Ca' Foscari
University
of Venice

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

Final Thesis

Model risk for risk measures: An application to the portfolio selection problem with two metaheuristics.

Supervisor
Ch. Prof. Marco Corazza

Graduand
Leonardo Colucci
856262

Academic Year
2020 / 2021

Contents

Introduction

The aim of this thesis is to assess Model Risk deriving from the application of two selected risk measures (Value at Risk and Expected Shortfall) to a realistic financial problem, so that I can compare their results in order to see which of the two is better in terms of Model Risk. More specifically, I want to see which of the two is a more reliable risk measure that better behaves in a realistic context.

The increasing complexity of financial instruments has caused the need for banks and financial institutions to employ very sophisticated models to deal with them. The implementation and use of these models must be done consciously, otherwise Model Risk (MR) may arise. The significant involvement of models in the Great Financial Crisis of 2007/2008 have highlighted the necessity of introducing a stringent regulatory framework to tackle this problem. Both US and EU regulators have issued guidelines in order to deal with MR, proposing the so called Model Risk Management framework. However, all these attempts are more focused in providing a qualitative approach to MR, and no compulsory and clearly defined procedure has been drafted to assess MR in a more quantitative way.

In Chapter 1, after having presented the general framework to deal with MR both in the US and EU, I will analyse the problem of MR for financial risk measures, considering the Value at Risk (VaR) and the Expected Shortfall (ES), and I will discuss some methods proposed in the literature to measure it.

Then, I decided to set up an experiment with the aim of assessing MR deriving from a realistic application of the two risk measures. More precisely, I will present an application of VaR and ES to the portfolio selection problem in three different scenarios, in order to assess their performances in different market conditions.

The portfolio selection problem, presented in Chapter 2, is a constrained optimization problem that aims to minimize an objective function (which is the selected risk measure) given a desired rate or return chosen by the investor. However, because I want a more realistic problem, I will introduce complex constraints, called mixed-integers constraints, that makes the portfolio selection a NP-hard problem, for which no exact method is able to find a solution.

Thus, in Chapter 3 I will introduce two approximated techniques that fall into metaheuristics algorithms' family. More in detail, I will apply for the resolution of the portfolio selection

problem the well known Particle Swarm Optimization (PSO) and the recently developed Grey Wolf Optimization (GWO), with the further aim of comparing their results in terms of Model Risk. In fact, because they are approximated methods, another component of MR deriving from their application will rise.

Then, in Chapter 4 I will present the experiment. I will apply the two metaheuristics algorithms to eleven securities taken from the S&P 500 index in order to define the optimal portfolios in three different scenarios: two consider crisis periods, respectively the financial crisis of 2007-2008 and the recent Covid-19 crisis, while the third considers a growing phase of the US economy in 2013-2014. Furthermore, I will contemplate two cases for each metaheuristics: one with each of the two risk measures as objective function to be minimized. Thus, I will have four models to analyse and four portfolios as outputs in each scenario: I will have a version of PSO with VaR as objective function, a PSO with ES, a GWO with VaR and a GWO with ES. The aim of this first analysis is to select the best algorithm in terms of Model Risk, and the three criteria used are:

- the best fitness value obtained, that states level of convergence of the algorithm towards the real optimum solution;
- the violations of the constraints;
- the computational effort and time required by the algorithm to fully execute the code.

Having selected the best algorithm, I will perform a further backtesting analysis, using only the portfolios found by this algorithm in the three scenarios, with the aim of assessing Model Risk for the two risk measures used. I will follow the approach of Boucher et al. (2014) that individuates the three criteria that a good risk forecast should possess:

- Expected frequency of violations;
- Suitable magnitude of violations;
- Absence of violations clustering.

In conclusion, I am able to provide a quantitative method to assess Model Risk linked both with the choice of the algorithm used to solve the portfolio selection problem and with the choice of the risk measure.

Chapter 1

Model Risk and Model Risk Management

Due to the increasing complexity of financial instruments, such as CDOs and ABSs [1]*, banks and financial institutions have to rely more consistently on mathematical and statistical models to evaluate, manage and control those instruments in order to help them make correct decisions. As a direct consequence, Model Risk arises, producing potential threats that must be addressed by the users of the model and by the Regulator. In this chapter, after having presented a brief summary of the main events that have shaped the environment around Model Risk, I will introduce the concept of model and Model Risk. Then, I will focus on the new normative environment that has been recently introduced, focusing on the European and US perspectives. As we will see later, both these two approaches provide a good qualitative framework to deal with Model Risk, known as Model Risk Management, but struggle when trying to identify quantitative methods to assess this risk. In the final paragraph, focusing on Model Risk deriving from financial risk measures (such as Value at Risk and Expected Shortfall), I will present some methods proposed in the literature to analyse and quantify Model Risk linked to these two risk measures.*

1.1 Introduction to Model Risk and Model Risk Management

As previously pointed out, the importance of statistical and mathematical models in the financial field has risen steeply in recent years. Almost every core activity of banks and financial institutions is based on the use of those models. Among the most important and debated ones there are models for measuring the risk of an investment, such as the Value at

[1] CDOs stands for Collateralized Debt Obligations, while ABSs stands for Asset Backed Securities. An ABS is a debt financial instrument backed by an underlying asset. A CDO is a peculiar type of bond composed of many ABSs.

Risk (VaR) and the Expected Shortfall (ES)[2]. From their application derive several crucial decisions: for example banks have to adequate their capital reserves with respect to their portfolio's riskiness, which is measured, among others, in accordance with VaR or ES. Thus, a good implementation and a deep understanding of those models is vital.

In this environment, the rapid development and growing use of the last years of complex algorithms, Artificial Intelligence (AI[3]) and Machine Learnings (ML[4]) techniques allow to automate an increasing number of complex processes, sometimes delegating to software, based on those techniques, the task of applying and managing important models, and thus of taking important decisions. Even if these advances may seem very promising, they of course must come with some risks.

Given that the use of models provides a lot of advantages in producing an easier representation of real world relationships and improving decision making process, it is acknowledged that relying blindly to them must come with a cost. In principle, all model outputs should be taken with an "healthy" degree of skepticism (Brown et al., 2015). In fact, the vast employment of models inevitably triggers the rising of some risks, known as Model Risks (MR). Those risks can be due to model errors, for example wrong specifications, implementations or coding of the model, or errors in the use of the model, for example when applying a model designed for a specific task in a different field that is beyond what it is supposed to do. Thus, like any other type of risk, MR should be evaluated, managed and regulated. The development of a strong framework to deal with Model Risk, the so called Model Risk Management, that both looks at qualitative and quantitative aspects, is one of the hot topics of the last decade and much more effort is required to achieve this goal by the US and EU regulators.

In the following subparagraph I will (briefly) present a summary of the crucial events that have contributed to the initial draft of the regulatory framework to deal with Model Risk.

[2] VaR and ES will be presented in more detail in paragraphs 1.4 of chapter 1 and 2.4 of chapter 2.

[3] According to Britannica, AI is "the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings." AI is a subset of Computer Science. Reference: https://www.britannica.com/technology/artificial-intelligence.

[4] According to Britannica, ML is the "discipline concerned with the implementation of computer software that can learn autonomously." ML is a subset of AI. Reference: https://www.britannica.com/technology/machine-learning.

1.1.1 Basel Agreements and the History of Model Risk

Now it is important to provide a short presentation of the core events that occurred in the recent years that have contributed to the development of the concept of Model Risk and to the normative framework for Model Risk Management. Figure 1 shows the timeline of these crucial events.

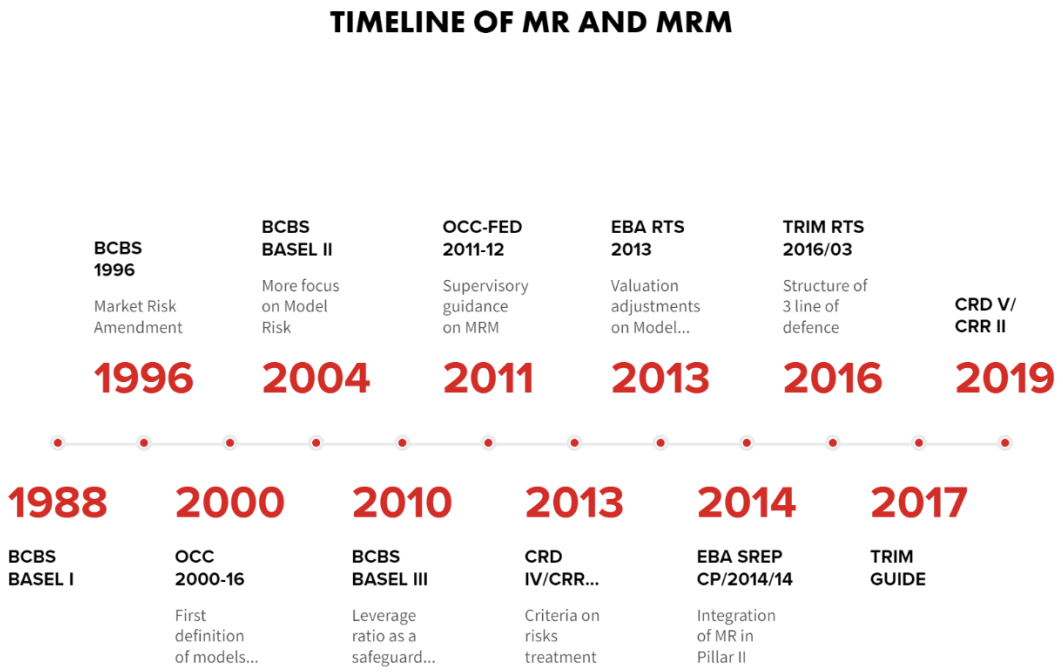


Figure 1: Timeline of the core events relative to Model Risk and Model Risk Management from 1988 to 2019.

The first event happens in 1988, when the Basel Committee on Banking Supervision (BCBS[5]) gathers for the first Basel Agreement. Basel I imposes new capital standards for all financial institutions: in particular, it provides a definition of capital as the composition of two elements, basic equity (Tier I) and supplementary capital (Tier 2). Tier 1 capital is composed of shareholders' equity and retained earnings and it is the primary and most liquid source of every bank. On the other hand, Tier 2 capital is composed of revaluation reserves, hybrid capital instruments and subordinated term debt. By nature, Tier 2 capital is less liquid than Tier 1 and it is a less reliable source for the bank. Moreover, Basel I provides the minimum

[5] It is an international authority instituted in 1974 by the heads of the Central Banks of ten of the most industrialized countries in the world. It has legal residence in Basel, in Switzerland. Its main aim is to harmonize prudential regulation and supervision of the Banking sector world wide.

capital requirement that every bank must maintain at any moment in order to absorb potential losses over time, which is given by the ratio of total capital over total risk weighted assets. Thus, in order to calculate this ratio every asset is multiplied by a weight in accordance to its riskiness. Those weights are respectively divided into four categories: 0% or safe, 20% or low risk, 50% or medium risk and 100% or high risk. Nonetheless, even if Basel I represents the first effort to regulate the banking sector, the only type of risk taken into account is Credit Risk, which is the risk that the counterparty of a contract would fail to comply with its obligations.

It is in 1996, with the Market Risk Amendment by the BCBS, that a new category of risk is introduced: Market Risk. This new typology includes all the risks deriving from unexpected and substantial changes in the market's valuation of a company's assets and liabilities due to changes in interest rates, exchange rates or other market movements. This constitutes a little step forward from Basel I and all its limitations. However, it was not sufficient because many other types of risk that were not considered (regulated).

The first definition of Model Risk is drafted in 2000 by the Office of Comptroller of the Currency (OCC[6]). Nonetheless, Model Risk is not included in any of the BCBS agreements, and not enough attention is devoted to it.

In 2004, Basel II agreement is introduced in substitution of the previous too 'simplistic' agreement. Basel II framework is divided into three pillars:

- Minimum capital requirements, based now on three types of risks;
- Supervisory review for risk review and capital assessment;
- Market discipline for public disclosure and transparency.

Besides Credit risk and Market risk, here a new risk is considered: Operational risk. This new category considers all the potential losses due to inadequate internal processes or external events like legal actions. In addition, Basel II also introduced the possibility for banks to develop their own internal models to assess the amount of capital to be kept, called the Internal Rate Based approach (IRB). With this new possibility, a potentially huge component of Model Risk, still not regulated, is introduced. However, as 2007/2008 events might highlight, Basel II is not found out to be fully effective in its aim.

[6] It is a Federal agency of the US government, instituted with the National Currency Act in 1863. Its aim is to regulate and monitor US banks, also outside US borders.

In fact, in response to the Great Financial Crisis[7], Basel III (2010) is drafted as an implementation of Basel II. According to Walter (2011), the main reasons for introducing Basel III are the magnitude and the frequency of banking crises in the last years and the hope that the benefits of implementing such a complex regulation would possibly outweigh the costs. Basel III increases the quality standard and the quantity of capital to be kept by banks: it distinguishes between two types of capital. Level 1 capital, which is the most important for ensuring continuity to the core activities of the bank, is composed of Core Equity Tier 1 (CET 1) and Additional Tier 1 (AT 1) capital. CET 1 is calculated as the ratio between total capital over risk weighted assets and it is composed of common shares and retained earnings. Basically, it represents the most liquid part of a bank's capital, and it ensures that the bank is able to withstand any possible financial distress. AT 1 instead is composed of all those instruments that are not common equity but that can be included into Level 1 category, like: Contingent Convertibles (or CoCos[8]) or hybrid securities that can be converted into equity. Level 2 capital is instead composed of Tier 2 capital, which constitutes the more illiquid part of capital, comprising reserves and hybrid instruments.

Basel III also introduces stress testing, liquidity risk supervision and capital buffers to tackle both procyclicality and countercyclicality[9]. Thus, in this context the concept of Model Risk became to have more relevance, but an explicit guidance was still lacking.

The first comprehensive guidance on Model Risk Management was drafted in 2011 by OCC and Federal Reserve (Fed) with the Supervisory Review (SR 11-7) on Supervisory Guidance on Model Risk Management.

Looking at the European environment, the introduction of the Capital Requirement Regulation (CRR 575/2013[10]) and the Capital Requirement Directive IV (CRD IV 36/2013[11]) in 2013 proposed new technical criteria concerning the assessment and treatment of risks for banks.

[7] The Crisis blew up in 2007 with the collapsing of the subprime mortgage bubble, however I will give more details later on in this chapter.

[8] CoCos are high yield debt instruments that can be converted into equity or stocks.

[9] With procyclicality I mean the mechanism through which the financial system contributes to amplifying cyclical fluctuations in the market, enhancing expansive phases but at the same time worsening recession phases. Vice versa, with countercyclicality I mean all the activities that do not follow the ongoing trend of the economy, but instead reverse it.

[10] Regulation (EU) No 575/2013 of the European Parliament and of the Council of 26 June 2013 on prudential requirements for credit institutions and investment firms.

[11] Directive (EU) No 36/2013 of the European Parliament and of the Council of 26 June 2013 on access to the activity of credit institutions and the prudential supervision of credit institutions and investment firms.

In the same year, the European Central Bank (ECB) published a Regulatory Technical Standard (RTS) concerning Prudent Valuation techniques and adjustments, more specifically tackling the problem of Model Risk quantification.

In 2014 the ECB in the context of the Supervisory Review and Evaluation Process (SREP[12]) integrated Model Risk as part of the Pillar II of the European Banking Union.

In 2016 the ECB, in the context of the Targeted Review of Internal Models (TRIM[13]), while adapting to the European framework the BCBS 2015-d328 guideline regarding Corporate Governance Principles for banks, proposed a three lines of defence structure to deal with Model Risk. The first line focuses on model development, aiming to strengthen this crucial step. The second line focuses on risks and model validation, while the third line deals with the internal control function, or the Audit function. However, this three lines structure will be discussed more in detail later on in this chapter.

In 2017 a complete guidance for the TRIM process was published by the ECB, giving an official interpretation to the existing legal framework (CRR, CRD IV and ECB guidelines). The aim of the TRIM process is the harmonization at European level of practices and methods to ensure the correct usage of internal models by banks and financial institutions.

Finally, in 2019 CRR II/CRD V were introduced, to substitute the CRR and CRD IV package, as a step forward through the implementation of Basel IV (Nielsen and Schulte-Matter, 2019). While CRR II is supposed to become fully operational from June 28 2021, CRD V has already come into effect on December 28 2020. For a full review on this topic see Nielsen and Schulte-Matter (2019).

In conclusion, we have seen an escalation of new regulatory frameworks and laws in recent years concerning Model Risk. However, the first regulatory effort to regulate Model Risk and to provide the management tools to deal with it comes from the USA, seeing the OCC-Fed collaboration in the frontline. Only later on the Euro-zone implements its own regulations (CRR/ CRD IV), and the EBA develops SREP and TRIM procedures to directly tackle this problem. However there is still further work to do in order to provide a more reliable Model Risk Management framework to deal both qualitatively and quantitatively with Model Risk, and the path is still long.

[12] The SREP procedure aims to assess and measure the risks that emerge from every bank in the Union, and it consists of an yearly basis monitoring procedure in order to provide to the bank the actions needed to be taken.

[13] According to the ECB, the TRIM "is a large-scale project conducted by the ECB in close cooperation with the NCAs" and "its aim is to reduce inconsistencies and unwarranted variability when banks use internal models to calculate their risk-weighted assets."

1.1.2 Model Risk

The OCC 2011-12[14], in accordance with the Fed provide a definition of model:

Definition. *Model*. The term model refers to a "quantitative method, system or approach that applies statistical, economic, financial or mathematical theories, techniques and assumptions to process data into quantitative estimates."

A model is a simplification of real world relationships. This simplification is both necessary, because no model can capture entirely the complexity of this web of relationships, and intentional, because many times what matters is just to focus on the most relevant aspects of those relationships.
Furthermore, following the OCC Guidance, we can identify three crucial components of a model:

- Information input component;
- Processing component;
- Reporting component.

The first component refers to all the assumptions and the data needed from the model to be applied. The processing component is the one that transforms raw data (inputs) into estimates (outputs). Finally, the reporting component is crucial for transforming the outputs into relevant information that could help decision making. All those three components are crucial, and the model's users must assess their quality constantly by looking at the seven properties that I am going to introduce soon.
It follows from the definition of model that relying on such a simplification of real world relationships comes with a cost. We can identify two types of costs: direct and indirect. Direct costs deal with all the resources used to develop and implement the model. On the other hand, indirect costs are related to all the possible negative consequences that the use of an incorrect model may cause. In order to mitigate the effect of indirect costs, it is important to assess model quality. Following the idea of Gabaix and Laibson (2008) it is possible to identify seven key properties that a good economic model should possess:

[14] OCC, Office of the Comptroller of the Currency: Supervisory Guidance on Model Risk Management, 2011.

- Parsimony;
- Tractability;
- Conceptual insightfulness;
- Generalizability;
- Falsifiability;
- Empirical consistency;
- Predictive precision.

Parsimonious models rely on few key assumptions and provide the user some degree of freedom when using them for their purposes. They are preferable "because they prevent the researcher from consciously or subconsciously manipulation of the model so that it over-fits the available facts" (Gabaix and Laibson, 2008). Tractability refers to the ease according to which a model can be analysed, implemented and used. Conceptual insightfulness is linked with the ability of the model to shape and describe the fundamental aspects of economic systems or behaviours that it is trying to mould. Generalizability is another crucial aspect that a good model should possess, meaning that it could be applied to a wide range of situations. Falsifiability is synonymous with prediction. "A model is falsifiable if and only if the model makes nontrivial predictions that can in principle be empirically falsified. If a model makes no falsifiable predictions, then the model can not be empirically evaluated" (Gabaix and Laibson, 2008). Empirical consistency is linked with the level of reliance of the model upon the data on which it is structured. If the data are not biased, then the model is likely to provide consistent predictions. Finally, predictive precision is linked with the accuracy of the forecasts made by the model.
Not managing adequately these components may potentially cause some threats. Thus, it is in this framework that a new concept arises: the concept of Model Risk (henceforth MR). MR was formally defined in 2011 by OCC-Fed 2011-12 as follow:

Definition. *Model Risk*. Model Risk is defined as "all possible adverse consequences that can result from correspondingly inadequate or incorrectly implemented models as well as from their outputs and reports."

As models become more sophisticated, for example through the implementation of Artificial Intelligence, and interconnected with each other, MR rises. We can identify two main reasons for which MR may occur:

- Fundamental errors in the model;
- Incorrect or inappropriate use of the model.

Fundamental errors may occur at any point in the design phase of the model or in the implementation phase, and may produce erroneous or inaccurate outputs that can finally lead to wrong decision making. This error type may be caused by incomplete data, flawed theory or assumptions, coding errors or decreasing predictive power due to environmental changes. Either if we are dealing with an accurate and robust model, i.e. with no fundamental errors, the risk of incorrect or inappropriate use is still looming. It is crucial for managers and all decision makers to understand the limitations of a model in order to ensure that the scope for which it is designed for and the field of application are both respected. Thus, in the last years, dealing with MR has become of crucial importance for all financial institutions. In particular, banks must now explicitly deal with MR as they do for all other types of risks: they should identify the impact and the sources causing MR to arise, they have to maintain a comprehensive documentation of every model, and finally they should challenge each model periodically and perform ongoing monitoring and benchmarking. However, it is important to remember that MR cannot be completely eliminated: yes, it can be diminished but a part of this risk can be seen as systematic. As a consequence, the need for tools capable of managing this risk arises. This requires a strong effort from the regulator to develop new guidelines and regulations and a tremendous endeavour for all financial institutions and firms to cope with them: it is crucial to develop a solid governance framework around each step of Model Lifecycle[15].

Other two elements that characterize a model should be taken into account when determining the intensity of the procedures used to evaluate MR:

- Materiality;
- Uncertainty.

Figure 2 gives a representation of these two concepts, showing a matrix that intersects materiality and uncertainty forming a grid upon which models are positioned. Models in the top right corner are the ones that should be managed with greater attention since they score

[15] Model Lifecycle links together all the main aspects of Model Risk Management. More precisely: proposal, development, testing, validation, approval, usage, performance monitoring and reporting. However, Model Lifecycle and Model Risk Management will be discussed with more detail later on in this chapter.

the highest value of both components. Materiality can be defined as the impact of the model into a bank's activities. It is linked both to the relevance of the model in the bank's framework and to its complexity. To assess materiality one should take into account the financial impact of the model, the context of use and the volume of resources managed through that model. A simple deterministic model with few inputs and assumptions (so with less likelihood of errors), used for a non core activity of the bank, like model C in Figure 2, would have a lower materiality impact than a complex stochastic model used for a core activity with several inputs and assumptions, like model A. The higher the materiality associated with a model, the higher will be its importance, and thus a more prudent development, calibration, use and monitoring should be put in place.

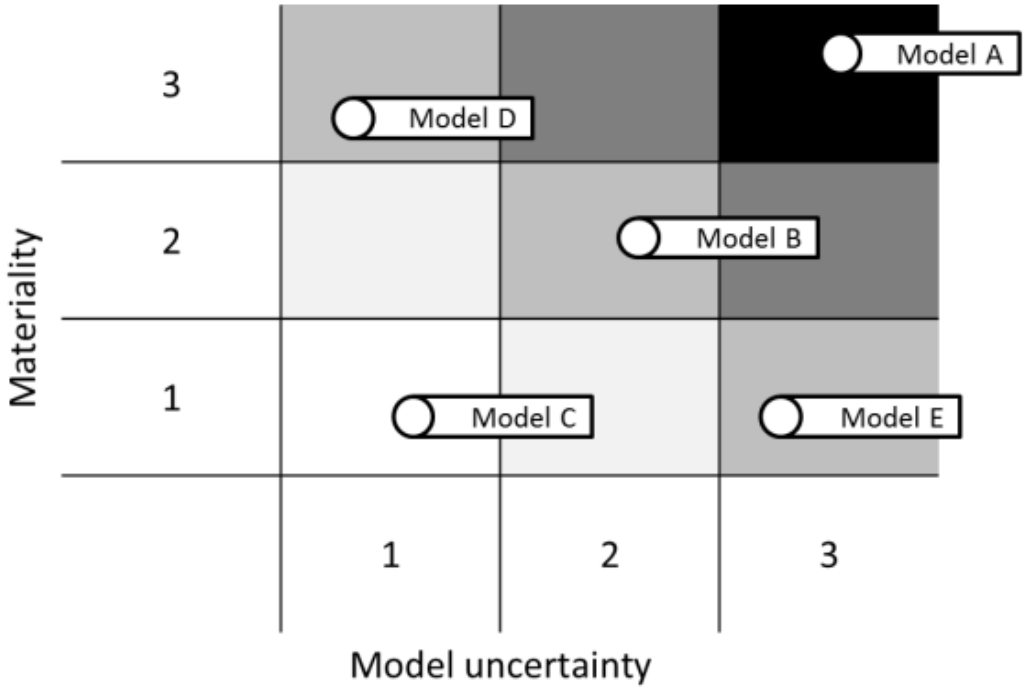


Figure 2: Graphical representation of model's materiality and uncertainty. Source: Bellof and Wehn (2018).

Uncertainty can be defined as the randomness of the phenomenon that the model is trying to capture: the higher is the level of predictability, the higher would be the caution needed to manage it. For example a stochastic model would have a higher impact in the uncertainty component, like model E or A in Figure 2.

Both qualitative and quantitative methods can be used to assess materiality and uncertainty. For example, with respect to qualitative methods to measure materiality banks can determine

through a score range the importance of the decisions made through the results of that model. Another qualitative approach to measure uncertainty instead, can be to assess the sensitivity of the results due changes in parameters or assumptions.

Now that I have presented the key elements of MR, it is important to provide some concrete examples of the impact that inadequate regulation and misapplication of models can cause. Models misuse and deficiencies, in fact, have largely contributed to the Great Financial Crisis.

1.1.3 Contribution to the Great Financial Crisis

The Great Financial Crisis began in 2007 with the collapse of the subprime mortgage bubble. Among the main causes of the Crisis we can recognise mathematical modelling: more specifically, the misuse of the Gaussian copula model is the one that mostly came under criticism (Watts, 2016). The Gaussian copula is a multivariate cumulative distribution function, and it is a method to model multivariate distributions considering the dependencies among variables. In fact, according to Sklar's theorem[16] any joint probability distribution function can be written in terms of univariate marginal distributions and of a copula, that captures all the information about the dependencies between variables.

This model is used in finance to evaluate the risk of Collateralized Debt Obligations (CDOs[17]). Usually CDOs are divided into different tranches with different risk-return characteristics. They can have a senior, mezzanine and equity tranches. The senior tranche has the highest credit rating among them because the losses that may arise from the default of the underlying are borne in the first place by the equity, then by the mezzanine and only as a last resort by the senior tranche. One thing that makes CDOs so appealing is that if the risky assets composing the pool are rated BBB it is possible through this technique to create a senior, super safe, tranche rated AAA. Furthermore, it is common for CDOs to contain several Asset Backed Securities (ABS[18]).

[16] This theorem is at the basis of the construction of multivariate stochastic models through a copula approach.

[17] According to Borsa Italiana a CDO is a complex synthetic investment product that is backed by a pool of loans and other assets and sold by institutional investors. Reference: https://www.borsaitaliana.it/borsa/glossario/collateralized-debt-obligation.html.

[18] According to Borsa Italiana an ABS is a type of financial investment that is collateralized by an underlying pool of income-generating assets. Reference: https://www.borsaitaliana.it/notizie/sotto-la-lente/assetbackedsecurities.htm#:~:text=Le%20Asset%20backed%20securities%20(o,tassi%20di%20interesse%20fissi%20o.

According to Donnelly and Embrechts (2010), it is widely accepted that the root of the crisis was CDOs of ABSs, which mainly consisted of American subprime mortgages. Here, the main issue is the transfer of mortgage default risk, through CDOs, from lenders directly into the financial market (banks, hedge funds and insurance companies). This process is called securitization[19], and allows banks to move from an "originate to hold" model, where they holds all the mortgages made (among with the risk) to an "originate to distribute" model, in which they sells these mortgages to other entities. More precisely, banks can pool together these mortgages and transfer them to a special-purpose vehicle (SPV), which is sponsored by the bank but it is at the same time independent from it. If the bank defaults the SPV is not involved. Thus, banks had a great an incentive in issuing CDOs, as they were able to remove risky underlying and return instead cash.

The proposal to use the Gaussian copula to model CDOs comes by Li (1999), and it suddenly receives great approval and success in the financial field due to its simplicity, becoming in a few years the standard industry. Credit Rating Agency (CRA) such as S&P, Moody's and Fitch adopt the Gaussian copula too. However, this model also includes some drawbacks that may have been ignored or considered not to be relevant by the people in the industry. According to Duffie (2008), the modelling of default correlation is currently the weakest part in the risk measurement and pricing of CDOs. This was particularly emphasized when trying to model tail events. Furthermore, the Gaussian copula provides inadequate modelling of default clustering, inconsistent implied correlations between tranches of CDOs, and no early warning signals. In the end, it does not capture the main features of what it is attempting to model.

To give an idea of the numbers of CDOs issued in that period, as Watts (2016) pointed out, at the beginning of the new millennium the total issuance of CDOs was less than $70 billion, as it can be seen in Figure 3. This was due to the fact that such a complex financial instrument requires an adequate model that can easily price them and measure its embedded risk. It is after the adoption of the Gaussian copula that the CDOs market skyrocketed in a few years. Just before the collapse of the bubble, in late 2006, the CDOs market soared to more than $500 billion per year. This rapid growth in the use of CDOs caused a chaotic environment in which no one more was able to understand every step behind in the chain of connections

[19] According to Investopedia, "Securitization is the procedure where an issuer designs a marketable financial instrument by merging or pooling various financial assets", such as contractual debt, residential/commercial mortgage, auto loans and so on, "into one group. The issuer then sells this group of repackaged assets to investors", in the form of bonds or CDOs. Reference: https://www.investopedia.com/terms/s/securitization.asp.

between the original loan and the final holder. If on the one hand sellers of CDOs and CRAs were abusing the Gaussian copula model without understanding the embedded risks, on the other hand buyers were trusting that model without adequate criticism and without performing any personal analysis.

When in the late 2000s subprime mortgages started to default, reaching a default rate above 14% in the US, a lot of CDOs tranches rated AAA by CRAs started to default, triggering a chain of events that intensified the default correlation of CDOs products, and eventually resulted in the greatest financial crisis since the Great Depression. The Gaussian copula was not able to capture these extreme tail events and a lot of financial products were mispriced. This is a clear example of the potential (extreme in this case) consequences that may arise from an inadequate control of MR. As we have seen, the Gaussian copula was used outside its field of application with the aggravating circumstance for the users of not posing sufficient attention to its limitations and assumptions. This Crisis is what pushed the regulator to develop guidelines for the treatment of MR: this is what is called Model Risk Management (MRM).

