Ca' Foscari
University
of Venice

Master's Degree Programme
Single Cycle (D.M. 270/2004)
in Economics and Finance-Finance

Final Thesis

PSO and BFO: two alternative metaheuristics for portfolio optimization problem

Supervisor
Prof. Marco Corazza

Graduand

Nicolò
Mazzucato
844766

Academic Year
2017 / 2018

Ad Anna

Contents

Introduction

As part of Modern Portfolio Theory, Portfolio management is a subject introduced by Markowitz back in 1950s. In Chapter 1 of this paper, we describe the Mean- Variance portfolio selection model proposed by Markowitz. Since its introduction, this model has been considered as the standard model. Although it has many advantages, with the years passing and the increased complexity of the markets, the model showed all its limits. The main drawbacks arises from the unrealistic assumptions at the base of the model. In other words, the assumptions are not able to present the real world and the risk measure used. Therefore, there was the need of a new class of risk measures, coherent risk measure, suitable for financial portfolios. The coherent measure of risk that belongs to this class chosen for this paper is the two-sided risk measure introduced by Chen and Wang in 2008. Chapter 2 describes the metaheuristics and in particular focuses on those chosen for this paper. Metaheuristic can be describe as trial and error optimization techniques able to find high level solutions to complex problems. Those high-level solutions, although high quality solutions, are not the optimal ones. However, metaheuristics find good solutions in a reasonable amount of time. In this paper we decided to choose bio-inspired metaheuristics, in particular Particle Swarm Optimization and Bacterial Foraging Optimization. In Chapter 3 we presented an alternative model to the one introduced by Markowitz, that is the realistic portfolio proposed by Corazza, Fasano and Gusso. This strategy allows to make the analysis more realistic by overcoming the limits of the model described in Chapter 2. However, in order to effectively solve the NP-hard problem that arises from the use of the two-sided risk measure combined with the realistic portfolio chosen, we applied an exact penalty method, which allows to transform the constrained problem into an unconstrained one. Finally, in Chapter 4, we applied PSO and BFO to solve the portfolio selection problem presented in the previous chapter. For this application, the data used are the daily closing prices of DAX 30 index from March 2014 to November 2018. The periods considered are eight, and each one consists of 8 in-sample months and 3 out-of-sample. In addition, we also analyzed the respect of the monotonicity property of the risk measure. Lastly, we carried out a comparison between the given respectively between P3SO and BFO.

Chapter 1

Portfolio Selection Theory

Portfolio selection is one of the cornerstones of economics and finance. The classical framework of portfolio selection was theorized by Markowitz in 1952 and it is called Mean-Variance portfolio selection model. This model represents one of the pillars of the modern portfolio theory. In this chapter we are going to describe the Markowitz model for portfolio selection as well as its limits and the alternative models which try to overcome the disadvantages.

1.1 Markowitz Model

As we mention in the introduction to this chapter, Markowitz who introduced the Mean-Variance framework for the portfolio selection, is the father of the Modern Portfolio Theory. The name Mean-Variance portfolio selection model comes from the two criteria used to select the best portfolio, the expected return (mean) and the standard deviation (variance). In fact, in its seminal work [1], Markowitz states that investors should perceive the expected return as a desirable factor and the variance as an undesirable factor. The latter factor of each asset of the portfolio eventually determines the level of risk of the entire portfolio. However, portfolio selection model proposed by Markowitz allows to select the best combination of assets with lowest variance and as high as possible return. In other words, for each level of risk of the entire portfolio the model is able to maximize the return by combining the different assets of the portfolio. It should be clear that a trade-off between risk and return arises. Furthermore, due to the diversification, the assets of the portfolio collectively have a lower variance than they have singularly.

Assumptions

In order to be meaningful, the Mean-Variance portfolio selection model needs some basic assumptions. We are going to describe them as follows:

- Frictionless markets. This assumption includes the absence of any transaction costs, taxation. Furthermore, the assets are assumed to be indefinitely divisible;
- Investor are price-taker thus, their action or decisions are not able to modify the distribution of security's returns;
- Investment opportunity is expressed by a probability distribution of returns and is measured over a unique holding period;
- Maximization of the investment's return is the objective of the investors;
- Rational and risk adverse investors are aware of the risk and demand higher return for accepting higher risks.

1.2 Portfolio selection model

In this section we are going to talk about the selection process of the model. After a thoroughly discussion of the measure of risk and return, we are going to analyze the mean-variance dominance criterion, that is the ratio underneath the selection process and finally the portfolio selection itself.

Markowitz model consists of three steps:

1. Identification of a tool which is able to effectively measure the uncertainty of the investment, that is an appropriate measure of risk;
2. Definition and selection of a criterion whereby classify the investment choices into an efficient set and an inefficient set. The two sets are mutually exclusive;
3. Given the investor's risk appetite, selection of the optimal portfolio.

1.2.1 Risk and Return of the Portfolio

Expected return and the risk of the portfolio are the two criteria through which the model works. However, given the stochastic nature of the investment choice the two measure are:

- Mean of the rates of return (single-period). In this contest, this statistical measure applied to the rates of the return is used to express the desired profitability of the investment choice.

- Variance of the rates of return (single-period). This measure serves to quantify the undesired risk of the investment choice.

We will break down the measures just presented above in order to make them mathematically meaningful. In order to do that, let *X* be a discrete random variable $X = (x_1, p_1), \dots, (x_i, p_i), \dots (x_M, p_M)$ where x_i with $i = 1, \dots, M$ is the *i-th* realization of *X*, the possible return of an asset, whereas p_i, with $i = 1, \dots, M$, is the probability that x_i occurs, where $0 \leq p_i \leq 1$ for all *i* and $\sum_{i=1}^{M} p_i = 1$. Given this information mean and variance can be described as:

$$E(X) = \sum_{i=1}^{M} x_i p_i$$

$$Var(X) = \sum_{i=1}^{M} \left(x_i - E(x)\right)^2 p_i$$

Of course, this is the case when the random variable *X* is discrete. Since, usually the rates of return are assumed to be continuous the variable *X* has a cumulative distribution function *F*(.) and a probability density function *p*(.). Thus, the two measures can be expressed as:

$E(X) = \int_{-\infty}^{+\infty} t dF(t)$ or $E(X) = \int_{-\infty}^{+\infty} t f(t)\, dt$

$Var(X) = \int_{-\infty}^{+\infty} \left(t - E(X)\right)^2 dF(t)$ or $Var(X) = \int_{-\infty}^{+\infty} \left(t - E(X)\right)^2 f(t) dt$

So far, we analyzed and discussed the mean and variance of the single assets. However, the portfolio is a combination of multiple assets. In order to obtain the mean of the entire portfolio, we need to define:

- R_i the random variable which represents the return of the *i-th* asset;
- x_i the part of the capital invested in the *i-th* asset;
- r_i expected rate of return of the *i-th* asset;
- σ_i^2 variance of the *i-th* asset.

Then, given the information described above, the mean and the variance of the entire portfolio which can be respectively expressed as follow:

$E(R_P) = \sum_{i_1}^{N} x_i r_i = r_P$

$$Var(R_P) = \sum_{i=1}^{N} x_i^2 \sigma_i^2 + 2 \sum_{i=1}^{N} \sum_{j=i+1}^{N} x_i x_j \sigma_{i,j} = \sum_{i=1}^{N} x_i^2 \sigma_i^2 + \sum_{i=1}^{N} \sum_{j=i+1}^{N} x_i x_j \rho_{i,j} \sigma_i \sigma_j = \sigma_P^2$$

Where:

- $\rho_{i,j} = [-1, +1]$ represents the Bravais-Pearson linear correlation coefficient between R_i and R_j
- $\sigma_{i,j} = \rho_{i,j} \sigma_i \sigma_j$ represents the covariance between R_i and R_j.

It may result useful for the next sections, already providing with the matrix notation of mean and variance of the portfolio:

- $r_P = \mathbf{x}'\mathbf{r}$
- $\sigma_P^2 = \mathbf{x}'\mathbf{V}\mathbf{x}$

1.2.2 Mean-Variance criterion

In this section we are going to present the fundamental mechanism which allows the model to separate among all the assets selected the efficient and inefficient ones. Considered *X* and *Y,* two random variables with means and variances equal to μ_X, μ_Y and σ_X^2, σ_Y^2, for the Mean-Variance dominance criterion we can state that *X* dominates *Y,* if and only if:

- $\mu_X \geq \mu_Y$
- $\sigma_X^2 \leq \sigma_Y^2$

Thus, we can say that when a portfolio, given the same or lower variance of other portfolios, has a greater expected return then it dominates. In other words, it is efficient.

As we introduced before, the main objective of the portfolio selection problem is to minimize the variance, among all the available portfolios. Therefore, we can provide with a first and basic algebraically version of the portfolio selection process when an investor can invest in *N* assets, that is nothing but a minimization of a convex function subject to linear constraints:

minimize $x_1, \dots, x_N \quad \mathbf{x}'\mathbf{V}\mathbf{x}$

subject to $\begin{cases} \mathbf{x}'\mathbf{r} = \pi \\ \mathbf{x}'\mathbf{e} = 1 \\ \mathbf{x} \geq 0 \end{cases}$

Where:

- $\mathbf{x}$ is the *N-column* vector of the proportion of capital invested in each of the *N* assets of the portfolio;
- $\mathbf{V}$ is the N x N variance-covariance matrix[1] between the assets;
- $\mathbf{r}$ is the *N-column* vector which storages the mean returns $r_1, \ldots, r_N$;
- $\mathbf{e}$ is a *N*-column vector of ones;
- π rate of return that investor wants the portfolio to earn;
- $\mathbf{x} \geq 0$ is the constraint that requires non-negative portions of capital invested in the portfolio[2];

The reason why the function is convex relies on the convexity of $\mathbf{x}'\mathbf{V}\mathbf{x}$ matrix due to the fact that, the matrix $\mathbf{V}$ is positive definite[3], as well as the convexity of the constraints. The unique solution of the problem is:

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

where:

- $\alpha = \mathbf{r}'\mathbf{V}^{-1}\mathbf{r}$
- $\beta = \mathbf{r}'\mathbf{V}^{-1}\mathbf{e}$
- $\gamma = \mathbf{e}'\mathbf{V}^{-1}\mathbf{e}$

Therefore, taking into consideration that the composition of the efficient frontier depends on: $\mathbf{V}$, **r** and μ, we have that:

$$E(R_{P^*}) = \ r_{P^*} = \mathbf{x}'\mathbf{r} = \mu$$

$$Var(R_{P^*}) = \sigma_{P^*}^2 = \frac{\gamma\pi^2 - 2\beta\pi + \alpha}{\alpha\gamma - \beta^2} \ ,$$

[1] We implicitly assume that the matrix is nonsingular. This property of the variance-covariance matrix allows that assets to be not perfectly correlated with the return of the portfolio.

[2] In other words, this constraint means that short-selling is not allowed.

[3] For our purpose, a positive definite matrix means that all assets are risky.

$$StDev(R_{P^*}) = \sigma^2_{P^*} = \left(\frac{\gamma\pi^2 - 2\beta\pi + \alpha}{\alpha\gamma - \beta^2}\right)^{1/2}$$

Where $Var(R_{P^*})$ displays a parabola in the variance-mean plane, whereas $StDev(R_{P^*})$ describes a hyperbola in the standard deviation-mean plane.

1.2.2 Portfolio Selection

Portfolio selection is the procedure aimed to select the optimal portfolio contingent to the investor's risk appetite. When investors have to select the portfolio that gives the highest rate of return, we need to keep in mind that they prefer high returns and they risk-adverse. Thus, they prefer high returns but stable and hate uncertainty. However, the utility function allows investors to carry out the portfolio selection. Technically speaking, the utility function *U* is a function defined on *R* (real numbers set) that represents different investor's wealth levels. In practical terms we take two random wealth variable *x* and and *y* and compare the values $E[U(x)]$ and $E[U(y)]$ and prefer the highest expected value. Moreover, it is an increasing continuous function because given two real values *x* and *y,* with $x > y$, $U(x) > U(y)$. Another feature of this function is that it is concave, in this way the risk aversion of the investor can be replicated. In fact in order to be concave, given an interval $[a, b]$ of real numbers where $0 \leq \alpha \leq 1$ and any *x* and *y* thus, $U[\alpha x + (1 - \alpha)y] \geq \alpha U(x) + (1 - \alpha)U(y)$ holds. The expected utility approach that consists in measuring individual utility function relying on the values assigned by investors to various risky alternatives, is consistent with the portfolio selection problem introduced by Markowitz if at least one of the following requirements is met:

- The utility function is a quadratic utility function[4], which can by mathematically expressed as:

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

[4] The quadratic utility function employed in the Markowitz's approach can by represented by:
$U(R_P) = R_P - \frac{a}{2}R_P^2$
Where:
— R_P is the random variable of the portfolio returns
— a is the coefficient that captures the risk aversion level of the investor. The conditions that is needs to respect are: being strictly positive and increase as the risk aversion of the investor increases.

- Assuming that the distribution of the assets' returns is a Gaussian distribution.

1.3 Criticisms to Markowitz Model

Although Markowitz portfolio selection problem is a cornerstone of the Modern Portfolio Theory is not free of criticisms and limits. In particular, most of the criticisms that we are going to describe in this section, move from the mean-variance framework.

However, among all the drawbacks the most evident ones are the unrealistic assumptions of the model. As pointed by Corazza and Favaretto in [2] among those unrealistic assumptions in the real world we have the following issues: frictions of the markets, transaction costs, taxes, non -indefinitely divisible assets and the absence of a minimum order size that does not allow investors to buy and sell securities' fractions. Another strong assumption not respected is the one referred to the stationarity of the world. In fact, this assumption allows investors to determine the optimal rate of return. In practice, investors' expectations are biased and subject to uncertainty. Moreover, in the Markowitz framework investors are assumed to be able to quantify their risk appetite and properly use the quadratic utility function.

Another limit of this model is the diversification effect on the portfolio risk. The inclusion of additional assets considerably reduces the portfolio variance. However, complexity related problems could arise due to the increased dimension of the variance-covariance matrix as result of the inclusion of more assets in the portfolio.

Another unrealistic assumption is the absence of asymmetries and then the subsequent hypothesis about the kurtosis in return distributions. As mentioned above, the distribution of return is assumed to be normal or gaussian but in this way extreme events are underestimated. Indeed, financial instruments are not normally distributed, rather their distributions are skewed and most important they have fat tails[5]. This feature entails that every type of utility function, different from the quadratic one, can't be used because returns are not normally distributed.

[5] Normal distribution has kurtosis equal to 3. The shape parameter that describes how much probability there is on the tails of the distribution, can be used as benchmark. In fact, heavy-tailed distributions such as Laplace

Moreover, another limit is the trade-off in the optimization process where the two criteria, minimization of the risk and maximization of returns are conflicting. The reason lies on the increase of the risk as the returns increase.

Finally, there is the criticism against the variance as measure of risk of the investment choice used in the Markowitz portfolio selection problem. The main limit is the nature of this measure. In fact, being a symmetric risk measure it equally weights down-side and up-side risk and even positive variance is considered risky. Of course, it is easy to understand that positive variance is not treated as an unfavorable fact due to possibility to earn higher returns.

1.4 Improvements of Markowitz Model

In this section we are going to present the improvements allowing portfolio selection problem to be more realistic. In order to achieve this objective, the improvements presented belong to two different categories. The first category refers to changing the constraints in order to contemplate some realistic facts such as transaction costs and taxes (market frictions) as well as the number of assets included in the portfolio. On the other hand, the second category considers the measure of risk. As respect to the first category, mix-integer constraints are able to make the portfolio more realistic and they can be divided into three categories:

- negotiation minimum asset lots related restrictions;
- restrictions concerning the maximum positive integer number of different assets that can be purchased by investors;
- restrictions related to the minimum positive integer number of lots of an asset that can be negotiated.

