table 4.9: Comparison between ABC and PSO regarded the number of assets $N_{a,p}(K_u)$ in the optimal portfolios over one year data from different time periods.

Period 1 - $\rho_{0.5,p}$	$p = 1$	$p = 2$	$p = 5$
<i>PSO</i> : $K_u = 10$	9	9	9
<i>ABC</i> : $K_u = 10$	6	9	5
<i>PSO</i> : $K_u = 30$	24	24	24
<i>ABC</i> : $K_u = 30$	12	9	7
Period 2 - $\rho_{0.5,p}$	$p = 1$	$p = 2$	$p = 5$
<i>PSO</i> : $K_u = 10$	8	8	8
<i>ABC</i> : $K_u = 10$	5	5	5
<i>PSO</i> : $K_u = 30$	25	25	25
<i>ABC</i> : $K_u = 30$	5	9	5
Period 3 - $\rho_{0.5,p}$	$p = 1$	$p = 2$	$p = 5$
<i>PSO</i> : $K_u = 10$	9	9	9
<i>ABC</i> : $K_u = 10$	5	5	5
<i>PSO</i> : $K_u = 30$	25	25	25
<i>ABC</i> : $K_u = 30$	7	9	9
Period 4 - $\rho_{0.5,p}$	$p = 1$	$p = 2$	$p = 5$
<i>PSO</i> : $K_u = 10$	9	9	9
<i>ABC</i> : $K_u = 10$	6	6	6
<i>PSO</i> : $K_u = 30$	25	25	15
<i>ABC</i> : $K_u = 30$	6	6	7

diversification using the PSO is higher than ABC, in any case both algorithms are consistent with the cardinality constraint and budget constraint introduced in the portfolio selection problem, see Tables A.3 A.6. The only constraint that presents more difficulty in being met is the one related to the minimum percentage d to be invested in each asset. In a period in which markets experienced sharp falls a good solution could be going “short”, but the considered model does not consider possibility of “short-selling”. A potential solution to this issue consist in assigning a very small percentage of investment, almost null, to securities on which we should go short.

In Tables 4.10 and 4.11 we compared the performance of the portfolio selected through the two approaches. Following the financial practice, we proceeded as follows: first we used one-year data of daily returns, from the portfolio selected by minimizing $P(X, Z; \epsilon)$, then we invested the selected portfolio for the next six months. For the purpose of our analysis we considered 4 time periods of one year length, called *in-sample*, to compute the portfolio for the test for the next six month data, called *out-of-sample*. The considered time periods for *in-sample* and *out-of-sample* were:

- Period 1: in-sample August 2005 - July 2006; out-of-sample August 2006 - January 2007;
- Period 2: February 2007 - January 2008; out-of-sample February 2008 - July 2008;
- Period 3: in-sample August 2008 - July 2009; out-of-sample August 2009 - January 2010;
- Period 4: in sample February 2010 - January 2011; out-of-sample February 2011 - July 2011.

The aim of doing so is to analyze the impact of different macroeconomic conditions on the performance of the portfolio and the results for this are shown in Table 4.10.

We noticed that the behaviour of the portfolio selected using ABC and PSO is almost the same in the case $K_u = 10$ and this is more evident in Period 1 for $\rho_{0.5,1}$ and in Period 4 for $\rho_{0.5,2}$ where we almost obtain the same return and the composition of the portfolio is quite similar: 6 assets for ABC and 9 assets for PSO.

Regarding the case in which $K_u = 30$, the behaviour is the same until period 3, in addition it is worth highlight that ABC tends to diversify less with respect to PSO, see Table 4.9.

In Period 1 returns for the selected portfolio by both PSO and ABC are positive, infact parameters for out-of-sample are estimated using assets prices during a period of an economic growth, just before the real estate subprime crisis due to Lehman Brothers bankruptcy. In case of $K_u = 30$ ABC performs slightly better than PSO.

In Period 2 portfolio returns are negative, infact the estimation of the parameters of the *out-of-sample* takes place during a period of time involving the failure of lehman brothers.

In Period 3 ABC performs better than PSO in both cases of K_u and its portfolio returns are increasing with p .

4.2. COMPARISON AND DISCUSSION BETWEEN PSO AND ABC 79

Table 4.10: Portfolio returns for PSO and ABC with $K_u = 10$ and for different time periods

Period 1 Out-of-sample 6 months			
	$p = 1$ (%)	$p = 2$ (%)	$p = 5$ (%)
PSO	0.1308	0.1310	0.1296
ABC	0.1346	0.1284	0.1342
Period 2 Out-of-sample 6 months			
	$p = 1$ (%)	$p = 2$ (%)	$p = 5$ (%)
PSO	-0.0932	-0.0932	-0.1029
ABC	-0.0735	-0.0619	-0.0468
Period 3 Out-of-sample 6 months			
	$p = 1$ (%)	$p = 2$ (%)	$p = 5$ (%)
PSO	0.0739	0.0685	0.0747
ABC	0.0856	0.0793	0.1176
Period 4 Out-of-sample 6 months			
	$p = 1$ (%)	$p = 2$ (%)	$p = 5$ (%)
PSO	-0.0429	-0.0385	-0.0329
ABC	-0.0158	-0.0357	-0.0541

In Period 4 portfolio returns are negative for both algorithms, as in case of Period 2, for the case $K_u = 10$ and this implies that investor can withdraw $-\rho_R$ amount of money without affecting the acceptance of his future position. In the case $K_u = 30$, it is interesting notice that portfolio selected through PSO provides positive returns for all values of p with respect to the portfolio selected by ABC. This may due to the fact that PSO portfolio is well diversified with respect to the ABC one, Table 4.9.

All the tests launched so far, for both algorithms, were run by setting a maximum number of iterations. In order to check the capability of the two algorithms in finding the best fitness, in the same amount of time, it was decided to set a timeout. In the following Tables 4.12 we compared the capability of PSO and ABC of exploring best solutions with different population sizes.