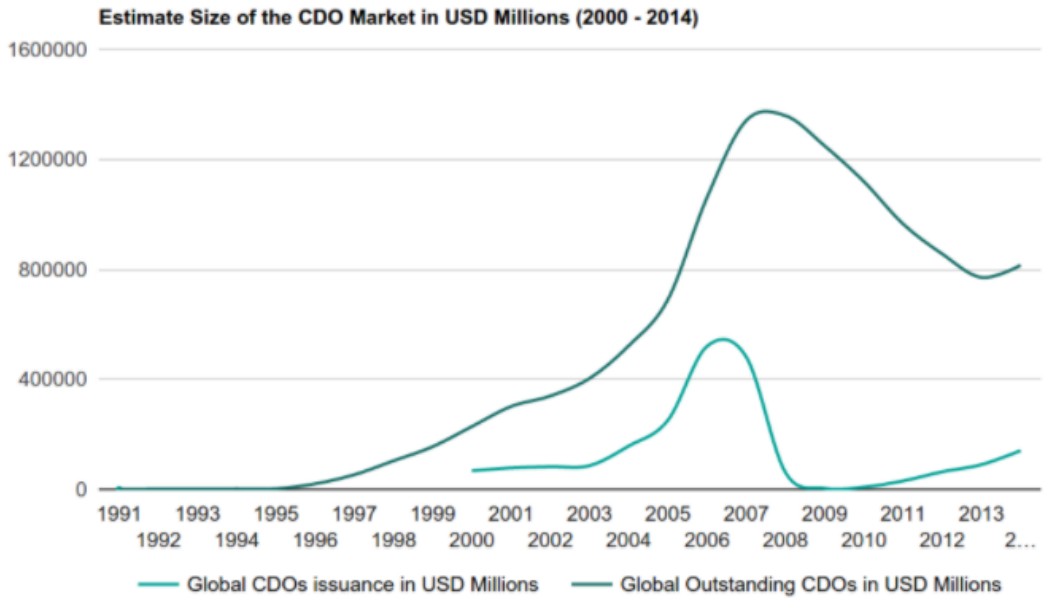


Figure 3: Size of the CDO market in USD Millions among the years, differentiating between CDOs issuance and outstanding CDOs. Source: http://www.turingfinance.com/recipe-for-the-financial-crisis/.

1.2 Model Risk Management in the US Framework

In this paragraph I will focus on the regulation behind MRM in the US. The BCBS d328 guideline, regarding Corporate Governance Principles for banks gives the definition of risk management:

Definition. *Risk Management*. "It is the process established to ensure that all material risks and associated risk concentrations are identified, measured, limited, controlled, mitigated and reported on a timely and comprehensive basis."

The main reference, as shown in subparagraph *1.1.1*, for MRM in the US was published on the 4th of April 2011, by OCC-Fed bulletin 2011-12 with the Supervisory Review (SR 11-7): on Supervisory Guidance on Model Risk Management. The Comprehensive Capital Analysis and Review (CCAR), introduced by the Fed to regulate large banks and financial institutions and to perform stress tests, is another important regulatory framework in the US and it is part of the Fed oversight process over MRM. According to this guidance, MR should be treated like any other type of risk with adequate respect and consideration by senior management (that for example have lacked in the Great Financial Crisis).
According to OCC-Fed bulletin 2011-12: "banks should objectively assess model risk using a sound model validation process, including evaluation of conceptual soundness, ongoing monitoring, and outcomes analysis." The key element here is the so called "effective challenge" principle, which consists of a critical analysis in order to assess the assumptions and the limitations of the model. An effective challenge consists of three elements:

- Incentives;
- Competence;
- Influence.

The first element refers to the incentives provided to the team that has the task of conducting the inspection of the model, in order to give them the right motivation to perform a meticulous analysis. Competence is linked to the expertise and skills of the team. Whereas, influence depends on the authority of the members of the team inside the organization and on the support provided by senior management.
The higher the level of separation between the development phase and the one responsible for testing the model (to better avoid conflicts of interest), the higher would be the incentive

to effectively challenge the model in order to assess its robustness and riskiness. Competence and Influence are also crucial for model developers, supervisors and users: they must be well informed and experienced in order to understand all the implications and assumptions that the model involves.

With the SR 11-7, OCC-Fed, requires that all financial institutions must cope with the following requirements, in particular they must:

- Create a strong model governance framework;
- Set up a comprehensive model inventory;
- Improve model validation process;
- Enhance ongoing model performance monitoring;
- Build a robust documentation of each model used;
- Operate an embedded control framework.

The approach provided by OCC-Fed to deal MRM is divided into three core areas:

- Model development, implementation and use;
- Model validation procedures;
- Model governance, policies and control.

Analysing the first area, model development and implementation phase is concerned with the design, the theoretical foundation and the logic behind the model. It is important that these elements are supported by sufficient literature and research. Model check is also important as it allows testing the model and providing some feedback to the developers regarding parameters estimation, assumptions and so on, before the model is fully approved and ready to be used. This second area is characterized by the validation procedure which is defined by OCC-Fed as:

Definition. *Model Validation*. It is the "practice of assessing the assumptions, underlying theoretical basis, and data used by models, as well as the processing, output, and reporting is one of the key ways to manage, mitigate, and control model risk."

Thus, having a sound model validation system in the organization helps to manage MR. It is a useful tool to identify potential model limitations and to report them quickly to managers so that measures to cope with these problems can be taken. It is crucial for the validation process to be independent from the development phase, so that the model could be effectively

challenged and tested in a critical way, without having pressures from developers or people that could have a stake in it. It is vital that the validation staff have the adequate knowledge, skills and expertise, but also powers and authority to complete their task appropriately. Furthermore, the validation process should be carried out on an ongoing basis to ensure that changes in the economic environment do not impact on the robustness and reliability of the model. For example, OCC-Fed states: "Computer code implementing the model should be subject to rigorous quality and change control procedures to ensure that the code is correct, that it cannot be altered except by approved parties, and that all changes are logged and can be audited." Ongoing monitoring function can be helped by a benchmark analysis that compares model's inputs and outputs with an alternative model: differences and misalignments should trigger an inquiry. Another technique that can be used to analyse a model's predictive power is backtesting: it consists in checking whether the output of the model is in line with the model forecasts. Usually one divides the sample time series of data used by the model into two subsamples, one effectively used for the development of the model, while the other used as a virtual future to test its predictive power. More precisely, this is exactly what I will do in chapter 4 when I will present the experiment for testing Model Risk for risk measures.

The third area is characterized by governance structure, policies and control. The board of directors and the senior management have a crucial role here. The latter is usually responsible for enforcing and maintaining an effective MRM framework; while the former is the ultimate responsible and delegates powers to senior management. Thus, it is the board of directors that should carry out policies to implement a sound MRM in the organization identifying clearly roles and responsibilities of each internal function. Finally, regarding the control aspect, an important role is the one of internal audit functions, that shall evaluate the effectiveness of the MRM framework implemented by the organization. It is crucial that the audit has its own powers and skills to perform its tasks. In addition, having a complete model inventory and documentation is essential as they both provide a database of information regarding all the models in function or retired, that for example can help the development or the validation phase.

All the aspects of the MRM framework are linked together in the so called Model Life Cycle, which is shown in Figure 4. Starting with the raw data, we have the development phase, in which the model is designed according to its purpose. Then, we move to the implementation phase, which requires adequate infrastructure to run tests and check model strengths, limitations and assumptions. Moving on we have the validation step, that as I said before

should have its independence from the developers in order to avoid conflicts of interests. If the model overtakes all these steps then it is approved and the firm can begin to use it, otherwise if some problems are detected, the model should pass through a re-development phase. Then, performance monitoring is conducted on an ongoing basis. And finally, this is linked to risk appetite, which is a metric chosen by the board of directors, in accordance to the risk capacity of the firm. Hence, it is important to implement good control functions for each stage of the model lifecycle.

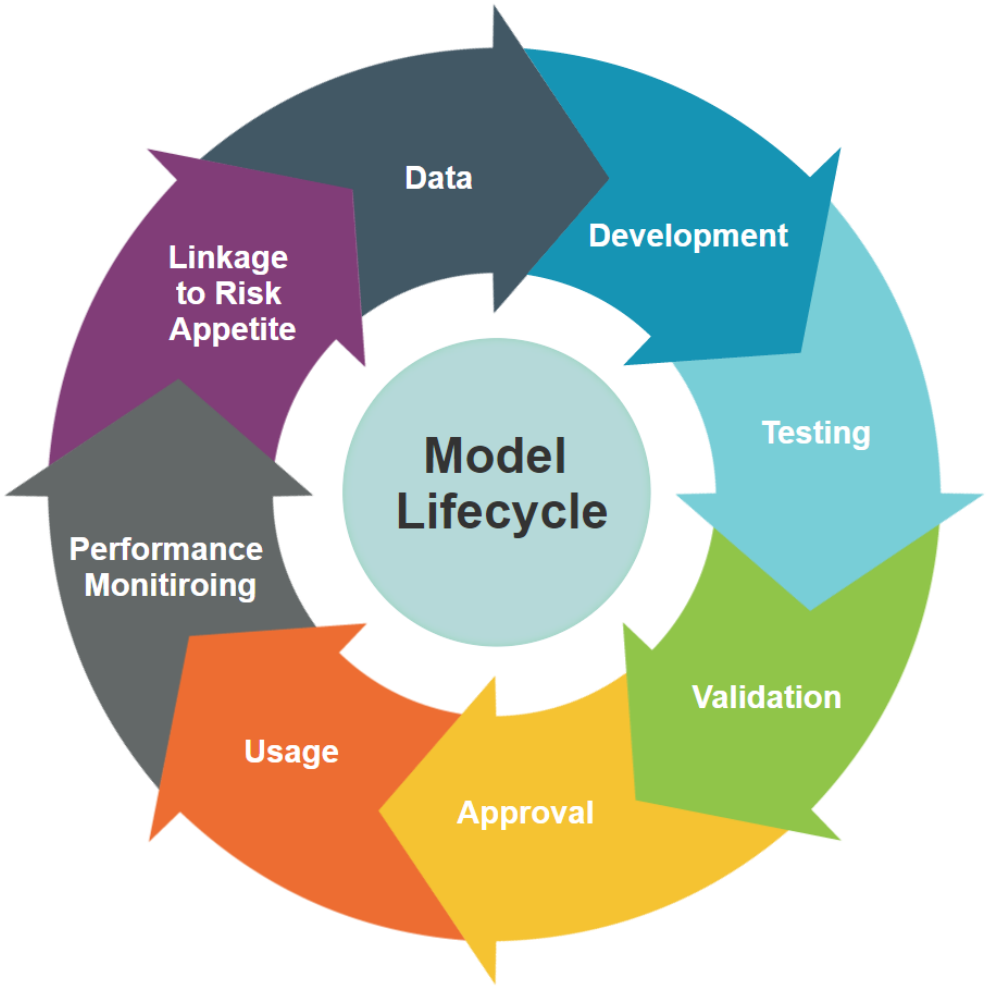


Figure 4: Representation of the core elements of Model Lifecycle.

Focusing on the most recent advances on MRM and on its future in the US, in 2019 the Algorithmic Accountability Act were introduced, requesting all the companies that use AI in order to implement "automated decision system impact assessments and data projection impact assessment" to pay particular attention to problems that may arise form "accuracy, fairness, bias, discrimination, privacy, and security." Thus, the Innovation Pilot Program was

proposed by the OCC in order to help promote FinTech innovation into firms, posing a strong effort to manage the risks that may arise. Furthermore, the new challenge for banks is now on implementing automated MRM processes, in particular focusing on the validation and ongoing monitoring phase.
In conclusion, MRM is now seen by many companies and institutions as a new way to gain competitive advantage. So, where MRM is effectively implemented now the focus is on producing efficiencies to become more competitive.

1.3 Model Risk Management in the EU Framework

Looking at the European regulatory framework regarding MR and MRM, the main reference is CRD IV/CRR, according to which, in Article 3.1.11[20] MR is defined as:

Definition. *Model Risk*. "The potential loss an institution may incur, as a consequence of decisions that could be principally based on the output of (internal) models, due to errors in the development, implementation or use of such models."

As we can see this definition of MR is slightly different from the one provided by OCC-Fed in the US. In fact, in the EU framework MR is treated as a relevant subcategory of operational risk. CRR Article 4.1.52[21] provides the definition of operational risk:

Definition. *Operational Risk*. "operational risk means the risk of loss resulting from inadequate or failed internal processes, people and systems, or from external events, and includes legal risk."

Thus, the concept of operational risk is linked with the risks that arise from the normal activity of the company. In addition, operational risks are part of the Internal Capital Adequacy Assessment Process (ICAAP), which was firstly established by Basel II agreement as part of Pillar I. While Pillar I sets the minimum capital requirement standards, Pillar II is focused on

[20] Directive (EU) No 36/2013 of the European Parliament and of the Council of 26 June 2013 on access to the activity of credit institutions and the prudential supervision of credit institutions and investment firms, amending Directive (EU) 87/2008 and repealing Directives (EU) 48/2006 and 49/2006.
[21] Regulation (EU) No 575/2013 of the European Parliament and of the Council of 26 June 2013 on prudential requirements for credit institutions and investment firms and amending Regulation (EU) No 648/2012 (1), 2013.

the supervisory review process and Pillar III is concerned on transparency and market discipline by increasing the disclosure of capital requirements and methods for risk assessment.

With the ICAAP, financial institutions can autonomously evaluate their capital adequacy requirements so that they can allocate capital and reserves basing their choices on a rigorous risk measurement process. Furthermore, it is also important to notice that if the US deals with MRM via regulation, in the EU this is done via supervision. The European Banking Authority (EBA), with the guideline 2014/13[22], has introduced the Supervisory Review and Evaluation Process (SREP) which consists in an annual evaluation of the data regarding the financial condition of the bank and a risk management assessment. ICAAP flows into the SREP evaluation. SREP enables a coherent analysis of a bank's risk profile and allows it to take the right surveillance measures. Summarizing, the treatment of MRM in the EU is done in accordance with SREP.

Moreover, as shown in *subparagraph 1.1.1*, the ECB has introduced in 2017 the Guide for the TRIM procedure[23]. This is the main cornerstone in the EU regulatory framework as it is specifically focused on internal models assessment, which was a particularly hot topic due to the possibility for banks to develop their own models through the IRB approach provided by Basel II. With this Guide the ECB is trying to harmonize the regulation and the practices regarding MRM in the Euro Area, instilling a common Risk Culture[24]. Thus, according to this Guide, "TRIM will encompass two aspects:

- Compliance with regulatory requirements related to internal models", specifically to "CRD IV/CRR, RTS, EBA guidelines and the approved ECB Banking Supervision manuals and guidelines…"
- "The reduction of unwarranted variability in Risk Weighted Assets (RWA) as it relates to internal model outcomes…"

However, there is no compulsory guideline on how to explicitly treat MR; TRIM only drafts the boundaries of a sound MRM framework but does not impose a specific solution. Thus, I

[22] EBA/GL/2014/13, Guidelines on Common Procedures and Methodologies for the Supervisory Review and Evaluation Process (SREP), 2014.

[23] European Central Bank, Guide for the Targeted Review of Internal Models, February 2017.

[24] BCBS d328 Guidelines on Corporate Governance Principles for Banks defines Risk Culture as: "a bank's norms, attitudes and behaviours related to risk awareness, risk taking and risk management and controls that shape decisions on risks. Risk culture influences the decisions of management and employees during the day-to-day activities and has an impact on the risks they assume."

want to present below some possible answers to this problem. Following the ideas proposed by Bellof and When (2018), we can evaluate MR in three different ways that compare the Risk Absorbing Capacity (RAC[25]) available with the capital reserves devoted to ICAAP purposes. The main issue is whether the bank should consider the uncertainty derived from the model usage as impacting the risk side, therefore the risks individuated for ICAAP purposes, or whether it should consider it as a capital reducing component, affecting the RAC of the bank.

The first approach individuates MR as a separate risk type, that is introduced into ICAAP risks classes. As a consequence the risk capital devoted to ICAAP purposes is increased by the amount devoted to MR. In this fashion, it is crucial to compare the required capital reserves for ICAAP with the available RAC, of course hoping that the available RAC is greater than the required reserves, otherwise severe measures should be taken by the bank to increase the capital reserves. This is shown in Figure 5. Considering MR as a separate risk category has the advantage of being clearly identifiable. However, the main downside of this approach is that creating a general MR "box", that aggregates MR arising from many different risk categories (that may have different units of measure), can be complex and dangerous because it may produce biases that would affect Model Risk estimation causing a wrong capital allocation.

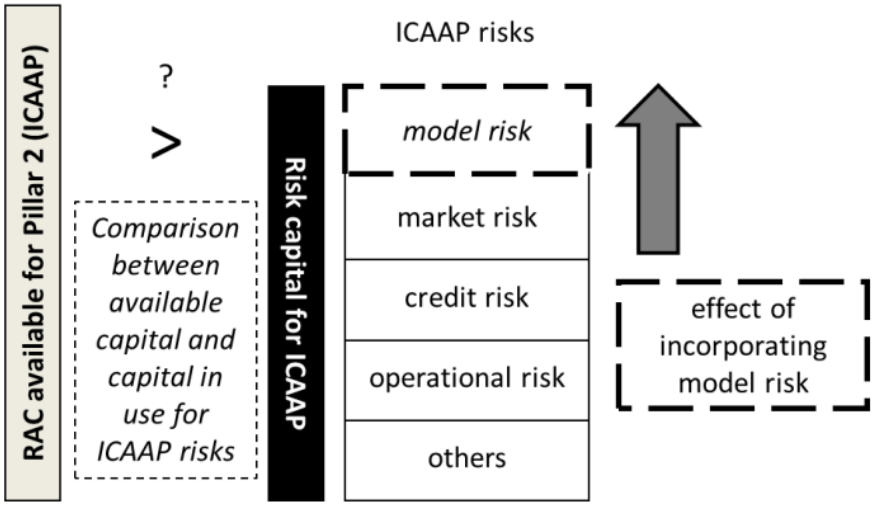


Figure 5: Representation of Model Risk as a separate risk category. Source: Bellof and When (2018).

[25] BCBS d328 Guidelines on Corporate Governance Principles for Banks defines Risk Capacity as: "the maximum amount of risk a bank is able to assume given its capital base, risk management and control measures, as well as its regulatory constraints."

To overcome this drawdown we have to move to the second approach, shown in Figure 6, that instead considers MR as a part of the existing typologies of risk. In particular, MR is no longer an heterogeneous, standing alone risk category that includes MR arising from many different sources, but instead it is divided into the three main risk classes: market risk, credit risk and operational risk. This produces the same effect as the previously mentioned approach: it increases capital reserves for ICAAP purposes, overcoming the main drawback of the first approach. However the main challenge of this method is to define an effective procedure to allocate the additional capital due to MR to each specific risk class. In fact, citing Bellof and When, "a simple pro-rata allocation is not necessarily adequate."

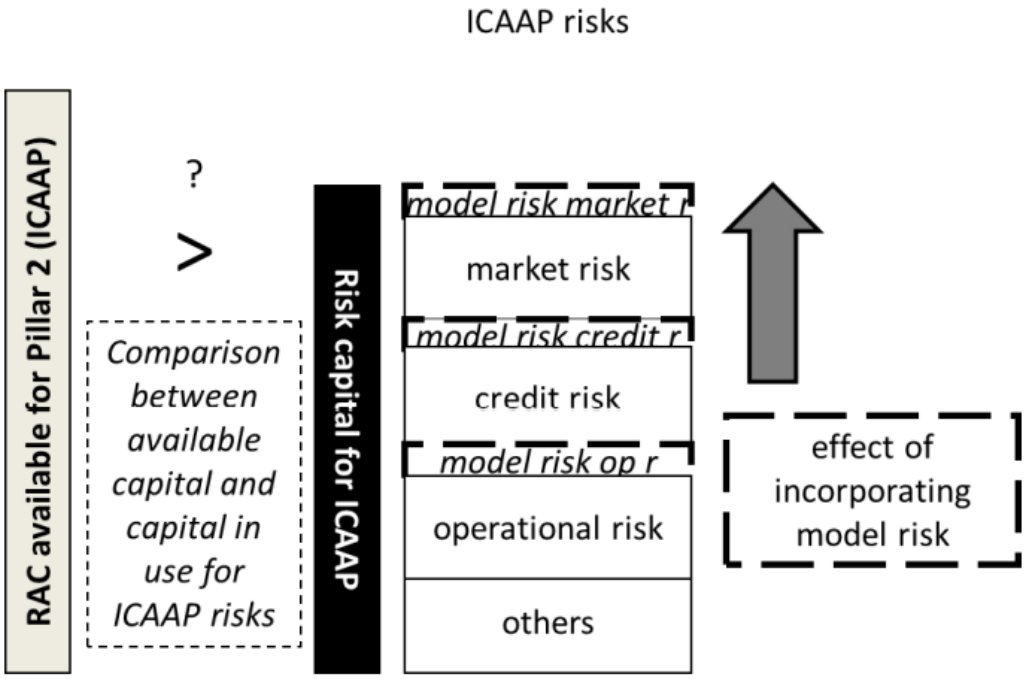


Figure 6: Representation of Model Risk as a part of the existing risk categories. Source: Bellof and When (2018).

Nonetheless, the last approach, shown in Figure 7, individuates MR as a separate capital buffer that is lowering directly the RAC available for the bank instead of increasing the capital reserves for ICAAP, as the two approaches before suggested. Because the RAC is lowered also the risk appetite of the bank has to be reduced. According to the authors, the main advantage of this approach is that it "facilitates the targeted management of the original risks associated with business activities."

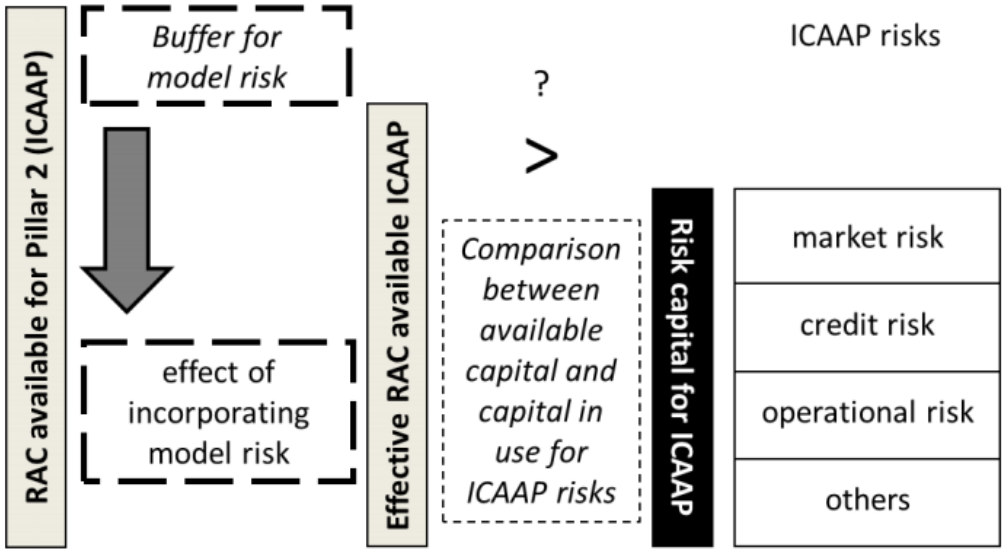


Figure 7: Representation of Model Risk as a separate capital buffer. Source: Bellof and When (2018).

Another typical approach, adopted in many other fields to deal with risk management in general, is the three lines of defence approach. This strategy was proposed for the first time by the BCBS d328 guideline. Looking at the European framework for example, this approach is used in preventing banks or financial institutions from breaching the Union law. The role of each line is to ensure the compliance with regulations, directives and guidelines at different levels of the organization. However, a three lines of defence approach can be also applied to deal with MR. The three lines consist of:

- Business units;
- Control function & Internal validation function;
- Internal audit function.

The first line has to ensure that the model complies with business requirements. It also has to ensure that the documentation regarding the model exists and that it is sufficiently detailed. The second line has to deal with the control and validation procedures, it has to perform tests and monitor effectively each step of the model lifecycle. Finally, the internal audit function has to measure the overall performance of the MRM framework put in place, providing adequate feedback to developers and users. This approach allows to effectively divide the roles and responsibilities regarding MRM. Looking at an example, if an error arises in the raw dataset used to build the model, or the wrong data is extracted from the database, the responsibility to address this problem is of the first line of defence, that should have put in

place adequate quality checks of the data at business units level. If a model is applied outside the scope it was designed for, the responsibility to address this problem is in the first and foremost of the audit function, that should perform the assessment of the effectiveness of model use. This approach should work in tandem with the Model Lifecycle, covering the risks that arise at each step of the life of the model.

Every approach, both in the US and EU framework, I have presented since here is more qualitative than quantitative and it is focused on establishing a sound MRM framework. However, the main drawdown of many qualitative approaches is that it is sometimes harder to provide a proper, unequivocal measure to assess its contribution to MR. A quantitative approach is preferable. In the next paragraph I will present two quantitative approaches to deal with MR deriving from the use of risk measures.

1.4 A Quantitative Approach to Model Risk

As I have pointed out before, neither the US nor the EU regulator have established an explicit method for the quantification of model risk. Yes, guidelines help to institute a sound MRM framework but no reference is made for quantitative methods to measure MR.

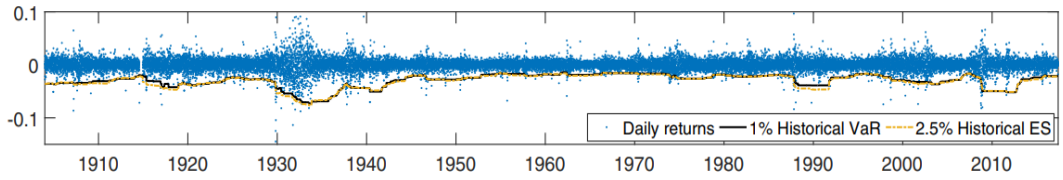


Figure 8: DJIA index daily returns plotted against daily historical 1% VaR and 2.5% ES estimates, from 28/12/1903 to 23/05/2017. Source: Lazar and Zhang (2019).

Since the aim of my work is to provide a method to try to quantify the amount of MR deriving both from the application of the correct risk measure and from the choice of the most appropriate and efficient algorithm[26] in the portfolio selection problem[27], in this paragraph I

[26] The two algorithms considered, the Particle Swarm Optimization and the Grey Wolf Optimization, will be presented in chapter 3.

[27] The portfolio selection problem is a constrained optimization problem that consists in choosing the best allocation for investor's money (the one that maximizes her expected utility) among a possible set of securities and by minimizing the risk deriving from this allocation. The theory of portfolio selection problem will be presented in detail in chapter 2.

will focus on risk measures. More specifically, I will consider the Value at Risk (VaR[28]) and the Expected Shortfall (ES[29]), and I will present a review of the most advanced techniques used in the literature to assess the MR associated with these two measures. Figure 8, taken from Lazar and Zhang (2019), shows the historical 1% VaR (dark blue line) and 2.5% ES (yellow dotted line) of the DJIA index in the period ranging from 28/12/1903 to 23/05/2017. As it can be seen, the blue dots falling below the two lines represent extreme losses that exceeded the predicted VaR and ES. For example, a big clustering of blue dots can be noted at the end of the 1920s, coinciding with the Wall Street crash and the beginning of the Great Depression. Another big clustering can be detected in the late 2000s, corresponding to the Great Financial Crisis.

Referring to Boucher et al. (2014), in this context we can define MR as "the uncertainty created by not knowing perfectly the data generating process (DGP)."

The two main approaches provided by the literature to deal with MR of risk measures are:

- The Bayesian model averaging;
- The worst-case approach.

The first approach, applied in the financial field by Brock et al. (2003), roughly consists of taking the average of the output of all the possible models that can be applied to solve a particular problem. Following the idea of Hoeting et al. (1999), the main issue is that when a manager selects a model to use, she may commit the mistake to assume that "the selected model had generated the data." If she does so, she "ignores the uncertainty in model selection" and this could lead to "overconfident inference and decisions that are more risky than one thinks they are." This issue can be solved by averaging the outcome of all the possible models that can be plausibly applied to the problem, providing a more stable estimate. However, two main drawbacks can be identified in using this technique: the first one is that the averaging approach requires to determine a priori a probability to be attached to each model. Secondly, it requires a huge computational effort as Markov Chain Monte Carlo (MCMC[30]) algorithms should be used.

[28] The VaR (that should not be mistaken as the acronym of the variance) is the maximum potential loss that derives from the holding of a financial asset, given a common holding period (usually 1 or 10 days) and a stated confidence level (usually 95% or 99%). An advantage in using VaR is that it expresses the risk in monetary terms.

[29] The ES is the average potential loss in which an investor can incur, given a common holding period and a stated confidence interval.

[30] MCMC methods are a particular category of algorithms for the sampling of probability distributions based on the construction of a Markov chain. Using these types of algorithms,

The second approach, proposed by Kerkhof et al. (2010), is more in line with the current practices in dealing with risk management. The idea of Kerkhof et al. is to include MR into risk measure (VaR and ES) calculation by adding an adjustment (called bias function) to the theoretical risk measure for each component of MR, through the computation of the worst case of these components. In particular, they divided MR into three elements:

- Estimation risk;
- Misspecification risk;
- Identification risk.

Estimation risk is linked with wrong or inaccurate estimates of the parameters of the model. Misspecification risk arises from the wrong model specification, while identification risk is concerned with the selection of the correct model for the task, because even very similar models with only few changes in the structure or in the assumptions can lead to very different results. The sum of these three components constitute the total MR. Then, they perform a backtest analysis to assess the accuracy of the correction, requiring that the corrected risk measure should pass three statistical hypothesis tests to assess if three required conditions that a good risk forecast should possess are not violated. According to Boucher et al. (2014) a good risk forecast should meet the following requirements:

- Expected frequency of violations;
- Absence of violation clustering;
- Suitable magnitude of violations.

Given the fact that when applying a risk measure such as the VaR or the ES we have to choose a confidence level α[31], the expected frequency of violations ensures that the number of wrong forecasts made though those risk measures are in line with the confidence level required. The absence of violation clustering ensures that wrong forecasts are not subsequent to each other, otherwise this would highlight that the model is not performing well, since there is a sort of dependence in the errors. Finally, the magnitude of violations ensures that when the forecast is wrong it is still in line with distributional assumptions of the model. Thus, when it is wrong the magnitude of the error should not exceed the expected one, according

it could be difficult to determine a priori how many steps are required for the process to converge, with an acceptable error, to the solution of the problem.