However, mixed-integer constraints are not free of disadvantages. This modification of the constraints makes become this mathematical programming problem a NP-complete problem[6]. However, another implication is related to the fact that in order to find the solution of such a mathematical programming problem is a NP-hard

distribution or Student’s t-distribution have values greater than 3. Therefore, their tails approach 0 more slowly and the probability assigned to outliers is greater than Normal distribution.

[6] They are problem where the process to find their solution is time consuming and the time grows as the size increases.

problem[7].

Although the solutions just presented are valuable alternatives for the purpose of this paper, we are going to focus on the improvements belonging to the second category. The definition of alternative risk measures is referred to this category. Among all the alternative measures of risk we start from the one proposed by Markowitz, the semi-variance. This measure can be defined in two ways, depending on the type of rate of returns. If we deal with discrete rate of returns, semi-variance can be expressed as follows:

$$semi - Var(R) = \sum_{i=1}^{M} (\min\{0, x_i - E(R)\})^2 p_i$$

On the contrary, when *R* (rate of return) is a continuous random variable the semi-variance can be expressed as:

$$semi - Var(R) = \int_{-\infty}^{+\infty} (\min\{0, t - E(R)\})^2 dF_R(t)$$

or deriving the cumulative distribution function of *R:*

$$semi - Var(R) = \int_{-\infty}^{+\infty} (\min\{0, t - E(R)\})^2 f_R(t) d(t)$$

Semi-variance allows to take into consideration only negative performances and upside risk is no longer considered. In this way, positive portfolio performances are now treated as upside potential since any rational investor will always invest on assets with performances above expectations. As stated by Markowitz [3] through this measure of risk it is possible to decrease losses, whereas the variance, as measure of risk, takes into consideration both positive and negative outliers along the distribution as equally undesired. Although this measure is considered an alternative solution to the variance approach has some critical drawbacks. Except to the fact that semi-variance is not able to offset all the problems linked to variance measure of risk, it entails more calculation complexity.

Beyond the measure of risk proposed by Markowitz for its portfolio selection problem, we are now going to present other alternatives.

The first measure of risk we are going to consider is Value at Risk (VaR). This

[7] They are problems that have a hardness level at least as NP-complete problems.

measure was introduced in 1994 and later approved by Basel Committee as answer to the need of covering institutions from market risk. Although VaR is consider a sophisticated risk measure, it does not give any insight of the maximum loss you can collect. Conversely, VaR provides information about minimum expected loss over a predetermined time with a given probability. Technically speaking, it defines the minimum loss given a certain confidence level ϵ over a predetermined time horizon. If the risky payoff is defined be *X* and the confidence level $(1-\epsilon)$ with $0 \leq \epsilon \leq 1$, then VaR can be expressed as follow:

$$VaR_{\epsilon}(x) = -\inf\{x|P(X \leq x) \geq \epsilon\}$$

Moreover, this VaR has two important properties missing in variance measure of risk:

- If *C* represents a riskless payoff, the transitional invariance property assures that:

  $$VaR_{\epsilon}(X + C) = VaR_{\epsilon}(X) - C$$

- If λ represents a positive constant, then positive homogeneity property assures that:

  $$VaR_{\epsilon}(\lambda X) = \lambda VaR_{\epsilon}(X)$$

Beyond the advantages of this measure of risk such as the ability to consider only downside risk and represent in a compact way the risk level, VaR has some drawbacks as well. Basically, in portfolio selection problem we require a measure of risk to be coherent. This means that the measure has to respect the following properties:

- positive homogeneity;
- sub-additivity;
- monotonicity;
- transitional variance.

Given the missing satisfaction of the sub-additivity property, VaR can’t be considered as a coherent measure of risk. In particular the diversification effect is not evident and sometimes it is possible that portfolio VaR can be higher than the summation of single asset VaR. This possibility is presented by the following formula:

$$VaR_{\epsilon}(X+Y) > VaR_{\epsilon}(X) + VaR_{\epsilon}(Y)$$

Where *X* and *Y* indicates two risky payoffs. Moreover, the failed satisfaction of this property entails the lack of convexity[8] making VaR not suitable for measuring real portfolio risks. Elliptic return joint distribution is the only case when VaR satisfies sub-adittivity.

The second drawback has been shown in the definition of VaR. Indeed, it tells the minimum loss that can be collected, whereas it does not shed any light on maximum loss. In other words, the loss can be higher than the one forecasted by the measure of risk. However, the criticisms against VaR concerning its limits was expressed by Szegö in [4]:

"VaR has become another "solution is search of a problem" and was wrongly adopted as risk measure".

In the following section we are going to present different coherent risk measures which respect the all the properties.

1.4.1 Coherent Risk Measures

Coherent risk measures, a new class of measures risk, are thought to be more appropriate to measure financial risk. In this section we are going to describe: Tail Conditional Expectation, Conditional Value at Risk, Expected Shortfall and Worst Conditional Expectation. However, following the definition provided by Szegö, a scalar measure of risk allows to compare different investment opportunities though their risk value. In fact, Szegö defines a measure of risk as $\rho : X \rightarrow R$, that the correspondence, ρ, between a space *X,* such as the returns and a non-negative real number [4]. However, a measure of risk in order to be coherent needs to satisfies the properties we introduced in the previous section, that is:

- *Positive homogeneity*: This property states the magnitude of the risk is linearly affected by the size of the investment and it can be presented as follows:

  $$\rho(\lambda X_i) = \lambda \rho(X_i)$$

  for all positive real number λ and for all X_i, with $i = 1, \dots, N$

[8] Convexity is necessary in rewarding the diversification, that is a fundamental aspect when the portfolio is being optimized.

- *Monotonicity:* This property allows risk measure to include and reflect the preference of the assets by investor. The monotonicity can be presented as:

  $$X_i \preccurlyeq X_j \Rightarrow \rho(X_i) \geq \rho\left(X_j\right)$$

  for all pairs of X_i and X_j, with $i, j = 1, \dots, N$.

- *Sub-additivity*: This property indicates that the inclusion of investment choices decreases the whole risk of the investment. This capability is also called risk contraction.

  $$\rho\left(X_i + X_j\right) \leq \rho(X_i) + \rho(X_j)$$

  for all X_i and X_j, with $i, j = 1, \dots, N$.

- *Transitional invariance:* This property is referred to the fact that the risk of the whole investment decreases whit respect to the risk of the risky investment choice when a riskless investment choice is added to a risky one. Mathematically, this property can be expressed as:

  $$\rho(X_i + X_{N+1}) = \rho(X_i) - \alpha$$

for all X_{N+1} riskless portfolios that have a return equal to α (a real number) and for all risk portfolios X_i, with $i = 1, \dots, N$.

Moreover, a risk measure which satisfies only *positive homogeneity* and *sub-additivity* properties is not coherent, but it is only convex.
We now describe the alternative coherent measures of risk.

- **Tail Conditional Expectation**

  This measure of risk, also called "*TailVar*", allows to compute the expected loss that could be collected in α worst case. The worst cases are expressed as a sample for the portfolio by the percentage $\alpha \in (0,1)$. Tail Conditional Expectation (TCE) can be expressed as the following quantile:

  $$TCE^{(\alpha)}(X) = -E\{X|X \leq x^{(a)}\}$$

  where:

  - *X* is a random variable that represents profit or loss in a *T* horizon;

- $x^{(a)}$ is the distribution quantile.

However, the TCE only holds for continuous portfolio's distribution functions. If general distribution functions are considered TCE could overestimate the probability of the event $\{X \leq x^{(a)}\}$, in other words it could be higher than the set of "worse cases", invalidating the purpose of the measure. Furthermore, the sub-additivity property could be violated. On the contrary, continuous distribution functions ensure TCE to be coherent.

- **Conditional Value at Risk**

Given a significant level $\alpha \in [0,1]\alpha \in [0,1]$, $CVaR_\alpha$ tells the mean,θ_α, of α-tail distribution of $z = f(x, z)$. The distribution function is:

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

where $\zeta_\alpha(x)$ is the VaR_α of the portfolio x. This is possible because solving a simple convex optimization problem allows to separate $CVaR_\alpha$ and VaR_α of the portfolio *x* itself [5]. We have to precise that $\Psi_\alpha(x,.)$ Is a function which is non-decreasing and right-continuous. Furthermore, $\Psi_\alpha(x,\zeta) \longrightarrow 1$ as $\Psi_\alpha(x,\zeta) \longrightarrow \infty$, that is the α-tail distribution $\phi_\alpha(x)$ is well defined through $\Psi_\alpha(x,\zeta)$. However, one of the main advantage of Conditional Value at Risk is the ability to quantify the loss that could occur in the tail of the distribution and it can be expressed by the minimization formula that can be incorporated in the minimization problem.

- **Expectected Shortfall**

Basically, Expected Shortfall is a universal measure of risk, always coherent, which leads to a unique solution of the problem. ES can be thought as the average loss in $(1-\alpha)\%$ of the worst cases. In order to give a mathematically representation let us consider *X* be the profit-loss of a portfolio, *T* the time horizon and $\alpha = A\% \in (0,1)$ the probability level. Then Expected Shortfall can be expressed as:

$$ES^{(\alpha)}(X) = -\frac{1}{\alpha}\left(E\left[X1_{\{X\leq x^{(\alpha)}\}}\right] - x^{(\alpha)}(P[X \leq x^{(\alpha)}] - \alpha\right)$$

where $x^{(\alpha)} = VaR$ and of course the part $x^{(\alpha)}(P[X \leq x^{(\alpha)}] - \alpha)$ is the part of the distribution beyond $\{X \leq x^{(\alpha)}\}$ the has the probability $\alpha = A\% \in (0,1)$. However, only when $P[X \leq x^{(\alpha)}] = \alpha$, it is possible to have $ES^{(\alpha)} = TCE^{(\alpha)}$ [6]. Moreover, when we deal with continuous random variable, ES is equal to CVaR.

- **Worst Conditional Expectation**

  Given the profit or loss of a portfolio represented by a random variable *X* that belongs to $\mathbb{R}$ in a probability space $(\Omega, \mathrm{A}, \mathrm{P})$ and an event *A,* Worst Conditional Expectation (WCE) can be expressed as follows:

  $$WCE_{\alpha}(X) = -inf\{E[X|A]P[A] > \alpha\}$$

  Although TCE has the great advantage to be really easy to apply, WCE is always a coherent measure of risk. However, the disadvantage respect to TCE is the need of knowledge all the probability space on order to be useful in practical terms. Moreover, Artzner et al. in [7] proved that $WCE_{\alpha} \geq TCE^{\alpha}$. In order to be clear we have to specify that, although WCE and TCE seem similar each they are not for the reasons we have described. However, Expected Shortfall is able to join the advantages of them.

1.5 Two-Sided Coherent Risk Measure

Although the alternative risk measures we presented above are mostly coherent, they are not free of limits and drawbacks. The most important one is the one-sided feature the share. In other words, they consider only one part of the distribution, that is the lower one. This limit is particular critical because of the inability of the measure of risk to replicate the real world where investors who buy and sell securities, for example stocks, are exposed to both demand-side risk and offer-side risk. An additional problem of those measures occurs when they try to fix the VaR's problem. Basically, they consider linear probability combination of losses beyond the VaR threshold, rather than considering higher orders of moments of a non-normal distribution. Finally, although those coherent measures of risk are theoretically effective, they are not usually applied in real world optimization selection problems.

A feasible solution in order to overcome those limits is presented by the coherent risk measure proposed by Chen and Wang [8]. This risk measure is able to consider at the same time negative and positive deviations from the expected return (mean). This is why is named two-sided coherent measure of risk and it provides some advantages. One of the main advantages of this measure of risk is its own ability to better describe the investors' attitude and deal with the features of the real world asset distribution. In fact, as we mentioned in the assumptions of the Markowitz portfolio selection problem, the normal distribution of the assets is not respected in the real world. In fact, financial assets do not follow a normal distribution and they present skewness and higher values of kurtosis respect to the one associated to the normal distribution (fat tails). Another advantage is its ability to carefully choose the order of the order of the norm of the downside random loss and the convex combination coefficient in order to manage the third and fourth moment of the distribution. Finally, it provides more stable (estimation error) and solid (from both trading sides) investment decisions by exploiting the entire domain of the loss distribution function.

Chapter 2

Particle Swarm Optimization and Bacterial Foraging Optimization

In this chapter, two metaheuristics, Particle Swarm Optimization (PSO) and Bacterial Foraging Optimization (PSO), are introduced and discussed. In the first part, some fundamental concepts such as metaheuristics and Swarm Intelligence are provided. While, in the second part the analytical formulation, the modifications, the variants and lastly a comparison between the two techniques are presented.

2.1 Metaheuristics

Many real world-optimization problems are NP-hard problems and, as such, they turn out to be intrinsically complex and high computing effort demand problems. Basically, they are comprised within a large range of fields, from technological one till scientific one. The complexity of these problems is given by: the sparse and large-dimension search spaces due to the hard constraints and the fact that they are multitude of objective problems which take into account hard-to-evaluate optimization functions. Furthermore, they are time varying problems and they also need to handle a huge amount of data. Based on the complexity of these problems and consequently on the time-consuming limit of operational research classical exact resolution methods, other techniques have been developed in order to overcome these issues. Metaheuristics represent new optimization methods to deal with real-world dimension problems more efficiently [9]. According to [10] metaheuristics evolved from heuristics whose etymology means “to find or to discover by trial and error”. In other words, heuristics can be described as a way by trial and error that allow to search for an acceptable solution in a reasonable time span even though there is no guarantee that the best solution is found as well as the algorithm work all the time. As mentioned above, metaheuristics represent an evolution of heuristics but unfortunately there is no broadly unique and accepted definition of it as stated by Blu, and Roli [11]. Therefore, to manage this issue it has been decided, for the purpose of this paper, to adopt the definition provided by Osman and Laporte [12]: “A metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions”. Hence, metaheuristics can be seen as a problem-independent-strategies that can be adapted to the exact problem. In this way metaheuristics can be a useful method to solve a wide range of problems because of their high applicability. Even though these optimization techniques do not ensure the optimal solutions to the problem, they are capable to calculate sub-optimal solutions by exploiting non-exacts solutions and enhancing them in order to have solutions which fit better the real-world problem requirements. At the same time those solutions are good-quality solutions. Furthermore, metaheuristics have the crucial capability to manage annoying issues related to data, such as uncertainty, approximation errors and missing information. As stated by Osman and Laporte, two fundamental features of metaheuristics are: exploitation and diversification[9]. The first term indicates the ability of the optimization technique to focus on the search in a local region by taking advantage of the accumulated search experience, whereas the second term refers to the generation of diverse solution by exploring the search space. Therefore, the opportunity to rapidly find regions within the search space where high quality solutions are avoiding the chance to waste time in regions already explored or in which there are poor quality solutions, is given by the proper balance between exploitation and exploration. Put it differently, the combination of them should guarantee the achievement of the global optimality preventing the risk that solutions being trapped at local optima. However, to correctly complete the overview of metaheuristics, it may be useful to discuss the main way to classify them. In particular, it has been decided to distinguish the metaheuristics according to the exploration and exploitation. Therefore, the two categories which are considered here are: trajectory-based[10] and population-based.

[9] They are also called intensification and diversification. These terms are basically used to indicate short-term strategies based on randomness rather than medium-term and long-term strategies which are linked to the use of some sort of memory.

[10] The term trajectory refers to kind of search process, in this case it is characterized by a trajectory.

- **Trajectory-based**

  The functioning of the search process of this type of metaheuristics works modifying and replacing the only candidate solution with the best one found in its neighborhood at each step[11]. Trajectory-based metaheuristics essentially have two main advantages with respect to population-based metaheuristics, that is speed and efficiency. In fact, they appear to be faster and more efficient than the others. The reason relies on the single solution they work with at each step. However, they are called exploitation-oriented method, due to the exploitation activity they carry out locally to enhance the solution [9].