We set the timeout to 300 seconds and the results for different population sizes for ABC and PSO are reported in Table 4.12. As it seen from table, ABC approach leads to better results for a lower population

Table 4.11: Portfolio returns for PSO and ABC with $K_u = 30$ and for different time periods

Period 1 Out-of-sample 6 months			
	$p = 1$ (%)	$p = 2$ (%)	$p = 5$ (%)
PSO	0.1289	0.1294	0.1249
ABC	0.1344	0.1334	0.1469
Period 2 Out-of-sample 6 months			
	$p = 1$ (%)	$p = 2$ (%)	$p = 5$ (%)
PSO	-0.0724	-0.0827	-0.0755
ABC	-0.1174	-0.1395	-0.0877
Period 3 Out-of-sample 6 months			
	$p = 1$ (%)	$p = 2$ (%)	$p = 5$ (%)
PSO	0.0615	0.0113	0.1076
ABC	0.1038	0.1080	0.1089
Period 4 Out-of-sample 6 months			
	$p = 1$ (%)	$p = 2$ (%)	$p = 5$ (%)
PSO	0.0804	0.0807	0.0749
ABC	-0.0493	-0.0294	-0.0617

Table 4.12: PSO and ABC - Best fitness values and $\rho_{0.5,2}$ in 300 seconds for different choices of the population size and $K_u = 30$.

Population Size	20	50	100	150
PSO	220.4156	2.70132	2.650049	2.658826
ABC	2.692354	2.679903	2.752702	15.46114

size with respect to PSO. While population size is 20, with 300 seconds timeout, best affinity is obtained with ABC. On the other hand as the population increases, best solutions are reached with PSO, but this is due to the time limit imposed to both algorithms, infact ABC need more time to reach better solutions as population size increases. However after a sufficient value for colony size, any increment in the value does not improve the performance of the ABC algorithm significantly, colony size of 50-100 can provide acceptable convergence for search [108].

4.2. COMPARISON AND DISCUSSION BETWEEN PSO AND ABC 81

In the following figures we report the convergence of PSO and ABC for different choices of number of particles and colony size with a fixed number of iterations: 5000 iterations for PSO and 2000 for ABC.

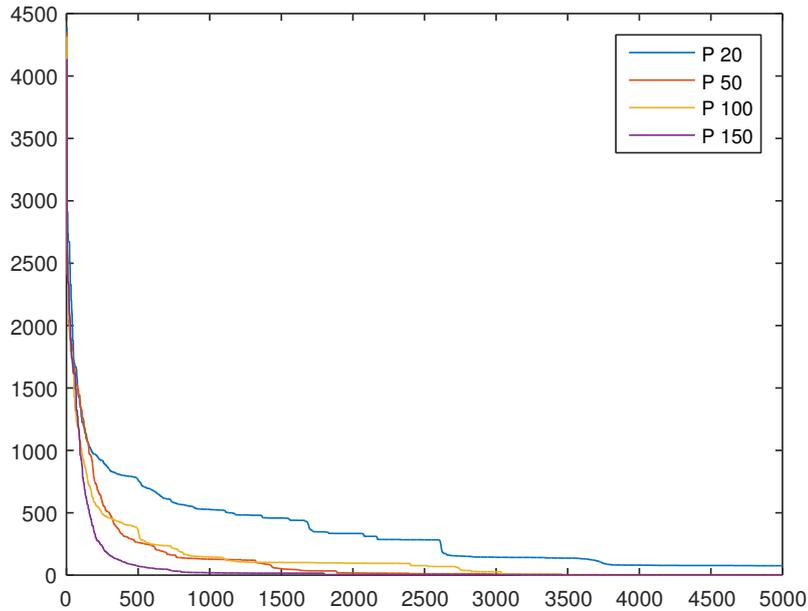


Figure 4.1: PSO - With 5000 iterations and for different choices of number of particles P.

In these two figures we can see that ABC needs a lower number of iterations to converge towards optimal solutions with respect to PSO.

From the following graph it can be noticed how ABC can reach good solutions even after a smaller number of cycles than the one used in the previous tests. In fact it can be seen that even after 1000 iterations ABC provides a good solution.

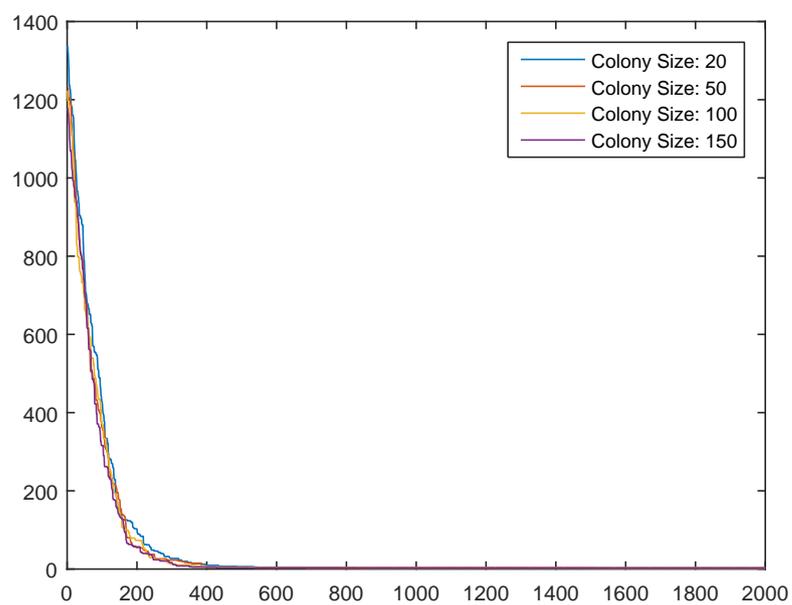


Figure 4.2: ABC - With 2000 iterations and for different choices of the colony size.