[31] The confidence level is the range of values between which it is estimated, with a given probability, that the true value of a random variable will lie.

to the model. Following this approach, hypothesis tests designed to deal with these three conditions are performed.

In the following subparagraph I will go more in detail presenting the approach of Boucher et al. (2014) for VaR and of Lazar and Zhang (2019) for ES.

1.4.1 Measuring Model Risk of Risk Measures

Having in mind that the final output of a VaR or ES model is a risk forecast, the approach proposed by Boucher et al. (2014) and of Lazar and Zhang (2019), respectively for VaR and ES consists in dynamically adjusting the risk forecast of these two models through a correction of the estimated value of the risk measure. This correction is done by looking at historical errors of the model, and by "using them to dynamically adjust the future forecasts" (Boucher et al., 2014).

The literature regarding MR for risk measures, as Boucher et al. and Lazar and Zhang do, focuses mainly on two of the three aspects of MR presented above: estimation risk and misspecification risk. See for example Escaciano and Olmo (2009, 2010 and 2011) in which they study backtesting procedures considering the effect of estimation risk. Alexander and Sarabia (2012) proposed another technique to quantify MR associated to VaR based on a correction on a maximum relative entropy based criterion.

I will now present the theoretical foundation of Boucher et al. and Lazar and Zhang approaches.

We can define the VaR, given a distribution function *F* and a confidence level α, as:

$$VaR_t(\alpha) = -inf\{q: F(q) \geq \alpha\} \tag{1.1}$$

where q is the quantile of F. In the same way we can define ES as:

$$ES_t(\alpha) = \frac{1}{\alpha}\int_0^{\alpha} VaR_t\,(u)\,du \tag{1.2}$$

As we can see by this formula, ES depends on VaR, and thus if the VaR is biased this error will also affect the ES forecast. Now, assuming that the Data Generating Process (DGP) is known and also the value of the true parameters are known (θ_0), we can define the theoretical VaR (ThVaR) and the theoretical ES (ThES) as follows:

$$ThVaR(\theta_0, \alpha) = -q_\alpha^F = -F_\alpha^{-1} \tag{1.3}$$

and

$$ThES(\alpha) = \frac{1}{\alpha}\int_0^{\alpha} ThVaR(\theta_0, u)\, du. \tag{1.4}$$

However if we assume that the DGP is known but the values of the true parameters are unknown, we have a discrepancy between the theoretical VaR and the estimated VaR caused by the estimation error. Thus, a bias function is introduced. The same happens for the ES. The relationship that links ThVaR with the estimated one is:

$$ThVaR(\theta_0, \alpha) = VaR\left(\hat{\theta}_0, \alpha\right) + bias\left(\theta_0, \hat{\theta}_0, \alpha\right) \tag{1.5}$$

where $\hat{\theta}_0$ is the estimate of θ_0, that are the real parameters. $bias\left(\theta_0, \hat{\theta}_0, \alpha\right)$ is the bias function, which is function of the real parameters, the estimated parameters and the confidence level α. It is a measure of the correction needed for the estimated VaR to be equal to the theoretical one.

From (1.5) which we can derive the expected bias of the estimated VaR as:

$$E\left(bias\left(\theta_0, \hat{\theta}_0, \alpha\right)\right) = ThVaR(\theta_0, \alpha) - E\left(VaR\left(\hat{\theta}_0, \alpha\right)\right) \tag{1.6}$$

In the same fashion we can rewrite the relationship between ThES and the estimated one as:

$$\frac{1}{\alpha}\int_0^{a} E\left(bias\left(\theta_0, \hat{\theta}_0, u\right)\right) du = ThES(\theta_0, \alpha) - E\left(ES\left(\hat{\theta}_0, \alpha\right)\right) \tag{1.7}$$

Nevertheless, for a more realistic scenario, the true DGP should be unknown as well. This complicates the scenario, making it impossible to directly measure MR. As a consequence, now it is impossible to perfectly correct the estimated risk measure for the biases. The idea of Boucher et al. (2014) and of Lazar and Zhang (2019) is to adjust the forecasts of the two models looking at model historical performances, so that a correction for future forecasts can be achieved through historical errors. Thus, they suggest implementing an imperfect corrected model that should be close enough to the theoretical one, and that is certified to have passed some statistical checks regarding the three properties that a good risk forecast should possess mentioned above.

Considering Boucher et al. (2014) approach we have that the imperfect model adjusted VaR (IMAVaR) is:

$$IMAVaR\left(\hat{\theta}_1, \alpha\right) = EVaR\left(\hat{\theta}_1, \alpha\right) + adj\left(\theta_0, \theta_1, \hat{\theta}_1, \alpha\right) \tag{1.8}$$

in which $EVaR\left(\hat{\theta}_1, \alpha\right)$ is the estimated VaR and $adj\left(\theta_0, \theta_1, \hat{\theta}_1, \alpha\right)$ is the minimum correction needed, so that:

$$IMAVaR\left(\hat{\theta}_1, \alpha\right) = sup\{VaR(\alpha)^*\} \tag{1.9}$$

where $VaR(\alpha)^*$ represents a set of corrected VaR. In this way the $IMAVaR(\hat{\theta}_1, \alpha)$ is the upper limit VaR that can be accepted by the portfolio manager. Thus, it is the less conservative VaR that is ensured to have passed the tests for the three properties that a good risk forecast should possess. Citing Boucher et al. (2014) "the IMAVaR$(\hat{\theta}_1, \alpha)$ corresponds to a model risk robust VaR that serves to calculate the correction we apply to the Estimated VaR." Thus, even if not optimally corrected (there still exist a discrepancy between the THVaR and the imperfectly corrected one) the IMAVaR allows to get a good estimate of the correct one, in the sense that it is ensured to possess all the three properties of a good risk forecast.

Furthermore, to enhance the quality of IMAVaR a backtesting procedure is implemented and it is required that IMAVaR must pass three statistical tests that are presented in the next section.

On the other hand, following the approach of Lazar and Zhang (2019) for the ES we have that the imperfectly corrected estimate that have passed backtesting ($ES^B_{i,j}$) is:

$$ES^B_{i,j}(\hat{\theta}_1, \alpha) = ES_i(\hat{\theta}_1, \alpha) + C^*_{i,j} \tag{1.10}$$

where the minimum correction $C^*_{i,j}$ is given by:

$C^*_{i,j} = min\{C_{i,j} | ES_{j,t}(\hat{\theta}_1, \alpha) + C_{i,j}$ passes the ith backtest, $t = 1, \dots, T, C_{i,j} \geq 0\}$,

in which the correction $C_{i,j} = (\theta_0, \theta_1, \hat{\theta}_1, \alpha)$ is the minimum to be made in order to have the ES passing each of the three backtests.

1.4.2 Backtesting VaR and ES

Recalling that a good risk forecast should meet the three following requirements, expected frequency of violations, absence of violation clustering in the tails and suitable magnitude of violations, now I am going to present the backtesting procedure used by Boucher et al. (2014) and Lazar and Zhang (2019).

A number of tests can be carried out to verify these three conditions. Referring to the VaR framework, we can define a test to measure the frequency of violations by using the unconditional coverage test, proposed by Kupiec (1995), which is a likelihood ratio (LR) test. This test is based on the comparison between the observed and expected amount of violations. With respect to equation (1.8) we can define the hit variable $I_t^{EVaR}(\alpha)$, which is obtained by the ex post observations of $EVaR(\hat{\theta}_1, \alpha)$, as:

$$I_t^{EVaR}(\alpha) = \begin{cases} 0 & if\ r_t < -EVaR\left(\hat{\theta}'a\right)_{t-1} \\ 1 & otherwise \end{cases} \tag{1.11}$$

where r_t is the return at time *t*. If $I_t^{EVaR}(\alpha)$ is assumed to be iid, then the total number of violations of VaR, Hit_t^{EVaR}, follows a Binomial distribution ($B(T,\alpha)$) and is defined as:

$$Hit_t^{EVaR}(\alpha) = \sum_{t=1}^{T} I_t^{EVaR}(\alpha) \sim B(T,\alpha) \tag{1.12}$$

and the LR test statistic ($LRfreq$) under the null hypothesis has the following asymptotic distribution:

$$\begin{aligned} LRfreq^{I_t^{EVaR}} &= 2\{log[\hat{\alpha}^{T_1}(1-\hat{\alpha}^{T-T_1})] - log[\alpha^{T_1}(1-\alpha^{T-T_1})]\} \\ &\rightarrow \chi^2(1) \end{aligned} \tag{1.13}$$

Looking at the test for the absence of violation clustering, as shown in Christoffersen (1998), we can apply the following LR test:

$$LRind^{I_t^{EVaR}} = 2\left[logL^{I_t^{EVaR}}(\pi_{01},\pi_{11}) - logL^{I_t^{EVaR}}(\pi,\pi)\right] \rightarrow \chi^2(1) \tag{1.14}$$

Finally, for testing the magnitude of violations the approach proposed by Berkowitz (2001) can be followed. We have to apply the following LR test:

$$LRmag^{y_{t+1}} = 2\left[L_m^{y_{t+1}}(\mu,\sigma) - L_m^{y_{t+1}}(0,1)\right] \rightarrow \chi^2(2) \tag{1.15}$$

To sum it up, a strong VaR estimate, that is ensured to pass the three properties of a good risk forecast, should pass each of the three tests:

$$\begin{cases} LRfreq^{I_t^{EVaR}} \rightarrow \chi^2(1) \\ LRind^{I_t^{EVaR}} \rightarrow \chi^2(1) \\ LRmag^{y_{t+1}} \rightarrow \chi^2(2) \end{cases} \tag{1.16}$$

Thus, the final objective, with respect to equation (1.8), is to find the minimal adjustment (q^*) for $IMAVaR$ to make it passess all these tests. Considering the sequence of forecasts $\left\{VaR\left(\hat{\theta},\alpha\right): t = [1,\dots,T]\right\}$ it is possible to built a set of values $q \in \mathbb{R}$ such that the sequence $\left\{VaR\left(\hat{\theta},\alpha\right) + q : t = [1,\dots,T]\right\}$ passes all the three backtests. Denoting the set of feasible adjustments $A_T(\alpha)$, the optimal adjustment is:

$$q_T^* = arg \underbrace{min}_{q \in A_T(\alpha)} \{q\} \tag{1.17}$$

However, since the estimation of the ES depends crucially on the estimation of the VaR, backtesting the ES is more difficult than for the VaR. In fact, if the VaR estimation is biased

this will also affect the correction made to ES. Following the work of Du and Escanciano (2016), we can define the test statistic for the expected number of violation test (unconditional coverage test) as a simple t statistic:

$$U_{ES} = \frac{\sqrt{n}\left({}^{1}/_{n}\sum_{t-1}^{n}\hat{H}_t(\alpha) - {}^{\alpha}/_{2}\right)}{\sqrt{\alpha\left({}^{1}/_{3} - {}^{x}/_{4}\right)}} \sim N(0,1) \tag{1.18}$$

where $\hat{H}_t(\alpha)$ are the estimated cumulative violations that can be written as:

$$\hat{H}_t(\alpha) = \frac{1}{\alpha}(\alpha - \hat{u}_t)|(\hat{u}_t \leq \alpha) \tag{1.19}$$

where $\hat{u}_t$ is the estimated probability level corresponding to the estimated distribution function ($\hat{F}_t$) of the daily returns (r_t) that can be written as:

$$\hat{u}_t = \hat{F}\left(r_t, \Omega_{t-1}, \hat{\theta}_1\right) \tag{1.20}$$

in which Ω_{t-1} is the set of the available information at time $t-1$ and $\hat{\theta}_1$ are the estimated parameters.

Du and Escanciano (2016) again, propose another test to assess jointly the expected frequency of violations and the absence of violation clustering, with a conditional coverage test. They have to test both hypotheses jointly since as said before backtesting ES is more complicated than for VaR. The general test statistic is:

$$C_{ES} = \frac{n^3}{(n-1)^2}\frac{\left(\sum_{t=2}^{n}\left(\hat{H}_t(\alpha) - {}^{\alpha}/_{2}\right)\left(\hat{H}_{t-1}(\alpha) - {}^{a}/_{2}\right)\right)^2}{\left(\sum_{t=1}^{m}\left(\hat{H}_t(\alpha) - {}^{\alpha}/_{2}\right)\left(\hat{H}_t(\alpha) - {}^{\alpha}/_{2}\right)\right)^2} \sim \chi^2(1) \tag{1.21}$$

Finally, Acerbi and Szekely (2014) propose a Z_2 test to test jointly the expected frequency of violations as:

$$Z_2 = \sum_{t=1}^{T}\frac{r_t I_t}{T\alpha ES_{\alpha,t}} + 1 \tag{1.22}$$

Where I_t is a dummy that equals to 0 if the VaR forecast is not violated and 1 otherwise. One particular thing of this Z_2 test is that it is a non-parametric test. Thus, in order to be implemented it only needs the magnitude of VaR violations ($r_t I_t$) and the predicted Expected Shortfall ($ES_{\alpha,t}$). In general, if the Z_2 is greater than the test statistic the model is rejected. Acerbi and Szekely found that the critical value of the test statistic, for $\alpha = 0{,}05$, is -0.7.

Chapter 2

The Portfolio Selection Problem

The aim of this chapter is to provide the basic concepts for the empirical work that I am going to present in the final chapter. Here I will introduce the portfolio selection problem, presenting the Modern Portfolio Theory (MPT), proposed in 1952 by the Nobel Prize winner Harry Markowitz, and discussing its characteristics, highlighting the main unrealistic assumptions upon which this model is based. Thus, in order to overcome these limitations and to deal with a more realistic model, I will introduce mixed-integers constraints that provide an indirect effect on transaction costs, giving a more realistic valence to the problem, but that make the optimization problem harder to be solved. Finally, I will introduce two advanced risk measures, the Value at Risk (VaR) and the Expected Shortfall (ES), that enables to overcome another limitation of the Markowitz model, that uses the variance instead. While the variance is considered a classical risk measure, the VaR can be considered a pre-modern one and the ES a modern one, since it complies with all the coherence axioms proposed by Artzenr et al. (1999).

2.1 Modern Portfolio Theory

Considering the microeconomic theory, investing is the final step of the Consumer theory. An individual that at time t consumes only a part of its capital, maximizing her expected utility function, has the opportunity to shift the remaining part for future time instants, say t_1, and so she has to invest in one or more risky assets. By choosing to invest she creates a portfolio, that from an operative point of view can be represented as a vector of dimensions $N \times 1$:

$$\boldsymbol{x}' = (x_1, x_2, \dots, x_N)$$

in which N indicates the number of securities and $x_1, x_2, \dots, x_N$ represents the percentage of capital allocated to each security. The first model developed to determine in a rational way

the composition of a portfolio is the Markowitz model (in "Portfolio Selection", 1952), for which Harry Markowitz in 1990 won the Nobel Prize for Economic Sciences. This model, known as the Mean-Variance model, is the base of the Modern Portfolio Theory.

2.1.1 Assumptions of the Markowitz Model

Before going in more detail with this model, it is important to present the main assumptions on which it is based:

- Frictionless markets, according to which there are no transaction costs, no taxation effects and all the assets are perfectly divisible.
- No institutional restrictions regarding the short selling.
- Each investment opportunity is measured over the same holding period.
- Perfect information, investors know perfectly the probability distribution of the assets' returns.
- Investors are price takers, in the sense that they cannot affect the probability distribution of returns.
- Investors are rational and risk averse, their aim is to maximize their rate of returns and if they have to choose between two titles with the same expected return, they always choose the one with the lowest risk.
- The criteria used to make decisions is the mean-variance one.

2.1.2 Portfolio Selection Process

It is possible to identify three different steps in the portfolio selection process:

1. Identification of an appropriate way to measure the uncertainty of an investment decision, which means to identify an appropriate measure of risk and expected returns.
2. Establish a criterion that enables to distinguish between efficient and inefficient investment choices, in order to identify the best portfolios.
3. Selection of a portfolio by the investor in accordance with her risk aversion. This is done by maximizing the investor's expected utility function.

With respect to the first step, according to Markowitz we can identify two tools: the mean and the variance of the rate of return of an investment. Reasonably the higher the mean (the

expected rate of return) the better will be for the investor; and vice versa the lower the variance the better.

Let X be a discrete random variable, where x_i with $i = 1, \dots, N$ is the i-th realization of X and p_i with $i = 1, \dots, N$ is the probability of occurrence of x_i. Thus, the expected value of this variable is:

$$E(x) = \sum_{1=1}^{N} x_i p_i \tag{2.1}$$

and the variance is:

$$Var(x) = \sum_{i=1}^{N} [x_i - E(x)]^2 p_i \tag{2.2}$$

Whereas, if X is a continuous random variable with probability density function $f(\cdot)$, then the expected value can be written as:

$$E(x) = \int_{-\infty}^{+\infty} t f(t)\, dt \tag{2.3}$$

and the variance is:

$$Var(x) = \int_{-\infty}^{+\infty} \big(t - E(x)\big)^2 f(t)\, dt \tag{2.4}$$

Since a portfolio can be defined as a weighted average of individual securities, we can write the expected return of a portfolio as:

$$E(R_p) = \sum_{i=1}^{N} x_i r_i = r_p = \boldsymbol{x'r} \tag{2.5}$$

where r_i is the expected rate of return of the i-th asset and $\boldsymbol{x'r}$ is the vectorial notation. Moreover the variance of R_p can be written as:

$$Var\big(R_p\big) = \sum_{i=1}^{N} x_i^2 \sigma_i^2 + 2 \sum_{i=1}^{N} \sum_{j=1+1}^{N} x_i x_j \sigma_{i,j} = \sigma_p^2 = \boldsymbol{x'Vx} \tag{2.6}$$

where $\sigma_{i,j} = \rho_{i,j} \sigma_i \sigma_j$ is the covariance between the i-th and the j-th asset and $\boldsymbol{x'Vx}$ is the vectorial notation.

Now, with respect to the second step, the efficiency criterion proposed by Markowitz is mean-variance-based. If we consider two random variables X and Y, we can state that X dominates Y, and so it is preferred to Y, if the following conditions are respected:

- $E(R_X) \geq E(R_Y)$;
- $Var(R_X) \leq Var(R_Y)$;
- At least one of these two inequalities is satisfied in the strong form.

This criterion enables the investor to compare in the mean-variance sense, part of investment choices. However, it is possible that between two options none dominates the other. In this case, where the mean-variance criterion is not able to distinguish, and each of them lie in the so called efficient frontier. Thus, the efficient frontier consists of all the efficient portfolios that given an expected rate of return minimize the variance, or that given a certain level of variance maximize the expected return. As a consequence, any rational investor will choose a portfolio that lies in the efficient frontier given his preferences.

Merton (1969) proposed the following optimization problem for the construction of the efficient frontier for N assets:

$$\begin{matrix} min_{x_1,\dots,x_N} & \boldsymbol{x'Vx} \\ s.t. & \begin{cases} \boldsymbol{x'r} = \pi \\ \boldsymbol{x'e} = 1 \end{cases} \end{matrix} \tag{2.7}$$

where:

- π is the investor's desired rate of return;
- $\boldsymbol{V}$ is the $N \times N$ variance-covariance matrix;
- $\boldsymbol{e}$ is a vector of ones of order N.

As it can be seen, these optimization problem has two constraints: $\boldsymbol{x'r} = \pi$ is the profitability constraint, according to which the investor chooses the desired rate of return; while $\boldsymbol{x'e} = 1$ is the budget constraint, that states that all the initial capital has to be invested among the possible assets.

In order to find the solution of this optimization problem two conditions have to be met: the $\boldsymbol{V}$ matrix of variances and covariances must be singular[32] and positive definite[33]; and the expected return of at least two assets must be different, i.e. $r_i \neq r_j$ for some $i, j = 1, \dots, N$. If these conditions are met a unique solution exists, and to find it we have to use the Lagrangian function:

$$L = \boldsymbol{x'Vx} - \lambda_1(\boldsymbol{x'r} - \pi) - \lambda_2(\boldsymbol{x'e} - 1) \tag{2.8}$$

[32] For a matrix to be singular it must be a square matrix whose determinant is equal to zero. Thus, a singular matrix is non invertible.

[33] The matrix $\boldsymbol{V}$ is positive definite if for any non null vector M, $\boldsymbol{x'Vx} > 0$.

where λ_1 and λ_2 are the Lagrange multipliers. Then, by imposing the first order partial derivatives equal to 0 and by solving the following equation system we find the solution of the optimization problem, x^*:

$$\begin{cases} \frac{\partial L}{\partial x} = 2\boldsymbol{x}'\boldsymbol{V} - \lambda_1 \boldsymbol{r}' - \lambda_2 \boldsymbol{e}' = 0 \\ \frac{\partial L}{\partial \lambda_1} = -\boldsymbol{x}'\boldsymbol{r} + \pi = 0 \\ \frac{\partial \mathrm{L}}{\partial \lambda_2} = -\mathbf{x}'\mathbf{e} + 1 = 0 \end{cases} \tag{2.9}$$

$$x^* = \frac{(\gamma \boldsymbol{V^{-1}} \boldsymbol{r} - \beta \boldsymbol{V^{-1}} \boldsymbol{e})\pi + (\alpha \boldsymbol{V^{-1}} \boldsymbol{e} - \beta \boldsymbol{V^{-1}} \boldsymbol{r})}{\alpha\gamma - \beta^2} \tag{2.10}$$

where α, β and γ are respectively: $\alpha = \boldsymbol{r}'\boldsymbol{V}^{-1}\boldsymbol{r}$, $\beta = \boldsymbol{r}'\boldsymbol{V}^{-1}\boldsymbol{e} = \boldsymbol{e}'\boldsymbol{V}^{-1}\boldsymbol{r}$ and $\gamma = \boldsymbol{e}'\boldsymbol{V}^{-1}\boldsymbol{e}$.

To sum it up, the mean, the variance and the standard deviation of the efficient portfolio are respectively:

$$E(Rp^*) = r_{p^*} = \boldsymbol{x}^{*\prime}\boldsymbol{r} = \pi \tag{2.11}$$

$$Var(Rp^*) = \sigma_{p^*}^2 = \boldsymbol{x}^{*\prime}\boldsymbol{V}\boldsymbol{x}^* = \frac{\gamma\pi^2 - 2\beta\pi + \alpha}{\alpha\gamma - \beta^2} \tag{2.12}$$

$$StdDev(Rp^*) = \sigma_{p^*} = \left(\frac{\gamma\pi^2 - 2\beta\pi + \alpha}{\alpha\gamma - \beta^2}\right)^{1/2} \tag{2.13}$$

As it can be seen easily by equation (2.12), the variance function describes a parabola in the mean-variance plane, whereas equation (2.13) describes a hyperbole in the mean-standard deviation plane. Figure 9 gives a representation of the frontier both in the mean-variance and in the mean-standard deviation plane in the case of $N = 15$ and no riskless asset. Every red dot represents a specific asset. It can be noticed that the efficient part of the frontiers is the one that lies above the vertex of the parabola and the hyperbole. Every point in this part of the frontiers constitute an efficient portfolio that can be chosen by the investor. Another interesting property is that by forming a portfolio of these fifteen assets it is possible to achieve a combination of those with lower risk that the risk of the less risky asset: this is shown graphically by the portion of the frontier that goes from the vertex until the first dot on the left (that represents the security with lower variance/standard deviation). This is the so called risk-reduction principle.

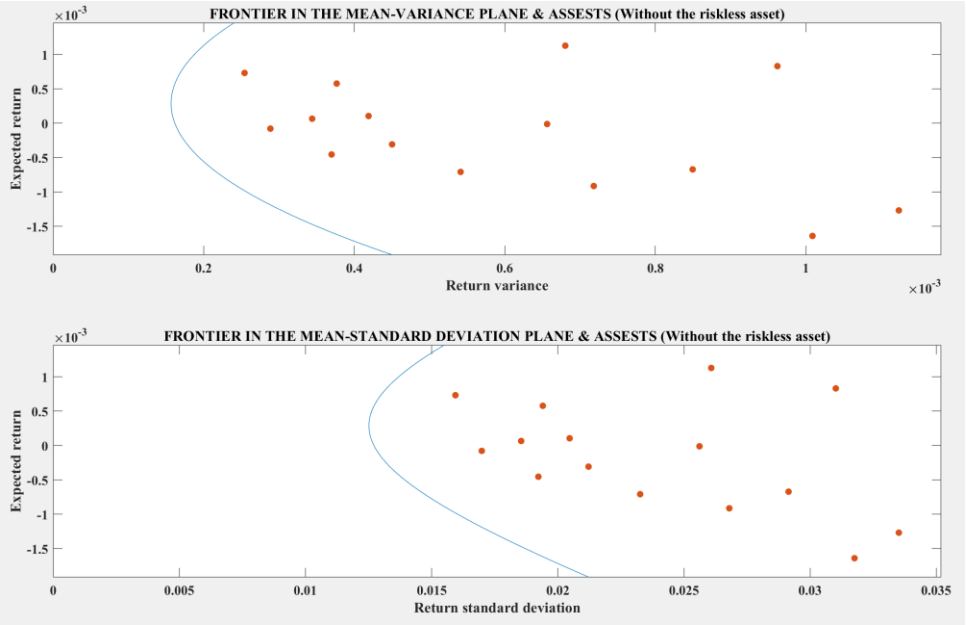


Figure 9: Graphical representation of the frontier in the mean-variance and mean-standard deviation plane for $N = 15$ and no riskless asset.

Furthermore, Figure 10 represents again the frontier both in the mean-variance and in the mean-standard deviation plane in the case of $N = 15$, but this time with the possibility of investing also in a riskless asset. By allowing to invest in a risk-free security, the frontier is now tangent to the y axis, because now it is possible to choose an allocation with no risk (thus with a certain return).

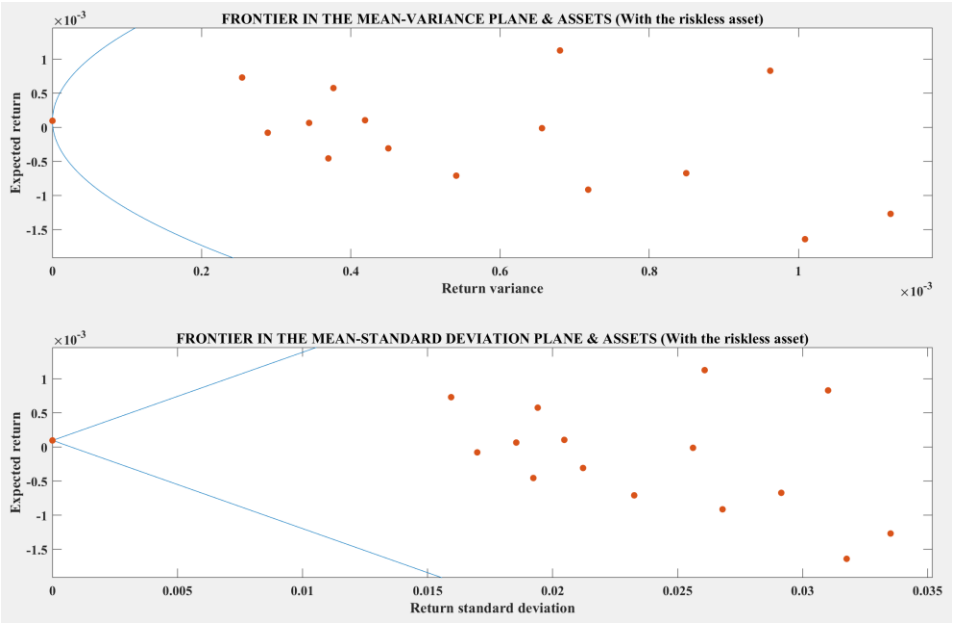


Figure 10: Graphical representation of the frontier in the mean-variance and mean-standard deviation plane for $N = 15$ plus one riskless asset.

2.1.3 Expected Utility Theory

With respect to the third point of the portfolio selection process, once the efficient frontier has been identified, now it is crucial to determine a criterion for the investor to pick her optimal portfolio among all the possible efficient ones considering her risk aversion. This is done in accordance with the Von Neumann and Morgenstern (1947) Expected Utility Theory. In this way the investor aims to choose the efficient portfolio which maximizes her expected utility function. A general utility function is consistent with the mean-variance dominance criterion if and only if one of the following two mutually exclusive conditions is respected:

- The expected utility function is quadratic.
- The joint probability distribution of returns is elliptical[34].

Thus, a generally adopted utility function is the quadratic utility one, U, with the following form:

$$U(R_p) = R_p - \frac{\alpha}{2} R_p^2 \tag{2.14}$$

where $R_p \leq \frac{1}{\alpha}$ is the random variable indicating the returns of the portfolio and $a > 0$ represents the risk aversion coefficient of the investor. The first addendum of (2.14) is linked with the profitability, while the second contains the subjective valuation of risk of the investor (through the a parameter). Looking at the first derivative of this function it is possible to notice that it is positive; while looking at the second derivative it is always negative by construction. Therefore, the properties of this utility function are that it is strictly increasing, since the aim of the investor is to maximize its profits, and concave, stating that to an increase of the capital would correspond a minor increase in the utility function the more is the capital owned by the investor.

Furthermore, calculating the expected value of (2.14) we have:

$$E[U(R_P)] = \mu_P - \frac{\alpha}{2}(\mu_P{}^2 + \sigma_P^2) \tag{2.15}$$

where μ_P and σ_p^2 are respectively the mean and the variance of the portfolio returns.

Furthermore, the generic indifference curve[35] of the quadratic utility function can be derived from equation (2.15):

[34] As for example the Normal and the Student-t distribution.

[35] An indifference curve is a curve connecting all the points with the same expected utility for the investor. So if two investment opportunities lie in the same curve they might present different risk-return characteristics, but they have the same expected utility.

$$k = \mu_p - \frac{\alpha}{2}(\mu_P{}^2 + \sigma_P^2) \tag{2.16}$$

where k is a fixed value of expected utility. By reformulating (2.16), it is possible to see that it is nothing more that the equation of a circumference with centre in $\left(0, \frac{1}{\alpha}\right)$ and with radius $\sqrt{\frac{1-2\alpha k}{\alpha^2}}$:

$$\sigma_P^2 + \mu_P{}^2 - \frac{2}{\alpha}\mu_p + \frac{2}{\alpha}\, k = 0 \tag{2.17}$$

Expression (2.17) is a convex function in the mean-variance plane increasing with respect to k. Hence, on the same indifference curve lie infinite investment choices characterized by the same expected utility. In conclusion, since we have demonstrated that the mean-variance criterion is consistent with the expected utility theory, it can be useful to collocate the efficient frontier in the same plane with the indifference curve. As shown in Figure 11, the optimal portfolio $\boldsymbol{X}^*$ is the tangency point between the efficient frontier and the highest indifference curve.