- **Population-based**

  On the contrary, these methods deal with a set or a population of candidate solutions at each step. In this case the technique allows to recombine the set of candidate solution in the attempt to compute better results. This means that at each iteration replacement process takes place, whereby some individuals of the population are replaced by new best solutions or some certain ones, contingent to quality-based criterion. One of the most appreciated features of these methods is the high level of diversification they can achieve, due to usage of huge amount of candidate solutions. In fact, they are called exploration-oriented methods [9].

2.2 Swarm Intelligence

A crucial concept in bio-inspired metaheuristics is played by Swarm Intelligence (SI). As a matter of fact, in the past years a huge number of models have been developed for swarm intelligence systems (PSO and BFO are only two of them). The term “swarm” derives from the nature and it can be described as the collection of animals like fishes, birds and insects such as ants, termites and bees performing collective behavior [13]. Basically, single components of the swarm are treated as simple entity even not smart by some authors. Although they can individually show a considerable

[11] Technically speaking, the metaheuristic, starting from an initial solution, generates a trajectory in the search space. However, the system dynamics depends on the of strategy chosen [11]

amount of complexity[12] it is not yet as much as the complexity of the social insect colonies [14]. Indeed, the work that social insect swarms can do is marvelous and clearly outperform the individual one. Quick examples derive from termites that are capable, once consider collectively, to build mounds allowing to maintain comfortable ambient temperature, levels of oxygen as well as carbon dioxide even when the nest grows. Whereas, termite when considered individually they are nothing but meager intelligent insects [15]. However, SI is a branch (although substantially new) of Artificial Intelligence (AI) and it can be described as the collective behavior of decentralized and self-organized swarms[13]. Although the first time this paradigm was used only in the field of cellular robotic systems by Beni, Hackwood, and Wang, it seems to be more appropriate to consider the wider use suggested by Bonabeau, Dorigo and Theraulaz. As stated in [14] "using the expression swarm intelligence to describe only this work seems unnecessarily restrictive: that is why we extend its definition to include any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animals societies". Considering the reasons that Bonabeau and Meyer [15] give to justify why social insects have been so successful, self-organization (SO) is crucial in the swarm systems. SO was firstly introduced in scientific fields such as physics and chemistry, but when applied to social insects underlines the collective behavior that arises from the interaction among individuals. To be more coherent with the purpose of this paper, SO can be seen as a collection of mechanisms by which the structures appearing at a global level derive from the interactions among its lower-level components. Moreover, the rules defining the interactions among swarm components guarantee that there is no external orders which may influence the behavior of the swarm itself. However, the four cornerstones of SO suggested by Bonabeau are:

- *Positive feedback*: they are empirical rules promoting the creation of convenient structures;

[12] Complexity is considered as the amount of task they can carry out. For example, these creatures can process a huge amount of sensory inputs and modelling their behavior given the stimuli they are receiving, or they are able to interact with nestmates as well making decisions according to the high volume of information.

[13] A decentralized swarm is a swarm where each element carries out and defined activity autonomously. Self-organized swarm is a swarm where thanks to cooperation among the elements of the swarm itself a common behavior arises without any order from a central point.

- *Negative feedback*: acts to stabilize the collective pattern by counterbalancing the positive feedback;
- *Fluctuations*: usually represented by random walks and errors, its scope is to confer randomness to emergent structures which allows to discover new solutions;
- *Multiple Interactions:* refers to the fact that multiple agents need to tolerate each other. In other words, the warm has to be composed by many agents the tolerate each as well as use their own results and other results.

However, according to Valle et al. in [16],in order to give an exhaustive definition of SI, the swarm in addition to be self-organized, it needs to be intelligent. Thus, five principles need to be considered:

- *Proximity Principle*: the population is supposed to perform simple time and space computations.
- *Quality Principle*: agents of the swarm should be able to respond to different factors in the environment.
- *Diverse response Principle*: population should not take paths that are too narrow
- *Stability Principle*: it refers to the capability of the population to not change its behavior because the environment changes.
- *Adaptability Principle*: the swarm's behavior should change when it is needed.

To recap, in the past decade scientists (such as biologists and so on) have started to pay always more attention on natural swarm systems because of their stunning efficiency and then in the late-80s introduced the insights coming from those natural systems to the field of AI[17]. As a consequence, Swarm Intelligence has become progressively an important research field since then for computer scientists, engineers as well as bioinformatics and economists. According to [13] the main reason lies on the important role played by those problems that the natural intelligent swarms can solve in engineering areas of the real world. Although a lot of efforts have been made in the past years to develop various models for SI like ant colony optimization (ACO) or Artificial Bee Colony, for the purpose of this paper two models, Particle Swarm Optimizaton (PSO) and Bacterial Foraging Optimization (BFO), will be considered.

2.3 Particle Swarm Optimization

Particle Swarm Optimization, or PSO, is a population-based optimization technique created by J. Kennedy and R. Eberhart in 1995. Back then their aim, as suggested by Poli [18], was to develop a computational intelligence inspired by the social behavior of bird flocks searching for food. In [19] Kennedy and Eberhart highlighted as the PSO algorithm started from simulations of a simplified social model in order to describe the apparently unpredictable movements of bird flocks, but they shortly evolved to an optimization technique. This optimization method can be formally described as a "population" of particles floating around the hyperspace of the problem at different individual stochastic velocities subject to their historical best position and neighborhood best position. However, individual and collective best positions are derived from a problem specific fitness function and the trajectories of the swarm of particles move towards optimal or near-optimal solution [16]. Besides the description, it is fundamental to grasp the pillars which PSO is built on, that can be basically summarize as: social concept, Swarm Intelligence (SI) and the computational characteristics. The first pillar, as stated by J. de Valle et al. (2008) is related to ability of the individuals to interact and mimic in order to learn from others' experience as well as to adapt to the surrounding environment. Furthermore, social concept refers directly also to the phenomenon that makes people to get similar each other thanks to the mutual social learning (known as culture) and the changes that allow agents to shift to more adaptive patterns of behavior. The second pillar, has been thoroughly described in the previous section, but just to recall it can be described as the collective behavior that emerges from a group of social insects. Finally, the third pillar is related to the link or even better the evolution and incarnation of cellular automata[14] (CA) that is a computational system thanks to the property of computing function and solving algorithmic problems [20].

[14] Cellular automata (CA) can be described as discrete computational systems deployed in representing general models of complexity and more specific non-linear dynamics. Basically, the system can be discrete or abstract. The first case consists of a finite set of simple cells or atoms evolving in parallel powered by state update functions or dynamical transition rules taking into account their neighbor cells' states. Whereas in the second case, cells are merely specified in mathematical terms.

2.3.1 PSO Functioning

In this section the functioning of the Particle Swarm Optimization technique is showed in detail as well as the description of the algorithm which the optimization method has been coded in. Technically speaking, the PSO consists of a swarm of particles, each representing a solution and valuating the objective function at their current location in problem's search space. The magnitude of the particles' movement is determined by the history of their current and best locations as well as the ones of the other particles and some disturbance, whereas the later iterations once all particles moved.

The particles are specified by three vectors and each one is a D-dimensional vector[15].They can be represented as follow:

- $\vec{x_i}$ represents the vector of the current positions and it can be thought as the coordinates of the solution of the search space. Its elements $x_i^k \in \mathbb{R}^n$ indicates the position of the *i - th* particle at step *k.*
- $\vec{p_i}$ represents the vector which stores the previous personal best positions p_i, or $pbest_i$,of the particles.
- $\vec{v_i}$ represents the vector of the current velocity. The element $v_i^k \in R^n$ indicates the current velocity of the *i – th* particles at step *k.*

In addition, p_g represents the global best position, or *gbest,* achieved until that moment by the whole swarm [18]. Before showing how the PSO algorithm works it is better to clarify what $pbest_i$ and $gbest_i$ actually are. As suggested by Clerc and Kennedy [21] the fitness function gives back values by using the coordinates of the particles as input. In other words, those values are $pbest_i = f(p_i)$ and $gbest_i = f(p_g)$ where *f* is the fitness function[16]. The implementation of the PSO algorithm assures the storing of the current position in proper vector $\vec{p_i}$, if and only if the current position is better than anyone found so far. Therefore, the goal can be summarized in searching for better solutions in order to bring up to date the individual best position vector $\vec{p_i}$. The mechanism that allows the updating process can be represented as

[15] Where D represents the search space's dimension. Furthermore, this paper considers real number space.

[16] In this paper the terms fitness function and objective function are interchangeably used even though actually a tiny little difference between them exists. Objective function is the function being optimized whereas the fitness function is the function used to guide the optimization process.

follows:

$$p_i^{k+1} = \begin{cases} p_i^k & \text{if } f(x_i^{k+1}) \geq f(p_i^k) \\ x_i^{k+1} & \text{if } f(x_i^{k+1}) < f(p_i^k) \end{cases} \tag{2.1}$$

As mentioned above, the objective function *f* uses the particle's coordinates as input, in this case the coordinates of the *i – th* particle at step *k*. Being consistent with the maximization problem, as big as possible values of the function are preferred.

According to [18] the structure of the original PSO algorithm can be explained as follows:

1. Initialize a population of particles randomly setting their positions and velocities within the D-dimensional search space.
2. Iterate:
3. Compute the value related to the fitness function for each particle
4. Compare the value obtained in the previous step with the $pbest_i$ for each particle. Store the current position x_i^k as $pbest_i$ into the vector $\overrightarrow{p_i}$ if and only if it is better than the previous personal best.
5. Select the global best positions (*gbest*) reached so far by the population
6. Use the following stochastic system of equation to update the individual position and velocity of each particle:

$$\begin{cases} v_i^{k+1} = v_i^k + U(0,\phi_1) \otimes \left(p_i^k - x_i^k\right) + U(0,\phi_2) \otimes (p_g - x_i^k) \\ x_i^{k+1} = x_i^k + v_i^{k+1} \end{cases} \tag{2.2}$$

Where:

- v_i^k and x_i^k are the current position and current velocity of the *i – th* particle at step *k*
- v_i^{k+1} and x_i^{k+1} are the current velocity and position of the *i – th* at step *k+1*
- p_g is a variable storing the global best position of the population.
- $U(0,\phi_i)$ is a vector of random numbers which follow a uniform distribution in $[0,\phi_i]$. For each iteration the generation process is carried out.
- $\otimes$ is the component-wise multiplication.

7. Once the number of iterations or other stop criteria are reached, end the iteration.
8. End iteration

The first of the three components of the velocity update equation is called “inertia” or “momentum”. This term literally indicates the inertia of the particles to keep going in the same direction of the previous step. The second component, $U(0,\phi_1)\otimes\left(p_i^k-x_i^k\right)$, is called “memory” component and represents the tendency of the particle to attracted towards the best position found by itself so far. However, the memory is scaled by the random number $U(0,\phi_1)$. The third component $U(0,\phi_2)\otimes(p_g-x_i^k)$ represents a sort of attraction towards the best position achieved so far by the entire population. Again, it is scaled by random number that derives from this uniform distribution $U(0,\phi_2)$. This third and last component is called “cooperation” or “social knowledge”. Recalling what has been mentioned at the very beginning of this chapter, the version proposed in this section is a basic PSO version. Indeed, only a minimum amount of parameters requires to be set up.

2.3.2 Parameters of the Algorithm

When it comes to implement the algorithm, two critical points require enough attention, that is the prevention of the explosion of the swarm and facilitation of the convergence of the particles. The former occurs when the velocities of the particle tend to infinity and as consequence the swarm dissolves. Whereas the later refers to the tendency of the swarm to converge towards the optimal solution in the search space. In order to limit these inconveniences, the set up of some parameters such as maximum velocity, acceleration constants and others has to be taken carefully into account. We are going to analyze in greater details the parameters in the following sections.

Acceleration parameters

The parameters characterizing the uniform distributions in (2.2), ϕ_1 and ϕ_2, are called acceleration parameters. As stated by Poli et al. [18], they are responsible to control the direction of the particles. Notably, ϕ_1 is responsible to control the move of the particles to their personal best, p_i^k, whereas the control on the particle towards the global best position p_g, is governed by ϕ_2. However, the choice of the values has to be taken carefully in order to prevent abnormal particles' behavior. In particular, small values of the parameters could hamper the movements of the particles. On the contrary, large values could make particles diverge. Thus, it seems clear that the choice of these parameters affects the behavior of the PSO itself. In fact, as showed in [16], it has been empirically obtained the different type of trajectories according to the acceleration parameter's values chosen. If the value of the variable ϕ, which can be expressed as $\phi = \phi_1 + \phi_2$ is set greater than 4 the trajectory diverges. On the contrary a value of $\phi = \phi_1 + \phi_2$ could represent an acceptable starting point. However, to different values of ϕ ranging within the interval [2,4] correspond different performances of the particles. Being more specific, small values of ϕ give a sinusoidal trajectory whereas approaching the higher bound, the trajectory looks like more a complex interwoven cyclic path. To ensure a bumpy and random move of the particles looking for the optimal solution, it is suggested to randomly choose values within the range. In this way a path will be pursued by particles and they will be provided with the possibility to be pulled back only after many iterations overcoming the issues that small values of acceleration parameters may limit the movements. However, as it will be explained in the next sections, in order to solve the opposite problem and avoiding too high values, V_{max} parameter can be fixed.

Maximum velocity

The velocity at which the particles move around the problem's search space is stochastically determined. Due to this feature, the trajectory might result uncontrolled and too swinging. To overcome this issue, the solution originally proposed was to damp the velocity by bounding each component of v_i. According to [16] a way to show how the criterion works could be the following one:

If $v_i^k > V_{max}$ then $v_i^k = V_{max}$

else if $v_i^k < -V_{max}$ then $v_i^k = -V_{max}$

However, the choice of $[+V_{max}, -V_{max}]$ itself does not solve the problem rather it may create even a bigger one. In many practical cases, the maximum velocity boundaries are chosen arbitrarily according to the kind of problem. However, this procedure is not the best one because it may be subject to mistakes such us misspecifications of the right values of the bounds. Therefore, as said previously, if the parameters values are too high the trajectory could diverge, and it may go beyond the optimal or good solution, whereas if the parameters values chosen is too small the move of the particle could be limited. In order to effectively tackle this parameter selection problem, a feasible solution could be represented by a dynamically changing V_{max}. The solution can be represented as follow:

$$V_{max} = \frac{x_{max} - x_{min}}{N} \tag{2.3}$$

where x_{max} and x_{min} represent the maximum and the minimum values achieved by the particles till that step and N is the number of intervals in the *k-th* dimension selected by the user. With this approach, it is possible to manage the problem related to the right choice of the parameters as well as uniform the velocity throughout all the dimensions.

2.3.3 PSO Variations

PSO represents a valuable optimization technique which is able to provide good or even near-optimal solutions saving a considerable amount of time. However, beyond the advantages this method brings, it suffers some drawbacks as well. Apart from the issues due to the wrong selection of the acceleration parameters and maximum velocity, there are other drawbacks that may affect the efficiency of the PSO algorithm. One of the most critical limits is the premature convergence of the particles towards a single point in the search space. Although convergence is an eligible property, premature convergence could not be enough to explore the hyperspace. It is easy to understand that in this case the PSO algorithm fails at finding the global optimum [22]. In order to adequately solve these problems, some variations of the velocity update equation are considered.

Inertia Weight

This modification of the original PSO was proposed bey Shi and Eberhart in [23].