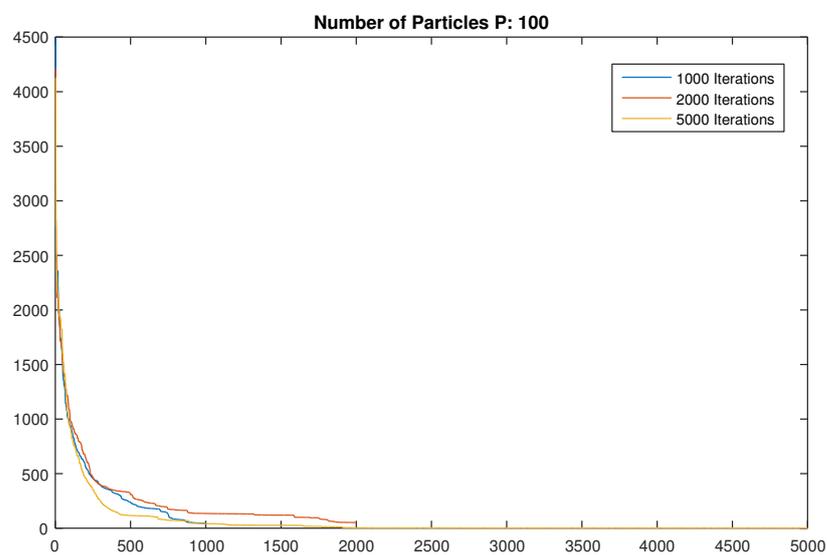


Figure 4.3: PSO with 100 particles.

4.2. COMPARISON AND DISCUSSION BETWEEN PSO AND ABC 83

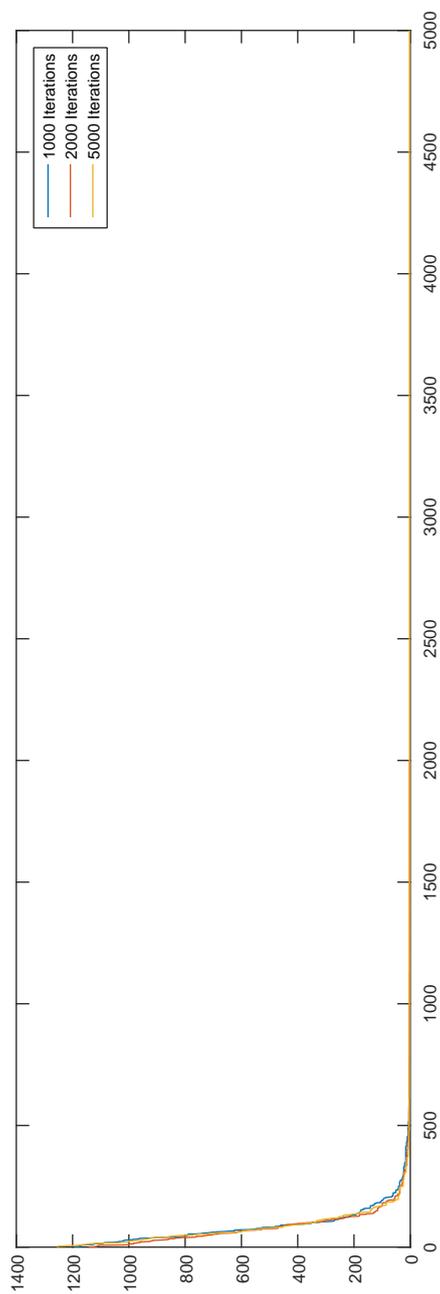


Figure 4.4: ABC with colony size of 100.

Conclusion

In the present work, two population-based algorithms, ABC and PSO have been applied to a complex portfolio selection model, with a two-sided risk measure. The complex portfolio is characterized by an upper-and-lower-moments-based coherent risk measure and a mixed-integer formulation. The problem presented in chapter 3 was solved using PSO and ABC. These two are bio-inspired metaheuristics that deal with optimization problems. It is worth notice that both algorithms are born as unconstrained optimization methods, indeed they are often modified in order to deal with constrained problems. Since the problem we were dealing with was a constrained one, in order to apply the two algorithms we combined them with an exact penalty scheme. This method transforms the considered mixed-integer portfolio selection problem into an equivalent nondifferentiable unconstrained minimization problem. In this way correspondence between the solutions of the original mathematical problem and the solutions of the exact penalty-based model is provided.

In the last chapter we compared the results provided by both algorithms. PSO and ABC are good in finding optimal solutions. We have noticed that ABC with respect to PSO provide better solutions with the same population size. But as the population size increases the time needed to the ABC to search optimal solutions increases as well, with respect to PSO. The convergence time required to search optimal solutions is significantly lower for PSO than the one needed by ABC.

PSO and ABC performed quite well in solving the complex portfolio selection problem. We have noticed that in case of $K_u = 30$ the number of maximum assets selected by the ABC is almost the same as in $K_u = 10$, this is evident in Table A.6 A.3. Infact, as result we have more diversified portfolio provided by the PSO with respect to the one found through ABC.

Performance of PSO and ABC can be improved by using different values for the parameters, for example it could be a good idea to increase the number of iterations and compare the results of both. Another idea could be to define a stopping criterion based on the value of the objective function.

The change in the algorithms could be obtained also by changing the control parameters of both, instead of keeping them fixed it could be a good idea to control them during the entire selection process. Regarding the ABC, in our work we used the basic version of the algorithm. It could be interesting compare the performance of ABC with respect to PSO by adopting the an alternative probability calculation method for the onlooker bee phase, seen in chapter 2. In any case, it could be interesting see the performance of the ABC for its modified versions proposed in the literature.

In order to improve the solution of the portfolio selection problem an interesting idea could be to initialize ABC with particles found through multiple runs of PSO or viceversa. This is mainly due to the fact that the two algorithms move towards optimal solutions in different ways. Infact when one of the two approaches it is not able to improve the solution anymore, a different way to move towards optimal solutions can be adopted.

Appendix A

Appendix

In the following tables we report the results obtained by running the PSO and ABC algorithms over 5 runs for each time period, and we also provide the tables related to the number of assets selected and the percentage of investment made by both algorithms in each of the asset.