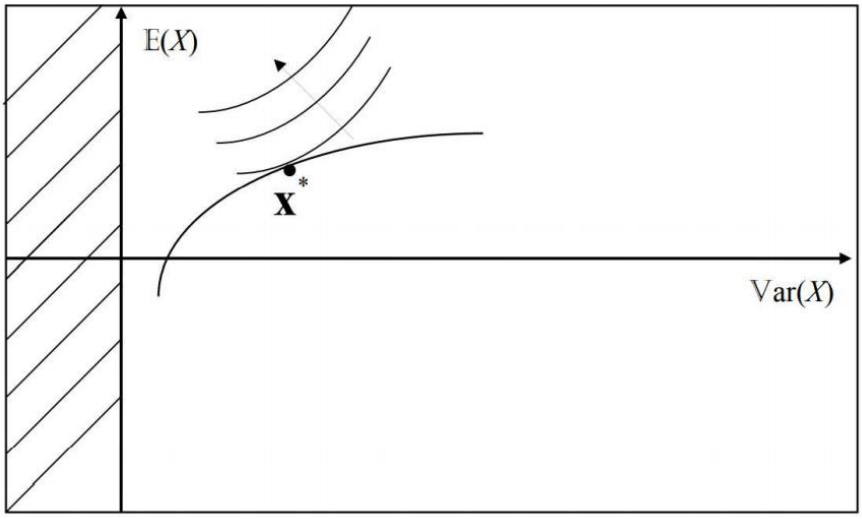


Figure 11: Optimal portfolio $\boldsymbol{X}^*$, represented as the tangency point between the efficient frontier and the highest indifference curve.

2.2 Improvements of the Markowitz Model

To improve the classic portfolio selection model, it is necessary to implement in the selection models some realistic aspects, generally in the form of constraints. It is important to remember from equation (2.7) that the two constraints in the Markowitz model are related to

the budget and the desired return. Improvements can be made by adding to these two the so called mixed-integers constraints. In linear programming[36] it is possible to identify three macro classes of problems:

- Linear continuous programming (LP);
- Integer linear programming (ILP);
- Mixed-integer linear programming (MILP).

To LP class belong all those optimization problems in which the variables are continuous, and that can take all the values inside their domain of existence. On the contrary, to ILP class belong all those problems in which all the variables are restricted to be integers, and that can take only integers values inside their domain of existence. Finally, to MILP class belong those problems in which only some of the variables are required to be integer, and thus contain both continuous and integer variables. Mixed-integers constraints belong to this third class. As we will see in section *2.3.1*, one integer constraint and two continuous constraints will be added to the portfolio selection problem.

The three main families of mixed-integers constraints are:

- Constraints to the minimum lots that must be negotiated in each transaction. For example it can be allowed to negotiate only integer numbers of lots of a specific security. A lot is generally composed by a specific number of shares, for example 100 or 200, but it can be greater depending on the asset.
- Constraints to the maximum positive integer number of different assets that can be negotiated. For example, a portfolio must be composed of at most 10 securities.
- Constraints to the minimum positive integer number of different lots that must be negotiated for a specific security. For example, when negotiating a security, the minimum number of lots to be bought or sold is 2.

By introducing these constraints, it is possible to give a greater operative valence to the portfolio selection problem: a portfolio manager who is facing the problem of choosing among hundreds of securities the ones in which to invest, might decide to limit the minimum and the maximum number of securities to be selected as well as the minimum and maximum percentage of capital to invest in each of them. In this way, mixed-integers constraints have

[36] Linear programming is a special case of mathematical programming that is focused on studying algorithms for the solution of linear optimization problems that are subject to linear equality and/or inequality constraints.

an indirect impact on controlling transaction costs, because considering too many securities would be expensive both in terms of time and money. For an example see Jobst et al. (2001), Andramonov and Corazza (2003) or Corazza and Favaretto (2007).

However, one implication of the introduction of these types of constraints is that the efficient frontier may lose some of its important properties: it could be that the frontier is no longer a continuous function, causing problems when choosing the optimal portfolio. Nonetheless, the main implication is that the optimization problem becomes more complex to be solved in terms of computational and time requirements. In fact, checking the eligibility of mixed-integers constraints in a mathematical programming problem is an NP-complete[37] problem. Furthermore, solving such a mathematical programming problem is an NP-hard[38] problem. As a consequence, the search for the optimum solution via exact methods, that guarantees that the best admissible solution is found, might not be possible and the employment of approximated techniques, that provide a good approximation of the best solution, might be needed. As pointed out by Richards and How (2005), for solving MILP problems there exists two main procedures:

- Branch and bound;
- Heuristics and metaheuristics.

Branch and bound method consists in decomposing the original problem into more sub-problems that are easier to be solved. In this way, a decision-tree structure is built in which to each node corresponds a sub-problem which is related though a branch to the problem from which it derives. The main advantage of this technique is that it is an exact method that allows one to find the global optimal solution among the admissible ones. However, it may not always be applicable. The other category includes heuristics and metaheuristics procedures. In general, they are approximated techniques that allows to find acceptable solutions that converge towards the global optimum, albeit they do not guarantee to reach it. As stated in Richards and How (2005), "heuristics methods are typically based on some random search procedure. This is attractive for non-convex problems, as the randomness reduces the susceptibility of the methods to getting stuck in local minima." More details about heuristics and metaheuristics will be given in chapter 3, where I will present the two algorithms applied to the portfolio selection problem, used for experiment with real data presented in chapter 4.

[37] NP-complete are problems that requires a huge effort and amount of time to be solved.
[38] NP-hard are problems at least as complex as NP-complete, but they can be even more.

2.3.1 Cardinality Constraints

An example of mixed-integer constraints are the cardinality constraints. They can be introduced to tackle indirectly the problem of transaction cost management by limiting the minimum and the maximum number of assets in which the portfolio can invest. More formally we have:

$$K_n \leq \sum_{i=1}^{N} z_i \leq K_m \quad with\ 1 \leq K_n \leq K_m \leq N \tag{2.18}$$

Where K_n is the minimum number of assets in the portfolio; K_m is the maximum number; N is the total number of titles in which we can invest; and $z_i \in [0,1]$ is a binary variable that assumes value of 1 if the *i*-th asset is included in the portfolio and 0 otherwise.

Furthermore other constraints, called bound constraints, that limits the percentage amount of wealth invested in each asset can be added:

$$z_i n_i \leq x_i \leq z_i m_i \quad with\ 0 \leq n_i \leq m_i \leq 1 \tag{2.19}$$

Where n_i is the minimum fraction of capital invested in the *i*-th asset and m_i is the maximum one. In order to guarantee that the two constraints presented above are compatible with each other, the following two inequalities for n_i and m_i must be satisfied:

$$n_i \leq \frac{1}{K_n} \tag{2.20}$$

$$m_i \leq \frac{1}{K_m} \tag{2.21}$$

To sum it up, now the optimization problem for the portfolio selection can be written as:

$$\begin{cases} min_{x,z}\ \rho(R) \\ s.t. \begin{cases} \hat{R} \geq \pi \\ \sum_{i=1}^{N} x_i = 1 \\ K_n \leq \sum_{i=1}^{N} z_i \leq K_m \\ z_i n_i \leq x_i \leq z_i m_i \quad \forall i \\ z_i(z_i - 1) = 0 \quad \forall i \end{cases} \end{cases} \tag{2.22}$$

where $\rho(R)$ is a generic risk measure and $z_i(z_i - 1) = 0$ is the reformulation of the integrality constraint $z_i \in [0,1]$.

2.3 Modern Risk Measures

As proposed by various authors like Szegö (2002), Ortobelli et al. (2005), Rachev et al. (2008) a risk measure can be seen as a function $\rho: X \rightarrow \mathbb{R}$ which assigns to the random variable X a non negative real number.

Thus, another improvement to the Markowitz model is to substitute the variance, which is a classical risk measure, with a more efficient one. As said before, the variance is not sensitive to tail events and it is not able to distinguish between positive variations and negative ones as pointed out by Rachev et al. (2008). In fact, when considering the risk associated with a portfolio it is more important to have a clearer understanding of the probability of downside movements (losses).

2.4.1 The Value at Risk

The Value at Risk (VaR), introduced for the first time by the J.P. Morgan in 1994[39] and successively adopted by the Basel Committee to calculate capital reserves, can be a valuable alternative to overcome these problems. The VaR, considered as a pre-modern risk measure, can be defined as:

Definition. *Value at Risk*. Given defined time horizon (called holding period) and a confidence level $\alpha \in [0,1]$, the VaR is the maximum potential loss that the portfolio can suffer in the α% of cases during the holding period.

Usually the confidence level for α are 95% and 99%. Figure 12 shows the concept of Var looking at the probability density function and at the probability distribution function of portfolio returns. As it can be seen the VaR is identified as the quantile of a probability distribution that satisfies the following property: $P(L > VaR_\alpha) = 1 - \alpha$. Where L is the profit-loss distribution. Another formulation, sometimes preferred, of the VaR is:

$$VaR_\alpha(X) = -\inf_x \{x | P(X \leq x) \geq \alpha\} \qquad (2.23)$$

The main advantage of the VaR is that it expresses the risk in monetary terms. Furthermore, it is able to capture tail events better than the variance. Nonetheless, it has some limits when the distribution of returns is not Gaussian. As shown by Embrechts (2000), Hu and Kercheval (2007) and Natarajan et al. (2008) when normality assumption is made, the portfolio fund by

[39] RiskMetricsTM - Technical Document, J.P. Morgan/Reuters, Fourth Edition (1996).

minimizing the VaR coincides with the one fund following Markowitz assumptions (thus by using the variance). However it has been shown that normality assumptions may distort VaR estimates, leading to an underestimation of the real risk. In addition, the VaR does not allow to estimate losses greater than the VaR itself. Furthermore, regarding risk measure properties (presented in more detail in the next section) the VaR does satisfy the translation invariance, monotonicity and positive homogeneity properties. However VaR lacks sub-additivity, in fact we can have that:

$$VaR_{\alpha}(X+Y) > VaR_{\alpha}(X) + VaR_{\alpha}(Y) \tag{2.1}$$

Thus, the risk deriving from a portfolio of two assets X and Y can in some cases be greater than the sum of the risk associated with each asset. This is not in line with the diversification assumption.

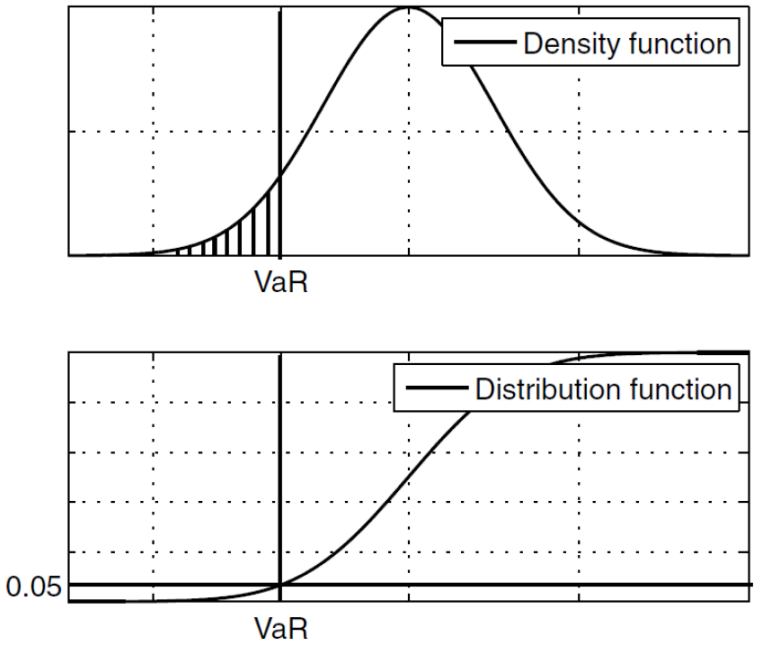


Figure 12: Graphical representation of the Value at Risk looking at the density function and at the cumulative distribution function of portfolio returns.

2.4.2 Coherent Risk Measures

Artzner et al. (1999) proposed the four properties that a coherent risk measure should possess. All the (coherent) risk metrics that respect these properties are known as modern risk measures. More in detail they are:

- Translation invariance: considering the random return X and a risk free asset and combining the two in a portfolio, its risk will decrease proportionally to the amount α invested in the safe asset.

$$\rho(X + \alpha) = \rho(X) - \alpha \tag{2.2}$$

- Sub-additivity : it is at the basis of portfolio diversification, meaning that the risk associated to the combination of two risky assets X and Y is lower than the sum of the risks associated with each of them.

$$\rho(X + Y) \leq \rho(X) + \rho(Y) \tag{2.3}$$

- Positive homogeneity : if the investment in X is augmented of a component $\lambda > 0$, then the risk associated with that investment will increase accordingly.

$$\rho(\lambda X) = \lambda\rho(X) \tag{2.4}$$

- Monotonicity : given two risky assets X and Y, if $X \leq Y$ then the risk associated to X will be lower than the risk associated to Y.

$$X \leq Y \rightarrow \rho(X) \leq \rho(Y) \tag{2.5}$$

2.4.3 The Expected Shortfall

The Expected Shortfall (ES), also known as Conditional Value at Risk (CVaR[40]), is considered a modern risk measure, as it satisfies all the 4 coherence axioms proposed by Artzner et al (1999). In financial terms it can be interpreted as the expected value of the losses conditioned to the fact that these losses exceed the VaR.

Definition. *Expected* Shortfall. Given the random variable X with probability density function $F(X)$ so that $P(X \leq x)$, and by introducing the inverse of $F(X)$ as $F^{-1}(\alpha) = \inf_x\{x|F(x) \geq \alpha\}$, that coincides with (2.18), then the ES can be expressed as:

$$ES_\alpha(X) = -\frac{1}{\alpha}\int_0^\alpha F^{-1}(p)\,dp \tag{2.6}$$

As it can be seen it is the average value under the left tail of the profit-loss distribution. Whereas, Acerbi and Tasche (2002) provided a more formal definition of ES even if it does

[40] By Rockafellar and Uryasev (2002).

not change substantially form (2.24). Given the profit-loss distribution X and specified the holding period, the ES with significance level $\alpha \in [0,1]$ is defined as:

$$ES_\alpha(X) = -\frac{1}{\alpha}\{E[X\mathbf{1}_{\{X\leq x^\alpha\}}] - x^\alpha(P[X \geq x^\alpha] - \alpha)\} \tag{2.7}$$

where x^α is the VaR_α.

Figure 13 provides a representation of the ES, and as it can be seen this risk measure is able to take into account all the losses exceeding the VaR, capturing in a better way the risk of tail events. For example Liang e Park (2007) provided evidence of the superiority of the ES with respect to the VaR as a measure of downside risk. Furthermore the ES is a universal risk measure, as it can be applied to every financial instrument.

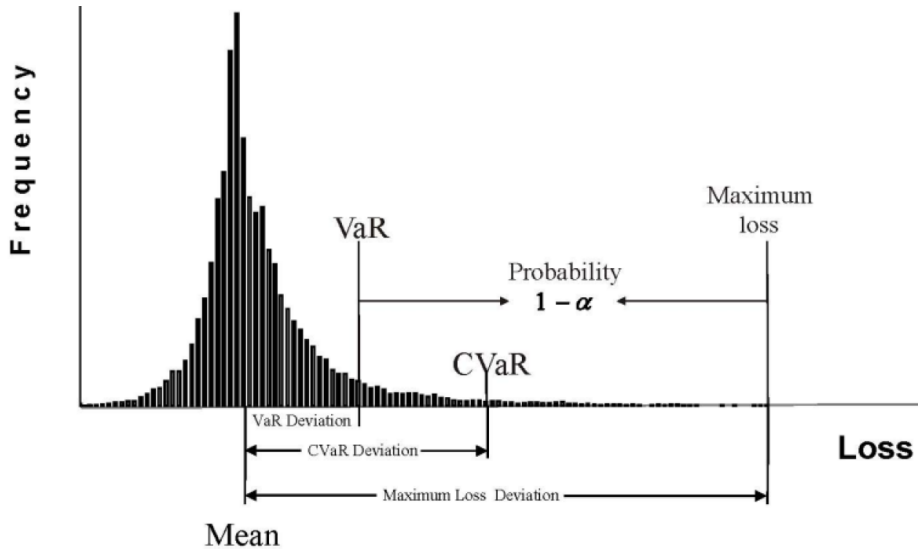


Figure 13: Graphical representation of the Expected Shortfall and comparison with the Value at Risk.

Chapter 3

Metaheuristics: a comparison between PSO and GWO

In this chapter I will present the two metaheuristics that will be applied in chapter 4 in order to solve the portfolio selection problem. As we have seen in the previous chapter, if we consider a more realistic problem it may not be possible to find a solution with exact techniques, thus metaheuristics algorithms are required. Here, after a brief introduction on metaheuristics' world in which I discuss the core concepts of exploration and exploitation and Swarm Intelligence, I will present two of these techniques in detail. More precisely, I will examine the Particle Swarm Optimization, which is one of the most applied and trusted techniques, and I will compare it with the recently developed Grey Wolf Optimization, which is one of the fastest growing algorithm of these years with a lot of applications in many fields. The aim of this comparison is to critically assess analogies and differences of these two models in order to provide a starting point for the measurement of Model Risk in chapter 4.

3.1 From Heuristics to Metaheuristics

Both Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO) are part of approximated algorithms[41] family, and more in detail they belong to metaheuristics algorithms. These techniques allow to solve Combinatorial Optimization (CO) problems, such as the portfolio selection problem even in the case of introduction of mixed-integers constraints, that as we have seen in the previous chapter makes the procedure to find a solution an NP-hard problem.

[41] With respect to computer science, approximation algorithms are efficient techniques that allow us to find approximate solutions to complex optimization problems (especially NP-hard problems). These techniques do not guarantee the optimal solution, but they provide a good approximation to that in a reasonable amount of time.

A Combinatorial Optimization problem can be defined as $P = (S, f)$ which is composed of:

- A set of variables $X = (x_1, \dots, x_n)$;
- Variables' domain $D_1, \dots, D_n$;
- An objective function f to be minimized, where $f: D_1 \times \dots \times D_n \rightarrow \mathbb{R}^+$;
- A number of constraints on the set of variables X;
- A set of all possible combinations given by:

$$S = \{s = [(x_1, v_1), \cdots (x_n, v_n)] | v_i \in D_i, \quad s\ satisfies\ all\ the\ constraints\} \tag{3.1}$$

where S is the solution space (or the search space) and every element in it is a candidate solution.

In case of minimization, in order to solve this CO problem we have to find the optimal solution $s^* \in S$ that minimizes the objective function so that:

$$f(s^*) \leq f(s) \forall s \in S \tag{3.2}$$

where s^* is the global optimum solution of (S, f).

Two methods can be identified to solve CO problems: exact algorithms or approximated algorithms (or heuristics). As pointed out in paragraph 2.2, exact methods are generally not suitable for the solution of NP-hard problems, since they provide a solution in an unreasonable amount of time[42].

By contrast, heuristic algorithms are able to approximately solve NP-hard problems providing a good approximated solution in a reasonable amount of time. Thus, even if they do not guarantee the optimal solution, they are characterized by flexibility, with a range of

[42] It is important to give the definition of temporal complexity: in computer science, the temporal complexity of an algorithm is a measure of the time required by the algorithm to be fully executed. In order to define a unit measure for the time requirement, the notation $O(f(n))$ can be introduced, where O is the order of magnitude. Thus, if we consider an initial number of input n, the complexity of an algorithm is $O(f(n))$ if the number of operations to fully execute it is a multiple of $f(n)$. In particular, an algorithm in polynomial time is an algorithm that has a time to be fully executed with an upper bound limit of the dimension of the input of the algorithm, so $T(n) = O(n^k)$, where k is constant. $T(n)$ is defined as the maximum amount of time required for any input of dimension n.
All the problems for which it is required an algorithm that needs a polynomial number of operations to solve them, belong to the P (Polynomial) problems class. Instead, for NP (Non deterministic Polynomial) problem class there are no algorithms of Polynomial class that are able to solve them. Finally, NP-hard problems are another class that has a complexity at least equal to the one of NP problems.

possible applications that is wider than exact methods. The etymology of the word "heuristic" comes from ancient Greek:

Definition. *Heuristic*. It comes from the Greek verb *heuriskein* that means "to find" or "to discover". In particular, heuristic algorithms are methods used to find a solution to complex optimization problems in a reasonable amount of time through a trial and error technique, that however do not guarantee that the best solution is reached.

It is possible to identify three main categories of heuristic algorithms:

- *Constructive algorithms*. They aim to build an admissible solution by aggregating, iteration after iteration, some adjustments to a starting solution. They generally follow a three step approach:
    1. Initialization: a starting input, which is nothing more than an initial partial solution, is generated.
    2. Selection of a new input: on the basis of a prefixed selection criterion a new input is aggregated to the partial initial solution found in step 1.
    3. Stop criterion: if the resulting solution is admissible the algorithm is stopped, otherwise step 2 is done again.

  The solutions found with this typology of algorithms are strictly related to the initial input and to the mechanism used to update partial solutions. These algorithms are relatively fast, however usually the solutions found with them are less "optimal" than other types of algorithm.

- *Improvement or local search algorithms*. Their mechanism is based on the enhancement of an admissible solution via an iterative procedure. As a consequence, they are usually more precise than constructive algorithms. Their functioning is based on 4 steps:
    1. Initialization: a possible initial solution is generated in order to trigger the search process.
    2. Identification of a criterion that enables to search around the current solution in order to find better ones. Following this criterion, the algorithm searches

in the neighbourhood[43] of the current solution, looking for a better one to be selected.

3. Selection of a new solution: the best solution in the neighbourhood is selected.
4. Stop criterion: If the new solution is better than the initial one step two is done again, otherwise the algorithm stops.

The solutions found with this typology of algorithms are strictly related to the initial input and to the criterion used to identify the neighbourhood.

- *Metaheuristics*. They are enhanced versions of heuristics since they are focused on finding the optimum by searching and exploiting the solution space efficiently and effectively in the most promising areas. As pointed out, constructive algorithms are focused on finding a suitable solution while improvement algorithms start with a suitable solution and then they try to improve it. Metaheuristics, on the other hand, try to enhance the quality of solutions identifiable with constructive and local search techniques in order to avoid local optimums. The Particle Swarm Optimization and the Grey Wolf Optimization belong to this category.

3.1.1 Metaheuristics

The term *"metaheuristic"* was first used by Glover (1986). The word *metaheuristic* is composed of the prefix *meta*, which means "beyond", and *heuristic* which as highlighted earlier means "to find" by trial and error. In the last 30 years the usage of these techniques has skyrocketed, causing some problems when trying to identify a unique definition of metaheuristic. In this thesis I will consider the definition provided by Sörensen and Glover (2013):

Definition. *Metaheuristic*. "A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization

[43] The neighbourhood of an individual (or solution) x is the space around it in which the algorithm is searching when trying to find better solutions to the current one. Every neighbourhood has a peculiar structure that influences the functioning of the algorithm as it states how information is exchanged between agents. Neighbourhood structures will be presented later in this chapter.

algorithms. The term is also used to refer to a problem-specific implementation of a heuristic optimization algorithm according to the guidelines expressed in such a framework."

Thus, if on one hand heuristics are algorithms that return in a reasonable amount of time a solution which is a good approximation of the optimal one, metaheuristics perform a deeper search in the most promising areas of the search space. These techniques allow us to solve any optimization problem more effectively than heuristic algorithms. The main advantages of metaheuristic approaches are their simplicity and flexibility that allow them to be adapted to any sort of optimization problem. Moreover, they are derivative-free methods, since they do not require the calculation of any derivative.
The two main concepts upon which metaheuristics are based are:

- *Exploration,* which refers to the searching phase around solution space for the most promising areas.
- *Exploitation*, which refers to the exploitation of the common experience about the search already done until that moment.

While exploration is linked to an individual behaviour, the single agent looks around for unexplored areas that seem promising to it, exploitation phase is linked to a social behaviour, in which multiple agents use and exchange their past experience. These two concepts are in trade off among them, if for example more emphasis is given to exploration, then the exploitation phase will suffer. Thus, it is crucial to find the right balance between exploration and exploitation: if on one hand it is important to identify quickly the most promising areas of the search space, on the other it is crucial to not waste time on areas already explored or non promising. However, it is right because of this interaction between individual and social behaviour that it is possible to enhance the quality of solutions. This is linked with the concept of Swarm Intelligence (SI) that I will present in the next section. However, briefly, with SI we mean a complex form of intelligence possessed by certain typology of the population of individuals. If these individuals are considered separately they are just capable of a very basic form of intelligence, but if considered jointly they are able to solve complex tasks that require a sophisticated level of intelligence through the exchange of information and cooperation.
Now I provide a brief presentation on the main characteristics according to which metaheuristics can be classified:

- *Nature inspired vs non-nature inspired*. This distinction is linked with the origin of the algorithm. Some metaheuristics are inspired by animals' behaviour observed in

nature: the Particle Swarm Optimization (PSO) is inspired by bird flocks and fish shoals; while the Grey Wolf Optimization (GWO) is inspired by the social hierarchy and hunting behaviour of wolves. Instead, other metaheuristics have nothing to do with *"nature"*, like the Fireworks Algorithms or the Tabu Search. However, it is possible to find in the literature hybrid algorithms that combine *nature* with *non-nature* ones, and that cannot be classified according to these criterions.

- *Population based vs single point search*. Here the difference is between the number of solutions that are treated simultaneously. Population based metaheuristics, when trying to find the optimum, generate and manage a set of solutions, called population. The population is generated randomly and then it is improved iteratively: if better solutions are found they substitute the previous ones. This process is carried out until a stop criterion is reached and hopefully the resulting set of solutions is close enough to the optimal one. On the other hand, single point search metaheuristics take into consideration just one solution and they describe its trajectory in the solution space. At each step the current solution is replaced with a new one which is hopefully a better one. Both PSO and GWO are population based metaheuristics. Examples of single point search algorithms are Simulated Annealing and Tabu Search.
- *Static objective function vs dynamic objective function*. In some algorithms the objective function remains the same in every searching step. Instead, in other metaheuristics the objective function changes while the searching is performed: this helps to prevent the algorithm from being stuck in local optimum areas of the searching space.
- *One neighbourhood structure vs multiple neighbourhood structures*. Some metaheuristics use just one structure for describing the topological conformation of the neighbourhood, assuming it to be static. Others instead modify dynamically the structure of the neighbourhood with the objective to diversify the search in the solution space.
- *Memory-usage vs memory-less methods*. Memory-usage algorithms keep track of past information (i.e. of past experience) and use it in the searching phase for updating the quality of the current solution. On the other hand, memory-less algorithms acts just like a Markow process[44] where the information used to

[44] A stochastic Markow process is a random process where the probability that determines the transition from the actual stage to the next one is only determined by the actual stage

determine the next step depends exclusively on the current information and not on the past one. However, among memory-usage algorithms it is possible to distinguish between those with long memory and those with short memory. While the first uses synthetic parameters relative to the past searching experience to move accordingly taking into consideration long time memory, the latter uses just recent information.

3.1.2 Swarm Intelligence

SI is a branch of Evolutionary Computation[45], that constitutes the family of algorithms for global optimization that simulates natural and biological phenomena. Beni et al. (1988), in the field of cellular robotic systems, use the term Swarm Intelligence for the first time. With SI they mean the collective behaviour that a group of individuals (swarm) put in place when they interact with each other following some rules.

While the single agent of the swarm may have only a basic form of intelligence, the entire swarm can reach a complex and high level form of intelligence. Thus, even if single agents are not sophisticated, by sharing information and personal experiences, directly or indirectly, with the swarm, they are able to learn and display complex forms of behaviour that enables them to solve hard tasks. It is possible to find a lot of examples of this mechanics in nature: think about insect behaviour, like ants or bees when seeking food, but also of other animals like the fish, birds or wolves. It is well known that wolves display a high level of cooperation when hunting the prey, they respect a set rules and mechanics for the encircling and capturing phase of the hunt. These behaviours are precisely what the GWO metaheuristics pretend to mimic. On the other hand the PSO is inspired by the behaviour of flock of birds and of shoals of fishes. Those groups of animals are able to optimally pursue objectives crucial for their existence, like food search or escape from predators even if the single agent may not be sophisticated.

The two main characteristics of SI are the decentralization and self-organization of the group. With decentralization we mean that in the group there is not any individual that has the task of supervising the functioning of the entire process. Instead, with self-organization we mean

and not by the past one. Thus it is a process with no memory: there is no impact of past information.

[45] Evolutionary Computation is one of the subsets of Computational Intelligence, which can be described as the ability of a computer to learn specific tasks from some data. Other branches of Computational Intelligence are Artificial Neural Networks, Fuzzy logic, Simulated Annealing, Chaos Computation, Tabu Search and Evolutionary Computation.

that the group aims to improve its capabilities by enhancing the organization of its components.

According to Biratti and Dorigo (2007) SI algorithms shows the following three properties:

- *Simultaneous actions*: defines the capacity of the swarm's components to manage a number of tasks in different locations (i.e. in different areas of the solution space) and at the same moment. Thus, it is possible to create auto-organized groups that simultaneously deal with different aspects of a certain problem. For example some agents of the swarm may implement individual behaviours (exploration phase) and others social behaviours (exploitation phase).
- *Scalability*: when there is an increment of the quantity of variables (of single agents of the swarm) the algorithm preserves its properties without having to recalibrate the interactions between particles. In SI algorithms interactions involve only neighbour agents, thus the number of interactions to be managed tend to increase less than proportionally with respect to the number of agents. Thanks to this principle it is possible to increment the performance of the algorithm simply by increasing the dimension of the swarm without the need of reprogramming it.
- *Error tolerance*: derives from the decentralization and self-organization properties of an SI system. Because the swarm is composed of a number of individuals with equal characteristics, where none of them has the role of controlling the swarm's behaviour, it is possible to remedy for their eventual errors by substituting these agents, without causing computational problems.

Furthermore, Kumar et al. (2008) proposed five properties that have to be respected for a swarm to be considered intelligent in the case of optimization problems:

- *Adaptability Principle*: when the behaviour of the swarm may cause computational problems, in terms of not balancing well between exploration and exploitation, it is better to change it.
- *Diverse Response Principle*: every agent when performing its task should not take ways that are excessively narrow.
- *Proximity Principle*: the population has to carry out simple time and space computations.
- *Quality Principle*: every agent should be reactive to different factors that influence the search space.

- *Stability Principle*: the swarm's behaviour has to remain stable and should not react to changes in the environment.