Their objective was to diminish the importance of maximum velocity parameter V_{max} but even more they want to better control the scope of the search by balancing the two components of the metaheuristic: exploitation and diversification. The modification that the two authors introduced consists of a new parameter in the velocity update equation as follow:

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

where the term ω is the “inertia weight”. As mentioned above, the scope of this parameter is to balance and control the exploration and exploitation components of the metaheuristics. In other words, the ability of the swarm to explore the search space as well as to focus and exploit a good solution found so far. What has been empirically found is that a high value of ω such us 0.9, allows particle to perform a thorough exploration, as if particles were moving in a low viscosity medium[17]. Reducing the value till 0.4 the swarm is no longer focused on the exploration but rather on the exploitation by focusing on the local optima. However, a value of the inertia weight greater than 1 makes the swarm considerably unstable, with the particles not coming back to local optima and diverging [18]. On the other hand, a value of ω equal to 0 makes particles moving without any memory of where they have been the step before. Indeed, as pointed out by Shi and Eberhart in [23], a value of inertia weight equal to 0 eliminates the first component in (2.2), thus the particles move around the same position and the search area shrinks. Notably, the PSO is very likely to find the global optimum if it is included in the initial search area. In fact, as said in [23]: “Therefore, it can be imagined that the search process for PSO without the first part is a process where the search space statistically shrinks through the generations. It resembles a local search algorithm”. Furthermore, the PSO algorithm has been found to better perform for changing values of inertia weight rather than fixed values of it. The reason is that varying values allows to control the local and global search. Typically, in order to achieve this goal a linearly decreasing inertia weight has been used. In this way it is possible to firstly let the particle explore the problem hyperspace and then to narrow the research preferring the exploitation mode. The value of a linearly decreasing inertia weight can be defined as follow:

[17] Low viscosity medium means a medium which has low resistance to gradual deformation.

$$\omega = \omega_{max} - \frac{\omega_{max}-\omega_{min}}{K} * k \qquad (2.4)$$

where:

- ω_{max} and ω_{min} represents the desired maximum and minimum values of the inertia weight;
- K represents maximum number of iterations;
- $k \in \{1, \dots, K\}$ represents the iteration number.

Although this approach guarantees fast convergence of the swarm it also has some drawbacks. Among the disadvantages the most serious one is that the change of mode between exploratory and exploitative, is irreversible and it is carried out the swarm loses the capability to search for new areas again. However, the value of inertia weight can be adjusted even adopting alternative approaches with respect to time-decreasing one such as a fuzzy system.

Constriction coefficients

This variation was introduced by Clerc and Kennedy in [21]. In this paper the authors showed how the optimization power of PSO can be improved as well as the control upon the explosion of the swarm by adding a coefficient called constriction coefficient X. Differently from the inertia weight, the constriction coefficient affects all the elements of the update velocity equation. Indeed, it is possible to show its influence on the particles' velocity, the equation (2.2) is modified as follow:

$$\begin{cases} v_i^{k+1} = X({}_i^k + U(0,\phi_1) \otimes \left(p_i^k - x_i^k\right) + U(0,\phi_2) \otimes (p_g - x_i^k)) \\ x_i^{k+1} = x_i^k + v_i^{k+1} \end{cases} \qquad (2.5)$$

where the coefficient X can be represented as follows:

$$X = \frac{2}{\left|2-\phi-\sqrt{\phi^2-4\phi}\right|}$$

in which:

$$\phi_1 + \phi_2 = \phi > 4.0$$

An important difference that arises comparing the original equation with the constriction parameters one, is that the value of the acceleration parameters can be set up greater than 4. When this variation of the PSO is used the value of ϕ is set to 4.1 and in this way the constriction parameter X is 0.7298. As consequence, the cognitive component $(p_i^k - x_i^k)$ and the social component $(p_g - x_i^k)$ are now multiplied by 1.49618 that is 0.7298 x 2.05. However, it is important to note that when the inertia weight ω is equal to 0.7298 as set by Clerc and Kennedy and ϕ equal to 1.49618, PSO with constriction coefficient and PSO with inertia weight are algebraically equivalent. Ultimately, this variation brings some advantages and disadvantages. The advantage is the improved convergence of the particles due to the damping effect upon the oscillations once the particles are within the optimal region. On the other hand, the main disadvantages are the wider cycles particles could do and the failure at converging when individual best performance and global best performance are in two different regions.

Fully Informed Particle Swarm.

In the original version, the particles of the PSO were supposed to have only two sources of information, that is personal best and global best. Although this traditional algorithm has been widely accepted over time, it presents a critical limitation: in this way particles are not allowed to exploit other sources of information such as the rest of the neighborhoods. A feasible solution was introduced by Mendes. In fact, its framework consists of particles which are affected by their neighbors. This variation of the PSO is called Fully Informed Particle Swarm. The velocity and position update equations are now the following ones:

$$\begin{cases} v_i^{t+1} = X(v_i^t + \frac{1}{K_i} \sum_{n=1}^{K_i} U^t \otimes (p_{nbr_k^i} - x_i^t) \\ x_i^{t+1} = x_i^t + v_i^{t+1} \end{cases}$$

where:

- K_i is the number of neighbors of the *i-th* particle;
- nbr_k^i is the *k-th* neighbor of the *i-th* particle.

The main drawback of this approach is high dependency of the PSO from the population topology, in other words the underlying framework governing the connections between particles. On the contrary this solution allows to save iterations respect to the original version of the PSO algorithm.

2.3.4 Population Topology

Exchanging of information is fundamental in PSO. The particles' purpose is to find the global optimum within the optimization domain and in order to do it they adjust their velocity and direction according to the information of the positions found by others. The fact that performance of particles is affected by particle's neighborhood is what allows common path of the swarm to arise [24]. However, the dimension of the neighborhoods implies the speed of the swarm to converge. Small neighborhoods are associated with slower convergence because of less exchange of information between particles. On the contrary, large neighborhoods speed up convergence because of high interaction between particles. As we introduced before, there are two types of neighborhoods:

- *global best (gbest):* the best solution found so far by the entire swarm if available to all the members of the swarm itself. This means that the network is fully connected;
- *local best (lbest):* the information is available only among immediate neighbors. Of course, the swarm topology plays a crucial role.

The two different population topologies can be appreciated in figure 2.1

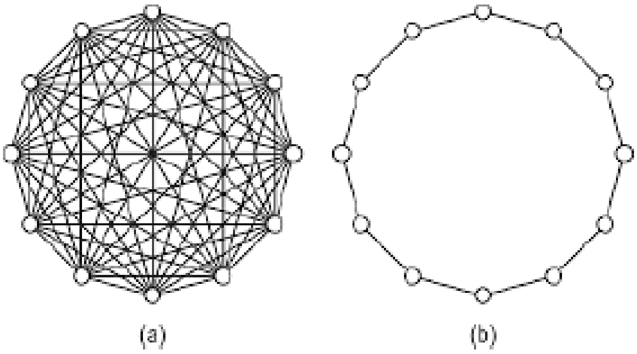

Figure 2.1 (*a*) Global best (*b*) Local best

As we said just above, there are some types of topology. We are going to illustrate some of them:

- *Star topology*: the network whereby the information flows, consists of a central particle which communicates to all the other particles. In this way the information of the central particle is shared with all the other particles. In other words, the central particle compares the best position found by all the other particles and consequently adjusts its position towards the best performance found. After that, the new position is shared with all the particles
- *Ring topology*: a ring is formed by the particles that are arranged in a circle. Each particle is connected with other two others in its neighbors. It is easy to understand that through this topology the information is exchanged among immediate neighbors till all the circle is completed and the information reaches the last particle. The main disadvantage is the slow convergence of the swarm due to the slow flow of information around the ring. However, the main advantage is that large parts of the search space are covered efficiently.

The choice among these two topologies is a trade-off between a slower convergence in the case of *lbest*, and a fast convergence but risky because of the possibility of being trapped in a local optimum in the case of *gbest*. However, there are other types of topologies beyond the ones presented in this paper, such as the Von Neumann topology, where particles are organized in a rectangular structure, which could outperform the *gbest*, as suggested by Kennedy and Mendes. A graphical representation of the topology is provided by figure 2.2. However, the selection of the most efficient neighborhoods has to be taken according to the specific problem to be solved.

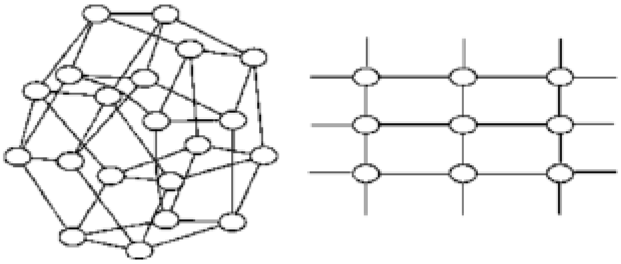

Figure 2.2: Von Neumann population topology

2.4 Bacterial Foraging Optimization

Bacterial Foraging Optimization is part of the newest swarm optimization methods for continuous optimization problems. The technique was presented by Passino in [25]. Since then it has increasingly drawn attention of experts and thanks to its high performance as optimizer and advantages. Furthermore, this optimization technique was successively used in many fields since its introduction such as control engineering, electric load forecast and many others. Although it seems to be a very powerful methods given the reduction in convergence time and the considerably high accuracy, the complete potential of the BFO remains unexplored [25]. The main idea behind this swarm-based algorithm is the mimic of the foraging strategy of a swarm of *Escherichia Coli* (*E. coli*) bacteria in a multi-objective function. When searching for nutrients, bacteria pursue the maximization of the energy obtained per unit time[18], as well as communicate information among each other through different signals. These ways to behave represent the two factors which the bacteria foraging decision process is based on. The movements the bacteria carry out after having taken the decisions, called chemotaxis, is precisely what BFO or Bacterial Foraging Optimization tries to replicate [26]. However, beyond the advantages listed above, this population-based technique presents some limits as well. In particular, drawbacks are related to the meager convergence tendency of the bacteria as well as the poor performance once the dimension of the search space increases when BFO is used in complex optimization problems [27]. This is the reason why researchers put effort to hybridize the original version of the algorithm, trying to overcome the limits. Among those variants introduced so far, one in particular seems to be working and it has been deeply studied in the past decades, that is the Cooperative search. The main feature of this approach is the increased number of search modules[19]. In this way a higher efficiency when searching for better solutions is ensured.

In order to have a better understanding of the functioning of BFO it may be fruitful to have a biological and physical background of *E. coli*. According to [28] *E. coli*

[18] The maximization of the energy intake per unit of time is the principal assumption of the foraging theory. Therefore, the function needed to be maximized is the following one: $\frac{E}{T}$. The importance of this concept could be clearer if the maximization of the above function is thought as a nutrient source that makes species to survive and allow animals to save time that can be spent in other crucial activities such as fighting, fleeing, mating and so on.

[19] More than one.

bacterium is made for 70% of its weight by water and consists of a plasma membrane, cell wall and capsule with cytoplasm and nucleoid inside. Furthermore, they occasionally engage in sex, whereas they synthesize and replicate in about 20minutes. However, in order to be able to move towards food sources and escape from noxious substances they have control system or guidance system that allows them to do it. The bacterium's movements throughout their entire life can be divided into two types:

- swimming: this movement is achieved by flagella[20]' counterclockwise rotations. Those rotations create a composite propeller that pushes the bacterium in one direction;
- tumbling: this movement derives from clockwise rotations. During this process, called "tumble interval", each flagellum rotates independently, thus the bacterium after a run almost immediately slows down due to the absence of inertia.

A crucial concept of the bacterial motile behavior is the motion patterns generated by bacteria, also called chemotaxes. The genesis of the chemotaxes is the decision of the bacterium about how long to run and how it responds to different conditions. Starting from a neutral condition such as the one represented by a substance without food or injurious substances the bacterium alternates tumbling and swimming modes following random directions. At this stage the bacterium is searching for foods. If the microorganism meets a nutrient gradient[21], he immediately changes the motile behavior increasing the time spent swimming and decreasing the time spent tumbling as long as he runs towards the concentration gradient. An important feature of *E. coli* is that, if it finds a region with constant concentration of food after a reasonable amount of time he adjusts its motile behavior as if he was in a neutral substance. In other words, he comes back to its original search behavior. This is a crucial biological property, because it always wants to find more food. Talking about the decision-making process more technically, it may result useful to talk about the underlying sensing and the balance of it. The receptor proteins of bacteria are very sensitive and provide them with "high gain" with a small attractant detection threshold. This means that the bacterium reacts very quickly to a minimum amount of nutrient. On the

[20] Flagella are lash-like appendages that protrudes form a bacteria cells.
[21] The term gradient is used to explain the gradual change of nutrient concentration.

contrary, the threshold for encountering neutral medium after being in a nutrient-rich is larger. However, the decision-making system has a sort of memory as well. Indeed, the bacterium while exploring the region, is able to remember the concentration one step before and compare it with the current region and considering the difference takes the decision. Beyond the changes of the motile behavior due to different levels of nutrients, there are also changes of the environment. The changes of the environment conditions dramatically affect the bacteria swarm by killing part of the swarm or split it into smaller groups and dispersed into new part of the environment. Those events are call elimination and dispersal events. Furthermore, those events could either jeopardized or improve the chemotaxis, because they may place bacteria in a region with more sources of food and higher levels of nutrients or in a very poor region. However, E. coli bacterium is the inspiration for BFO that also an evolutionary algorithm, but predominantly it is a population-based method, that is the swarm intelligence plays a crucial role. That is why it is important to introduce the group behavior. Especially in a semisolid nutrient medium, bacteria will shift from the center drawing rings selecting the gradient created by consumption of the nutrient by the group. In particular, when cells release the attractant chemical component[22] after having metabolized high levels of nutrient, bacteria assemble as groups moving following concentric patterns. The attractant chemical substance is used as a chemical signal that allows the swarm moves together.

In the following sections the Bacterial Foraging Optimization technique will be presented as well as the original version of the algorithm. Furthermore, the main steps characterizing the behavior of the bacteria swarm and the derivation of the main formula will be presented. Finally, a thorough comparison between PSO and BFO will be carried out.

2.4.1 Original version of BFO

In this section the functioning of Bacterial Foraging Optimization Algorithm will be explained. Basically, the main goal it is to provide a mathematical representation of the foraging strategy of the bacteria that represents the biological foundation of this optimization technique. To ensure a detailed and thorough description, we decided to

[22] Passino (2002) uses succinate as nutrient and aspartate as scrap.

us work made by Jun Li et al. in [29] as guideline.
The strategy whereby the bacteria search for food consists of four parts:

- chemotaxis;
- reproduction;
- elimination and dispersal;
- swarming.

Below we are going to provide a detailed analysis of each of these elements.

Chemotaxis

As described in the previous section, chemotactic component represents exactly the motion of the bacterium that consists of swimming and tumbling. Briefly recalling, swimming means a unit walk, in the same direction as the last step whereas tumbling means a unit walk, in a random direction. As such, favorable conditions like nutrient rich environments entail bacteria to swim in the same direction. On the contrary, unfavorable conditions like noxious rich environments implicate bacteria to change direction by tumbling. In order to translate the biological chemotaxis into computational chemotaxis, we firstly specify some parameters. Therefore, considering S as the bacteria population dimension and the position of the *i–th* bacterium as a candidate solution of the problem represented by $\theta^i(j,k,l) \in \mathrm{R}^p$, thus the bacterium motion can be expressed as follow:

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(i) \qquad (2.6)$$

where:

- $\theta^i(j,k,l)$ represents the *j-th* chemotactic *k-th* reproductive and *l-th* dispersal step of the *i-th* bacterium;
- $C(i)$ represents the length of the tumble step;
- $\phi(i)$ represents the random direction of the tumble step.

However, following [27], the direction of the tumble movement $\phi(i)$, as represented in (2.6),can be written in the extended form as follow:

$$\phi(i) = \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

where:

- $\Delta(i)$ represents the direction vector of the direction of the *j-th* chemotactic step[23].

In other words, the equation (2.6) can be represented in the following way as well:

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{2.7}$$

In the following section we are going to present the bacteria reproduction.