We can see from the tables that the cardinality constraint and the budget constraints are fulfilled by both algorithms. This implies that both algorithms are able to solve constrained problem providing a good solution for the portfolio selection problem. The only difference between the two regarded the fact that ABC even in case of $K_u = 30$ do not select more than 11 assets in order to build the optimal portfolio. On the other hand portfolio obtained through the PSO is more diversified.

Table A.1: PSO - Best fitness and the $N_{a,p}$ for $a = 0.5$ and $p = 2$ over 5 runs for one year time period for $K_u = 10$ and $K_u = 30$.

Period 1	Run 1	Run 2	Run 3	Run 4	Run 5	Average
Fitness: $K_u = 10$	2.6377	2.6622	2.6450	2.6689	2.6577	2.6543
$N_{a,p}(K_u)$	9	9	10	8	9	9
Fitness: $K_u = 30$	2.6867	2.6555	2.6696	2.6578	2.6471	2.6634
$N_{a,p}(K_u)$	17	24	20	26	18	21
Period 2						
Fitness: $K_u = 10$	0.0102	0.0066	0.0069	0.0058	0.0066	0.0073
$N_{a,p}(K_u)$	10	8	10	8	8	9
Fitness: $K_u = 30$	0.0058	0.0129	0.0062	0.0069	0.0067	0.0077
$N_{a,p}(K_u)$	14	25	20	26	18	20.6
Period 3						
Fitness: $K_u = 10$	0.6731	0.7352	0.6264	0.7926	0.8221	0.7298
$N_{a,p}(K_u)$	10	9	9	7	8	9
Fitness: $K_u = 30$	0.68185	0.71825	0.77706	0.76311	0.74901	0.73786
$N_{a,p}(K_u)$	14	25	21	28	19	21
Period 4						
Fitness: $K_u = 10$	3.2814	3.3852	3.3134	3.3058	3.3103	3.3193
$N_{a,p}(K_u)$	10	9	8	8	8	8.6
Fitness: $K_u = 30$	3.2942	3.2911	3.3243	3.3106	3.3138	3.3068
$N_{a,p}(K_u)$	16	24	20	30	18	21.6

Table A.2: PSO - Percentage of capital invested in each asset over 5 runs with $K_u = 30$ for Period 1.

N = 50	Run 1	Run 2	Run 3	Run 4	Run 5
x_1	0.0300	0.0309	0	0	0
x_2	0.0677	0	0	0	0.0324
x_3	0	0	0	0.0300	0
x_4	0	0	0	0.0312	0
x_5	0.0343	0	0	0	0
x_6	0.0583	0.0308	0	0.0313	0
x_7	0	0	0.0307	0.0308	0.0310
x_8	0	0.0317	0	0.0538	0
x_9	0.0316	0.0332	0.0301	0	0
x_{10}	0.0415	0.0304	0	0.0300	0
x_{11}	0.0336	0	0	0.0309	0.0327
x_{12}	0	0.0673	0.1696	0	0.0327
x_{13}	0	0	0	0	0
x_{14}	0	0	0.0844	0.0305	0.0301
x_{15}	0	0.0327	0	0	0.1353
x_{16}	0	0.0309	0	0	0.1006
x_{17}	0	0.0323	0	0.1893	0
x_{18}	0	0	0.0302	0.0342	0
x_{19}	0.1305	0	0.0815	0.0352	0
x_{20}	0	0	0.0376	0.0302	0
x_{21}	0	0	0.0318	0.014	0.0310
x_{22}	0	0	0	0.0316	0
x_{23}	0	0	0.0305	0.0325	0
x_{24}	0.1009	0.1381	0.0303	0	0.0313
x_{25}	0	0	0	0	0

Table A.3

N = 50	Run 1	Run 2	Run 3	Run 4	Run 5
x_{26}	0	0.0325	0	0.0353	0
x_{27}	0	0	0	0.0300	0.0421
x_{28}	0	0.0304	0	0	0
x_{29}	0.0369	0	0.0304	0	0
x_{30}	0	0.0336	0	0	0.0928
x_{31}	0.0312	0	0	0.0314	0
x_{32}	0.0743	0.0303	0	0.0301	0
x_{33}	0	0	0	0	0
x_{34}	0	0.0332	0	0.0302	0
x_{35}	0	0.0300	0.0304	0	0
x_{36}	0.0953	0.0921	0.0316	0	0.1011
x_{37}	0.0712	0	0.0321	0	0
x_{38}	0.0832	0	0	0	0.0323
x_{39}	0	0.0305	0	0	0
x_{40}	0.0336	0	0.0316	0	0
x_{41}	0	0.0508	0	0	0
x_{42}	0	0	0	0.0315	0.0344
x_{43}	0	0.0534	0	0	0
x_{44}	0	0.0309	0.0369	0.0310	0
x_{45}	0	0	0.1588	0.0301	0.0657
x_{46}	0	0.0307	0.0300	0.0301	0.1079
x_{47}	0.0457	0	0	0	0.0308
x_{48}	0	0.0323	0.0302	0.0322	0
x_{49}	0	0.0306	0.0308	0.0345	0.0354
x_{50}	0	0	0	0	0
$\sum_{i=1}^{50} x_i$	1.0000	0.9999	1.0000	1.0000	1.0000
z_i	17	24	20	26	18

Table A.4: ABC - Best fitness and the $N_{a,p}$ for $a = 0.5$ and $p = 2$ over 5 runs for one year time period for $K_u = 10$ and $K_u = 30$.