3.2 Particle Swarm Optimization

As pointed out before, PSO is a bio-inspired and population-based metaheuristic introduced for the first time by Kennedy and Eberhart (1995). PSO is a Swarm Intelligence-based algorithm able to solve unconstrained optimization problems and inspired by the behaviour of flock of birds and shoal of fishes. The original idea of Kennedy and Eberhart was to exploit the social interactions between the elements of the swarm rather (SI) than the abilities of a single individual. The current PSO is composed of a swarm of particles that iteratively moves into the searching space with the aim of finding the optimal solution that, depending on the problem, minimizes or maximizes an objective function, called fitness function. Every particle of the swarm uses and exchanges information relative to its best position (called *pbest*) and to the best position of all the other members of the population (called *gbest*). That information directly impacts the direction and velocity of the next movement of the particle. Furthermore, each particle has a memory in which it is able to store that information. In this way the particles are able to calibrate their own behaviour improving their individual search and also the one of the entire swarm.

Look for example at the behaviour of a bird that has to follow certain rules in order to remain in the flock but at the same time can decide to move away from the swarm when it discovers some food. In this way it decides between a social behaviour and an individualistic one, but both of them are aimed to improve the wealth of the flock. Those are exactly the already mentioned concepts of exploration and exploitation.

In the problem that I will present and analyse in the 4th chapter, each particle corresponds to a peculiar portfolio of stocks that is a candidate solution to the problem. Furthermore, since PSO is able to solve only unconstrained problems, I need to reformulate the portfolio selection problem in unconstrained form[46]. Thus, the fitness function, according to which the quality of the solutions is measured, is a function of both the penalty function and the risk

[46] The reformulation of the portfolio selection problem in unconstrained form will be presented in Chapter 4 following the Exact Penalty method. However, it briefly consists in rewriting all the constraints in terms of their violation, and in summing these violations together in the penalty function (that will be added, as another component, to the fitness function).

measure chosen: I will apply both the VaR and ES to each of the two metaheuristics, PSO and GWO.

Going more in detail, each particle is characterized by three vectors of dimensions *d*, where *d* are the dimensions of the searching space. Considering a population of *M* particles in a multidimensional space, at the *k*-th iteration, the three vectors of the *i*-th particle, where $j \in \{1, \dots, M\}$, are:

- $X_i^k \in \mathbb{R}^d$, is the position of the *i*-th particle at the *k*-th iteration represented in a *d-dimensional* space:

$$X_i^k = \left(x_{i,1}^k, x_{i,2}^k, \dots, x_{i,d}^k\right) \tag{3.3}$$

  Thus, this vector will contain the percentage of wealth to be invested in each asset of the portfolio.

- $V_i^k \in \mathbb{R}^d$, is the velocity of the *i*-th particle at the *k*-th iteration represented in a *d-dimensional* space:

$$V_i^k = \left(v_{i,1}^k, v_{i,2}^k, \dots, v_{i,d}^k\right) \tag{3.4}$$

- $P_i \in \mathbb{R}^d$, is the best position (*pbest*) visited by the particle since the *k*-th iteration:

$$P_i = \left(p_{i,1}, p_{i,2}, \dots, p_{i,d}\right) \tag{3.5}$$

Furthermore, each particle knows the global best (*gbest*) position visited by the swarm since the *k*-th iteration, which is $G \in \mathbb{R}^d$:

$$G = (g_1, g_2, \dots, g_d) \tag{3.6}$$

Initially, the *M* particles are generated randomly in the solution space in order to provide a good exploration in the initial steps of the algorithm. At each iteration every particle evaluates the quality of its current position (the quality of the possible solution) by mean of the fitness function, $f\left(X_i^k\right)$. Since in my framework the fitness function evaluates the risk of the portfolio, the aim of each particle is to reduce this function by finding the best position (thus the optimal portfolio). The PSO updates at each iteration the best positions found by each particle, called *personal best*, $f(P_i)$. Then, the algorithm takes also into consideration the best position found by the entire swarm, called *global best*, $f(G)$. Each particle of the swarm determines its successive move considering its current position but also the overall best position of the swarm. This information allows the algorithm to converge quickly to the optimal solution of the optimization problem.

3.2.1 The original structure of PSO

Below it is reported the original structure of PSO algorithm:

1) Initialization of the population of particles, attaching to each of them random position and velocity in the *d-dimensional* searching space.
2) Starting of the loop:
   a) Determination of the fitness function $f\left(X_i^k\right)$ relative to the current position X_i^k of each particle;
   b) If the value of the fitness function of the current position is better than the *personal best*, $f\left(X_i^k\right) \leq f(P_i)$, then the personal best is updated with the current position $P_i = X_i^k$.
   c) Identify the best particle in the neighbourhood, indicated by *G*, which is the one with the smallest value of the fitness function, $f(G)$.
   d) Update the position X_i^{k+1} and the velocity V_i^{k+1} of each particle using the following equations:

$$\begin{cases} V_i^{k+1} = V_i^k + U(0, \Phi_1) \otimes \left(P_i - X_i^k\right) + U(0, \Phi_2) \otimes \left(G - X_i^k\right) \\ X_i^{k+1} = X_i^k + V_i^{k+1} \end{cases} \quad (3.7)$$

   where:

   - $\left(P_i - X_i^k\right)$ is the difference between the best personal position and the current position of the *i-th* particle. It represents the cognitive component, and it is a sort of memory of the particle that allows it to move towards the best region explored (the one with lower fitness value);
   - $\left(G - X_i^k\right)$ is the difference between the best global position of the swarm and the current position of the *i-th* particle. It represents the social component and it makes the particle move towards the best position explored by the swarm;
   - $U(0, \Phi_i)$, with $i \in \{1,2\}$, is a vector of random numbers Uniformly distributed in the interval $[0, \Phi_i]$. Multiplying the cognitive and social component for a random number allows to avoid to be stuck into local optimums;
   - Φ_1 and Φ_2 are two acceleration coefficients. Φ_1 expresses the effect produced by the *pbest* position and it is also called cognitive acceleration

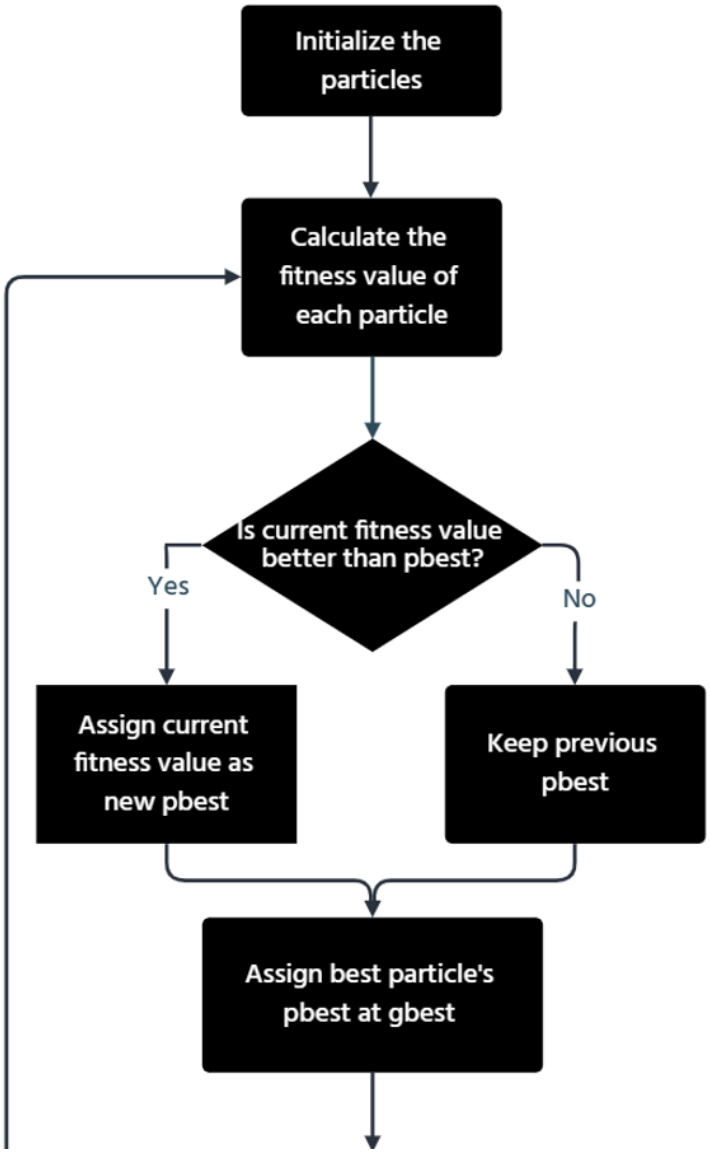
coefficient. Φ_2 expresses the effect produced by the *gbest* position and it is also called social acceleration coefficient. This two coefficients control how each particle moves in the search space;

- $\otimes$ is the tensor product[47].

e) If the prefixed stop criteria is met, like the fitness function has reached a sufficiently good value or the maximum number of iterations is reached, then the algorithm passes to step 3, otherwise it returns to step 2.

3) End of the loop.

PSO algorithm can be represented by the following flowchart:



[47] It is a bilinear operator, a function that combines two vectorial spaces into the same field to produce an element of a third vectorial space, which is linear in every argument.

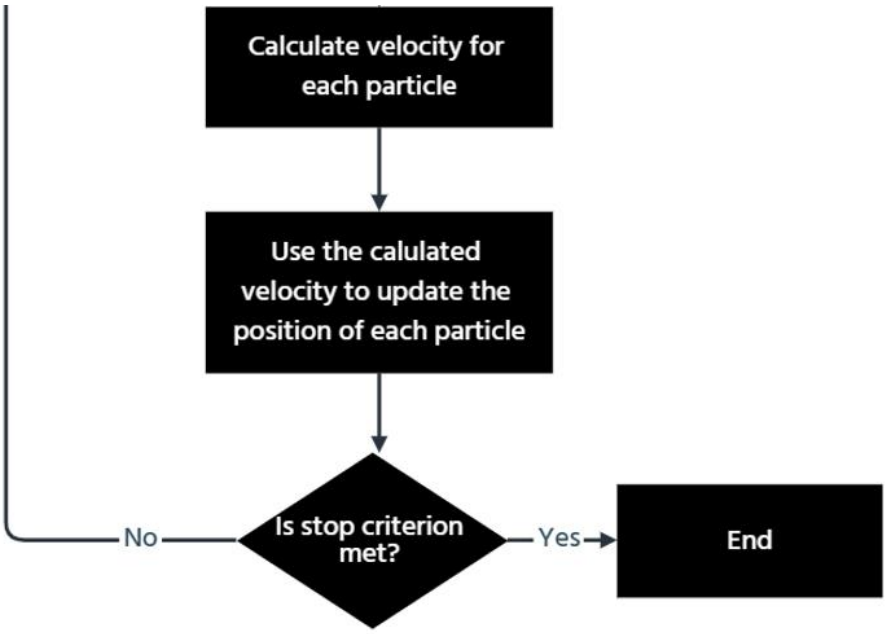


Figure 14: Flowchart of the structure of the Particle Swarm Optimization.

Focusing on the parameters of the PSO, the two acceleration coefficients, Φ_1 and Φ_2, have a huge impact on the dynamic of each particle since they determine the attraction force to the *pbest* and *gbest*. A wrong setting of these parameters could cause the instability of the whole algorithm, determining an uncontrolled increment of the velocities of the particles. In the first study of the PSO they were both set equal to 2, but this did not guarantee the stability of the process as the velocity of the particles might diverge. Thus, an idea was to impose a range of maximum velocity for each particle, so that $V_i^k \in [V_{min}, V_{max}]$. However, because the setting of these parameters influences directly the balance between exploration and exploitation, and there are no specific rule of thumb for the determination of V_{max}, it may become a complex task to set them correctly.

Another parameter that has to be considered and balanced is the number of particles composing the swarm. An excessively large amount of particles would cause a useless computational effort, on the other hand a too small number would undermine the effectiveness of the algorithm in finding the best solution. According to Blackwell et al. (2007) a number between 20 and 50 particles would be the best choice.

To overcome these issues some adjustments of the original version of the PSO were proposed in the literature.

3.2.2 PSO with Inertia Weight

A first improvement to PSO algorithm, called the Inertia Weight, is the one of Shi and Eberhart (1998). They modify the equation that defines the velocity of each particle introducing a new parameter ω as:

$$\begin{cases} V_i^{k+1} = \omega V_i^k + U(0, \Phi_1) \otimes \left(P_i - X_i^k\right) + U(0, \Phi_2) \otimes \left(G - X_i^k\right) \\ X_i^{k+1} = X_i^k + V_i^{k+1} \end{cases} \quad (3.8)$$

where ω is the inertia weight.

The role of ω is to balance between exploration and exploitation, providing more stability to the model and preventing the explosion of the velocity without having to set V_{max}. A great value of ω would increase the influence of the current velocity V_i^k on the future one, V_i^{k+1}. On the contrary, a small value decreases the influence of the current velocity on the next one, giving more weight to the personal best position and global best position.

A criterion that allows to rationally implement the Inertia Weight approach is to assign to ω linearly decreasing value as the number of iterations increases. In fact, in the initial phases of the loop it is better to focus on exploration (by giving a bigger weight to ω) in order to search all around the solution space. Then, as the number of iterations increase, it is better to reduce the value of ω, giving more importance to the exploitation phase, searching around the most promising areas. A possible way to implement such a strategy is the one used by Corazza et al. (2013) in which they set ω as:

$$\omega^k = \omega_{max} - \frac{\omega_{min} - \omega_{max}}{K} k \quad (3.9)$$

where :

- ω_{max} is the fixed maximum value;
- ω_{min} is the fixed minimum value;
- K is the maximum number of iterations;
- k is the current number of iterations.

The general rule is to assign to ω_{max} and ω_{min} respectively values of 0.9 and 0.4. Furthermore, in order to ensure the convergence towards the optimum the following relationship between Φ_1 and Φ_2 must be respected:

$$\frac{\Phi_1 + \Phi_2}{2} - 1 < \omega \quad (3.10)$$

Looking at the literature, Blackwell et al. (2007) investigate the use ω and they find that the performance of the PSO significantly improves with this technique. They also propose to include a random component to ω.

3.2.3 PSO with Constriction Coefficient Factor (CFA)

Another approach, very used in practice, is the one of Clerc and Kennedy (2002). They propose to introduce a coefficient χ that multiplies the entire velocity equation with the aim of preventing the instability of the model guaranteeing the convergence of the particles to the optimum, again without having to set the troublesome parameter V_{max}. Thus, the equation for updating the velocity is:

$$\begin{cases} V_i^{k+1} = \chi(V_i^k + U(0,\Phi_1) \otimes \left(P_i - X_i^k\right) + U(0,\Phi_2) \otimes \left(G - X_i^k\right)) \\ X_i^{k+1} = X_i^k + V_i^{k+1} \end{cases} \quad (3.11)$$

where:

- $\chi = \frac{2}{\Phi - 2 + \sqrt{\Phi^2 - 4\Phi}}$;
- $\Phi = \Phi_1 + \Phi_2 \ \ with \ \ \Phi > 4.$

Contrary to ω in the Inertia Weight approach, that affects only the current velocity, χ affects all the components of the right part of the equation. Clerc and Kennedy poses $\Phi = 4.1$ with $\Phi_1 = \Phi_2 = 2.05$. Thus, by substituting into equation (3.11) we found that $\chi = 0.7298$. As a consequence the current velocity is multiplied by 0.7298, while the two component linked to exploration and exploitation are multiplied by a random number in the interval $[0, 1.496118]$ since $0.7298 \times 2.05 \cong 1.496118$.

It has to be noticed that the Inertia Weight and the CFA variants of the PSO coincides if ω is posed equal to χ and $\Phi_i = \chi\Phi_i$. In this way $\omega = 0.7298$ and the two acceleration coefficients become $\Phi_1 = \Phi_2 = 1.496118$.

3.2.4 Fully Informed Particle Swarm (FIPS)

Finally, the last variant presented is the one proposed by Kennedy and Mendes (2002 and 2003) in which they modify the way in which information is exchanged between particles. According to Blackwell et al. (2007) in the original version of the PSO the only two information exchanged, that modify the velocity equation of each particle, are the current best

position (*pbest*) and the global best position (*gbest*) of the entire swarm. It is clear that part of the available information remains unused during the entire process: for example all the past visited positions of each particle are deleted since only the best visited remains in the particle's memory. Thus, Kennedy and Mendes propose a version of the algorithm in which every particle takes into account all the available information in its neighbourhood. This is done by modifying the way particles interact with each other, thus by modifying their neighbourhood. They propose to change the velocity equation as:

$$\begin{cases} V_i^{k+1} = \chi\left(V_i^k + \frac{1}{\mathrm{K_i}}\sum_{\mathrm{n=1}}^{K_\mathrm{i}} \mathrm{U}(0,\Phi) \otimes \left(\mathrm{P_{nbr_n^i}} \mathrm{X_i^k}\right)\right) \\ X_i^{k+1} = X_i^k + V_i^{k+1} \end{cases} \tag{3.12}$$

where:

- χ is the constriction coefficient;
- Φ is the acceleration coefficient;
- K_i is the number of neighbour of the *i-th* particle;
- $\mathrm{nbr_n^i}$ represents the *n-th* neighbour of the *i-th* particle;
- $\mathrm{P_{nbr_n^i}}$, for each *n*, represents the best position reached by each particle in the neighbourhood of the *i*-th particle.

As it can be noticed, the velocity of the *i*-th particle is updated through the use of a weighted average of the best positions of the neighbour particles. In this way, the information used in the FIPS version of the PSO is more complete because it is provided by each of the neighbouring particles and not only by the *gbest*. Kennedy and Mendes show that by adopting the FIPS it is possible to reach better solutions than the normal PSO with less iterations. However, it can be seen that this version coincides with the original PSO if we impose $K_\mathrm{i} = 2$.

3.2.5 Topologies of the Population

Now that we have introduced the concept of neighbourhood of a particle it is worth to present the main characteristics of some of them. As pointed out in the previous paragraph, the choice of the structure of the neighbourhood is crucial because it determines how the particles exchange information and thus the efficiency of the algorithm. It is possible to distinguish between two main classes of topologies:

- *Static topologies*: their structure does not change as iterations increase. The three main examples of this structure are:
  - *Local best* topology, also called *lbest*. Proposed by Kennedy and Eberhart in (1995) it is characterised by a ring structure (Figure 15) where each particle is connected with the two adjacent ones, and thus exchange information only with them. Even if with this topology the searching process is slower, it results to be less vulnerable than other topologies to the attraction towards local optimums.

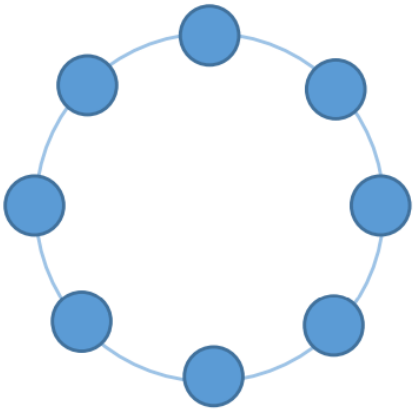


Figure 15: Representation of local best topology.

  - *Global best* topology, also called *glbest*. In this topology each particle is connected with every other particle of the population (Figure 16). It is a so called fully-connected structure in which the entire population is used as a neighbourhood of a particle. Differently from the previous topology, here there are no sub populations (the one formed between a particle and its two neighbours).

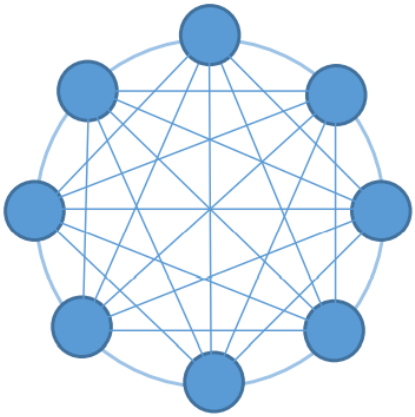


Figure 16: Representation of Global best topology.

The *glbest* topology allows a faster convergence towards the optimum if compared with the *lbest*, however there is a higher risk of convergence towards local optimums. If on one hand, a higher level of connection between particles allows more information to be exchanged, on the other one it slows the performance of the algorithm as the number of particles increases.

- *Von Neuman* topology. It can be considered a half way between the *lbest* and the *glbest*. In this case each particle is linked with the two adjacent ones but also with the ones under and above it. Kennedy and Mendes (2002) have investigated and proved the superiority of this topology on the other two previously presented.

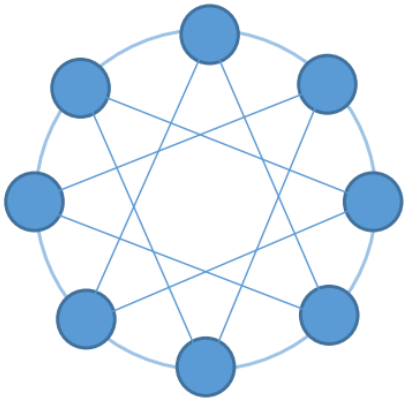


Figure 17: Representation of Von Neuman topology.

❖ *Dynamic topologies*: their structure does change when the number of iterations increases.

- Suganthan (1999) proposes to start the search with the *lbest* topology and then slowly increment, as the number of iterations increase, the number interaction between particles to finally reach the *glbest* topology. The main idea is to exploit the *lbest* in the first stages of the search since it performs better in the exploration phase, and only successively move to the *glbest* that performs better in late stages due to a faster convergence towards the optimum.
- Peram et al. (2003) propose to use a weighted Euclidean distance to identify the particle with which each component of the swarm should interact with.

The aim is to avoid the premature convergence of the traditional PSO algorithm. This correction allows each particle to move in the direction of the ones with the best fitness functions. Going in detail, each particle has to select its neighbour on the basis of a Fitness Distance Ratio (FDR):

$$FDR = \frac{Fitness(X_j) - Fitness(P_i)}{|X; -P_i|} \tag{3.13}$$

where P_i is the *i-th* neighbour of the *j-th* particle. Equation (3.13) ensures to reduce the probability that a particle will interact with one that is too far away from it.

- Jason and Middenford (2004) propose a hierarchy-based topology, in which the next move of every particle is influenced from its best past position and from the position of the particle right above it in the hierarchy of the best positions visited by the swarm. More in detail, they suggest to list (rank) in descending order (from the best position, thus the one with lowest fitness value, to the worse, with highest fitness value) all the positions visited by the particle in that specific iteration.
- Clerc et al. (2009) introduces "Tribes" topology, where the characteristics of the topology and the dimension of the population change with the number of iteration depending on the results obtained. In particular, the initial population is divided into sub-populations (tribes) composed of a variable number of particles. These sub-groups are able to establish intra-tribe communication (each particle stores its best location and is aware of the best location found in its sub-population) as well as an inter-tribe communication (each tribe knows the position of the best particle of all the other tribes). Furthermore Clerc et al. distinguish between good tribes and bad tribes depending on the outcomes of the last iteration: if none of the particles of a tribe was able to improve its result the tribe is considered bad; otherwise, if at least one particle was able to improve its position then the tribe is considered good. Thus, considering that "the most time consuming part of a PSO algorithm is the evaluation of the objective function" it is crucial to "carry out as few evaluations … as possible" (Clerc et al, 2009). For doing this, the number of particles has to be reduced. Following this idea, they decided to select and move the particles with worst results in the best sub-

populations to other worse sub-populations, in this way the chance of improving the more promising tribes is increased. By moving these particles the topology structure is changed as well.

3.3 Grey Wolf Optimization

The Grey Wolf Optimization (GWO) is a recently developed population-based and bio-inspired metaheuristic that due to its performance and versatility to solve a vast number of optimization problems has gained a lot of attention in the literature. As pointed out by Faris et al. (2018) the fields of application thus far range between engineering, machine learning, medical and bioinformatics, environmental applications, networking and image processing. Figure 18 gives a pie-chart representation.

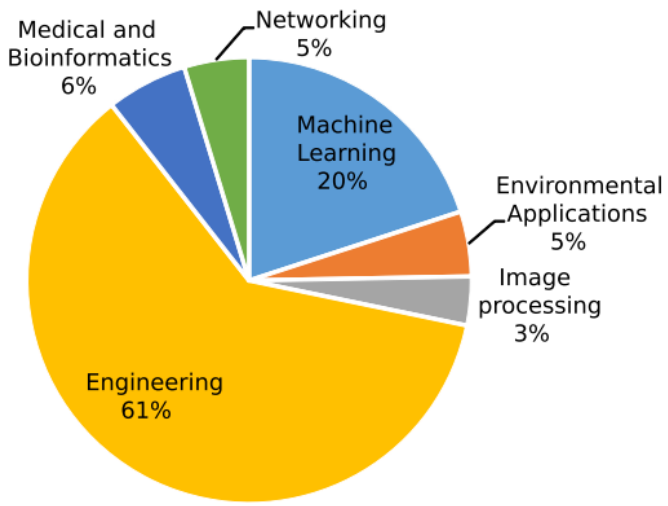


Figure 18: Pie-chart of the main areas of application of the Grey Wolf Optimization. Source: Faris et al. (2018).

Created by Mirjalili et al. (2014) the GWO is inspired by the social behaviour and the hunting technique of grey wolves. With respect to the first aspect, grey wolves are considered apex predators in nature, since they are on the top of the food chain, and they exhibit a strict hierarchical system inside the pack, as shown in Figure 19. On the top of this social order there are the alphas: they are the leaders and they take important decisions for the herd. The role of an alpha is crucial for managing the pack, and even if it may not be the strongest wolf in it, it is the one with the better qualities to rule it. The beta wolf is the second in the hierarchy and it is the most probable candidate to become alpha. Beta is subordinated to alpha but helps

him in taking decisions and it dominates the other wolves of the pack. Delta wolves are the third in the line and they are submitted to alpha and beta but they dominate omega. They are usually hunters, scouts, sentinels or elder wolves that used to be alpha or beta. On the bottom of the pyramid there are omega wolves. They are considered scapegoats and have the lowest rights inside the pack.

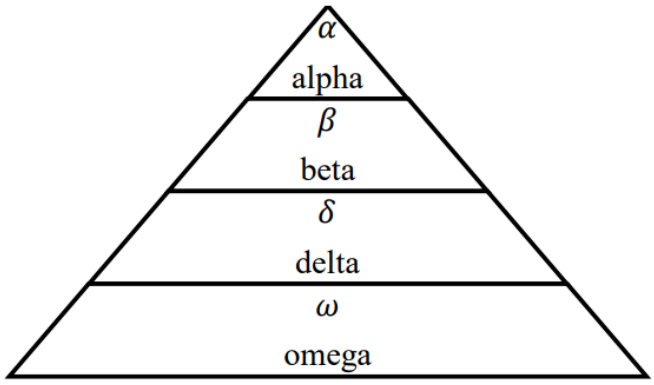


Figure 19: Hierarchical structure of grey wolves in nature. Source: Köybasi and Yazici (2020).

With respect to the second aspect that influence the GWO, Muro et al. (2011) consider three main steps in the hunting behaviour of grey wolves:

- Tracking and chasing the prey;
- Pursuing the prey and encirclement;
- Attack towards the prey.

Like in PSO, where each particle represents a candidate solution, in GWO each wolf (i.e. each searching agent) represents a possible solution of the optimization problem. At the beginning, the initial positions of grey wolves are randomly initialized in the solution space. In order to mimic the social hierarchy of grey wolves, GWO ranks these positions (considering the fitness value of each) in descending order, from the best position to the worst. Thus, the best three solutions in this list are respectively alpha, beta and delta wolves. The remaining solutions are considered omega wolves. For example, considering 50 searching agents, the best three are respectively alpha, beta and delta, while the remaining 47 are omega wolves. At each iteration the solutions are updated and ranked again in the list, so if the algorithm finds better solutions, alpha, beta and delta wolves can be changed with other searching agents that have obtained better fitness values.

On the other hand, to mimic the hunting behaviour of grey wolves, GWO adopts only a few very simple mathematical equations. GWO reasonably assumes that alpha, beta and delta wolves have a good idea of the location of the prey[48]. In fact, as said before, these three wolves represent the best three solutions so far, that are presumably the closest to the prey. Thus, the main idea of GWO is to oblige all the other searching agents to follow these three wolves during the hunt. However, because in an abstract search space we have no idea about the real location of the prey, according to Mirjalili et al., it is safe to assume that its location will reasonably be the average position between alpha, beta and delta, as shown in Figure 20. So, the prey is located in the average point between these three wolves and according to the hierarchical mechanism explained before, as iterations increase and better solutions are found, the three best wolves will change, and as a consequence also the new estimated location of the prey will change. Thus, the prey is "moving" in the solution space as the iterations increase, and through this mechanism all the wolves will converge towards it. For this reason, each omega wolf updates its estimate about the position of prey with respect to the position of the best three wolves following these equations, that represents the encirclement phase of the hunt:

$$\begin{aligned} \overrightarrow{D_\alpha} &= \left|\overrightarrow{C_1} \cdot \vec{X}_\alpha(t) - \vec{X}(t)\right| \\ \overrightarrow{D_\beta} &= \left|\overrightarrow{C_2} \cdot \vec{X}_\beta(t) - \vec{X}(t)\right| \\ \overrightarrow{D_\delta} &= \left|\overrightarrow{C_3} \cdot \vec{X}_\delta(t) - \vec{X}(t)\right| \end{aligned} \tag{3.14}$$

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha(t) - \overrightarrow{A_1} \cdot \overrightarrow{D_\alpha} \\ \vec{X}_2 &= \vec{X}_\beta(t) - \overrightarrow{A_2} \cdot \overrightarrow{D_\beta} \\ \vec{X}_3 &= \vec{X}_\delta(t) - \overrightarrow{A_3} \cdot \overrightarrow{D_\delta} \end{aligned} \tag{3.15}$$

where:

- t is the current iteration;
- $\vec{X}_\alpha$, $\vec{X}_\beta$ and $\vec{X}_\delta$ are the position vectors of the best three wolves;
- $\vec{X}$ is the position vector of a generic wolf;

[48] The prey is the optimum solution of the optimization problem. During the iterations, as shown in Figure 20, the location of the prey coincides with the average position between the best three wolves. As better locations are found, the best three wolves are updated and as a consequence also the estimated position of the prey. After each iteration, through the hunting mechanism, the wolves will converge towards it.

- $\overrightarrow{D_\alpha}$, $\overrightarrow{D_\beta}$ and $\overrightarrow{D_\delta}$ are vectors that measure the distance respectively between alpha, beta and delta and all the other wolves;
- $\vec{A}$ and $\vec{C}$ are two coefficient vectors equal to:

$$\vec{A} = 2a \cdot \vec{r}_1 - \vec{a} \tag{3.16}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{3.17}$$

$$\vec{a} = 2 - t * \left(\frac{2}{t_{\max}}\right) \tag{3.18}$$

where $\vec{a}$ is a vector whose elements are linearly decreased from 2 to 0 as the number of iterations increases; $\vec{r}_1$ and $\vec{r}_2$ are two random vectors uniformly distributed in the interval $[0, 1]$.