Reproduction

In this section we are going to describe the reproduction step. It is easy to understand that this step is the reason why BFOA is classified also as evolutionary algorithm, that is a branch of population-based algorithms. Indeed, through reproduction step this metaheuristic tries to replicate mechanisms belonging to the biological evolution like replication and selection of the species. Technically speaking at the end of each chemotactic step, the health status of each bacterium is evaluated through the following formula:

$$J^i_{health} = \sum_{j-1}^{N_c} P^{i,j,k,l} \tag{2.8}$$

where:

- N_c represents the total number of steps which the motile phase, or chemotactic process, consists in;
- $P^{i,j,k,l}$ represents the single value of the cost function of the *i-th* bacterium at the *j-th* chemotactic, *k-th* reproduction and *l-th* dispersal step.

That is nothing but the corresponding sum of cost functions of the *i-th* bacterium. However, according to the value of the health status, bacteria are ordered and only half of them are eligible to survive and access the reproduction phase. Each surviving bacterium splits up in to two new entities that are placed at the same positions. Whereby this procedure, the population dimension remains constant and more bacteria are placed at the best positions in the problem hyperspace. In other words,

[23] $\Delta(i)$ is equal to the last chemotactic step only when the bacterium is swimming, otherwise it is a random number within [-1,1].

this step allows to refine the search and speed up the convergence.

Elimination and Dispersion step

As mentioned in the general description of BFO, in this process part of the bacteria swarm is killed or moved to another location. Elimination and dispersion steps remedy the critical limit of the metaheuristic, that is to get stuck around local optima without finding global optima or better solutions. The main reason why bacteria may be trapped in poor solutions is simply that chemotaxis and reproduction are not sufficient to allow microorganisms to explore the search space. However, each bacterium is not always dispersed and eliminated. In fact, each bacterium has a probability equal to P_{ed} of being part of this process and once the bacterium is eliminated another one is dispersed in the environment. Before going on with description it may be useful to underline the advantages and disadvantages related to the values P_{ed} can assume. A high probability implies bacterium has more chances to move away from local optima. On the other hand, greater probability of escaping given to bacterium dramatically reduce the convergence speed, creating an "escape" issue. It is clear that a trade-off referred to the right choice of this parameter arises. In order to solve escaping-problem it is sufficient not selecting the single bacterium close to or at the global optimum which would be dispersed otherwise. In this way it is possible to optimize the single bacterium avoiding him to uselessly move away. In order to implement this procedure, the elimination-dispersal operator selects bacteria, with a probability of P_{ed} only among ones who have a fitness value lower than certain threshold. In other words, poor candidate solutions will be dispersed and eliminated, whereas the good ones will not. In addition, by increasing the generations and by gradually decreasing the dispersal and elimination steps, it is possible to further speed up the convergence and improve the efficiency of the metaheuristic.

Swarming

Being a particular type of swarm intelligence, BFO reproduces the social behavior of bacteria who inspire the metaheuristic. Just to make thing clearer, we recall that bacteria, while moving, release attractant signals which are captured by other population components and all the swarm moves in that direction. However, if bacteria release repellent counterparts move away. Thus, the optimization technique replicates this behavior through the following equation:

$$J_{cc}\left(\theta, P(j,k,l)\right) = \sum_{i=1}^{S} J_{cc}(\theta, \theta^{i}(j,k,l) =$$
$$\sum_{i=1}^{S}\left[-d_{attractant}\exp\left(-\omega_{attractant}\sum_{m=1}^{D}\left(\theta_m - \theta_m^i\right)^2\right)\right] +$$
$$\sum_{i=1}^{S}\left[-h_{repellant}\exp\left(-\omega_{repellant}\sum_{m=1}^{D}\left(\theta_m - \theta_m^i\right)^2\right)\right]$$

(2.9)

where:

- $J_{cc}(\theta, \theta^i)$ represents the value of the corresponding cost or fitness function;
- $d_{attractant}$, $\omega_{attractant}$, $\omega_{repellant}$ and $h_{repellant}$ represent the coefficients of the function that need to be chosen;
- S and D represent respectively the total number of bacteria and the total number of parameters to be optimized.

After the description of each fundamental element which characterizes the functioning of the metaheuristic, we are going to present the original algorithm introduced by Passino in [25].

2.4.2 Bacterial Foraging Optimization Algorithm

Up to know, BFO metaheuristic has been thoroughly described in every part, taking into consideration the mechanism whereby it works. Now we are going to present the algorithm[24] through which the BFO can be implemented. However, it has to be clarified that the algorithm presented in this section refers to the original version introduced by Passino in 2002.

In order to the initialize the algorithm, we have to choose the following parameters:

- *p* represents the dimension of the real number set *R*;
- *S* represents the population dimension;
- N_c represents the number of chemotactic steps;

[24] Algorithm is an unambiguous specification that allows to solve a class of problems.

- N_s represents the maximum number of steps being taken in the same direction by the bacterium if the value of the cost function at the current position is lower than the one at the previous step[25];
- N_{re} represents the maximum number of reproduction steps;
- N_{ed} represents the number of elimination-dispersal events:
- P_{ed} represents the probability that each bacterium will be subject of elimination-dispersal event;
- $C(i)$ represents the thumble step size.

In addition, if the algorithm includes the swarming, there will be necessity to set up the following parameters as well: $d_{attractant}$, $\omega_{attractant}$, $\omega_{repellant}$ and $h_{repellant}$:

- $d_{attractant}$ represents the depth of the attractant released by the bacterium. In other words, a measure to quantifies the attractant;
- $\omega_{attractant}$ represents the measure to quantifies the diffusion rate of the chemical attractant signal;
- $h_{repellant}$ represents the magnitude of the noxious substance effect;
- $\omega_{repellant}$ represents the width of the repellant.

In addition, after setting all the parameters above we must select the starting position of each bacterium θ^i. Of course, it should be pretty much obvious that choosing the initial position in areas around optimum value would bring a clear advantage. However, the choice of the initial position is not easy to be taken when there is no knowledge where the optimum value really is. Therefore, bacteria are randomly distributed around the hyperspace problem.

According to [28] the algorithm is:

1. Elimination-dispersal loop: $l = l + 1$
2. Reproduction loop: $k = k + 1$
3. Chemotaxis loop: $j = j + 1$
    a. For $i = 1,2 \ldots, S$, take chemotactic step for bacterium *i* as follows.
    b. Calculate $J(i, j, k, l)$.

[25] This concept is related to the bacterium exploitation of the problem hyperspace. This means that if $J(i, j + 1, k, l) < J(i, j, k, l)$ then the bacterium will swim in the same direction for a length of $C(i)$.

c. Let $J_{last} = J(i,j,k,l)$ to save this value given that it is possible to find a better cost value in the next run.

d. Tumble: Generation of a random vector $\Delta(i) \in R^p$ where each element of the vector $\Delta_m(i), m = 1,2, \dots p$ corresponds to a random number within the interval [-1,1].

e. Move:

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

f. Compute $J(i,j+1,k,l)$ and then let

$$J(i,j+1,k,l) = J(i,j+1,k,l) + J_{cc}(\theta^i(j+1,k,l), P(j+1,k,l)$$

g. Swim[26]:

   i. Let m=0 (counter for swim length).

   ii. While $m < N_s$

      - Let $m = m + 1$.
      - If $J(i,j+1,k,l) < J_{last}$, let $J_{last} = J(i,j+1,k,l)$ and let $\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$ using $\theta^i(j+1,k,l)$ to compute new $J(i,j+1,k,l)$ as in f.
      - Let $m = N$. This is the end of the while statement.

h. If $i \neq S$ then go to the next bacterium $(i+1)$.

4. If $j < N_c$ go to step 3. If so, continue chemotaxis because bacterium keeps living.

5. Reproduction:

   a. For the given *k* and *l*, and every $i = 1,2, \dots, S$ let

   $J^i_{health} = \sum_{j+1}^{N_c+1} J(i,j,k,l)$

   Where J^i_{health} represents the health of the *i-th* bacterium. In order to sort bacteria, they need to be ascendingly ordered according to their health function value.

   b. The bacteria with the highest health value J^i_{health} , S_r bacteria die and the other S bacteria splits up.

[26] Where for simplicity only bacteria numbered {1,2,...,i} have swum, whereas bacteria numbered {i+1,i+2,...S} have not.

6. If $k < N_{re}$ then go to step 2. This means that the number of specified reproduction steps have been achieved, so the next generation in the chemotactic loop starts.
7. Elimination-dispersal: For $i = 1,2, \dots S$, with probability equals to P_{ed}, eliminate and disperse each bacterium. In order to keep the number constant, for each bacterium that is eliminated another one is dispersed in another location within the hyperspace problem.
8. If $l < N_{ed}$the go to step 1: otherwise end.

2.4.3 Algorithm Parameter Setting

In the previous section, all the parameters required to be specified have been listed and they have to be chosen really carefully. In fact, in this section we are going to explained main advantages and disadvantages in order to properly choose the right parameter values.

When it comes the first and basic parameter, *S* (population size), there is a trade-off between the magnitude of the population size. In fact, an increased population size brings a certain grade of computational complexity that could not be easy to overcome. On the contrary, larger size together with a random initial bacteria positions improves the effectiveness of the metaheuristic of the finding the global optimum by increasing the chances that some bacteria already start in optimum value regions. The tumble step size $C(i)$ is another parameter that needs to be carefully chosen. Too large values imply that bacteria could jump over local optima without sopping on them. Too small values could slow down the convergence and the risk of bacteria may be trapped in local optima arises. For those reasons, it is not suggested to select values of $C(i)$ that are biologically inspired. Talking about swarming, different values of attractant width, $\omega_{attractant}$, balances the individual and swarm activity. In particular, $\omega_{attractant}$ high and deep the cells are more likely to swarm. On the contrary, small width induces bacteria to search on their own instead of altogether. Moreover, the choice of N_c clearly affect the metaheuristic behavior. Large values of chemotactic steps N_c allow bacteria to exploit and explore the optimization domain but those large values increase the computational complexity as well. Small values of N_c could jeopardize the efficiency of BFO by causing a premature convergence towards local optima. The maximum number of steps in the same direction once the bacterium is in a position where the cost function is lower than the step before N_s, can be thought as a deviation of a random walk. The computational complexity could increase even due to large values of the maximum number of reproduction steps N_{re}. In particular, taking into consideration the reproduction procedure, this parameter affects the tendency of the algorithm to ignore regions with poor solutions and focus on regions where better solutions. However, if the value is too small the algorithm may suffer of premature convergence. Large values of elimination-dispersion parameter N_{ed} increase the ability of bacteria to search for good solution in more regions but, the main disadvantage is the higher computational complexity. Finally, the probability P_{ed} aids BFO to not be trapped into local optima but rather to focus on global optima. However, it should be pretty much clear that large values of this parameter may degrade the effectiveness of the algorithm overworking the random search.

2.5 Comparison between PSO and BFO

In this section a briefly comparison between the two metaheuristics will be presented. As we mentioned before, even though the PSO and BFO are somewhat young optimization techniques, they were not introduced at the same time. Starting from the oldest one, PSO was developed in 1995 by Kennedy and Ebehrart whereas the youngest one is BFO that was introduced by Passino in[25]. Even though the timespan between them is not so big, PSO has more studies underneath and more variants were successively developed and applied throughout years. On the contrary, BFO has a lot of unexpressed potential that needs to be found out. However, both PSO and BFO belong to the same family of population-based optimization techniques, in particular they are two swarm intelligence techniques.

Going more in depth, although BFO shows features of evolutionary computation techniques such as, reproduction, selection and elimination, the two metaheuristics presents many similarities.

The most evident common feature between them is the similarity between particles and bacteria. In fact, both represent candidate solutions to the optimization problem. Particles and bacteria move within the problem hyperspace looking for global optima solutions and while doing this they act like a swarm. The main difference in the social behavior is the way through which population elements exchange information in order to follow a common path. Whereas in BFO bacteria release attractant to signal other bacteria to swim in the same direction that can be computationally expressed through the combined cell-to-cell attraction and repelling effect in (2.9), in PSO a crucial role is played by the population topology. In the original version, a particle does not exchange information with all the particles swarm rather with its neighbors. Even the difference among the kinds of neighborhood, *gbest* or *lbest,* affects the swarm behavior. However, fully informed particle swarm variant tried to overcome those limits. Another important difference is the motile behavior between particles and bacteria. Particles move following a function in which the stochastic velocity is adjusted according to the distance between the best position personally found by the particle itself and the distance to the global position found by the entire swarm. On the other hand, bacteria move alternating swim and tumble, without stochastically adjusting the velocity. Bacterium keeps swimming in the same direction if the values of the cost function increasingly get better step by step, whereas he tumbles if the values of the cost function get worse. Another difference between PSO and BFO is played by elimination-dispersal operator. In fact, thanks to the probability P_{ed}, some bacteria move to another position increasing the chance to better explore the optimization domain. However, considering a huge number of iterations, particles are able to eventually go everywhere. Of course, this method could be time consuming.

After presenting the two metaheuristics we are going to analyze, in the next chapter, the measure of risk and the portfolio selection model.

Chapter 3

Realistic portfolio selection model and two-sided risk measure

Based on the wide range of coherent measures of risk, for the purpose of this work we decided to use the two-sided coherent risk measure introduced by Chan and Wang in [8]. As we mentioned in the first chapter, this measure of risk besides of being coherent, it takes into consideration positive and negative deviations from the mean returns and it is capable to replicate the asymmetry and fat tails of the distribution as well. Furthermore, this measure allows to reflect the investor's risk attitude. However, in this chapter after the first part where we analyze the measure of risk chosen, we are going to describe the portfolio model which that will be applied. For the purpose of this work the realistic portfolio selection model has been taken into consideration [30].

3.1 Breaking down two-sided coherent risk measure

As we mentioned in chapter 1, variance can not be considered as an appropriate measure of risk due to the non-normality distribution of the returns. Moreover, variance measures dispersion and assigns equal weights to potential losses and gains. Thus, this risk measure allows to effectively deal with downside and upside risk. In this section we are going to describe in greater detail respect to chapter 1 the two-sided coherent measure of risk.

Firstly, we set up the time period equal to one, this means that we assign 0 to the current date and *D* to the future date. Secondly, we specify the stochastic variable *Y* defined on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$, the asset payoff, as the risk that has to be measured. This means that positive values of *Y* represent a random profit for the investor, whereas negative values of *Y* indicate a random loss and the risk measure $\rho(Y)$ represents the minimum amount of extra cash added to *Y* that allows the position to be acceptable for the holder. In other words, when $\rho(Y) > 0$ the investor needs to add an amount of extra cash equal to $\rho(Y)$ in order to accept his future position. On the contrary, when $\rho(Y) < 0$ the investor is allowed to withdraw an amount of cash equal to $\rho(Y)$ without affecting the acceptance of his future position.

In order to represent this measure of risk, let's define $\|Y\|_p = (\,E[Y|^p|])^{1/p}$, where $E[\cdot]$ is the mean, or expected value of the random variable *Y,* with $p\ \in [1, +\infty)$ which represents the negative risk. Moreover, we define Y^- as the maximum (-*Y,*0) and $Y^+ = (-Y)^-$. Lastly, we define $\sigma_p^{\pm}(Y) = \|(Y - E[Y])^{\pm}\|_p$. The two sides of the return distribution that are taken into account by this risk measure are, the downside denoted by the random variable $(Y - E[Y])^-$ and the upside represented by the random variable $(Y - E[Y])^+$. However, given $p\ \in [1, +\infty)$ and $0\ \leq a \leq 1$ we can define a new risk measure, $\rho_{a,p}\colon L^p(Q) \longrightarrow \mathbb{R}$, as follows:

$$\rho_{a,p}(Y) = a\sigma_1^+(Y) + (1-a)\sigma_p^-(Y) - E[Y]$$

$$= a\|(Y - E[Y])^+\|_1 + (1-a)\|(Y - E[Y])^-\|_p - E[Y] \qquad (3.1)$$

where:

- a is global factor that balances the good volatility and bad volatility. By changing the value of the weight a, it is possible to have risk neutrality or risk preference. In particular, by selecting the following values, $p = 1$ and $a = 0.5$, we obtain risk neutrality. On the contrary, if we choose the value of a such that $0.5 < a \leq 1$ we obtain risk preference. We can conclude that, the smaller the value of the weight a, the bigger the weight on the lower $p - \mathrm{th}$ partial moment. In other words, an investor who is more cautious;
- p represents the global risk factor of the investment itself. Large values of p are associated with investors who are risk adverse. On the other hand, small values of p are associated with risk adverse investors. Moreover, by carefully selecting suitable values of this parameter is also possible consider the skewness and the kurtosis (leptokurtosis in this case) of the returns.