Period 1	Run 1	Run 2	Run 3	Run 4	Run 5	Average
Fitness: $K_u = 10$	3.4368	4.5669	4.5419	3.1068	2.9998	3.0533
$N_{a,p}(K_u)$	7	9	8	6	7	7.4
Fitness: $K_u = 30$	2.73982	2.78457	2.98063	2.61320	2.78744	2.78113
$N_{a,p}(K_u)$	8	9	9	10	11	9
Period 2						
Fitness: $K_u = 10$	0.00639	0.00503	0.00554	0.00576	0.00612	0.00577
$N_{a,p}(K_u)$	5	5	5	5	5	5
Fitness: $K_u = 30$	0.01069	0.02057	0.00773	0.00721	0.01475	0.01219
$N_{a,p}(K_u)$	9	9	5	5	8	7
Period 3						
Fitness: $K_u = 10$	0.84308	0.72710	0.65744	0.90089	0.75058	0.77582
$N_{a,p}(K_u)$	8	5	5	7	5	6
Fitness: $K_u = 30$	0.73400	0.83048	0.68340	0.72019	0.73981	0.74158
$N_{a,p}(K_u)$	7	9	7	5	7	7
Period 4						
Fitness: $K_u = 10$	4.01587	3.64537	6.32763	3.35543	4.45253	4.35937
$N_{a,p}(K_u)$	6	6	8	6	7	6.6
Fitness: $K_u = 30$	3.29197	3.33508	3.31272	3.46672	3.44169	3.36964
$N_{a,p}(K_u)$	8	6	5	8	7	6.8

Table A.5: ABC - Percentage of capital invested in each asset over 5 runs with $K_u = 30$ for Period 1.

N = 50	Run 1	Run 2	Run 3	Run 4	Run 5
x_1	0	0.0435	0	0	0
x_2	0.1967	0	0	0	0
x_3	0	0	0.1321	0	0
x_4	0	0	0.0002	0	0.0488
x_5	0	0.0433	0	0,1688	0
x_6	0	0	0	0	0
x_7	0	0	0	0.0345	0
x_8	0	0	0	0	0
x_9	0	0	0	0	0.0309
x_{10}	0	0	0	0,2	0
x_{11}	0	0	0	0	0.1032
x_{12}	0	0	0	0	0
x_{13}	0	0	0.0024	0	0
x_{14}	0	0	0	0,2	0
x_{15}	0	0	0	0	0.0923
x_{16}	0	0.0502	0.1649	0	0
x_{17}	0	0	0	0	0
x_{18}	0	0	0	0	0
x_{19}	0	0	0.1937	0	0
x_{20}	0.00002	0	0.0915	0.0293	0,1484
x_{21}	0	0	0	0	0.0295
x_{22}	0	0	0	0	0,1195
x_{23}	0	0	0	0	0
x_{24}	0	0	0	0,2	0
x_{25}	0.2	0.0869	0.1581	0.0384	0

Table A.6

N = 50	Run 1	Run 2	Run 3	Run 4	Run 5
x_{26}	0	0	0	0	0
x_{27}	0,1368	0,1760	0	0.0512	0
x_{28}	0	0	0.0002	0.003	0
x_{29}	0	0,1271	0,2	0	0
x_{30}	0	0.0726	0	0	0
x_{31}	0.0004	0	0.0564	0	0
x_{32}	0	0	0	0	0
x_{33}	0	0	0	0	0
x_{34}	0	0	0.0002	0	0
x_{35}	0,0853	0	0	0	0
x_{36}	0	0	0	0	0,1008
x_{37}	0,1583	0	0	0	0
x_{38}	0	0	0	0	0.0746
x_{39}	0,1016	0	0	0	0
x_{40}	0.0005	0	0	0	0
x_{41}	0	0	0	0	0
x_{42}	0	0	0	0	0
x_{43}	0	0	0	0	0
x_{44}	0	0	0	0	0
x_{45}	0.0823	0	0	0	0
x_{46}	0.0378	0,2	0	0	0,1293
x_{47}	0	0	0	0	0
x_{48}	0	0,2	0	0.0485	0
x_{49}	0	0	0	0	0,1218
x_{50}	0	0	0	0	0
$\sum_{i=1}^{50} x_i$	1.0000	1.0000	1.0000	1.0008	0.9996
z_i	8	9	7	10	11

Bibliography

- [1] M. Corazza, G. Fasano, and R. Gusso, “Particle swarm optimization with non-smooth penalty reformulation, for complex portfolio selection problem,” *Applied mathematics and computation*, vol. 224, pp. 611–624, 2013.
- [2] Z. Wenping, Z. Yunlong, C. Hanning, and Z. Beiwei, “Solving multiobjective optimization problems using artificial bee colony.” <http://dx.doi.org/10.1155/2011/569784>, 2011.
- [3] Z. Chen and Y. Wang, “Two-sided coherent risk measures and their application in realistic portfolio optimization,” *Journal of Banking and Finance*, vol. 32(12), pp. 2667–2673, 2008.
- [4] H. M. Markowitz, “Portfolio selection,” *Journal of Finance*, vol. 7, pp. 77–91, 1952.
- [5] Investopedia, “Mean-variance analysis — Investopedia.” <http://www.investopedia.com/terms/m/meanvariance-analysis.asp>. Accessed: 15-12-2015.
- [6] Investopedia, “Short selling — Investopedia.” <http://www.investopedia.com/terms/s/shortselling.asp>. Accessed: 15-12-2015.
- [7] J. Neumann and O. Morgenstern, *Theory of games and Economic Behavior*. Princeton University Press, 1944.
- [8] D. Luenberger, *Investment Science - 2nd Edition*. Oxford University Press, 2010.
- [9] H. M. Markowitz, *Portfolio selection: Efficient diversification of investments*. Springer, 1991.
- [10] E. Canestrelli and C. Nardelli, *Criteri per la selezione del portafoglio*. Torino, Giappichelli, 1991.