Thus, according to equations (3.14), the first step of each omega wolf is to calculate the distance $\overrightarrow{D}$ between him and the positions of each of the best three wolves. Notice that $\vec{X}_\alpha$, $\vec{X}_\beta$ and $\vec{X}_\delta$ are multiplied by a vector ($\vec{C}$) whose elements are random numbers, stochastically changed at each iteration. Then, with equations (3.15) each omega wolf calculates its next move (its next position) with respect to each of the three best wolves, alpha, beta and delta, as if it decides to follow one of them at the time. For example, $\vec{X}_1$ is its next position if he decides to follow alpha, and so on. Also in this case there is a vector ($\vec{A}$) that multiplies each distance, whose elements are random numbers that stochastically change at each iteration. In this way it is possible to better search around these promising areas of the solution space.
However, as previously pointed out, omega wolves estimate the position of the prey (and thus calculate their next move) with respect to the position of the best three wolves, and not just with respect to the position of one of them. Therefore, finally, each omega calculates its next move ($t + 1$) according to this equation:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{3.19}$$

where $\vec{X}(t+1)$ is a vector describing the next position assumed by the omega wolf; $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ are its new positions if it had decided to follow respectively alpha, beta and delta one at the time. Thus, according to equation (3.19) each omega wolf updates its position with respect to the one of these three by making a simple average of their positions, combining three vectorial components. In this way, at each iteration the wolves converge towards what is a reasonably accurate estimate of the prey's location. "In other words alpha, beta, and delta

estimate the position of the prey, and other wolves update their positions randomly around the prey" (Mirjalili et al., 2014).

The effect of equations (3.14), (3.15) and (3.19) on the hunting behaviour of an omega wolf is represented graphically in Figure 20.

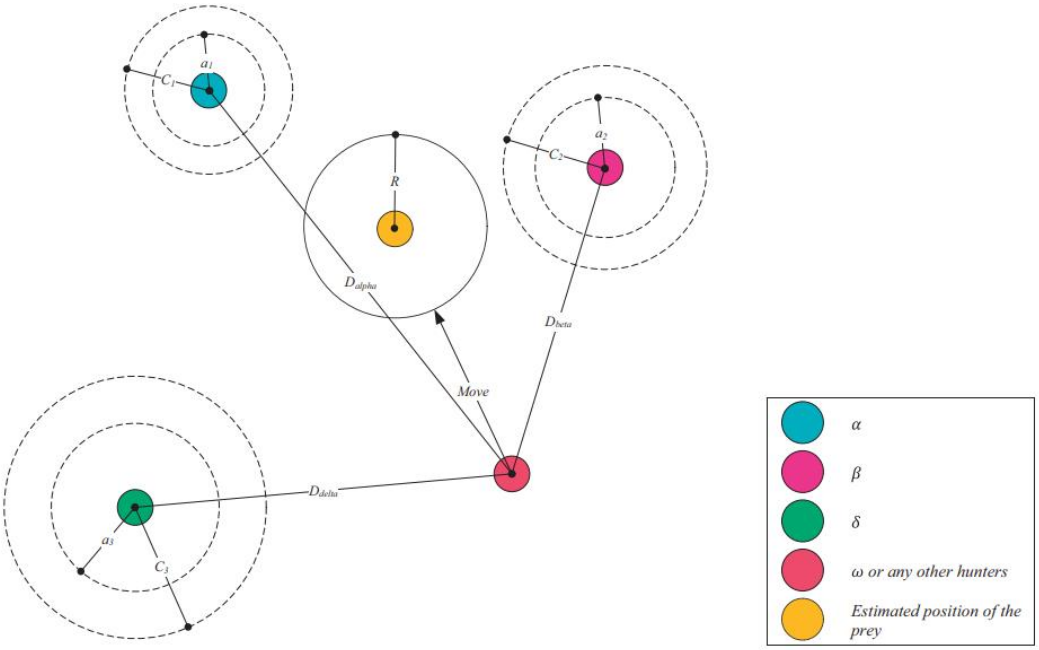


Figure 20: Graphical representation of the encirclement phase of the hunt. Source: Mirjalili et al. (2014).

Focusing now on the parameters of GWO, $\vec{A}$ and $\vec{C}$ are the most important ones as they allow to balance between searching for the prey (exploration) and attacking the prey (exploitation), which represent the other two characteristics of grey wolves hunting technique. The value of the elements of $\vec{a}$ is linearly decreased form 2 to 0, and as a consequence also the value of the elements of $\vec{A}$ (from equation (3.16)), which is ranging in the interval $[-2, 2]$, is also decreasing. Indicating with A a generic element of $\vec{A}$, when $A > 1$ or $A < -1$ the wolf diverges from the prey, giving more emphasis to exploration. Instead, for values of A belonging to $[-1, 1]$ the wolf converges towards the prey, prioritizing exploitation. As it can be seen in Figure 21, because the value of $\vec{A}$ is decreasing as the number of iterations increase, the exploration phase is prioritized in the first part of iterations, while in the second part the algorithm is focused on exploitation. This behaviour guarantees a good balance of GWO algorithm.

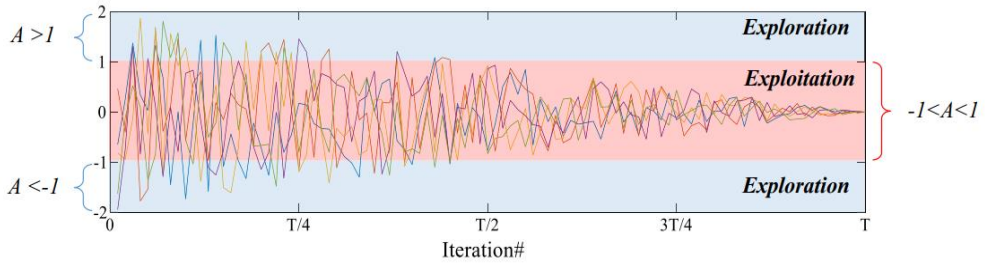


Figure 21: Graphical representation of the effect of changes in parameter A. Source: Faris et al. (2018).

The other important vector of parameters, that favours exploration phase, is $\vec{C}$. It can be seen from equation (3.17) that it is multiplied by a random vector $\vec{r}_2 \in [0, 1]$. Thus, $\vec{C}$ is a stochastic vector that contains numbers ranging between $[0, 2]$. Indicating with C a generic value of $\vec{C}$, then for values of $C > 1$ the effect of the position of the best three wolves in equation (3.14) is emphasized, while for values of $C < 1$ this effect is deemphasized. The main function of $\vec{C}$ is to provide a more random behaviour to the GWO algorithm, favouring exploration and the avoidance of local optimums. From a realistic point of view, this vector of parameters may represent the effect of an "obstacle" that the wolves have to face when searching the prey and that slows their hunt.

3.3.1 The Structure of GWO Algorithm

The structure of GWO algorithm is as follow:

1) Random initialization of the grey wolves population, $X_i(i = 1,2, \dots, n)$, in the searching space.
2) Initialize vectors of parameters $\vec{a}$, $\vec{A}$ and $\vec{C}$
3) Calculate of the fitness value of each searching agent and assign the three best values to alpha, beta and delta wolves, respectively X_α, X_β and X_δ.
4) Starting of the main loop:
    a) For each searching agent (omega wolves) update its current position by equation (3.14)
    b) Update parameters $\vec{a}$, $\vec{A}$ and $\vec{C}$.
    c) Calculate the fitness value of each searching agent and update X_α, X_β and X_δ with the new three best values, if any.

d) If a prefixed stop criterion is met go to step 5, otherwise skip to next iteration and return to step 4.

5) End of the main loop.

6) Return X_α, which is the position of the alpha wolf that represents the best solution found.

GWO algorithm can be represented by the following flowchart:

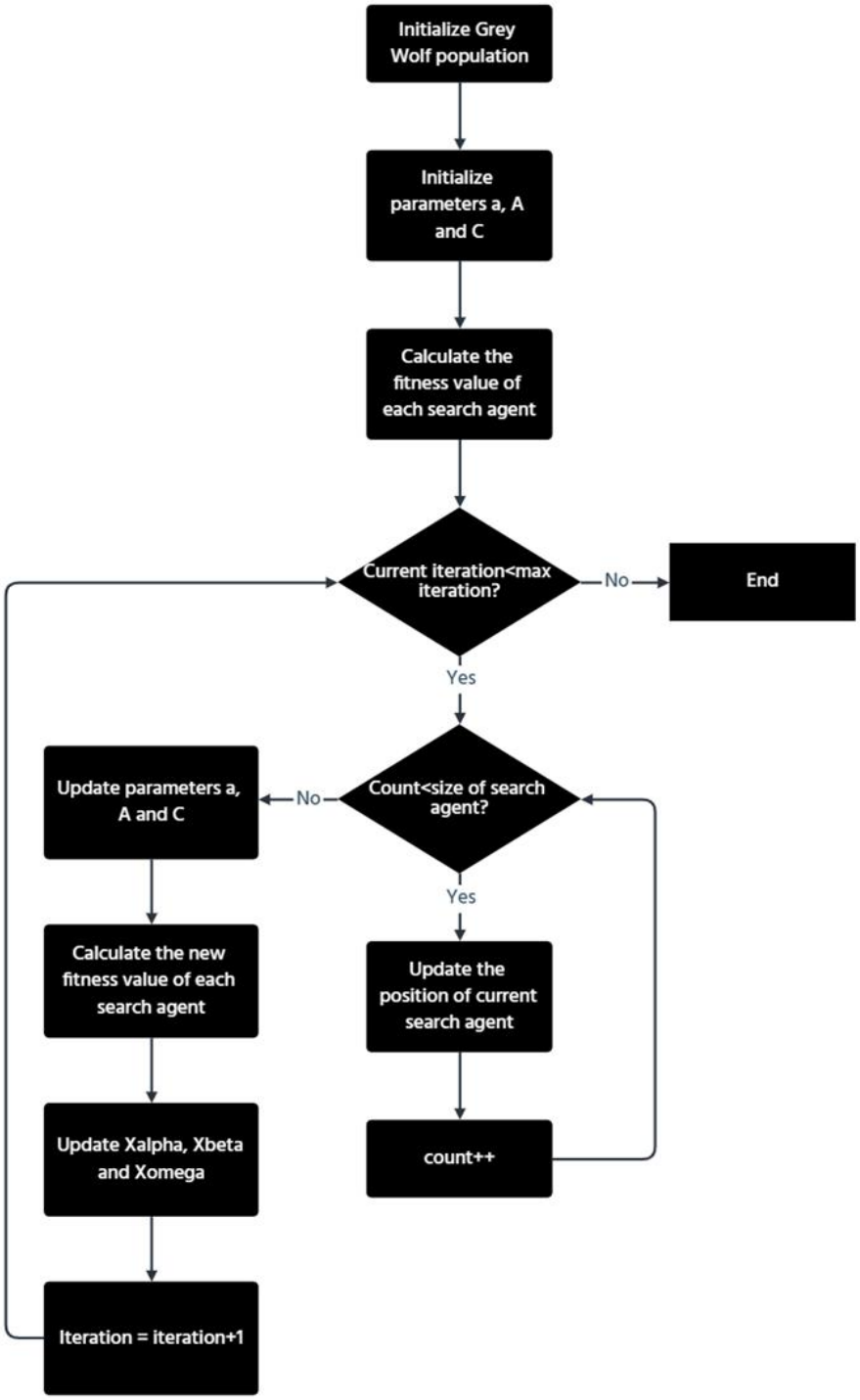


Figure 22: Flowchart of the structure of GWO algorithm.

3.3.2 Main Advances in the Literature

Faris et al. (2017) provide an excellent summary of the most recent advances and improvements of the original version of the GWO. Focusing on the updating mechanisms of the algorithm, that aims to provide a better balance between exploration and exploitation, we can identify several works:

- Mittal et al. (2016) propose a modified version of GWO in which the exploration phase is enhanced by imposing that the vector of parameters $\vec{a}$ follows an exponential decay function instead of equation (3.19):

$$\vec{a} = 2\left(1 - \frac{t^2}{t_{\max}^2}\right) \tag{3.20}$$

  They find better performances with respect to exploration with this new parameter.

- Long et al. (2016) propose another non linear equation to update $\vec{a}$:

$$\vec{a} = \left(1 - \frac{t}{t_{\max}}\right) \cdot \left(1 - \mu \frac{t}{t_{\max}}\right)^{-1} \tag{3.21}$$

  where μ is a non linear modulation index ranging from $(0, 3)$. They find a better balance between exploration and exploitation.

- Dudani and Chudasama (2016) instead propose a new way to update the vector of positions of wolves in contrast with equation (3.14).

$$\vec{X}(t+1) = \left(\frac{1}{t}\right)^{\left|(bestf(t) - f_i(t))(bestf(t) - worstf(t))\right|} \tag{3.22}$$

  where $f(t)$ is the value of the fitness function. The main advantage of this approach is that it uses a fewer number of parameters and that it is not required to define the initial ones.

- Malik et al. (2015) also focus on updating equation (3.14) and they propose to use a weighted average instead of a simple average of the position of the three best wolves:

$$\vec{X}(t+1) = \frac{\omega_1 \vec{X}_1 + \omega_2 \vec{X}_2 + \omega_3 \vec{X}_3}{(\omega_1 + \omega_2 + \omega_3)} \tag{3.23}$$

  where $\omega_1 = A_1 \cdot C_1$, $\omega_2 = A_2 \cdot C_2$ and $\omega_3 = A_3 \cdot C_3$. This version shows better performances when applied to multimodal functions.

Focusing now on population structure and hierarchy of GWO, Yang et al. (2016) propose a different hierarchy system. They divide the initial population of grey wolves in two sub-

populations. The first one is the "cooperative hunting group" and the second one is the "random scout group". The aim of the first group is to perform a deeper exploitation process, while the aim of the second group is to carry out a wider exploration.

The last aspect to be considered is the hybridization of the GWO algorithm. Singh and Singh (2017) propose a coevolutionary mixed hybrid algorithm between PSO and GWO called HPSOGWO. HPSOGWO is coevolutionary because they structure this algorithm in such a way that PSO and GWO work in parallel, and not one after the other. While, "it is mixed because there are two distinct variants that are involved in generating final solutions of the problem" (Singh and Singh, 2017). The positions of alpha, beta and delta are updated by the following equations:

$$\begin{aligned} \vec{D}_\alpha &= \left|\overrightarrow{C_1} \cdot \overrightarrow{X_\alpha} - \omega \cdot \vec{X}\right| \\ \vec{D}_\beta &= \left|\overrightarrow{C_2} \cdot \overrightarrow{X_\beta} - \omega \cdot \vec{X}\right| \\ \vec{D}_\delta &= \left|\overrightarrow{C_3} \cdot \overrightarrow{X_\delta} - \omega \cdot \vec{X}\right| \end{aligned} \tag{3.24}$$

Then they combine the PSO with GWO by updating the velocity equation of PSO in the following fashion:

$$\begin{cases} V_i^{k+1} = \omega((V_i^k + \vec{r}_1\overrightarrow{C_1} \cdot \left(\overrightarrow{X_1} - X_i^k\right) + \vec{r}_2\overrightarrow{C_2} \cdot \left(\overrightarrow{X_2} - X_i^k\right) + \vec{r}_2\overrightarrow{C_2} \\ \qquad \cdot \left(\overrightarrow{X_2} - X_i^k\right)) \\ X_i^{k+1} = X_i^k + V_i^{k+1} \end{cases} \tag{3.25}$$

As it can be seen, ω is a constriction coefficient. Instead of the two classical random component of PSO, $U(0, \Phi_i)$ and $U(0, \Phi_i)$, they multiply each term by $\vec{r}_i\overrightarrow{C_\iota}$. They found out that this hybrid algorithm is more reliable and provides better solutions than normal GWO.

3.4 A Comparison Between the Two Metaheuristics

Finally, a quick comparison between PSO and GWO is presented in this paragraph.

Like PSO, GWO is also a SI based algorithm and it is considered one of the fastest growing metaheuristics of recent years. A lot of improvements have been proposed as shown in the previous section, but many others can be done.

GWO has a simpler structure than PSO, and the underlying mathematical equations are also very easy to be implemented.

Contrary to PSO, that utilizes a vector of positions and a vector of velocities for the searching agents, GWO needs only one matrix of positions in which the current locations of searching agents are ranked from the best to the worst (the best three represents alpha, beta and delta wolves). In this way GWO requires less memory than PSO.

Furthermore, in GWO the only solutions saved are the first three, while PSO registers the best position of each particle (*pbest*) and the global best position obtained by the swarm (*gbest*). This however can be criticised, as a substantial part of available information in GWO is not used. An idea to overcome this limitation could be to work on population topology or on its hierarchical structure, as proposed by Yang et al. (2016).

Moreover, GWO, differently from PSO, has only two vectors of parameters to be adjusted: $\vec{a}$ and $\vec{C}$. As shown previously they help to balance between exploration and exploitation and they are easier to be set than the ones of the PSO.

Finally, in many circumstances, GWO when applied to solve different unimodal and multimodal functions, as for example in Mirjalili et a. (2014) or in Köybasi and Yazici (2020), has been found to be more effective than standard PSO, with a faster convergence towards the optimum and a better balance between exploration and exploitation.

However, both these algorithms are able to solve only unconstrained optimization problems, thus a reformulation of the portfolio selection problem in an unconstrained form is needed.

Chapter 4

Application of PSO and GWO to the Portfolio Selection Problem

In this final chapter I will present the experiment. More precisely, I will apply the two metaheuristics, presented in Chapter 3, to the portfolio selection problem considering the two risk measures presented in Chapter 2, in order to assess Model Risk, presented in Chapter 1, deriving from the two algorithms and from the two risk measures. In the first paragraph I will discuss the reformulation of the portfolio selection problem in order to be solved through the two metaheuristics, since as pointed out earlier they are able to solve only unconstrained problems. In the second paragraph I will present the settings of the portfolio optimization problem and of the two metaheuristics. Then, in the third paragraph I will present the experiment, consisting of the application of the two metaheuristics to 11 securities traded in the S&P 500 considering three different scenarios. PSO and GWO are both applied in two different versions, one using the VaR as an objective function to be minimized and the other using the ES. Thus, in total I have four models and four efficient portfolios as output. Finally, in the fourth paragraph I will apply to the efficient portfolios of the best selected algorithm found in the previous step the backtesting framework for risk measures discussed in Chapter 1, in order to assess Model Risk for financial risk measures.

4.1 Reformulation of the Portfolio Selection Problem

As mentioned in the previous chapter, neither the PSO nor the GWO was born to solve constrained optimization problems. Since the portfolio selection problem in equation (2.22) is by its own nature a constrained problem, we have to reformulate it in an unconstrained form in order to apply the two metaheuristics. In the literature there are many techniques for

doing this[49], here I choose the exact penalty method. Briefly, it consists in the introduction of a penalty function, that is built by adding to the objective function (which in my case is the risk measure that the algorithm tries to minimize) another term that contains the sum of the violations of all the constraints and that add a penalization to each violation. In this way, by minimizing the objective function, the algorithm minimizes jointly the risk measure and the violations of all the constraints. For doing this we have to rewrite each constraint in terms of its violation. Nonetheless, the main issue here is the coincidence between the solution of the constrained and the unconstrained problem, in other words how well the unconstrained version approximates the constrained one. As we will see, the coincidence is ensured by a wise choice of the penalization term.

4.1.1 The Exact Penalty Method

More in detail, the exact penalty methods, introduced by Luenberger (1984), consists in rewriting a general constrained optimization problem of the form:

$$\begin{aligned} & min \quad f(x) \\ & s.t. \quad \begin{cases} h(x) = 0 \\ g(x) \leq 0 \end{cases} \end{aligned} \tag{4.1}$$

where:

- $f(x)$ is a continuous function in $\mathbb{R}^n$;
- $h(x)$ is a vector containing the m equality constraints;
- $g(x)$ is a vector containing the p inequality constraints;

in a unconstrained optimization problem of the form:

$$\min f(x) + \frac{1}{\varepsilon} P(x) \tag{4.2}$$

where:

- ε is a positive number called penalty coefficient.
- $P(x)$ is the penalty function with the following characteristics:
  - $P(x)$ is continuous in $\mathbb{R}^n$;

[49] According to Koziel and Michalewicz (1999) it is possible to identify four different approaches: methods based on preserving feasibility of solutions, methods that distinguish between feasible and infeasible solutions, methods based on penalty functions, hybrid methods.

- $P(x) \geq 0 \ \ \forall x \in \mathbb{R}^n$;
- $P(x) = 0$ if and only if $x \in S$[50];

The role of $\frac{1}{\varepsilon}$ is to amplify the numeric value of each violation. This coefficient must be chosen carefully, since it ensures the correspondence between the solution of the constrained and unconstrained problem.

The Exact Penalty Theorem states that if the constrained problem is reformulated in an unconstrained version with the following penalty function:

$$P(x) = \sum_{i=1}^{m} |h_i(x)| + \sum_{j=1}^{p} \max\{0, g_j(x)\} \tag{4.3}$$

and if a generic point x^* satisfies the second order conditions[51] in order to be a local minimum of the constrained optimization problem, than that point will also be a local minimum of the unconstrained optimization problem if the following property is satisfied:

$$\frac{1}{\varepsilon} > \max\{|\omega_i|, \mu_j\}, \quad \text{with } i = 1 \dots m \text{ and } j = 1 \dots p \tag{4.4}$$

where ω_i and μ_j are two vectors composed by the Lagrange multipliers respectively of the m constraints in $h_i(x)$ and of the p constraints in $g_j(x)$. This theorem guarantees the coincidence between the solutions of the two versions of the optimization problem. However it does not provide any hints for the setting of ε parameter.

Looking back to the portfolio selection problem, following this approach, the constraints of equation (2.22) can be rewritten in terms of violations as:

$$\sum_{i=1}^{N} x_i = 1 \rightarrow \left| \sum_{i=1}^{N} x_i - 1 \right| \tag{4.5}$$

$$\hat{R}_p \geq \pi \rightarrow \max\{0, \pi - \hat{R}_p\} \tag{4.6}$$

$$k_l \leq \sum_{i=1}^{N} z_i \rightarrow \max\left\{0, k_l - \sum_{i=1}^{N} z_i\right\} \tag{4.7}$$

[50] S is a vectorial subspace of $\mathbb{R}^n$.

[51] For a constrained optimization problem the second order conditions state that if the hessian matrix (that contains the second derivatives of the objective function) is positive (or negative) definite, then the stationary point identified is a local minimum (or a maximum) of the function.

$$\sum_{i=1}^{N} z_i \leq k_u \rightarrow \max\left\{0, \sum_{i=1}^{N} z_i - k_u\right\} \tag{4.8}$$

$$z_i l_i \leq x_i \rightarrow \sum_{i=1}^{N} \max\{0, z_i l_i - x_i\} \tag{4.9}$$

$$x_i \leq z_i u_i \rightarrow \sum_{i=1}^{N} \max\{0, x_i - z_i u_i\} \tag{4.10}$$

$$z_i(z_i - 1) = 0 \rightarrow \sum_{i=1}^{N} |z_i(z_i - 1)| \tag{4.11}$$

It is possible to notice that the constraints written in inequality form in equation (2.22) have been split into two components as in equation (4.7) and (4.8) or equation (4.9) and (4.10). In this way, the constrained portfolio selection problem can be rewritten in unconstrained form in the following manner:

$$\min_{X,Z,\varepsilon} P(X, Z, \varepsilon) \tag{4.12}$$

where:

$$\begin{aligned} P(X, Z, \varepsilon) = {} & \rho_{\alpha,p}(Rp) \\ & + \frac{1}{\varepsilon}\left[\left|\sum_{i=1}^{N} x_i - 1\right| + \max\{0, \pi - \hat{R}_p\} + \max\left\{0, k_l - \sum_{i=1}^{N} z_i\right\}\right. \\ & + \max\left\{0, \sum_{i=1}^{N} z_i - k_u\right\} + \sum_{i=1}^{N} \max\{0, z_i l_i - x_i\} \\ & \left. + \sum_{i=1}^{N} \max\{0, x_i - z_i u_i\} + \sum_{i=1}^{N} |z_i(z_i - 1)|\right]. \end{aligned}$$

As we can see $P(X, Z, \varepsilon)$ contains both the risk measure ($\rho_{\alpha,p}$) and the sum of the violations of each constraint. If none of the constraints is violated, the algebraic sum of the terms in square brackets is 0 and thus no penalization is added to the risk measure. On the other hand, if one of the constraints is violated, the sum of the terms in brackets is positive, and it is amplified by the penalization quantity $\frac{1}{\varepsilon}$. Thus, a positive quantity would be added to the risk measure. This would increase the value of the objective function (fitness function) that we actually want to minimize. Thus, if a constraint is violated the quality of that solution is decreased, and usually it is discarded by the algorithms during the search process.

The main issue of this approach remains the correct setting of $\frac{1}{\varepsilon}$ in order to ensure the coincidence between the solution of the constrained and of the unconstrained problem, since as pointed out before the Exact Penalty Theorem gives no hints about it. For this reason, when I propose the parameters' settings of the two metaheuristics in the next paragraph, I run a specific test for the setting of epsilon.

4.2 Settings of the Problem

In this paragraph I am going to present the optimization problem that I am going to solve. As pointed out in the previous chapters I will apply the two metaheuristics presented in chapter 3, the PSO and the GWO, to the portfolio selection problem, discussed in chapter 2. I will consider the two risk measures presented in chapter 1 and 2 as objective functions to be minimized: the Value at Risk and the Expected Shortfall. In this way, in total I consider two versions of the PSO, one with VaR and the other with ES, and analogously two versions of the GWO, one with VaR and the other with ES. Thus, the output of the four models will be four efficient portfolios. By following this method, I will be able to compare the performance of the four portfolios in relation to Model Risk deriving both from the choice of the algorithm and from the choice of the risk measure.

For the experiment, I choose to focus on the US market, taking eleven large-cap securities from the S&P 500, listed in Table 1, where each of them represents a specific market sector. For each security I have downloaded the time series of the adjusted closing prices from Bloomberg, and I have divided each of them into three different sub-periods, each of which corresponds to a peculiar market scenario that will be presented shortly. Furthermore, each sub-period is divided into two subsamples: in-sample period of the duration of one year, and out-of-sample period of the duration of six months. In this way it is possible to test the algorithms by building a portfolio on one year of data and then to test the portfolio's performance on a virtual future of six months of data. For comparability reasons, the length of the three scenarios has been set equal, however, as we will see shortly, two of them consider crisis periods. Therefore, looking at a portfolio manager standpoint, during crisis periods the time interval for the estimation of the model and for the testing phase can be reduced at her discretion, for example considering six months in-sample and three out-of-sample, or taking three months in-sample and one month out-of-sample. In this way, due to the high volatility of markets during crisis periods, it is possible to rebalance the portfolio

more often in order to match current market conditions, reducing its riskiness and increasing its performances.

Then, for each security I calculate the logarithmic net returns[52] as follows:

$$R_{i,t} = \ln\left(\frac{P_{i,t}}{P_{i,t-1}}\right) \tag{4.13}$$

where:

- $R_{i,t}$ is the log return for the *i*-th security at time t;
- $P_{i,t}$ is the price of the *i*-th security at time t;
- $P_{i,t-1}$ is the price of the *i*-th security at time $t-1$.

S&P 500

SECTOR	SECURITY	MARKET TICKER
Communication services	Google	GOOG
Consumer discretionary	Amazon	AMZN
Consumer staples	Walmart	WMT
Energy	ExxonMobil	XOM
Financials	JP Morgan	JPM
Health care	Johnson & Johnson	JNJ
Industrials	3M	MMM
Information technology	Apple	AAPL
Materials	DuPont	DD
Real estate	Americal Tower	AMT
Utilities	Duke Energy	DUK

Table 1: The eleven securities divided for the market sector taken from the S&P 500.

More in detail, the three sub-periods are:

- From 02/07/2007 to 30/06/2008 in sample period (252 observations), from 01/07/2008 to 31/12/2008 out of sample period (128 observations); for a total of 380 observations.
- From 01/07/2013 to 30/06/2014 in sample period (252 observations), from 01/07/2014 to 31/12/2014 out of sample period (128 observations); for a total of 380 observations.

[52] I use the logarithmic net returns instead of the percentage net returns because they provide a better representation of the underlying dynamic of financial prices.

- From 02/01/2019 to 31/12/2019 in sample period (252 observations), from 02/01/2020 to 30/06/2020 out of sample period (125 observations); for a total of 377 observations.

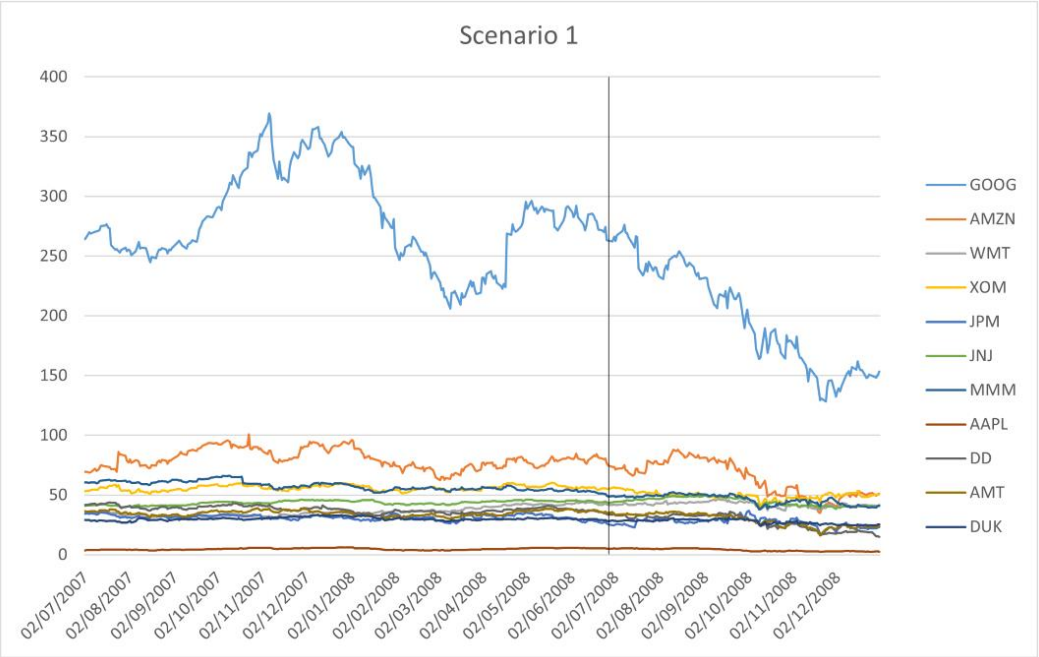


Figure 23: Historical daily closing prices in USD dollars of the eleven securities in the first sub-period.