In order create the measure of risk $\rho_{a,p}(Y)$ we take the convex combination of $1$and $-\mathrm{norm}$, where the former is the positive deviation of the returns from the mean $(Y - E[Y])^+$ and the latter the negative deviation $(Y - E[Y])^-$. However, the inclusion of the expected value $E[Y]$ insures that the risk measure respects the coherence properties. A proof of the coherence is that the following theorems, as defined in [8], hold:

Theorem 3.1 For any $p : 1 \leq p \leq \infty$ and $a \in [0,1]$, the measure of risk $\rho_{a,p}$, as represented in (3.1) is a coherent risk measure.

Theorem 3.2 The risk measure $\rho_{a,p}$ is non-decreasing with respect to p and non-increasing with respect to a.

Theorem .3.2 is fundamental to explain how the investor reacts to risk. In fact, non-decreasing property with respect to p insures that, given a fixed value of a, higher values of p indicate that investor considers *Y* riskier. This means that higher values of p entail larger values of $\rho_{a,p}$, respect when the investor consider lower values of p. However, larger values of p reflects the propensity of the investor of being more carefully towards negative risk. The reason is that risk factor p is referred to the negative tail of the asset returns distribution, that is to the negative risk. On the other hand, non-increasing property of this measure of risk with respect to a, insures that, for fixed values of p, higher values of a are associated with a lower risk aversion of the investor.

Furthermore, the most remarkable difference between this measure of risk and one-sided risk measures is that, since $E((Y - E[Y])^+) = E((Y - E[Y])^-)$, if the investor minimizes $E((Y - E[Y])^+)$ he minimizes his dispersion from the mean. However, the main advantage of this measure of risk is that $\rho_{a,p}(Y)$ contributes in finding robust optimal portfolios, with respect to others one-sided risk measure such as CVaR.

3.2 Portfolio Selection Model

The portfolio selection model chosen for this paper is one proposed by Corazza, Fasano and Gusso [30]. However, making a valid and effective portfolio selection in the real-world stock markets means solving a NP-hard problem, due to some issues. The first issue is to choose a risk measure which satisfies the coherence properties and represents non-normal distribution of returns. In order to effectively solve this issue, we choose the two-sided coherent risk measure. Secondly, the model is required to take into consideration the rules applied to the portfolio management industry. Bounds on the number of stocks that can be traded is an example of such rules. This issue can be solved by introducing some appropriate constraints in the model. The last issue is that the use of risk measure together with the constraints used to replicate the rules used in the management industry, makes the portfolio selection problem a NP-hard problem. The solution to this problem is obtained by the combination of an exact penalty method and two metaheuristics taken into consideration this paper, PSO and BFO. The exact penalty method is used because it allows to transform nonlinear, non-differentiable and mixed-integer problem, into an equivalent minimization problem, in term of solutions, without any constraints. However, it has to be pointed that the problem still presents nonlinearity, non-differentiability and non-convexity but, the use of algorithms which do not include derivatives, such as PSO and BFO, allows to approximately compute a global minimizer of the exact penalty-based model.

Return and budget constraint

Return and budget constraint represent two of the most important constraints due to their essential role. Furthermore, they are present in many portfolio selection models. In order to present the return and budget constraint, we first start with some definitions. Let's consider the possibility to choose from N assets where for $i = 1, \dots, N,\ x_i \in \mathbb{R}$ is the weight of the $i - \text{th}$ portfolio asset, with $X^T = (x_i, \dots, x_N)$ and r_i which represents the real value of random variable where its expected return, $E[r_i]$, is represented by $\hat{r}_i$. If we move from the single asset to the whole portfolio it is possible define the return of the whole portfolio $R \in \mathbb{R}$ and the expected value $\hat{R}$ as:

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

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

We can now define the return constraint as:

$$\hat{R} \geq l, \quad \text{with } l > 0$$

Where l is the minimum expected return the investor wants the portfolio to earn. It is easy to understand that in this way it is possible to select the portfolio with the minimum risk among all the portfolios on the efficient frontier.

Budget constraint, which allows all the capital available to the investor to be invested, can be defined as:

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

Cardinality constraint

Among the advantages that cardinality constraint brings, it allows to indirectly control transaction cost by limiting the number of assets included in the portfolio together with the proportion of the portfolio held in a given asset. Furthermore, it allows to include not too small or too large subset of the available assets. As a consequence, cardinality constraint permits to indirectly control transaction costs by helping fund managers to solve the issue of selecting assets from hundreds of them.

In order to define the cardinality constraint, let $Z^T = (z_1, \dots, z_N) \in \{0,1\}^N$ be a binary vector, with z_i that can assume only values equal to 1 if the $i - \mathrm{th}$ asset is included in the portfolio and 0 if it does not.

Thus, the cardinality constraint can be defined as follows:

$$K_d \leq \sum_{i=1}^{N} z_i \leq K_u, \qquad \text{where } 1 \leq K_d \leq K_u \leq N \tag{3.2}$$

where:

- K_d represents the minimum number of assets in the portfolio;
- K_u represents the maximum number of assets in the portfolio.

Moreover, it is also required that the portfolio does not include too small or large fraction of an asset. In order to ensure this property, it is required to fix a minimum and maximum fractions, d_i and u_i, to allocate in each asset:

$$z_i d_i \leq x_i \leq z_i u_i, \text{ where } 0 \leq \mathrm{d} \leq \mathrm{u} \leq 1 \tag{3.3}$$

In order to ensure that cardinality constraint and the establishment of minimum and maximum fractions are compatible, parameters d and u must satisfy:

$$d \leq \frac{1}{K_d} \text{ and } u \geq \frac{1}{K_u}$$

After the presentation of the constraints which make the portfolio realistic, we are going to describe the portfolio selection model.

Portfolio Selection Model

The portfolio selection model, whose objective is the minimization of the two-sided risk measure $\rho_{a,p}$, consists of combining the constraints described before. Thus, the portfolio selection model chosen for this paper can be defined as follows:

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

$$\text{subject to: } \hat{R} \geq l$$

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

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

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

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

Penalty function method

As we stated above, the penalty method transforms the constrained optimization problem, into an equivalent optimization problem, in terms of solutions, without any constraints. In order to achieve this approximation, we add to the objective function a penalty which entails costs sufficiently high if the constraints are violated. The parameters entitled to quantify the severity of the penalty is ϵ.

However, penalty function has two drawbacks as follows:

- the accuracy whereby the unconstrained problem approximates the constrained problem. Basically, if the parameter ϵ tends to infinity, the solution of the unconstrained problem converges to the one of the constrained problem.
- the solution method when the unconstrained problem's objective function contains the penalty term. In fact, the larger the value of the parameter ϵ, the more precise the approximation is. Unfortunately, the structure of the

unconstrained problem becomes more unfavorable and the consequence is a longer time required to find a good solution.

Although the penalty method has some drawbacks it also has the important advantage of solving constrained problems without the need of refined algorithms. With the purpose of describing the penalty method introduced by Luenberger and Ye in [31] and how it works, let’s firstly define the following problem:

$\text{minimize } f(x)$

$\text{subject to } h_j(x) = 0,\ \ g_i(x) \leq 0$

where:

- The function f is a continuous function on $\mathbb{R}^n$;
- $h_j(x)$ with $j = 1, \dots, m$, represents a set of m constraints expressed in equality form;
- $g_i(x)$ with $i = 1, \dots, p$, represents a set of p constraints expressed in equality form.

Then, the penalty function method replaces the optimization problem described here above with an unconstrained problem as:

$\text{minimize } f(x) + \epsilon P(x)$

where:

- ϵ represents a positive constant;
- The function P is a continuous function on $\mathbb{R}^n$.

However, the continuous function P needs to satisfy the two following properties:

- $P(x) \geq 0\ \ \forall\, x \in \mathbb{R}^n$;
- $P(x) = 0\ \ \text{iff}\, x \in S$.

Whereas the absolute-value penalty function is expressed as:

$$P(x) = \sum_{j=1}^{m} \left| h_j(x) \right| + \sum_{i=1}^{p} \max(0, g_i(x))$$

For the purpose of this paper the penalty method chosen is the exact penalty method that guarantees a perfect correspondence between the solution of the original constrained problem and the derived unconstrained or penalized problem [30]. The Exact Penalty Theorem [31] permits this perfect correspondence:

Theorem 3.3 Let's suppose:

- the local minimum x^* satisfies the second-order sufficiency conditions of the constrained problem;
- λ and μ be the Lagrange multipliers;

Then for $\epsilon > \max|\lambda_i|, \mu_i : i = 1, \dots, m \; j = 1, \dots, p$, x^* is a local minimum of the absolute-value penalty objective.
This theorem, as we stated before, allows the perfect correspondence between constrained and unconstrained problems. Nonetheless, it does not give any insight about the value of the parameter ϵ.

Portfolio Selection Problem with Exact Penalty Method

In this section we are going to reformulate the portfolio selection problem as an unconstrained optimization problem by applying the exact penalty method. Given that penalty function is applied only when a constraint is violated, we are going to describe all the constraints. The constraints can be reformulated as follows:

- $\hat{R} \geq l \rightarrow \max\{0, l - \hat{R}\} = 0$
- $\sum_{i=1}^{N} x_i = 1 \rightarrow \left|\sum_{i=1}^{N} x_i - 1\right| = 0$
- $K_d \leq \sum_{i=1}^{N} z_i \rightarrow \max\{0, K_d - \sum_{i=1}^{N} z_i\} = 0$
- $\sum_{i=1}^{N} z_i \leq K_u \rightarrow \max\{0, \sum_{i=1}^{N} z_i - K_u\} = 0$
- $z_i d \leq x_i \rightarrow \sum_{i=1}^{N} \max\{0, z_i d - x_i\} = 0$
- $x_i \leq z_i u \; \rightarrow \sum_{i=1}^{N} \max\{0, x_i - z_i u\} = 0$
- $z_i \in \{0,1\} \rightarrow \sum_{i=1}^{N} |z_i(1 - z_i)| = 0$

The portfolio selection model with exact penalty function becomes:

$\text{minimize}_{X,Z} \, P(X, Z, ; \epsilon)$

where:

$$P(X, Z, ; \epsilon) = \rho_{a,p}(R) + \frac{1}{\epsilon}[\max\{0, l - \hat{R}\} + \left|\sum_{i=1}^{N} x_i - 1\right| + \max\left\{0, K_d - \sum_{i=1}^{N} z_i\right\} \\ + \max\left\{0, \sum_{i=1}^{N} z_i - K_u\right\} \\ + \sum_{i=1}^{N} \max\{0, z_i d - x_i\} + \sum_{i=1}^{N} \max\{0, x_i - z_i u\} + \sum_{i=1}^{N} |z_i(1 - z_i)|]$$

where ϵ is the penalty parameter.

However, it is possible to reformulate the measure of risk as well. Following this idea, for any portfolio $X = (x_1, \dots, x_N)^T$ the risk measure can be expressed as:

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

where:

- $r_{i,t}$ represents the return of the stock computed using its price time series: $r_{i,t} = \frac{p_{i,t+1} - p_{i,t}}{p_{i,t}}$
- $\hat{r}_i$ represents the expected value of the stock return $r_{i,t}$ and it is estimated by using the historical data: $\hat{r}_i = \frac{1}{T}\sum_{t=1}^{T} r_{i,t}$

After the transformation of the portfolio selection model into an unconstrained one, we are going to apply the two metaheuristics chosen: PSO and BFO.

Chapter 4

Application to the DAX 30 Stock Index

In the following chapter we will present the application and its result of the metaheuristics presented in the previous chapters. The solution procedure will be implemented on the german stock index DAX 30 and the period of time that has been taken into consideration is from April 2012 until November 2018

4.1 Problem setting

The daily closing prices of the of the assets included in the DAX 30, which is based on the stocks of the 30 companies with the highest capitalization liquidity and the number of stocks, have been taken into consideration for the analysis. The time length of the data is from April 2015 until November 2018. However, in this analysis only 26 of the 30 stocks populating the index have been considered. This because 4 of them had missing values in the request period.

The assets included in the stock are presented in Table 4.1.

Adidas	Fresenius
Allianz	Fresenius Medical Care
BASF	HeidelbergCement
Bayer	Henkel vz
Beiersdorf	Infineon
BMW	Linde
Continental	Merck
Covestro	Münchener Rückversicherungs-Gesellschaft AG
Daimler	RWE
Deutsche Bank	SAP
Deutsche Borse	Siemens
Deutsche Post	thyssenkrupp
Deutsche Telekom	Volkswagen (VW) vz
Lufthansa	Vonovia
EON	Wirecard

Table 4.1: DAX 30 stocks

The period of time of the analysis has been organized in 4 periods of 11 months each. However, each of those periods has been divided into 2 sub-periods: the in-sample period with a length of eight months and the out of sample period with a length of three months. The following scheme represents how the analysis period has been organized:

- Period 1: in-sample: April 2015 – November 2015
  out of sample: December 2015 – February 2016
- Period 2: in sample: March 2016 – October 2016
  out of sample: November 2016 – January 2017
- Period 3: in sample: February 2017 – September 2017
  out of sample: October 2017 – December 2017
- Period 4: in sample: January 2018 – August 2018
  out of sample: September 2018- November 2018

The two periods have different purposes. The in-sample period is used to estimate the parameter of the model. The out of sample, on the contrary, is used to test and verify the model. In other words, the out of sample period is the virtual future where the investment strategy is applied and it allows to evaluate the goodness of the model. The reason behind is that the risk and expected return of the portfolio in the first period is expected to approximately be the same even the second period and the percentage of investment suggested in the in-sample period should return the best portfolio also for the next future.

For the purpose of this analysis the daily returns have been obtained using the following formula:

$$r_{i,t} = \ln\left(\frac{p_{i,t+1}}{p_{i,t}}\right)$$

while, the expected value of the return has been computed using the following formula:

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

where $t = 1, \dots, T$ and T is the time horizon considered, and $i = 1, \dots, 26$ is the asset considered.

However, for this analysis we used MATLAB R2018 and we performed the computation on a Notebook ASUS equipped with a Intel Core i7, with 4gb ram.

Problem specific parameter setting

In this section we are going to presents the correct setting of parameters related to the constraints. However, in order to make comparable the results of both algorithms we decided to choose the same value of the parameters. The values chosen are the following:

- Minimum expected return desired from the portfolio: the mean return of the in-sample period;
- Minimum fraction allowed in each asset: $d = 2\%$;
- Maximum fraction in each asset: $u = 20\%$;

- Minimum number of holding assets:$K_d = 5$;
- Maximum number of holding assets, two values are equal to: $K_u = 10$ and $K_u = 20$;
- Global risk factor: $a = 0.5$;
- Local risk factor: $p = 2$.

PSO parameter setting

The parameter set up for this analysis are the ones suggested by the literature:

- Inertia weight $\omega = 0.7298$;
- Cognitive acceleration coefficient: $\phi_1 = 1.49681$;
- Social acceleration coefficient: $\phi_2 = 1.48961$;
- Number of particles: 52;
- Number of iterations: 1000.