- [11] S. Rachev, D. Martin, B. Racheva, and F. Stoyanov, *Stable ETL optimal portfolio and extreme risk management*. Springer - Physika Verlag, 2007.
- [12] L. Knetsch and B. F. J. Borges, “Tests of market outcomes with asymmetric valuations of gains and losses: smaller gains, fewer trades, and less value,” *Journal of Economic Behaviour and Organization*, vol. 33(2), pp. 185–193, 1998.
- [13] M. Corazza and D. Favaretto, “On the existence of solutions to the quadratic mixed-integer mean-variance portfolio selection problem,” *European Journal of Operational Research*, vol. 176, pp. 1947–1960, 2007.
- [14] R. Michaud, “The Markowitz optimization enigma: Is optimized optimal ?,” *Financial Analysts Journal*, vol. 45, pp. 31–42, 1989.
- [15] H. Konno and H. Yamazaki, “Mean-absolute deviation portfolio optimization model and its application to Tokyo Stock Market,” *Management Science*, vol. 37(5), pp. 519–531, 1991.
- [16] F. Black and R. Litterman, “Global Portfolio Optimization,” *Financial Analysts Journal*, vol. 48, pp. 28–43, 1992.
- [17] S. Rachev, S. Ortobelli, S. Stoyanov, F. Fabozzi, and A. Biglova, “Desirable properties of an ideal risk measure in portfolio theory,” *International Journal of Theoretical and Applied Finance*, vol. 11(1), pp. 19–54, 2008.
- [18] G. Szego, “Measures of risk,” *Journal of Banking and Finance*, vol. 26(7), pp. 1253–1272, 2002.
- [19] D. A. Peiro, “Skewness in financial returns,” *Journal of Banking and Finance*, vol. 6, pp. 847–862, 1999.
- [20] S. Ortobelli, S. T. Rachev, S. Stoyanov, F. J. Fabozzi, and A. Biglova, “The proper use of risk measures in the portfolio theory,” *International Journal of Theoretical and Applied Finance*, vol. 8, pp. 1107–1133, 2005.
- [21] P. Artzner, F. Delbaen, J. M. Heath, and D. Eber, “Coherent measures of risk,” *Mathematical Finance*, vol. 9, pp. 203–228, 1999.
- [22] R. T. Rockafellar and S. Uryasev, “Optimization of conditional value-at-risk,” *Journal of Risk*, vol. 2, pp. 21–41, 2000.

- [23] A. G. Quaranta and A. Zaffaroni, “Robust optimization of conditional value at risk and portfolio selection,” *Journal of Banking and Finance*, vol. 32(10), pp. 2046–2056, 2008.
- [24] R. T. Rockafellar and S. Uryasev, “Conditional value-at risk for general loss distributions,” *Journal of Banking and Finance*, vol. 26(7), pp. 1443–1471, 2002.
- [25] C. Acerbi and D. Tasche, “Expected shortfall: A natural coherent alternative to value-at-risk,” *Banca Monte dei Paschi di Siena*, vol. 31(2), pp. 379–388, 2002b.
- [26] R. B. Myerson, *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.
- [27] C. Blum and A. Roli, “Metaheuristics in combinatorial optimization: Overview and conceptual comparison,” *ACM Computing Surveys*, vol. 35(3), pp. 269–308, 2003.
- [28] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms - 2nd Edition*. Luniver Press, 2010.
- [29] F. Rothlauf, *Design of Modern Heuristics*. Springer-Verlag Berlin Heidelberg, 2011.
- [30] F. Glover and M. Laguna, “Tabu search,” Boston: Kluwer, 1997.
- [31] G. Tollo and A. Roli, “Metaheuristics for the portfolio selection problem,” *International Journal of Operations research*, vol. 5, pp. 13–35, 2008.
- [32] F. Glover and K. Srensen, “Metaheuristics,” *Scholarpedia*, vol. 10(4), p. 6532, 2015. revision 149834.
- [33] Wikipedia, “Parallel metaheuristic — Wikipedia, the free encyclopedia.” https://en.wikipedia.org/wiki/Parallel_metaheuristic. Accessed:29-09-2015].
- [34] S. Kirkpatrick, C. Gelatt Jr., and M. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220(4598), pp. 671–681, 1983.
- [35] P. Hansen, N. Mladenovic, and J. Perez, “Variable neighbourhood search: methods and applications,” *4OR*, vol. 6(4), pp. 319–360, 1983.
- [36] J. H. Holland, *Adaptation in natural and artificial systems*. University of Michigan Press, 1975.

- [37] M. Dorigo, *Optimization, Learning and Natural Algorithms (in Italian)*. PhD thesis, Dipartimento di Elettrotecnica, Politecnico di Milano, Milan, Italy, 1992.
- [38] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.
- [39] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Kayseri, Turkey, 2005.
- [40] R. Storn and K. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," *J. of Global Optimization*, vol. 11(4), pp. 341–359, 1997.
- [41] "Metaheuristic network." <http://www.metaheuristics.net/index.php%3Fmain=1.html>. Accessed: 2015-09-29.
- [42] J. McCarthy and J. P. Hayes, "Some philosophical problems from the standpoint of Artificial Intelligence," in *Machine Intelligence 4*, pp. 463–502, Edinburgh University Press, 1969.
- [43] D. Karaboga, B. Gorkemli, C. Oztruk, and N. Karaboga, "A comprehensive survey: artificial bee colony (abc) algorithm and applications," *Artificial Intelligence Review*, vol. 42(1), pp. 21–57, 2012.
- [44] Y. Xin-Shi, B. Gebrail, and N. . Sinan, Melih, *Metaheuristics and optimization in civil engineering*. New York: Yale University Press John Wiley, 2015.
- [45] G. Beni, B. Hackwood, and J. Wang, "The engineering of cellular robotic systems," in *Proc. IEEE Int. Symp. Intelligent Control*, pp. 70–75, 1988.
- [46] J. Kennedy, Y. Shi, and R. C. Eberhart, *Swarm Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.
- [47] E. Bonabeau, M. Dorigo, and G. Thulaz, "Swarm intelligence: From natural to artificial systems," Oxford University Press, 1999.
- [48] M. Millonas, *Swarms, Phase Transitions, and collective intelligence*. AdisonWesley Publishing Company, 1994.
- [49] S. Binitha and S. S., Siva, "A survey of bio inspired optimization algorithms," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 2(2), pp. 2231–2307, 2012.