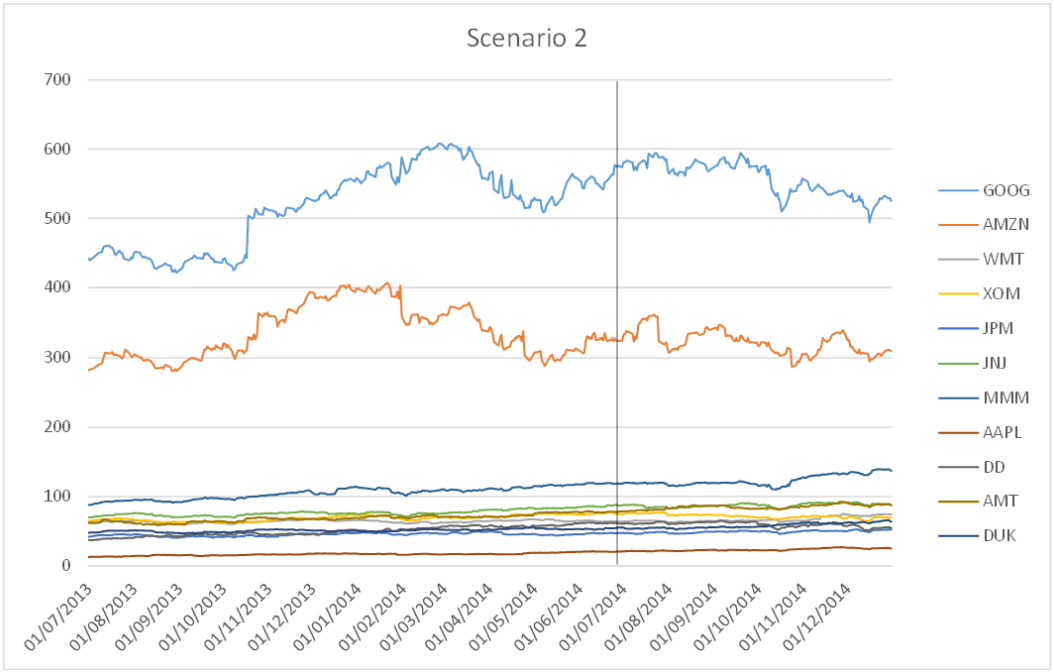


Figure 24: Historical daily closing prices in USD dollars of the eleven securities in the second sub-period.

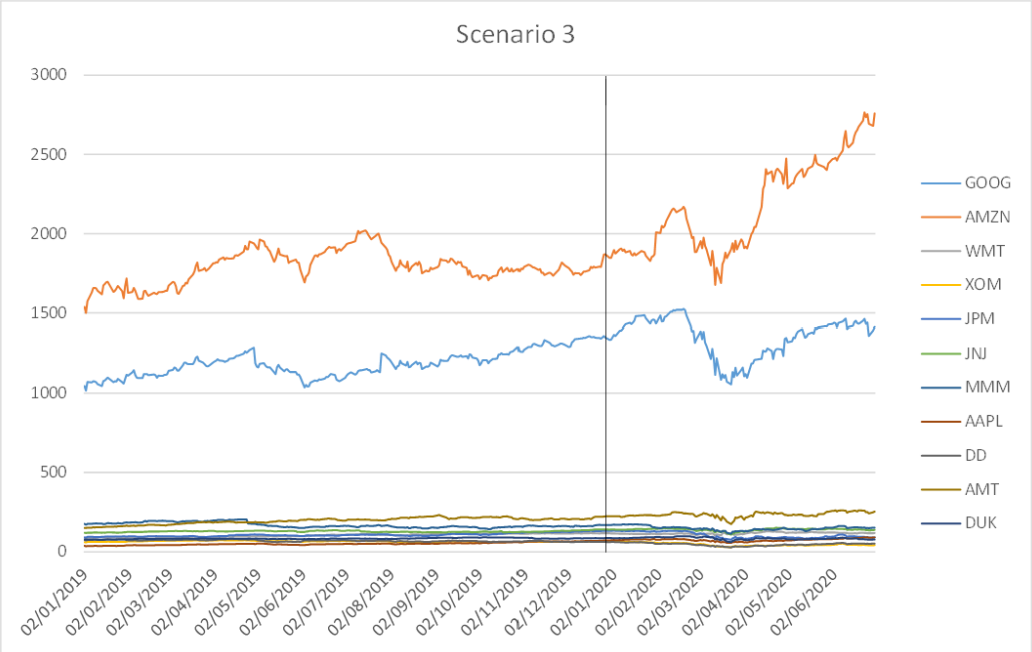


Figure 25: Historical daily closing prices in USD dollars of the eleven securities in the third sub-period.

The first sub-period takes into consideration the Great Financial Crisis that hardly hit the US economy, including in the out-of-sample period the Lehman Brothers bankruptcy[53] and the stock market crash of September 29 of 2008 where the Dow Jones index fell by 7% and the S&P 500 fell by 8.8% in just one day. Figure 23 represents the historical prices of the eleven securities in the first sub-period; the black vertical line divides between the in-sample period and the out-of-sample period. Thus, especially in the out-of-sample period, I expect bad performances on every portfolio, probably with negative returns and very high levels on both VaR and ES, highlighting the current instability of the market.

The second sub-period represented in Figure 24, on the other hand, takes into consideration a growing phase of the US economy, that starts to recover after the crisis: in 2014 the GDP[54] of the country rose by almost 2.5% with respect to the previous year. The monthly job growth has been the strongest since 1999. The unemployment rate at the end of the fourth quarter of 2014 was around 5.8%, which is far lower than the 10% reached during the Great Financial Crisis. What I expect here is a relatively stable period with good performances on every portfolio and low values of risk measures.

[53] On September 15 of 2008, Lehman Brothers filed for bankruptcy, remaining the largest case in US history.

[54] The Gross Domestic Product is the most important indicator of the performance of the economy of a country.

Finally, the third sub-period, represented in Figure 25, takes into consideration the financial crisis due to Covid-19 outbreak, especially in the out-of-sample period. Between the end of February and the beginning of March 2020, the US stock market crashed again with the Dow Jones and the S&P 500 indexes making the sharpest fall after the 2008 crisis. The unemployment rate spiked to almost 15% in April 2020 with respect to the 3.6% registered at the end of 2019. Similarly to the first sub-period, I expect very negative performances of the portfolios in the out-of-sample period with high values of the risk measure. The only title that after the initial plunge of March 2020 seems to recover quickly and also to grow in the following months is Amazon. Thus, I expect that portfolios investing in this security may have mitigated the losses due the crisis.

The choice of these three sub-periods is designed to stress the risk measures' forecasts in order to assess the Model Risk deriving from their application to two crisis scenarios and a growth one.

Now it is important to define the preliminary information needed by the two algorithms used, focusing on the constraints and on the parameters that have to be set. The codes of both algorithms, presented in appendix A, have been run on MATLAB R2020b on Windows 10 with Intel Core i5-1035G4 processor and 8GB of RAM.

4.2.1 Constraints Setting

In this subparagraph I illustrate the setting of the parameters for the constraints presented from equation (4.6) to (4.10) and for the setting of the risk measures both from the PSO and the GWO. These parameters are kept constant and equal for both the algorithms during the computation of the objective function. More in detail, they are:

- The desired daily expected return (equation (4.6)) is set equal to the portfolio mean return over the in-sample period, considering assets as equally weighted: $\pi = \sum_{i=1}^{N} \hat{r}_i \, / N$ where $\hat{r}_i$ is the *i*-th asset's mean return over the in-sample period and it is equal to $\hat{r}_i = \frac{1}{T} \sum_{i=1}^{T} r_{i,T}$. In this way the desired daily expected return depends form the historical time series of the in-sample prices of each security, allowing it to change depending on the sub-period considered;
- The minimum number of assets of the portfolio (equation (4.7)) is set equal to: $k_l = 5$;

- The maximum number of assets of the portfolio (equation (4.8)) is set equal to: $k_u = 9$;
- The minimum percentage of wealth invested in each security (equation (4.9)) is set equal to: $l_i = 3\%$;
- The maximum percentage of wealth invested in each security (equation (4.10)) is set equal to: $u_i = 20\%$;
- The confidence level for both risk measures, VaR and ES, is set equal to: $\alpha = 5\%$;
- Initial capital invested: $C = 100\$$. I set it equal to 100 in order to have VaR and ES results expressed in percentage terms, in accordance with the other variables used for the composition of the portfolio, $x_1, x_2, \dots, x_N$[55]. Thus, VaR and ES outputs will highlight the loss in percentage terms to which the portfolio is subjected to.

4.2.2 PSO Parameters' Setting

I choose to use the inertia weight version of the PSO presented in *subparagraph 3.2.2* of chapter 3 since it provides a more stable model, preventing the explosion of the velocity without having to set its upper limit, which as discussed earlier can be tricky.
Now I provide the list of the parameters' setting for this metaheuristic:

- Inertia weight: $\omega = 0.7298$;
- Cognitive acceleration coefficient: $\Phi_1 = 1.49618$;
- Social acceleration coefficient: $\Phi_2 = 1.49618$;
- Penalty parameter: $\varepsilon = 1e - 05$;
- Number of particles: $p = 70$[56];
- Number of iterations: $niter = 3000$.

The parameters governing the equations for the position and the velocity of each particle (equation (3.7)) have been chosen in accordance with literature's advice. Thus, inertia weight, cognitive and social acceleration coefficients have been chosen according to what presented in the previous chapter.

[55] Recalling from paragraph 2.1 that they express the percentage wealth to be invested in each security.
[56] To ensure a good exploration of the searching space it is best to use a number of particles that is at least the double of the number of securities.

The penalty parameter epsilon, ε, that ensures the coincidence between the solution of the original constrained problem and that of its reformulation in unconstrained form, has been chosen empirically, since the Exact Penalty Theorem does not provide any help. I run the PSO algorithm[57] five times for a different value of epsilon in order to test such values. Then, I took the average of the results of the five runs and calculated the normalized fitness value, along with their standard deviation. The results are reported in Table 2. As we can see the slowest value of the normalized fitness is when $\varepsilon = 1e - 05$, thus I chose to use this value.

EPSILON	NORMALIZED FITNESS	STANDARD DEVIATION
1	0.2525061	0.4760110
0.1	0.2754862	0.3604500
0.01	0.3359008	0.1743508
0.001	0.2264751	0.3699218
0.0001	0.2157207	0.2201669
0.00001	0.1354620	0.2527649
0.000001	0.1844512	0.2535577
0.0000001	0.2789033	0.8783957

Table 2: Results for the setting of epsilon parameter in the PSO algorithm.

Once this parameter is set, I proceed with the setting of the number of particles and of the number of iterations. The correct setting of these two parameters is important since it directly impacts the efficiency of the algorithm, with respect to the quality of the solution and to the computational time. Increasing the number of particles or the iterations too much, without having a proportional improvement of the solution, would cause a waste of computational effort and time.

For the number of particles, that represents the number of searching agents, I run the PSO algorithm five times with different numbers (30, 40, 50, 60, 70) and then I choose again the one with lower fitness value, that corresponds to 70 particles.

I have done the same test for the number of iterations, choosing 3000. Figure 26 shows the convergence curve[58] of the PSO algorithm with Expected Shortfall: as we can see the

[57] with 50 particles and 1000 iterations.

[58] It is the curve plotted with each value of the final fitness function obtained at each iteration, it shows the convergence of the algorithm' solution towards the optimum as the iterations increase.

convergence is very fast initially, and then it almost stops very close to 0 for the largest part of the remaining iterations. This suggests that a smaller number could have been used, because for a considerable part of the final iterations the solution is not improving, while we are wasting computational time. However for comparability reasons with the GWO algorithm, that as we will see shortly has a far slower convergence, 3000 iterations have to be chosen.

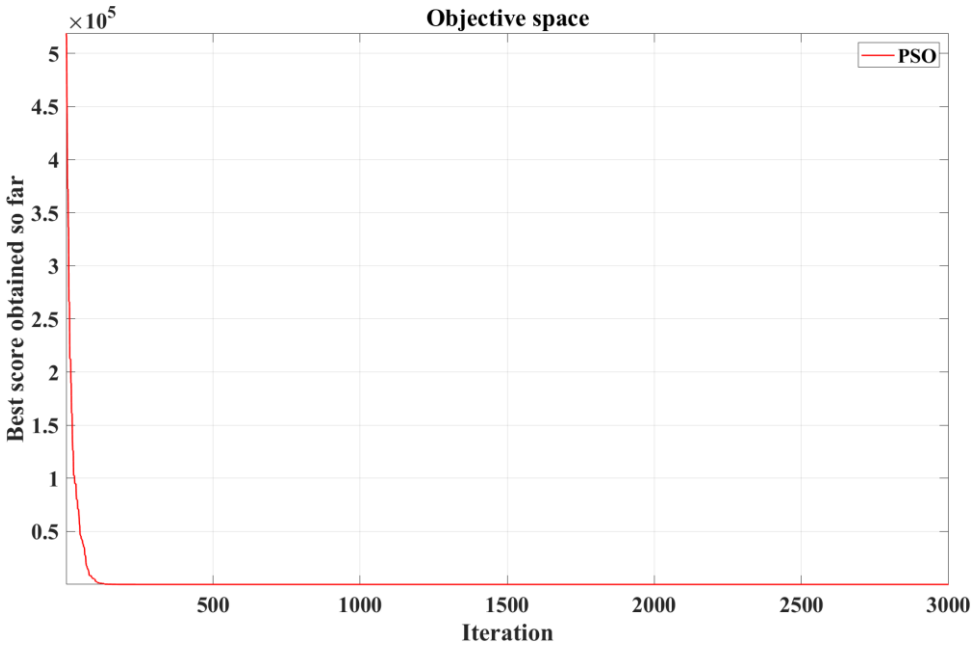


Figure 26: Convergence curve of the PSO algorithm with Expected Shortfall, with 70 particles and 3000 iterations.

4.2.3 GWO Parameters' Setting

Focusing on the GWO, I chose to use the original version of the algorithm presented in paragraph 3.3 of chapter 3. The parameters to be set are:

- Lower bound: $lb = -1e18$
- Upper bound: $ub = 1e18$
- Number of variables; it corresponds to the number of securities taken into consideration: $dim = 11$;
- Penalty parameter: $\varepsilon = 1e - 05$;
- Number of searching agents: $SearchingAgents\_no = 70$;
- Number of iterations: $Max\_iteration = 3000$.

The upper and lower bound have been chosen in order to not impact the behaviour of searching agents in the solution space, as it is for the PSO. By reducing these bounds, the algorithm would reallocate all the searching agents that go beyond them. Thus, by setting them equal to very large and small numbers this mechanism has been neutralized.

To chose the penalty parameter epsilon, I run the same test of the PSO and found out that the lowest normalized fitness value is in correspondence to $\varepsilon = 1e - 05$.

EPSILON	NORMALIZED FITNESS	STANDARD DEVIATION
1	0.308360	0.152421
0.1	0.298832	0.165099
0.01	0.369005	0.189747
0.001	0.322043	0.208138
0.0001	0.310252	0.143093
0.00001	0.296688	0.165456
0.000001	0.343073	0.204618
0.0000001	0.315252	0.131414

Table 3: Results for the setting of the epsilon parameter in the GWO algorithm.

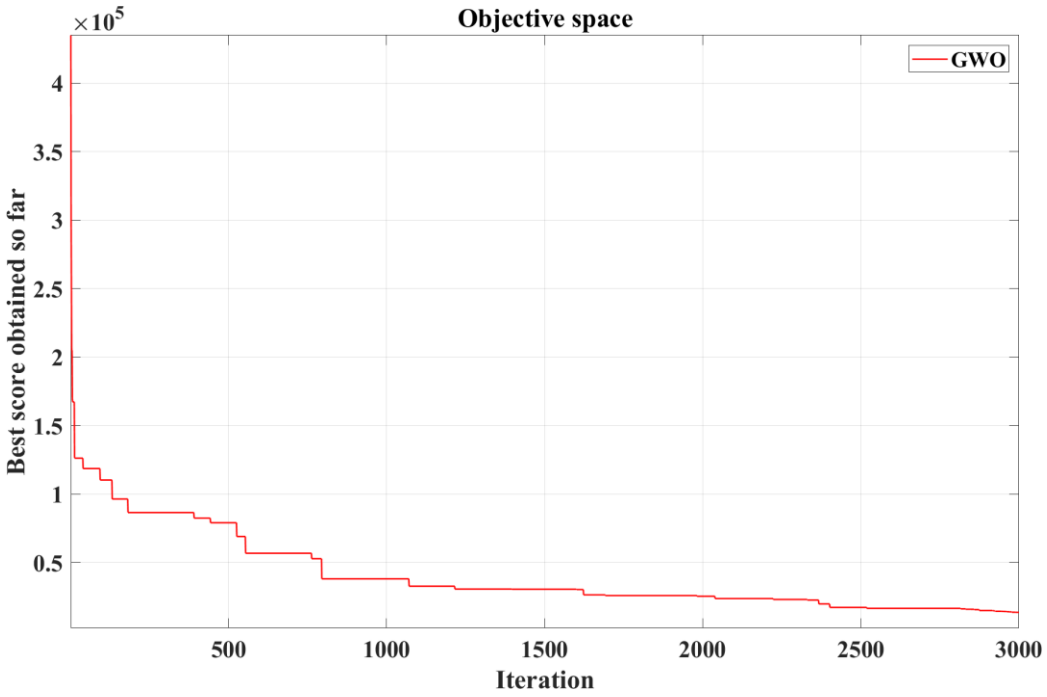


Figure 27: Convergence curve of the GWO algorithm with Expected Shortfall, with 70 particles and 3000 iterations.

The number of searching agents, for comparability reasons, corresponds to the number of particles in the PSO. However, I have run the same test of the PSO and I found out that by increasing the number of particles the quality of solutions increases together with the computational time. Thus, 70 searching agents is once more a good trade-off.
Again for comparability reasons, the number of iterations is set equal to 3000. Nonetheless, in this case it could have been greater because as shown in Figure 27, the convergence is far slower than in the PSO, and even after 3000 iterations it struggles to go to 0. However, 3000 is a good trade-off between the solution's quality and computational time.

4.3 Application and Discussion

Now that the initial tuning of the two algorithms is completed, I begin the presentation of the experiment.
For each sub-period I run the four algorithms (PSO-VaR and PSO-ES, GWO-VaR and GWO-ES) 5 times each on the in-sample period to build the efficient portfolio, and then I select the run with lowest fitness value. Next, I use the out-of-sample period as a virtual future to test the performance of the portfolio built, focusing on the expected return and on the value of the risk measure. I report the data in tables and discuss them in relation to the specific sub-period. Furthermore, I analyse the performance of the two algorithms, comparing the results obtained in relation to Model Risk, especially considering:

- the best fitness value obtained after the five runs;
- the violations of the constraints;
- the computational effort and time required by the algorithm to fully execute the code.

Since our aim is to minimize the fitness value[59], if the final output of an algorithm has a high fitness level this would suggest a worse convergence towards the real optimal solution, highlighting that the solution fund may be in reality far from the real one. Thus, the risk of taking the wrong decisions by making an inefficient allocation is looming.
Constraints' violations are linked with the fitness value obtained. If a large amount of them is heavily violated, the fitness value would rise, highlighting again the worse quality of the

[59] As shown in *subparagraph 4.1.1*, it is the objective function to be minimized, containing both the risk measure and the sum of the violations of each constraint, penalized by the epsilon parameter.

solution found. Here, it is important to consider how many of the constraints are violated and by how much they are violated. In fact, as previously pointed out, a higher magnitude of violations would be amplified by the ε parameter in the computation of the fitness value (as equation 4.12 suggests), causing its value to increase.

Computational effort is another component, linked to Model Risk, that has to be taken into consideration: if an algorithm is able to achieve better solutions (thus with lower fitness values) with less iterations and fewer searching agents, it should be considered more efficient. As pointed out in the previous sections (*4.2.2* and *4.2.3*) where I Made the setting of the two algorithms' parameters, increasing the number of iterations or the number of searching agents would have a direct impact on computational time. More specifically, fixing the number of iterations and repeating the steps required by the algorithm for a higher number of searching agents, will increase the time required. In the same way, fixing the number of searching agents and increasing the number of iterations will increase the time required by both algorithms. In fact, the computational time requirement is a crucial aspect, because having correct information more rapidly may directly impact on the investment decisions taken by the portfolio manager.

Then, in paragraph 4.4, considering the best portfolios among the ones obtained by the algorithms I perform Model Risk assessment for risk measures through a backtesting analysis in the three sub-periods, focusing on the three properties that a good risk forecast should possess, presented in paragraph 1.4[60].

4.3.1 First Sub-period

As pointed out in the previous section, the first sub-period takes into consideration the period of the Great Financial Crisis between mid 2007 and 2008.

Table 4 reports the best results obtained by the two algorithms, when applying both the VaR and the ES, in the five runs in terms of best fitness value; furthermore it reports also the number of assets selected, the values of the risk measures (both in-sample and out-of-sample), the average return of the portfolios (both in-sample and out-of-sample), and the violations of the constraints.

[60] Expected frequency of violations, absence of violation clustering and suitable magnitude of violations.

OUTPUT OF THE PROBLEM

Output	PSO-VaR	PSO-ES	GWO-VaR	GWO-ES
Best fitness	0.020714983	0.029448534	7.65E+03	8.35E+03
Number of selected assets	6	9	5	6
Risk measure in-sample	2.071498274	2.944684713	2.341363633	2.73018614
Mean return in-sample	1.03E-04	5.30E-05	-1.63E-04	-1.06E-05
Risk measure out-of-sample	5.568303355	7.369277529	5.933193277	6.74748208
Mean return out-of-sample	-1.83E-03	-2.25E-03	-2.61E-03	-1.87E-03
Constraints				
Budget	1.04E-14	2.10E-12	9.39E-05	7.97874E-05
Return	0	0	1.16E-04	0
Min number of assets	0	0	0	0
Max number of assets	0	0	0	0
Min investment %	8.33E-24	2.80E-14	1.02E-03	3.03E-02
Max investment %	3.65E-24	1.61E-13	2.06E-02	1.92E-02
Z	2.96E-22	1.63E-11	5.47E-02	3.39E-02

Table 4: Results and constraints violations form the application of the PSO and GWO to the first sub-period.

Starting from the fitness value it can be seen that the PSO algorithms are able to obtain far better results than the GWOs. This is in line with what showed in Figure 26 and 27, where the PSO converges much faster touching almost 0 in the first 1000 iterations. Whereas, the GWO convergence is slower, and neither 3000 iterations are sufficient to move it close to 0. This could be considered an important component of Model Risk in relation to the two algorithms, that has to be taken into account when choosing between the two, because the PSO allows a better convergence towards the real optimal solution in less iterations, thus with less computational effort. Furthermore, the higher fitness values obtained by the GWO are due to the fact that it violates more constraints than the PSO, and the magnitude of these violations is greater on average. Thus, the results of the PSO are more reliable because they are in line with the requests of the portfolio manager. For example, the PSO-VaR violates the budget constraint only by 1.04E-14, whereas the GWO-VaR violates it by 9.39E-05. It can be also noticed that every algorithm violates the budget, the minimum and maximum invested percentage and the Z constraints; instead none of them violates the constraints regarding the minimum and the maximum number of assets of the portfolio and the return constraint (excluding the GWO-VaR).

Now, focusing on the performances of the portfolios, the risk measures in-sample are almost in line for the two algorithms: considering the VaR, the GWO shows a slightly higher level of VaR than the PSO, with respectively 2.341 against 2.071. On the other hand, considering the ES, the results are also coherent, because the ES measures are greater than the VaR. However, in this case the PSO registers a higher ES than the GWO.

As expected, the values of the risk measures out-of-sample are higher with respect to the in-sample period, highlighting the greater risk in the market of the next six months. For example, the PSO-ES registers a value of 2.944 in-sample and a value of 7.369 out-of-sample.

With respect to the portfolios' mean returns in-sample, the PSO is able to outperform the GWO obtaining positive returns both for the VaR and the ES, while the GWO's portfolios obtain negative returns in both cases. As expected, in the out-of-sample window every portfolio's return is worse than in the other period. If PSO's portfolios have negative returns, GWO's ones have even more negative returns than in the in-sample-period.

ASSET ALLOCATION

Security	PSO-VaR	PSO-ES	GWO-VaR	GWO-ES
Google	0.152868738	0.041235233	0	0.201291418
Amazon	0.186637838	0.171292896	0.196391998	0.206411838
Walmart	0	0.176161488	0.216418401	0.203489321
ExxonMobil	0.171845772	0.181609964	0	0
JP Morgan	0	0.039606437	0	0
Johnson & Johnson	0.196431324	0.033900751	0	0
3M	0	0.141949941	0.195647199	0.207642277
Apple	0	0.095474597	0	0.000639443
DuPont	0	0.118768693	0.199595562	0
American Tower	0.096485000	0	0.192040784	0
Duke Energy	0.195731327	0	0	0.18060549
Tot percentage	1	1	1	1
Number of assets	6	9	5	6

Table 5: Asset allocation in the first sub-period.

Table 5 presents the asset allocation of the four metaheuristics. As we can see, PSO-VaR selects 6 assets, PSO-ES selects 9 assets, GWO-VaR selects 5 assets and GWO-ES selects 6 assets. The portfolio PSO-ES is the one that diversifies mostly, investing in 9 assets. It can be noticed that every portfolio suggests to invest heavily on Amazon, especially the GWO's

ones. Furthermore, it can be seen that GWO's portfolios violate the constraints of the minimum and the maximum percentage invested in each asset. For example, even if the upper bound is set to 20% of initial capital, the GWO-VaR allocates 21.642% to Walmart. On the other hand, the GWO-ES allocates only 0.064% to Apple. This strongly violates the lower bound, set to a minimum of 3%. In addition, the GWO-VaR portfolio can be seen as an equally weighted portfolio composed of 5 securities.

Finally, the last aspect to be considered, valid for every sub-periods, is the computational time required by the two algorithms to fully execute the code. This can be seen as another point when assessing Model Risk for the two algorithms. For example, considering the case of quantitative trading[61], where even a fraction of a second can be crucial for the effectiveness of the trade, having good information as quickly as possible is essential. Thus, given the settings presented in sections *4.2.2* and *4.2.3*, the PSO code is executed on average in 16 seconds, while the GWO code is executed on average in 30 seconds. Therefore, also in this case the PSO seems to prevail over the GWO.

4.3.2 Second Sub-period

The second sub-period takes into consideration the growing phase of the US economy of mid 2013 and 2014.

Table 6 presents the results obtained by the two algorithms in both their versions. With respect to the fitness value, also in this case the PSO achieves better results than the GWO for the same reason explained previously: it violates less constraints and with a lower magnitude. None of the PSO versions violate the budget constraint, while all the metaheuristics violate the Z and the minimum and maximum invested percentage constraints.

As expected, the risk measures in-sample and out-of-sample are lower than in the first sub-period. Furthermore, there is no remarkable difference, as in the first sub-period, between the performances in-sample and out-of-sample. However, the GWO's portfolios show a strange result: the ES both in-sample and out-of-sample is lower than the VaR. This result is not in line with the characteristics of the two risk measures presented in paragraph 2.3, because by definition the values obtained by the ES must be greater than the VaR. This could be another

[61] It is an investment technique that relies on mathematical and statistical analysis in order to automate the investing process and the decision making phase, thus requiring a lot of computational effort and fast information delivery.

factor, for the Model Risk assessment, that may suggest the better reliability of the PSO algorithm over the GWO.

The mean returns in-sample of every portfolio are all positive, and the best one is obtained by the PSO-VaR with 1.19E-03. Also in the out-of-sample period all the mean returns are positive, with the exception of the GWO-VaR, which shows a negative return of -5.39E-06. This is caused by asset allocation: a reason may be having invested heavily on Google and Amazon, which grew in February and March 2014 but then moved sideways for the remaining period, with a slower downward tendency in the out-of-sample period, as shown in Figure 24.

OUTPUT OF THE PROBLEM

Output	PSO-VaR	PSO-ES	GWO-VaR	GWO-ES
Best fitness	0.009893089	0.013436364	3.92E+03	8.04E+03
Number of selected assets	7	8	5	5
Risk measure in-sample	0.989308931	1.343636400	1.396042564	1.293802682
Mean return in-sample	1.19E-03	9.92E-04	9.26E-04	1.07E-03
Risk measure out-of-sample	1.467763461	1.777452136	1.935680869	1.525163506
Mean return out-of-sample	3.56E-04	4.29E-04	-5.39E-06	5.35E-04
Constraints				
Budget	0	0	2.23E-05	2.68E-04
Return	0	0	2.62E-06	0
Min number of assets	0	0	0	0
Max number of assets	0	0	0	0
Min investment %	3.97E-26	1.05E-26	9.50E-04	1.98E-03
Max investment %	7.03E-25	2.37E-24	2.44E-03	1.31E-02
Z	4.84E-24	1.22E-23	3.58E-02	6.51E-02

Table 6: Results and constraints violations form the application of the PSO and GWO to the second sub-period.

Table 7 presents the asset allocation of the four metaheuristics. The number of assets selected for the PSO-VaR, PSO-ES, GWO-VaR and GWO-ES are respectively: 7, 8, 5 and 5. Thus, PSO's portfolios selects more securities than GWOs, allowing a greater diversification of the investment; and the portfolio that diversify mostly is the PSO-ES. It can be noticed that GWO's portfolios are almost equally weighted. Nonetheless, allocation more than 20% in some securities, they violates in some cases the constraint of the maximum percentage allowed to be invested. The only security that is picked by every algorithm is Duke Energy.

In fact, this security shows a steady and steep growth form the end of 2009 until today (where it crosses 100 $ per share), moving in the window considered from an adjusted closing price of 47.885 $ on 01/07/2013 to 63.856 $ on 31/12/2014.

ASSET ALLOCATION

Security	PSO-VaR	PSO-ES	GWO-VaR	GWO-ES
Google	0.156308310	0	0.199386901	0.20057737
Amazon	0	0.061666138	0.199139346	0
Walmart	0.072603018	0	0	0
ExxonMobil	0.062193961	0.192942035	0	0
JP Morgan	0	0.10676389	0.200185287	0
Johnson & Johnson	0	0	0	0.197746091
3M	0	0.136843601	0	0
Apple	0.186902514	0.098771526	0	0.188912595
DuPont	0.199879122	0.109689945	0.201516572	0
American Tower	0.124136487	0.173263794	0	0.211097434
Duke Energy	0.197976589	0.12005907	0.199749576	0.201398742
Tot percentage	1	1	1	1
Number of assets	7	8	5	5

Table 7: Asset allocation in the second sub-period.