Before starting the analysis, we performed some preliminary tests in order to properly choose the value of the penalty parameter ϵ.As mentioned in the previous chapter this parameter is fundamental because it guarantees the corrispondency between the unconstrained and constrained optimization problem. We decided, in order to be compliant with the best practice, to run this preliminary analysis only on the first period as indicative of the whole period. In order to identify the right value of ϵ we run the algorithm 10 times for 5 different values of the parameter of interest using 52 particles. The number of particles has been set equal to 52 since we have 52 variables: that is 26 asset variables, and 26 binary variables z which tells if a particular asset has been included or not in the portfolio. -For each value of the penalty parameter, we computed the average normalized fitness and the relative standard deviation, and the parameter associated to the lowest standard deviation value has been selected, as shown in table 4.2.

ϵ	Normalized Fitness	Standard Deviation
0.0001	0.897584175	0.2992167
0.00001	0.869491352	0.3027447
0.000001	0.827959206	0.2910785
0.0000001	0.867235283	0.303813
0.00000001	0.847425359	0.2993551

Table 4.2: Output for different values of ϵ

As shown in Table 4.2 we selected the penalty parameter equal to 0.000001 given that it presents the lowest standard deviation.

BFO parameter setting

Conversely to PSO, the literature does not offer enough insights about the BFO parameter setting. The unproper choices of the values could lead to insufficient numbers of iterations, making the algorithm not able to find high quality solutions or increasing to much the computation time and the BFO would work inefficiently. Therefore, in order to effectively solve these problems, we run many trials of the algorithm combining different values of the algorithm. The main goal of this test it was to find the right combination which allows the algorithm to find acceptable solutions in a reasonable amount of time. The parameters have been set as follows:

- number of bacteria: 52;
- number of chemiotaxis steps: $N_c = 400$;
- number of swimming steps: $N_s = 4$;
- number of reproduction steps: $N_{re} = 4$;
- number of elimination-dispersal steps: $N_{ed} = 4$;
- elimination-dispersal probability: $P_{ed} = 0.3$;
- size of the step taken by the tumble: $C(i) = 0.02;$
- size of the step taken in each run: $C(t) = 0.002$.

As we did with PSO we performed a preliminary analysis in order to choose the right value of the penalty parameter ϵ of the fitness function. In order to make this preliminary analysis comparable with the previous one, we run 10 times the algorithm for 5 different values of ϵ on the first period, using 52 particles. The reasoning behind the choice of the particle is the same of the PSO. The Table 4.3 shows the results of the analysis.

ϵ	Normalized Fitness	Standard Deviation
0.0001	0.466912643	0.4109212
0.00001	0.535473847	0.4113192
0.000001	0.659274961	0.4662657
0.0000001	0.577591981	0.4227383
0.00000001	0.594749056	0.429676

Table 4.3: Output for different values of ϵ

The results, conversely to PSO, suggest to select a value for ϵ equal to 0.0001. In fact, the third column which lists the values of the standard deviation of the normalized fitness for different values of the penalty parameters, associates the lowest standard deviation to the first element of the first column. However, for BFO we decided to skip the performance analysis that compares different numbers of bacteria and different number of iterations. Taking into consideration the way higher computation time than PSO, and increased number of bacteria or iterations lead to an inefficient BFO that would be time consuming. For this reason, we decided to keep the values as shown above.

4.2 Application and discussion

In this section we present the relevant results of the application of the two metaheuristics to the data. The application has been carried out in order to test and compare the ability of the two metaheuristics, PSO and BFO, to effectively find the global minimum solution of the optimization problem and the effectiveness of the portfolio obtained. However, in this analysis we considered different investors' taste for risk.

The analysis has been divided in two parts: in the first one we applied the PSO, while in the second one we applied the BFO.

The periods have been singularly analyzed for different values of the maximum holding asset K_u, respectively equal to 10 and 20, while the minimum holding asset K_d has been kept constant. Furthermore, we used different values of the local risk factor p, as mentioned above, in order to consider different risk preferences of the investor. The values considered in this analysis are: $p = 1, p = 2, p = 5$.The parameter p is referred to the negative variance. Therefore, small values of the local risk factor represent an investor who is risk seeker or in other words less careful about negative risk. On the contrary, high values of the local risk factor represent an investor who is risk averse, that is more worried about risk. The other parameter of the risk measure, the global risk factor a, has been kept constant to 0.5 for all the analysis. However, for each combination of K_u and p, we run the algorithm 10 times and we took the average. The reason is because metaheuristics are stochastic techniques and not exact optimization methods.

Period 1	p = 1	p = 2	p = 5
$\rho_{0.5,p}; K_u = 10$	0.002224	0.008191	0.007539
$N_{a,p}(K_u)$	**8**	**8**	**8**
$\rho_{0.5,p}; K_u = 20$	0.010249	0.007224	0.007592
$N_{a,p}(K_u)$	**13**	**16**	**12**
Period 2	p = 1	p = 2	p = 5
$\rho_{0.5,p}; K_u = 10$	0.002259	0.00531	0.001248
$N_{a,p}(K_u)$	**9**	**9**	**9**
$\rho_{0.5,p}; K_u = 20$	0.001918	0.001151	0.002478
$N_{a,p}(K_u)$	**13**	**12**	**12**
Period 3	p = 1	p = 2	p = 5
$\rho_{0.5,p}; K_u = 10$	0.002686	0.001477	0.000761
$N_{a,p}(K_u)$	**9**	**9**	**7**
$\rho_{0.5,p}; K_u = 20$	0.001945	0.002115	0.001272
$N_{a,p}(K_u)$	**14**	**13**	**15**
Period 4	p = 1	p = 2	p = 5
$\rho_{0.5,p}; K_u = 10$	-3.55E-05	0.001088	0.007796
$N_{a,p}(K_u)$	**8**	**8**	**10**
$\rho_{0.5,p}; K_u = 20$	0.003554	0.006957	0.004502
$N_{a,p}(K_u)$	**12**	**11**	**14**

Table 4.4: Monotonicity of the risk measure $\rho_{0.5,p}$ for $a = 0.5$ and different values of p and K_u with eight in sample moths, where $N_{a,p}(K_u)$ represents the number of assets included in the optimal portfolio.

As we mentioned above, each value listed in Table 4.4 is the average of 10 runs of the risk measure of the best portfolio. In other words, the values taken into consideration are the ones referred to the minimum risk. This because those values are the one of the particles which performed better in the entire swarm. However, rows two and three list the number of the assets included in the optimal portfolio $N_{a,p}(K_u)$. Table 4.4 highlights that the Theorem 3.2, monotonicity theorem of the risk measure is not always respected. In fact, the preliminary test shows that $\rho_{a,p}$ is not non-decreasing with respect to different values of the local risk factor. As we described in chapter 3, the theory suggests that a risk-averse investor, who is more concerned about negative risk, uses a higher value of the local risk factor and considers the investment risky. A risk-seeker investor, on the contrary, uses a lower value of the local risk factor and considers an investment less risky. In general, the monotonicity seems to be not respected when the local risk factor p increases from 2 to 5 for different periods and different values of K_u.

The monotonicity is not respected neither between the values of p equal to 1 and 2 such as in periods 1 and 2 for value of K_u equal to 20. On the contrary, the only case where the monotonicity is respected for all the different values of p it is the period 4 with the maximum number of assets included in the optimal portfolio K_u equal to 10. However, it is important to notice that the cardinality constraint, which is responsible to limit the number of assets in the optimal portfolio, is always respected in each period for all the values of p and K_u. Furthermore, the minimum number of assets included in the optimal portfolio equal to 5 is always respected. However, an encouraging aspect is that the measure of risk in all the periods is always higher for values of K_u equal to 10 than for values of K_u equal to 20. This means that our algorithm is able to perform a satisfying exploration in the set of $K_u = 10$ and the subset $K_u = 20$ Although the monotonicity is not always respected, since all the other constraints, such as the budget one, are never violated, we decided to continue the analysis.

Table 4.5 shows the results of the analysis of the other part of Theorem 3.2, that is the non-increasing property of $\rho_{a,p}$ with respect to the global risk factor a. Furthermore, it has to be specified that in this analysis we kept $p = 2$, while the values of a are respectively $a = 0, a = 0.5, a = 1$.

Period 1	a = 0	a=0.5.	a=1
$\rho_{0.5,p}; K_u = 10$	0.020342	0.002654	-0.0005
$N_{a,p}(K_u)$	**8**	**8**	**9**
$\rho_{0.5,p}; K_u = 20$	0.003973	0.001788	-0.00036
$N_{a,p}(K_u)$	**11**	**10**	**13**
Period 2	a = 0	a=0.5.	a=1
$\rho_{0.5,p}; K_u = 10$	0.0021	0.001797	0.000102
$N_{a,p}(K_u)$	**8**	**8**	**9**
$\rho_{0.5,p}; K_u = 20$	0.006752	0.000785	0.002105
$N_{a,p}(K_u)$	**13**	**16**	**15**
Period 3	a = 0	a=0.5.	a=1
$\rho_{0.5,p}; K_u = 10$	0.00132	0.001945	-8.33E-05
$N_{a,p}(K_u)$	**8.**	**10**	**10**
$\rho_{0.5,p}; K_u = 20$	0.001245	-0.00037	-0.00035
$N_{a,p}(K_u)$	**15**	**10**	**13**
Period 4	a = 0	a=0.5.	a=1
$\rho_{0.5,p}; K_u = 10$	0.004542	0.006298	-0.00013
$N_{a,p}(K_u)$	**8**	**9**	**9**
$\rho_{0.5,p}; K_u = 20$	4.63E-02	0.002135	0.001879
$N_{a,p}(K_u)$	**13**	**11**	**17**

Table 4.5 of $\rho_{a,p}$ for $p = 2$ and different values of a and K_u, eight in-sample months, where $N_{a,p}(K_u)$ represents the number of assets included in the optimal portfolio.

From the results listed in Table 4.5 it is clear to see that is not always respected the non-increasing property of a. The theory in fact suggests that when the global risk factor is low the negative risk assumes more importance in the risk measure and lower values of a refer to risk-averse investors. On the other hand, higher values of a refer to investors who are risk-seeker. In period 3 for $K_u = 20$ the non-increasing property is not respected for a value of a equal to 0.5. Moreover, the property is not respected in period 4 as well. In fact, in this period for $K_u = 10$ and $a = 0.5$ the risk measure increases instead of decreasing. Except for those cases the non-increasing property of the risk measure always holds.

In this part we are going to analyze and comment the results obtained in the out-of-sample period. We have an optimal portfolio provided by the PSO for each of the four periods. Each portfolio has been evaluated in the in-sample period and verified in the out-of-sample period. For different values of K_u and p we computed the percentage return of the resulting portfolios. Even in this analysis we followed the same logic, the values presented are the average of the 10 runs.

When it comes the returns, Table 4.6 shows that the returns are pretty much low. In this application, we set the return constraint equal to the mean return of the in-sample period. Although Germany recovered greatly after the 2007 economic crisis and the DAX 30 index reflected it, the period of our analysis from April 2015 till November 2018 is the period where the index slowed down and after a peak in 2017 it started to rapidly decrease testing the medium-term minimum. We can say that the results obtained were expected. Looking more carefully to the results obtained, in the first period, the worst one, there no difference between portfolios with $K_u = 10$ and $K_u = 20$. Although the expected return constraint is always violated the best return is obtained by the combination of $p = 1$ and $K_u = 10$. The second period is the period where the PSO performed better than in the other ones. In this period the optimal portfolios have performances greater than the mean return of the in-sample period and so the minimum expected return constraint is always respected. However, the best performance, that is a daily return equal to 0.14%has been obtained combining the local risk factor p equal to 1 and the maximum assets included in the optimal portfolio K_u equal to 10. Period 3 is the second-best investment period. In this period the portfolio performances are close to the in-sample return 0.8%, but they remain below. However, the portfolio returns are comprised in a range of 0.00% - 0.7%. The best portfolio in this period, with a return of 0.7% is obtained by the combination of $p = 1$ and $K_u = 20$. Finally, period 4 has poor performances like the first one. Although, the in-sample return is negative, -0.1%, portfolio returns are even worse. In fact, the worst portfolio, given by the combination of $p = 2$ and $K_u = 20$, has a return of -0.23% which is 22 basis point worse than the in-sample return. To summarize, the best performance is given by the portfolio in period 2 with the combination of $p = 1$ and $K_u = 10$ able to provide a return of 0.14%. However, we can also notice that in period 1 2 and 3 risk-averse investors are able to earn higher returns rather than risk-seeker investors, who are able to collect better performances only in period 4.

Period 1	p=1	p=2	p=5
in-sample return		-0.03%	
$K_u = 10$	-0.20%	-0.24%	-0.29%
$K_u = 20$	-0.23%	-0.24%	-0.23%
Period 2	p=1	p=2	p=5
in-sample return		0.06%	
$K_u = 10$	0.14%	0.09%	0.07%
$K_u = 20$	0.08%	0.09%	0.08%
Period 3	p=1	p=2	p=5
in-sample return		0.08%	
$K_u = 10$	0.06%	0.00%	0.05%
$K_u = 20$	0.07%	0.06%	0.03%
Period 4	p=1	p=2	p=5
in-sample return		-0.01%	
$K_u = 10$	-0.14%	-0.12%	-0.16%
$K_u = 20$	-0.21%	-0.23%	-0.16%

Table 4.6: Portfolio out-of-sample returns with different values of p and K_u for all the four periods.

In this part we present the analysis made for BFO and the results obtained. The analysis is organized in the same way as for the one made for PSO.

Therefore, for the analysis of the monotonicity of the risk measure with respect to the local risk factor p, as we did before, we considered separately the four in sample periods and for each of them different values of the maximum number of holding assets K_u, respectively 10 and 20. In order to be clear, we remember that different values of p correspond to different levels of concern with respect to negative risk. In particular, small values of p indicate an investor who is risk-seeker. On the contrary, high values of p indicate an investor who is risk-averse. The analysis has been implemented keeping the value of $a = 0.5$ in order to be consistent with the previous one. Table 4.7 shows the results.

Period 1	p = 1	p = 2	p = 5
$\rho_{0.5,p}; K_u = 10$	0.005717	0.007221	0.012939
$N_{a,p}(K_u)$	**10**	**10**	**10**
$\rho_{0.5,p}; K_u = 20$	0.006002	0.008153	0.013321
$N_{a,p}(K_u)$	**16**	**14**	**15**
Period 2	p = 1	p = 2	p = 5
$\rho_{0.5,p}; K_u = 10$	0.003602	0.008592	0.017752
$N_{a,p}(K_u)$	**9**	**10**	**10**
$\rho_{0.5,p}; K_u = 20$	0.004628	0.00613	0.016694
$N_{a,p}(K_u)$	**15**	**17**	**15**
Period 3	p = 1	p = 2	p = 5
$\rho_{0.5,p}; K_u = 10$	0.001741	0.002883	0.005673
$N_{a,p}(K_u)$	**10**	**10**	**10**
$\rho_{0.5,p}; K_u = 20$	0.001541	0.003052	0.004934
$N_{a,p}(K_u)$	**13**	**15**	**19**
Period 4	p = 1	p = 2	p = 5
$\rho_{0.5,p}; K_u = 10$	0.003187	0.004425	0.010399
$N_{a,p}(K_u)$	**10**	**10**	**10**
$\rho_{0.5,p}; K_u = 20$	0.004712	0.004975	0.009535
$N_{a,p}(K_u)$	**16**	**16**	**15**

Table 4.7: Monotonicity of the risk measure $\rho_{0.5,p}$ for $a = 0.5$ and different values of p and K_u with eight in sample moths, where $N_{a,p}(K_u)$ represents the number of assets included in the optimal portfolio.