- [50] I. Boussaid, J. Lepagnot, and P. Siarry, “A survey on optimization metaheuristics,” *Information Sciences*, vol. 237(2), pp. 82–117, 2013.
- [51] R. Poli, J. Kennedy, and T. Blackwell, *Particle swarm optimization*. 2007.
- [52] E. Ozcan and C. Mohan, “Particle swarm optimization: Surfing the waves,” in *IEEE Congress Evol. Comput.*, vol. 3, pp. 70–75, 1999.
- [53] G. Kumar, S. Mohagheghi, J. C. Hernandez, and Y. Del Valle, “Particle swarm optimization: Basic concepts, variants and applications in power systems,” *IEEE*, vol. –, pp. 171–195, 2008.
- [54] R. Eberhart and Y. Shi, “Particle swarm optimization: Developments, applications and resources,” in *Proc. IEEE Congr. Evol. Comput*, vol. 1, pp. 81–86, 2001.
- [55] X. Hu, R. Eberhart, and Y. Shi, “Recent advances in particle swarm,” in *Proc. IEEE Congr. Evol. Comput*, vol. 1, pp. 90–97, 2004.
- [56] M. N. R. Bahareh Nakisa, Mohd Zakree Ahmad Nazri and S. Abdullah., “A survey: Particle swarm optimization based algorithms to solve premature convergence problem,” *Journal of Computer Science*, vol. 10(9), pp. 1758–1765, 2014.
- [57] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *Proc. Of the IEEE Congress on Evolutionary Computation*, pp. 69–73, 1998.
- [58] A. Abrahm and A. K. S. Das, “Particle swarm optimization and differential evolution algorithms: Technical analysis, applications and hybridization perspectives,” 2008.
- [59] Y. Shi and R. C. Eberhart, “Parameter selection in particle swarm optimization,” in *Proc. 7th Int. Conf. Evol. Program. (EP)*, pp. 591–600, 1998.
- [60] Y. Shi and R. C. Eberhart, “Empirical study of particle swarm optimization,” in *Proc. IEEE Congr. Evol. Comput. (EP)*, vol. 3, pp. 1945–1950, 1999.
- [61] A. P. Engelbrecht, *Computational Intelligence: An introduction*. John Wiley and Sons, 2007.

- [62] M. Clerc, “The swarm and the queen: towards a deterministic and adaptive particle swarm optimization.,” *Proc. 1999 ICEC*, vol. —, pp. 1951–1957, 1999.
- [63] M. Koshino, H. Murata, and H. Kimura, “Improved particle swarm optimization and application to portfolio selection,” *Wiley Periodicals, Inc. Electronics and Communications in Japan*, vol. 90(3), pp. 13–25, 2007.
- [64] J. Kennedy and R. Mendes, “Population structure and particle swarm performance,” in *Proceeding of the 2002 Congress on Evolutionary Computation CEC*, pp. 1671–1676, 2002.
- [65] J. Kennedy and R. Mendes, “Neighbourhood topologies in fully-informed- and best-of-neighbourhood particle swarms,” in *Proc. IEEE Int. Workshop Soft Computing in Industrial Applications*, pp. 45–50, 2003.
- [66] E.-G. Talbi, *Metaheuristics - From Design to Implementation*. John Wiley and Sons, 2009.
- [67] J. Kennedy, “Small worlds and mega-minds: Effects of neighbourhood topology on particle swarm performance,” in *Proc. IEEE Congress Evol. Comput.*, vol. 3, pp. 1931–1938, 1999.
- [68] J. Kennedy and R. Mendes, “Population structure and particle swarm performance,” in *Proc. IEEE Congress Evol. Comput.*, vol. 2, pp. 1671–1676, 2002.
- [69] D. Karaboga and B. Akay, “A survey: algorithms simulating bee swarm intelligence,” *Artificial Intelligence Review* 31, pp. 61–85, 2009.
- [70] J. Bansal, H. Sharma, and S. Jadon, “Artificial bee colony algorithm: a survey,” *International Journal of Advanced Intelligence Paradigms*, vol. 5(1/2), pp. 123–159, 2013.
- [71] C. Wei, M. Hui, Y. Yiping, and S. Mengrong, “Application of artificial bee colony algorithm to portfolio adjustment problem with transaction costs.” <http://dx.doi.org/10.1155/2014/192868>, 2014.
- [72] M. Tuba and N. Bacanin, “Artificial bee colony algorithm hybridized with firefly algorithm for cardinality constrained mean-variance portfolio selection problem,” *Applied Mathematics and Information Sciences*, vol. 8(6), pp. 2831–2844, 2014.

- [73] B. Akay and D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization," *Information Science*, vol. 192, pp. 120–142, 2012.
- [74] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (abc) algorithm," *Applied Soft Computing*, vol. 8(1), pp. 687–697, 2008.
- [75] D. Haijun and F. Qingxian, "Bee colony algorithm for the function optimization," *Science Paper Online*, August.
- [76] P. Tsai, J. Pan, B. Liao, and S. Chu, "Enhanced artificial bee colony optimization," *International Journal of Innovative Computing, Information and Control*, vol. 5(12), 2009.
- [77] K. Mina, Husseinzadeh, N. Nasim, and K. Ali, Husseinzadeh, "Dis-abc: A new artificial bee colony algorithm for binary optimization." <http://dx.doi.org/10.1016/j.asoc.2011.08.038>, 2011.
- [78] D. Karaboga and B. Basturk, "Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems," in *Proc. Of the 12th International Fuzzy Systems Association World Congress on Foundations of Fuzzy Logic and Soft Computing IFSA '07*, pp. 789–798, Springer, Berlin, 2007.
- [79] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186(2), pp. 311–338, 2000.
- [80] D. Karaboga and B. Akay, "A modified artificial bee colony (abc) algorithm for constrained optimization problems," *Applied Soft Computing*, vol. 11(3), pp. 3021–3031, 2011.
- [81] A. Bilal, "Chaotic bee colony algorithms for global numerical optimization," *Expert Systems with Applications*, vol. 37(8), pp. 5682–5687, 2010.
- [82] S. Rahnamayan, "Opposition-based differential evolution," *IEEE Transaction on Evolutionary Computation*, vol. 12(1), pp. 64–79, 2008.
- [83] K. Xiangyu, L. Sanyan, and W. Zhen, "A new hybrid artificial bee colony algorithm for global optimization," *International Journal of Computer Science*, vol. 10(1), pp. 287–301, 2013.

- [84] S. Kumar, V. Sharma, and R. Kumari, “Improved onlooker bee phase in artificial bee colony algorithm,” *International Journal of Computer Applications*, vol. 9(6), pp. 31–39, 2014.
- [85] Y. XS, “Engineering optimizations via nature-inspired virtual bee algorithms,” in *Mira J, Ivarez JR (eds) Artificial intelligence and knowledge engineering applications: a bioinspired approach*, vol. 3562, pp. 317–323, Springer, 2005.
- [86] H. Webbe and M. Farooq, “The wisdom of the hive applied to mobile ad-hoc networks,” in *Swarm Intelligence Symposium, 2005*, pp. 341–348, Proceedings 2005 IEEE, 2005.
- [87] H. Webbe, M. Farooq, and Y. Zhang, “Beehive: An efficient fault-tolerant routing algorithm inspired by honey bee behaviour,” in *Dorigo M, Birattari M, Blum C., Gambarella LM, Mondada F, Stutzle T (eds) ANTS workshop. Lecture notes in computer science*, vol. 3172, pp. 83–94, Springer, Berlin, 2004.
- [88] P. Lucic and D. Teodorovic, “Bee system: modeling combinatorial transportation engineering problems by swarm intelligence,” in *Preprints of the TRISTAN IV triennial symposium on transportation analysis, Sao Miguel, Azores Islands*, pp. 441–445, Portugal, 2001.
- [89] D. Teodorovic and D. M, “Bee colony optimization - a cooperative learning approach to complex transportation problems,” in *Proceedings of the 16th mini-EURO conference on advanced OR and AI methods in transportation*, pp. 51–60, 2005.
- [90] Wikipedia, “Optical communication — Wikipedia, the free encyclopedia.” https://en.wikipedia.org/wiki/Optical_communication, 2015. Accessed: 07-02-2016.
- [91] A. Baijal, S. Chauhan, and T. Jayabarathi, “Application of pso, artificial bee colony and bacterial foraging optimization algorithms to economic dispatch: An analysis,” *International Journal of Computer Science*, vol. 8(4), pp. 467–470, 2009.
- [92] D. S., A. Biswas, S. Dasgupta, and A. Abraham, “Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications,” *Foundations of Computational Intelligence Volume 3*, vol. 203, pp. 23–55, 2009.

- [93] S. Oliveira, R. Schirrun, and A. C. C. de Medeiros, “On the performance of an artificial bee colony optimization algorithm applied to the accident diagnosis in a pwr nuclear power plant,” *International Nuclear Atlantic Conference - INAC*, 2009.
- [94] Wikipedia, “Genetic algorithm — Wikipedia, the free encyclopedia.” https://en.wikipedia.org/wiki/Genetic_algorithm. Accessed: 28-01-2016.
- [95] Wikipedia, “Cinii — Wikipedia, the free encyclopedia.” <https://en.wikipedia.org/wiki/CiNii>. Accessed: 28-01-2016.
- [96] A. Bullinaria, John and A. K., “Artificial bee colony training of neural networks,” *Nature Inspired cooperative strategies for Optimization (NCSO 2013)*, vol. 512, pp. 191–201, 2013.
- [97] R. Rao, S. V. L. Narasimham, and M. Ramalingaraju, “Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm,” *Int J Electr Power Energy Syst Eng*, vol. 1(2), pp. 116–122, 2008.
- [98] X. Shi, Y. Li, H. Li, R. Guan, L. Wang, and Y. Liang, “An integrated algorithm based on artificial bee colony and particle swarm optimization,” *In 2010 sixth international conference on natural computation (ICNC)*, vol. 5, pp. 2586 – 2590, 2010.
- [99] C. Acerbi, “Spectral measures of risk: A coherent representations of subjective risk aversion,” *Journal of Banking and Finance*, vol. 26, pp. 1505–1518, 2002.
- [100] T. Fischer, “Examples of coherent risk measures depending on one-sided measures,” *European Journal of Operational Research*, vol. 185(3), pp. 1542–1547, 2001.
- [101] R. T. Rockafellar, S. Uryasev, and M. Zabarankin, “Generalized deviations in risk analysis,” *Finance and Stochastics*, vol. 10, pp. 51–74, 2006.
- [102] F. Delbaen, “Coherent risk measures on general probability spaces,” in *Advance in Finance and Stochastics. Essays in Honor of Dieter Sondermann*, pp. 1–37, Springer, 2002.
- [103] D. Tasche, “Expected shortfall and beyond,” *Journal of Banking and Finance*, vol. 26, pp. 1519–1533, 2002.

- [104] T. Fischer, “Risk capital allocation by coherent risk measures based on one-sided moments.,” *Insurance: Mathematics and Economics*, vol. 32(1), pp. 135–146, 2003.
- [105] K. E. Parsopoulos and M. N. Vrahatis, “Particle swarm optimization method in multiobjective problems.,” *Proceedings ACM Symposium on Applied computing*, pp. 603–607, 2002.
- [106] D. Luenberger, *Linear and Non-linear programming*. Adison Wesley Publishing Co., Menlo Park, 2 ed., 1984.
- [107] “Artificial bee colony (abc) algorithm.” <http://mf.erciyes.edu.tr/abc/software.htm>. Accessed: 10-09-2015.
- [108] D. Karaboga and B. Basturk, “On the performance of artificial bee colony (abc) algorithm,” *Applied Soft Computing*, vol. 8, pp. 687–697, 2007.