The results clearly show that the Theorem 3.2, which states that risk measure is non-decreasing for different values of p holds. In particular, we can see that the risk measure is non-decreasing with respect to p holds for each period and values of the cardinality constraints. In fact, in all four periods $\rho_{0.5,1} < \rho_{0.5,2} < \rho_{0.5,5}$ is respected. In other words, the higher the value of p the higher the value of the measure of risk. Furthermore, this evidence is theoretically meaningful since a more cautious investor to negative risk utilizes higher values of p. On the contrary, an investor who is risk seeker will use lower values of p since he considers the investment less risky. However, it is important to notice that these results show an important difference respect to PSO. As shown in Table 4.4, the measure of risk almost never respects the non-increasing property with respect to p. On the contrary using BFO Theorem 3.2 always hold. This crucial difference led us to rely much more on the results provided by BFO rather than PSO. In addition, the results in table 4.7 show that the cardinality constraint always hold for different values of $K_u = 10$ and $K_u = 20$.

Even for BFO we performed the analysis on the respect of the non-increasing property of the risk measure $\rho_{a,p}$ with respect to parameter a. As for PSO, we set the same values of a, that is $a = 0$, $a = 0.5$ and $a = 1$, while the parameter p has been kept fixed and equal to 2.

Period 1	a = 0	a=0.5.	a=1
$\rho_{0.5,p}; K_u = 10$	0.016219	0.00866	0.006825
$N_{a,p}(K_u)$	**10**	**10**	**10**
$\rho_{0.5,p}; K_u = 20$	0.011336	0.008591	0.006089
$N_{a,p}(K_u)$	**16**	**17**	**13**
Period 2	a = 0	a=0.5.	a=1
$\rho_{0.5,p}; K_u = 10$	0.009052	0.005514	0.004133
$N_{a,p}(K_u)$	**10**	**10**	**9**
$\rho_{0.5,p}; K_u = 20$	0.009887	0.006392	0.004183
$N_{a,p}(K_u)$	**11**	**14**	**15**
Period 3	a = 0	a=0.5.	a=1
$\rho_{0.5,p}; K_u = 10$	0.00502	0.002605	0.001047
$N_{a,p}(K_u)$	**9**	**10**	**10**
$\rho_{0.5,p}; K_u = 20$	0.004539	0.002973	0.001618
$N_{a,p}(K_u)$	**13**	**16**	**18**
Period 4	a = 0	a=0.5.	a=1
$\rho_{0.5,p}; K_u = 10$	0.008436	0.006184	0.005307
$N_{a,p}(K_u)$	**10**	**10**	**10**
$\rho_{0.5,p}; K_u = 20$	0.007359	0.005692	0.00375
$N_{a,p}(K_u)$	**15**	**17**	**13**

Table 4.8 of $\rho_{a,p}$ for $p = 2$ and different values of a and K_u, eight in-sample months, where $N_{a,p}(K_u)$ represents the number of assets included in the optimal portfolio.

The results once again show that the non-increasing property of a holds for each of the four in-sample periods and different values of K_u. In fact, $\rho_{0,2} < \rho_{0.5,2} < \rho_{1,2}$ is always respected. As mentioned in the previous analysis of the Theorem 3.2 we remember that lower values of a represent an investor who is more careful with respect to negative risk. On the contrary, higher values of a reflect an investor who is less concerned on negative risk. However, the results in Table 4.8 highlight that, differently from PSO, with BFO the Theorem 3.2 is always respected. This important evidence allows us to rely much more on the results and performances of BFO than the ones of the PSO.

In this part of the analysis we show and comment the results of the performance of the portfolio selected during the out-of-sample periods. However, as in the analysis of the PSO, even for BFO the Table 4.9 shows the expected returns, expressed in percentages, of the optimal portfolios found by the algorithm for different values of K_u and p. The expected returns listed in the Table 4.9, in order to be consistent with the analysis of the PSO, are the average of the 10 runs.

Period 1	p=1	p=2	p=5
in-sample return	-0.03%		
$K_u = 10$	-0.31%	-0.22%	-0.33%
$K_u = 20$	-0.34%	-0.24%	-0.29%
Period 2	p=1	p=2	p=5
in-sample return	0.06%		
$K_u = 10$	0.08%	0.21%	0.16%
$K_u = 20$	0.11%	0.09%	0.15%
Period 3	p=1	p=2	p=5
in-sample return	0.08%		
$K_u = 10$	0.07%	0.01%	0.00%
$K_u = 20$	0.00%	0.02%	0.06%
Period 4	p=1	p=2	p=5
in-sample return	-0.01%		
$K_u = 10$	-0.10%	-0.17%	-0.26%
$K_u = 20$	-0.21%	-0.15%	-0.14%

Table 4.9: Portfolio out-of-sample returns with different values of p and K_u for all the four periods.

It is clear that, even with BFO, the out-of-sample returns are very low. Except for period 2 the return constraints, which requires the portfolio in each period to outperform the mean return of the in-sample period is always violated. Starting from the first period, we can see that the final portfolios have the worst expected returns and in particular they are worse than the ones obtained in same period by the PSO.

However, portfolios with $K_u = 20$ have poorer performance respect to portfolios with $K_u = 10$, both in terms of evarage expected returns and in absolute values. In fact the worst performance is given by the combination of $p = 1$ and $K_u = 20$ with an expected return of -0.34%, while the best performance is provided by the portfolio that results from the combination of $p = 2$ and $K_u = 10$ with an expected return equal to -0.22%. The second period is the period that has the best performances and the return constraint is always respected. Furthermore, the final portfolios performed better than the ones of the PSO for the same period. The portfolio with the best performance is the one with $p = 2$ and $K_u = 10$, which provides an expected returns of 0.21%, while the worst expected return, 0.08%, is provided by the portfolio with $p = 1$ and $K_u = 10$. However, the results show that, in this period, the best performances are obtained with $K_u = 10$. Period 3 is the second best period in terms of performance. Although cardinality constraints never holds, we can see that expected returns of portfolios with $p = 1$, $K_u = 10$ and with $p = 5$, $K_u = 20$ are very close to the mean return of the in-sample period equal to 0.08%. On the contrary the worst performances are provided by portfolios given by $p = 1$, $K_u = 20$ and $p = 5$, $K_u = 10$ with an expected returns of 0.00%. Finally, the fourth period represents the second worst period in terms of performances. Indeed, the return constraint never holds and the final portfolios have performances lower than -0.01%. The worst performance in this period is given by the portfolio with $p = 5$ and $K_u = 10$, that has an expected return equal to -0.26%. On the other hand, the portfolio that performs better in the final period is the one resulting from the combination of $p = 1$ and $K_u = 10$ with an expected return of -0.10%. In order to summarize, the best performance equal to 0.21% is given by the portfolio with $p = 2$ and $K_u = 10$ in period 2, while the worst performance is provided by the portfolio with $p = 1$ and $K_u = 20$ in period 1. Moreover, given the results obtained and listed in Table 4.9 we can conclude that the performance of BFO are similar to the ones of PSO but more extreme. When BFO collects expected returns greater than the mean return of the in-sample period, such as in period 2, it outperforms PSO, being able to provide higher expected returns. However, when BFO realizes the worst performances, such as in period 1 and 4, they are worse than the ones obtained with PSO.

Conclusions

The main goal of this analysis was to optimize the portfolio selection problem through the application of two metaheuristics, BFO and PSO. In order to be as much realistic as possible we tried to replicate the conditions of real-world financial markets. In fact, the portfolio selection model chosen in this paper includes realistic assumptions such as transaction costs, taxes and other real-world financial problems that are not included in the original Markowitz model. Furthermore, with the purpose of fitting the real-world conditions, we selected a measure of risk different from the one originally used by Markowitz. Among the risk measures classified as coherent, we decided to use the two-sided risk measure. Basically, the choice of the risk measure allows an investor to consider at the same time positive and negative variance and to assign different weights to them in order to reflect the different attitude towards risk. This means that through the combination of the two parameters characterizing the measure of risk, the global risk factor a and the local risk factor p, it is possible to represent risk-averse investors and risk-seeker investors. Finally, the reason of using metaheuristics for this optimization problem is the huge advantage that they are able to provide in terms of computation time.

If we look at the results obtained, we can say that they are not fully satisfactory. The PSO is not able to respect the return constraint in 3 of the four periods considered in the analysis, that is the expected returns of the final portfolios do not exceed the mean return of the in-sample period. The best portfolio gives a daily expected return of 0.14%, while the worst portfolio gives a daily expected return of -0.29%.However, an explanation to the poor performances of the metaheuristics could be represented by the data considered for this application. In fact, if we analyze the DAX 30 index in the period between April 2015 and November 2018 struggled to increase. This hypothesis could be justified from the fact that, in the first and fourth period the daily mean return of the 26 assets taken into consideration is negative. Furthermore, in the valuation of the results it has to be taken into consideration the fact that the measure of risk does not always respect the Theorem 3.2, by which risk measure has to be non-decreasing with respect to parameter p and non-increasing with respect to parameter p. This issue could seriously jeopardize the goodness of the results obtained. In addition, if PSO is improved by increasing the number of iteration, the results could be better. We are quite confident that a higher number of iterations, such as 6000 or 10000, would provide better performances of the portfolios. However, we have to be aware that increasing the iterations could not represent always the best solution since it could dramatically increase the computation time.

With regards to BFO, the results are more or less aligned with the ones provided by PSO but the main difference is that they are much more reliable. In fact, using BFO the Theorem 3.2 of the two-sided risk measure holds in each period considered. Of course this aspect is fundamental when comparing the performances of the two metaheuristics. BFO is able to outperform PSO in period 2 when both respect the return constraint. On the contrary, BFO provides lowest daily-expected portfolio returns than PSO. However, the main disadvantage of the BFO is the computation time. In fact, investors are inclined to use metaheuristics to optimize portfolio selection problems and accept good solutions that are not the optimal ones though, as long as the computation time is reasonable. Therefore, the longer time needed by the BFO to carry out the optimization could be taken into consideration when it comes to decide which metaheuristics to chose. In particular if we decide to improve it by increasing the number of iterations the metaheuristics could become time consuming. However, taking into consideration pros e cons and the fact that with BFO the monotonicity of the risk measure is always respected we would recommend to use BFO rather than PSO.

The choice of the BFO is also supported by the increasing interest of the researchers towards this metaheuristics. Indeed, since its introduction BFO has increasingly drawn attention not only in fields like engineering but also in finance. Furthermore, the development is still ongoing given that the metaheuristics is still young and it has a lot of unexpressed potential.

To conclude, taking into consideration the results obtained from the analysis, we suggest to use BFO for portfolio optimization problems rather than PSO.

Acknowledgements

I would like to deeply thank my supervisor professor Marco Corazza, who assisted me throughout all this work. His willingness to help me although I was working allowed me to bring this work to light.

I would like also to thank my parents. I will always be grateful to them for having let me study throughout these years without worrying about anything else.

Finally, I would like to thank all my friends who supported me in this demanding period.

Bibliography

[1] H. M. Markowitz, "Portfolio Selection," *Journal of Finance ,* vol. 7, pp. 77-91.

[2] M. Corazza and D. Favaretto, "On the existence of solutions to the quadratic mixed-integer," *European Journal of Operational Research,* vol. 176, pp. 1947-1960, 2006.

[3] H. M. .. Markowitz, Portfolio Selection: efficient diversification of investments, Basil Blackwell, 1991.

[4] G. Szego, "Measure of risk," *European Journal of Operational Research,* no. 163, pp. 5-9, 2005.

[5] R. T. Rockafellar and S. Uryasev, "Conditional value-at risk for general loss distributions," *Journal of Banking and Finance ,* vol. 26, no. 7, pp. 1443-1471, 2002.

[6] C. Acerbi and D. Tasche, "On the coherence of expected shortfall," *Journal of Banking and Finance,* vol. 26, pp. 1487-1503, 2002.

[7] P. Artzner, F. Delbaen, J. Eber and D.Heath, "1999," *Mathematical Finance,* vol. 9, pp. 203-228.

[8] Z. Chen and Y. Wang, "Two-sided coherent risk measures and their application in relistic portfolio optimization," *Journal of Banking and Finance ,* vol. 32, pp. 2667-2673.

[9] S. Nesmachnow, "An overview of metaheuristics: accurate and efficient methods fro optimization," *Int. J. Metaheuristics,* vol. 3, no. 4, pp. 320-346, 2014.

[10] X.-S. Yang, Nature-Inspired Metaheuristcs Alrgorithms Second Edition, Cambridge: Luniver Press, 2010.

[11] C. Blum and A. Roli, "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison," *ACM Computing Surveys,* vol. 35, no. 3, pp. 268-308, 2003.

[12] L. Osman and G. Laporte, "Metaheuristics: A bibliography," *Ann. Oper. Res.,* vol. 63, pp. 513-623, 1996.

[13] D. Karaboga, B. Gorkemli and C. Ozturk, "A comprehensive survey: artificial bee colony (ABC) argorithm and applications," Springer Science+Business Media B.V., 2012.

[14] E. Bonabeau, M. Dorigo and G. Theraulaz, Swarm Intelligence From Natural to Artificial Systems, New York: Oxford UNiversity Press, 1999.

[15] E. Bonabeau and C. Meyer, "Swarm Intelligence: a Whole New Way to Think About Business," *Harvard Business Review,* pp. 107-114, May 2001.

[16] J. d. Valle, G. K. Venayagamoorthy, S. Mohagheghi and R. G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in," *IEEE Transactions on Evolutionary Computation,* pp. 171-195, 2008.

[17] H. Ahmed and J. Glasgow, "Swarm Intelligence: Concepts, Models and Applications," Queen's University, Kingston, Ontario, Canada, 2012.

[18] J. K. T. B. R. Poli, "Particle Swarm Optimization An overwiev," Springer Science + Business Media, LLC, 2007.

[19] R. E. J. Kennedy, "Particle Swarm Optimization," in *IEEE International Conference on Neural Networks*, 1995.

[20] E. N. Zalta, U. Nodelman, C. Allen and R. L. Anderson, "Cellular Automata," *Stanford Encyclopedia Philosophy,* vol. Fall 2017 Edition , 2017.

[21] M. Clerc and J. Kennedy, "The Particle Swarm—Explosion, Stability, and Convergence in a Multimedisonal Complex Space," *IEE Transactions on Evolutionary Computation,* vol. 6, no. 1, pp. 58-73, 2002.

[22] B. Nakisa, M. Z. A. Nazri, M. N. Rastgoo and S. Abdullah, "A SURVEY: PARTICLE SWARM OPTIMIZATION BASED ALGORITHMS TO SOLVE PREMATURE CONVERGENCE PROBLEM," *Jurnal of Computer Science ,* vol. 10, no. 9, pp. 1758-1765, 2014.

[23] Y. Shi and R. Eberhart, "A modified Partiscle Swarm Optimizer," IEE, Indianapolis, 1998.

[24] A. Engelbrecht, Computational Intelligence: An introduction., John Wiley and Sons, 2007.

[25] V. Sharma, S. Pattnaik and T. Garg, "A Review of Bacterial Foraging Optimization and Its Applications," in *National Conference on Future Aspects of Artificial intelligence in Industrial Automation*, 2012.

[26] S. Das, A. Biswas, S. Dasgupta and A. Abraham, "Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications," https://www.researchgate.net/publication/226084136, 2014.

[27] H. Chen, Y. Zhu and K. Hu, "Cooperative Bacterial Foraging Optimization," *Discrete Dynamics in Nature and Society,* vol. 2009, no. 815247, pp. 1-17, 2009.

[28] K. M. Passino, "Biomimicry of Bacterial Foragin for Distributed Optimization and Control," *IEE Control System Magazine ,* vol. June 2002, pp. 52-67, 2002.

[29] J. Li, J. Dang, F. Bu and J. Wang, "Analysis and Improvement of the Bacterial Foraging Optimization Algorithm," *Journal of Computing Science and Engeneering,* vol. 8, no. 1, pp. 1-10, 2004.

[30] M. Corazza and R. G. G. Fasano, "Particle Swarm Optimization with non-smooth penalty reformulation, for a complex portfolio selection problem," *Applied Mathematics and Computation,* vol. 224, pp. 611-624, 2013.

[31] D. Luenberger and Y. Ye, Linear and Nonlinear Programming, Springer, 2016.