4.3.3 Third Sub-period

The third sub-period represents again a crisis period: the very recent one due to Covid-19 outbreak in 2020. Here, as in the first sub-period, the biggest impact of the crisis is in the out-of-sample period.

Table 8 presents the results obtained by the two algorithms in both their versions. Looking at the fitness value, also in this case the PSO achieves better results than the GWO. The Z and the minimum and maximum invested percentage constraints are violated by every metaheuristics. However, they are violated heavily by the GWO's portfolios.

As expected, the risk measures in-sample are lower than those out-of-sample, where the crisis began. For example, the PSO-ES shows a value of 1.610 in-sample, which is close to the values reported in the second sub-period, and 7.011 out of sample. Thus, they all behave well according to the period. Furthermore, this extreme increase in the value of the risk measures is even bigger than the one in the first sub-period, where for example the PSO-ES moves from a value of 2.945 in-sample to 7.369 out-of-sample, highlighting the sudden and

unexpected explosion of the Covid-19 crisis. The mean returns in-sample are all positive, and the higher one is obtained by the GWO-ES with 1.20E-03. Focusing on the out-of-sample mean returns the PSO's portfolios perform worse than the GWOs, that are able not only to outperform them but also to achieve a positive return.

OUTPUT OF THE PROBLEM

Output	PSO-VaR	PSO-ES	GWO-VaR	GWO-ES
Best fitness	0.012160583	0.016103576	3.25E+03	6.27E+03
Number of selected assets	7	8	5	5
Risk measure in-sample	1.216058295	1.610357557	1.71732941	2.249590938
Mean return in-sample	9.70E-04	9.70E-04	4.80E-04	1.20E-03
Risk measure out-of-sample	4.157367409	7.011465069	4.300108186	6.92870834
Mean return out-of-sample	-7.32E-04	-5.87E-04	4.82E-04	6.28E-04
Constraints				
Budget	0	6.66E-16	8.75E-05	3.22E-04
Return	0	0	3.19E-04	0
Min number of assets	0	0	0	0
Max number of assets	0	0	0	0
Min investment %	4.49E-26	1.05E-17	5.57E-04	1.52E-03
Max investment %	4.24E-25	6.01E-17	1.24E-02	3.46E-03
Z	3.62E-24	3.32E-15	1.91E-02	5.74E-02

Table 8: Results and constraints violations form the application of the PSO and GWO to the third sub-period.

According to Table 9 that shows the asset allocation, this might be due to the fact that the GWO's portfolios invest heavily on Google and especially on Amazon, that after the initial plunge is able to recover in few months and finally skyrocketed, according to Forbes, due to the diffused lockdowns all around the world that incentivize people to order goods from home for daily necessities. The number of assets selected for the PSO-VaR, PSO-ES, GWO-VaR and GWO-ES are respectively: 7, 8, 5 and 5. Also in this case, the PSO's portfolios are more diversified than the GWOs. The only security that is selected by every algorithm is American Tower. Actually, this security shows a huge growth in the period considered, moving from an adjusted closing price of 150.3573 $ on 02/01/2019 to 254.5403 $ on 30/06/2020. It can be also noticed that GWO's allocations sometimes violate the upper bound of the percentage to be invested in each security.

ASSET ALLOCATION				
Security	**PSO-VaR**	**PSO-ES**	**GWO-VaR**	**GWO-ES**
Google	0.195566671	0	0.187510703	0.201370336
Amazon	0	0.073745028	0.202951784	0.199898843
Walmart	0.185869893	0.154918753	0	0
ExxonMobil	0.113505417	0.188257962	0	0.200093251
JP Morgan	0.184731915	0.114424648	0	0
Johnson & Johnson	0.041882052	0	0	0
3M	0.10121882	0.063002302	0.203196339	0
Apple	0	0.116212438	0	0.200580232
DuPont	0	0	0.203320700	0
American Tower	0.177225233	0.118153394	0.202932933	0.197735392
Duke Energy	0	0.171285475	0	0
Tot percentage	1	1	1	1
Number of assets	7	8	5	5

Table 9: Asset allocation in the third sub-period.

4.4 Risk Measures Backtesting

As pointed out in the previous sections, PSO's portfolios generally are more efficient than the GWOs: they are able to achieve better fitness values, violating on average less constraints and with lower magnitude. Thus, for this backtesting analysis[62] I consider only PSO's portfolios previously built (PSO-VaR and PSO-ES). The aim of this further analysis is to assess which of the two risk measures used is better in terms of Model Risk. To do this, I consider these two portfolios in the same three sub-periods previously presented. However, because I need a higher number of observations for the application of the models that I am going to present, in this case I do not make the distinction between out-of-sample and in-sample periods, instead I consider the time series in its entirety.

In order to assess Model Risk deriving from the two risk measures, I consider the three properties that a good risk forecast should possess, presented in paragraph 1.4:

- Expected frequency of violations;
- Suitable magnitude of violations;
- Absence of violations clustering.

[62] A backtesting analysis aims to test quality and the reliability of a predictive model on historical data. The main idea is that if a model behaves well on historical data it is likely to provide good forecasts.

For each of them I provide a method to measure it directly on historical data. For doing this, using the variance-covariance method[63] I implement one static approach and two dynamic approaches, that will be presented shortly, to model on historical data both the VaR and the ES forecasts. The two dynamic models are an Exponentially Weighted Moving Average (EWMA) and a Generalized Autoregressive Conditional Heteroskedasticity (GARCH). These two models and the static approach are applied to determine the "historical" forecasts of the two risk measures in order to see if what they have predicted is in line with the actual return in that specific day[64]. In this way, I am able to historically check the performances of the two risk measures:

- I calculate the number of violations counting all the cases where the historical loss exceeds the value predicted by the risk measure. It has to be mentioned that the percentage number of violations should be lower than the confidence level required (in my case 5%), otherwise this would suggest a misbehaviour of the risk measure.
- I calculate the average magnitude of violations determining for all the violation cases the average amount of the losses that exceed the predicted one. In this way, I am able to compare the average magnitude of VaR violations with the ES ones.
- I check for violation clustering using a *k*-mean clustering algorithm through the function *kmean* in MATLAB[65]. According to Arthur and Vassilvitskii (2007) it is an iterative algorithm that partitions the initial dataset into *k* non overlapping subsets (clusters) by seeking "to minimize the average squared distance between points in the same cluster." In this way it is possible to individuate *k* clusters in the data series. The measure used for the distance is the Squared Euclidean Distance:

$$d(x, c) = (x - c)(x - c)' \tag{4.14}$$

where: $x \in X$ is an observation of X, which is the numeric matrix containing the data, and c is a centroid[66]. Thus, by applying this algorithm to the historical series that contains only the violations of risk measure's forecasts, I detect three centroids

[63] It is a parametric technique for measuring the VaR and the ES that relies on the calculation of the variance-covariance matrix of portfolio's returns and that assumes they follow a multivariate normal distribution. It is a very simple technique to be applied, however the main drawback of this approach is that it relies on normality assumption, that might be too restrictive, not considering for example the problem of fat tails.
[64] Because I have daily data, also the forecasts of the two risk measures will be daily.
[65] For the complete references about the functioning and the structure of this algorithm see: https://it.mathworks.com/help/stats/kmeans.html#bue6nc4-1.
[66] It is the point whose coordinates are the mean values of the coordinates of the points of the subset. Thus, each centroid individuates the centre of each cluster.

(in my application I decided to select $k = 3$ for every sub-period and for both portfolios) highlighting three possible clusters in each series. The presence of a cluster of violations underlines a bad behaviour of the risk measure, because it means that in a relatively short time period, the forecasts are violated several times, causing unpredictable huge losses to the owner of the portfolio.

More in detail, according to the variance-covariance method, the formulas used for the VaR and the ES forecasts computation are respectively:

$$VaR = -\left(\Phi^{-1}(1-\alpha)\right) \times \sigma_R \tag{4.15}$$

$$ES = -\left(\frac{\varphi\left(\Phi^{-1}(1-\alpha)\right)}{\alpha}\right) \times \sigma_R \tag{4.16}$$

where:

- α is the confidence level;
- σ_R is the standard deviation of portfolio returns;
- $\Phi^{-1}(1-\alpha)$ is the inverse of cumulative distribution function (CDF) of a standard Normal distribution, thus it is the corresponding quantile to $1-\alpha$;
- φ is the probability density function (pdf) of a standard Normal distribution.

A static approach calculates the value of VaR or ES just once for the investigated time series, assuming it to be constant for the entire time series of returns, as I have done in the previous paragraph when calculating the value of the two risk measures for the portfolios found through the two metaheuristics in-sample and out-of-sample. For this reason, looking at formulas (4.15) and (4.16), σ_R is calculated just once and kept constant for the entire period. Thus, the value of the risk measures does not change according to market conditions: for example in the crisis periods presented in the first and third sub-periods, where the volatility of returns is higher, the value of the two risk measures should move accordingly, and be increased as σ_R increases, however this is not captured by this method.

On the other hand, implementing a dynamic approach means to impose a certain time-dependent structure in the variance-covariance matrix of portfolio returns. Thus, in this case σ_R will be time dependent. In this way it is possible to adequate the value of the risk measure to the corresponding market situation at each instant of the time series. For example, a period with higher volatility will correspond to higher values of the risk measures since we are introducing a dependence in the variance-covariance matrix with past information. To better

show this, I present the formulas of two possible approaches for modelling the dynamics of σ_R: an EWMA and a GARCH(1,1).

The EWMA models the variance as:

$$\sigma_t^2 = (1-\lambda) \cdot R_{t-1}^2 + \lambda \cdot \sigma_{t-1}^2 \tag{4.17}$$

where $\lambda = 0.94$ is a decay factor, generally ranging from 0.99 to 0.90. A lower value of it tends to give a higher weight to recent information. As we can see the current variance (at time t) is a function of portfolio return at time $t-1$, multiplied by $(1-\lambda)$, plus the variance at time $t-1$, multiplied by time λ.

On the other hand, introducing a GARCH (1,1)[67] model means to structure the time dependence of the variance in this way:

$$\sigma_t^2 = \omega + a \cdot R_{t-1}^2 + b \cdot \sigma_{t-1}^2 \tag{4.18}$$

where ω, a and b are three positive parameters. Thus, the main difference is that EWMA has only one parameter (λ), while GARCH has three. Instant time by instant, equations (4.17) and (4.18) have to be substituted into equations (4.15) and (4.16) to introduce those time dependent components, otherwise the static approach will derive.

Figure 28 shows the VaR forecasts made through these three models when applied to the PSO-ES portfolio to the first sub-period. As we can see, the static approach (VaR-ex1 in the legend) is just a straight line for the entire duration of the time series. On the other hand, EWMA (VaR-ex3 in the legend) and GARCH(1,1) (VaR-ex4 in the legend) reacts to current market conditions, thus where the volatility is higher, as for example in the last part of the time series around October 2008, their forecasts are more negative than during the relatively more stable period before. Around October 2008, it is particularly evident, at least in three points, that the blue line (representing the actual returns of the portfolio) is even more negative than the risk forecasts made through these models. Thus there are some violations of the VaR forecasts made through these models. They are actually what I intend to measure: how many violations there are in the series, their magnitude and if they are clustered.

Figure 29 shows the same things but for the ES forecasts. It can be easily noticed that around October 2008 the risk forecasts made through the three models are more negative than in the previous case, being more in line with the actual returns of the portfolio. This may suggest a better behaviour of the ES with respect to the VaR.

[67] The two numbers in brackets state the number of lags to be introduced respectively for the variance term, σ_{t-i}^2, and for R_{t-i}^2.

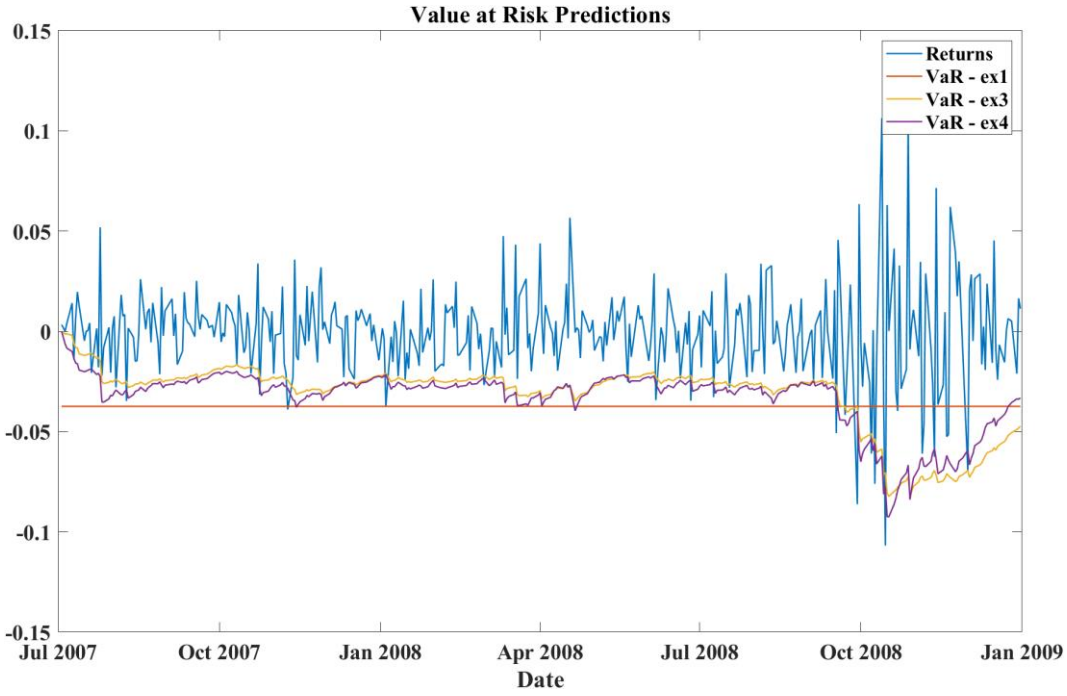


Figure 28: Historical estimates of the VaR according to the three methods for the PSO-ES portfolio in the first sub-period plotted against historical returns. VaR-ex1 is the static approach, VaR-ex3 is the EWMA and VaR-ex4 is the GARCH(1,1).

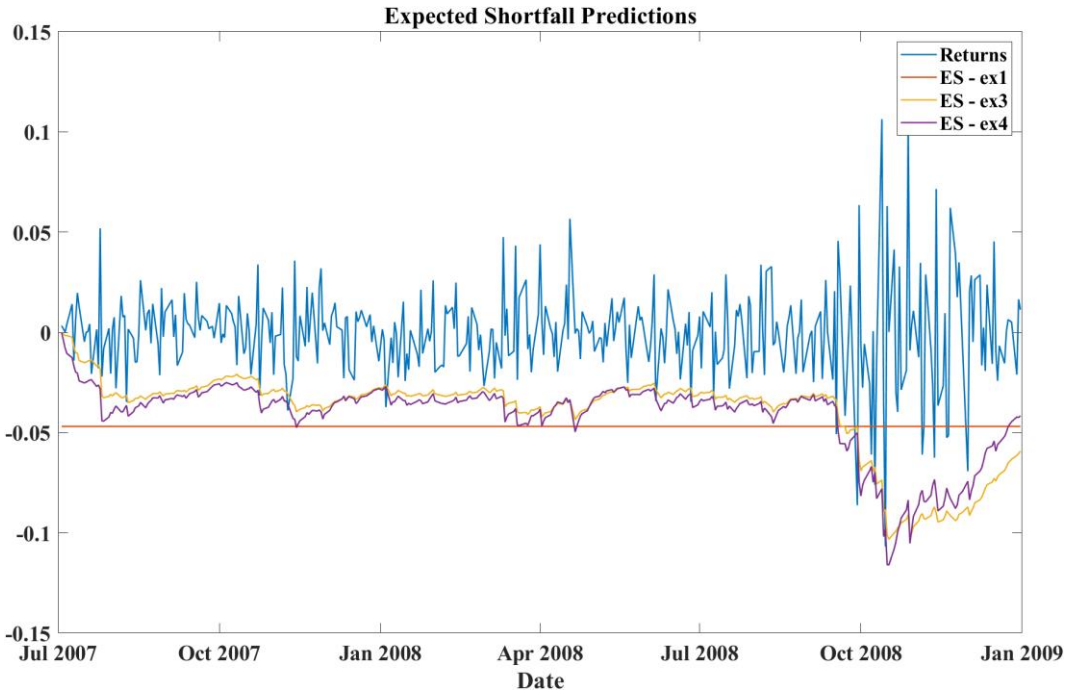


Figure 29: Historical estimates of the ES according to the three methods for the PSO-ES portfolio in the first sub-period plotted against historical returns. ES-ex1 is the static approach, ES-ex3 is the EWMA and ES-ex4 is the GARCH(1,1).

Appendix B shows the MATLAB code used for the backtesting analysis.
Appendix C shows the figures of the risk forecasts of the three approaches (static, EWMA and GARCH(1,1)) for the two risk measures plotted against portfolio returns in the three sub-periods for PSO-ES portfolios.
Appendix D shows the figures of the cluster analysis for the two risk measures applied to the three sub-periods for PSO-ES portfolio.
I decided to report only PSO-ES portfolio' results on Appendix C and D because PSO-VaR's are very similar to these, and their display would not have added any further insight to the analysis.

4.4.1 First Sub-period Backtest

Table 10 shows the results of the checks for the expected frequency of violations in the first sub-period. As we can see, according to all the three methods applied, the VaR forecasts are violated more than the ES ones for both portfolios. Considering only the VaR, among the three approaches the static one violates less than the other two, with respectively only 17 and 14 violations over 380 observations. Furthermore, in percentage terms, the number of violations is lower than 5%, which is consistent with the confidence level required. This may suggest that the static approach might be preferred to the other two, however as pointed out previously, a method like this does not react with respect to current market conditions, thus in crisis periods (like in the first sub-period) it is not able to adapt to extreme downtrends of stock prices. Looking at a concrete example, because banks have to allocate reserves with respect to the riskiness of their portfolios, during crisis periods they should allocate more reserves than in steady periods (where they can have more free capital to be invested), and a static approach is not able to capture these dynamics. Thus, it may be risky to rely on such a method.
Furthermore, the EWMA and GARCH(1,1) approaches for the VaR violate the 5% limit, being not consistent with the confidence level set. On the other hand, the ES is consistent with the confidence level for every model for both portfolios.
As expected, looking at the average magnitude of violations in Table 11, the static approach has higher magnitude both for the VaR, where the average violation is 9.82% for the PSO-VaR portfolio and 9.78% for the PSO-ES one, and for the ES, where it is 1.81% for the PSO-VaR and 2.09% for the PSO-ES. Not to mention, the magnitude of VaR violations is far

higher than ES. This suggests that the ES adapts better to changes in market conditions, being able to predict the downward movements in a better way.

FREQUENCY OF VIOLATIONS

	PSO-VaR			PSO-ES		
	Static	**EWMA**	**GARCH(1,1)**	**Static**	**EWMA**	**GARCH(1,1)**
VaR	17	24	19	14	24	19
VaR%	4.49%	6.33%	5.01%	3.69%	6.33%	5.01%
ES	11	12	9	10	14	8
ES%	2.90%	3.17%	2.37%	2.64%	3.69%	2.11%

Table 10: Results of the three approaches in terms of expected frequency of violations applied to the first sub-period.

MAGNITUDE OF VIOLATIONS

	PSO-VaR			PSO-ES		
	Static	**EWMA**	**GARCH(1,1)**	**Static**	**EWMA**	**GARCH(1,1)**
VaR	9.82%	7.50%	8.88%	9.78%	6.54%	8.07%
ES	1.81%	1.32%	1.21%	2.09%	1.00%	1.04%

Table 11: Results of the three approaches in terms of average magnitude of violations applied to the first sub-period,

VIOLATIONS CLUSTERS

	PSO-VaR			PSO-ES		
	Static	**EWMA**	**GARCH(1,1)**	**Static**	**EWMA**	**GARCH(1,1)**
VaR	(264.00; 0.094)	(312.90; 0.108)	(296.27; 0.107)	(348.33; 0.095)	(119.33; 0.051)	(323.63; 0.117)
	(144.50; 0.084)	(57.625; 0.047)	(127.00; 0.066)	(92.000; 0.076)	(282.15; 0.082)	(43.400; 0.052)
	(331.57; 0.100)	(185.29; 0.064)	(13.000; 0.051)	(318.60; 0.104)	(100.33; 0.039)	(210.17; 0.056)
ES	(264.00; 0.006)	(15.000; 0.008)	(289.20; 0.017)	(349.60; 0.012)	(100.33; 0.007)	(317.00; 0.016)
	(349.60; 0.015)	(283.25; 0.025)	(13.000; 0.001)	(306.00; 0.038)	(287.00; 0.014)	(110.50; 0.007)
	(317.60; 0.024)	(120.20; 0.007)	(139.33; 0.005)	(320.50; 0.035)	(15.000; 0.005)	(242.00; 0.003)

Table 12: Coordinates of the centres of the three centroids for the cluster analysis in the first sub-period.

Considering the cluster analysis, the results are shown in Appendix D (first sub-period) and in Table 12, which presents the coordinates of the locations of the three centroids determined

for each of the three models. The first term in the brackets indicates the coordinate on the *x* axis, which is the time instant; the second term instead indicates the coordinate on the *y* axis, which is the average magnitude of the violations inside the cluster.

Though a wise comparison of the data reported in the table and the charts reported in Appendix D it is possible to see if to each centroid would actually correspond a cluster (because the number of centroids, *k*, individuated by the algorithm has been set equal to three by default, even if the time series may show more or less clusters). For example, recalling that in the first sub-period we have 380 observations, 252 in-sample and 128 out-of-sample, it can be seen from Figure 30 that, for the PSO-ES portfolio when considering VaR forecasts, the static approach individuates three clusters, the first one on observation 348, another on observation 318 and the third cluster on observation 92. However, from Figure 30 it can be seen that the one on observation 92 is composed of only one violation, thus it is not a cluster, while the other two, that are located in the out-of-sample period (where the epicentre of the crisis is) can be considered as one cluster because of their proximity. Applying this reasoning method we can analyse all the other results. An evidence that confirms the results obtained in Table 11 is that looking at the values on the *y* axis of ES clusters for both portfolios, they are lower than VaR clusters' ones, highlighting the lower average magnitude of violations of ES forecasts over VaRs. In addition, comparing Figure 30 with Figure 31 (as well as other charts of Appendix D for the other sub-periods) it can be noticed that on average the higher concentration of the clusters is in the out-of-sample period, where the epicentre of the crisis is.

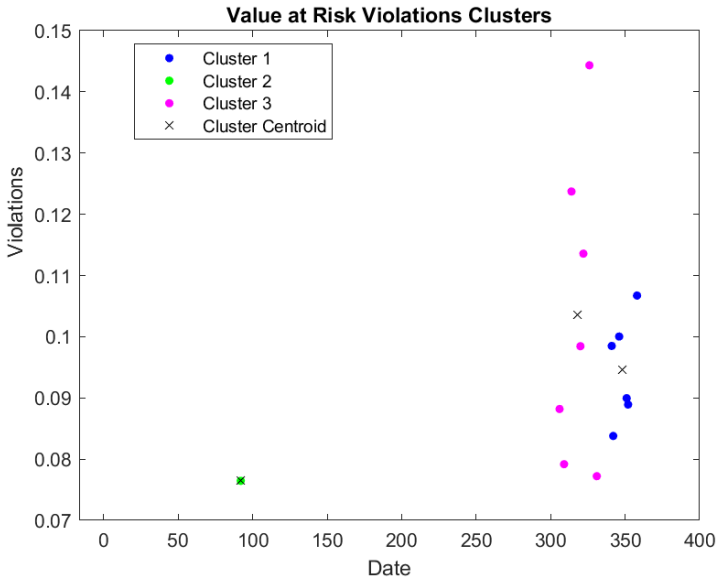


Figure 30: Cluster analysis for VaR violations of the static approach for the PSO-ES portfolio in the first sub-period. See appendix D for all the charts relative to the other models.

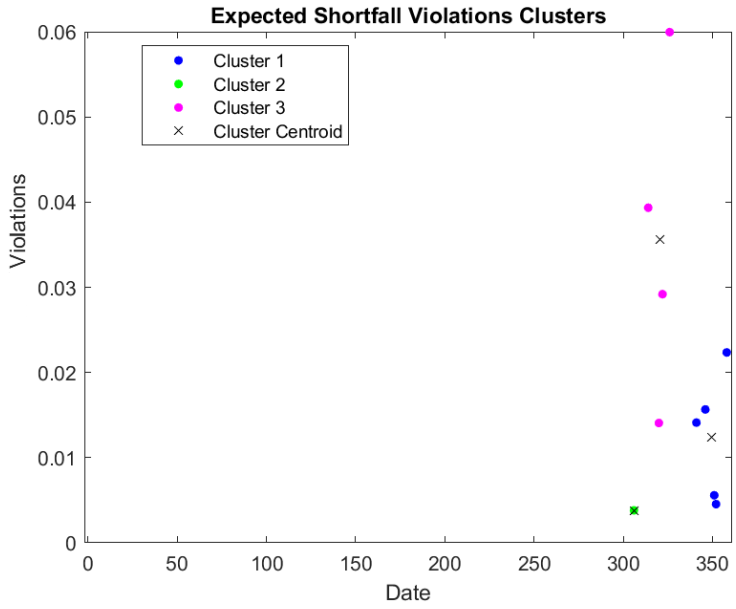


Figure 31: Cluster analysis for ES violations of the static approach for the PSO-ES portfolio in the first sub-period.

4.4.2 Second Sub-period Backtest

Table 13 shows the frequency of violations in the second sub-period. Focusing on VaR forecasts, the only method that violates the confidence level is the EWMA for both portfolios, with respectively 7.12% and 6.86% of violations. On the other hand, all the ES forecasts' violations are below the 5% bound and the model that violates them less is the GARCH(1,1). Looking at Table 14, the ES average magnitude of violations are far lower than the VaR. For the VaR case, the static approach has the highest magnitude. By comparing results of Table 11 and 14, it can be noticed that in the crisis sub-period the average magnitude of violations is higher than in the second sub-period. For example, considering the VaR forecasts for the GARCH(1,1) in the PSO-ES portfolio, they report an average magnitude of violations of 8.07% in the first sub-period and 3.35% in the second one.

Focusing now on the cluster analysis, we can see that the results shown in Table 15 are similar to the ones presented in Table 12. However by comparing the charts in the first sub-period with the ones in the second sub-period of Appendix D, it can be noticed, better than by the comparison of the results in the tables, that the clusters in the second sub-period are more dispersed. This can be due to the fact that it is not a crisis period, like in sub-period one, where the huge amount of the violations happened during a relatively small amount of days in the out-of-sample period (where there is the epicentre of the crisis). However also in this case it can be noticed that looking at ES forecasts, the y axis clusters' coordinates are lower than the

VaR ones, suggesting a lower average magnitude of violations. These results are in accordance with what was found in Table 14.

FREQUENCY OF VIOLATIONS

	PSO-VaR			PSO-ES		
	Static	**EWMA**	**GARCH(1,1)**	**Static**	**EWMA**	**GARCH(1,1)**
VaR	15	27	14	15	26	17
VaR%	3.96%	7.12%	3.69%	3.96%	6.86%	4.49%
ES	7	14	6	9	12	7
ES%	1.85%	3.69%	1.58%	2.37%	3.17%	1.85%

Table 13: Results of the three approaches in terms of expected frequency of violations applied to the second sub-period.

MAGNITUDE OF VIOLATIONS

	PSO-VaR			PSO-ES		
	Static	**EWMA**	**GARCH(1,1)**	**Static**	**EWMA**	**GARCH(1,1)**
VaR	4.09%	3.08%	3.81%	3.54%	3.04%	3.35%
ES	0.81%	0.52%	0.66%	0.81%	0.62%	0.91%

Table 14: : Results of the three approaches in terms of average magnitude of violations applied to the second sub-period,

VIOLATIONS CLUSTERS

	PSO-VaR			PSO-ES		
	Static	**EWMA**	**GARCH(1,1)**	**Static**	**EWMA**	**GARCH(1,1)**
	(326.25; 0.038)	(173.00; 0.050)	(20.500; 0.016)	(333.20; 0.031)	(102.33; 0.032)	(126.20; 0.036)
VaR	(197.40; 0.043)	(323.53; 0.031)	(333.83; 0.037)	(152.00; 0.038)	(335.89; 0.027)	(316.00; 0.032)
	(146.00; 0.050)	(27.439; 0.019)	(179.67; 0.045)	(240.25; 0.038)	(224.63; 0.033)	(200.00; 0.033)
	(176.20; 0.009)	(173.00; 0.078)	(133.00; 0.007)	(146.67; 0.013)	(306.83; 0.005)	(236.00; 0.008)
ES	(369.00; 0.006)	(333.83; 0.004)	(369.00; 0.003)	(343.5; 0.002)	(32.000; 0.003)	(365.00; 0.002)
	(273.00; 0.004)	(19.333; 0.0026)	(273.00; 0.008)	(236.00; 0.008)	(165.60; 0.008)	(145.50; 0.014)

Table 15: Coordinates of the centres of the three centroids for the cluster analysis in the second sub-period.

4.4.3 Third Sub-period Backtest

Table 15 reports the frequency of violations in the third sub-period. Looking at VaR's forecasts, both the EWMA and the GARCH(1,1) approaches slightly violate the confidence level. The static approach again is the one with lower number of violations for the VaR's forecasts. Focusing on the ES, the frequency of violations is lower than 5% in every model for both portfolios.

Looking at Table 16, the VaR average magnitude of violations is far higher than the ES as in every other sub-period. However, in general, the magnitude of violations is similar to the one of the first sub-period in fact they both consider crisis periods. However, because the first sub-period considers a longer period of instability than the third sub-period, where the Covid-19 crisis breaks out only in the out-of-sample period, this explains the slightly higher values reported in Table 11.

FREQUENCY OF VIOLATIONS

	PSO-VaR			PSO-ES		
	Static	**EWMA**	**GARCH(1,1)**	**Static**	**EWMA**	**GARCH(1,1)**
VaR	17	24	19	14	19	20
VaR%	4.52%	6.38%	5.05%	3.72%	5.05%	5.32%
ES	13	12	11	10	16	13
ES%	3.46%	3.19%	2.93%	2.66%	4.26%	3.46%

Table 16: Results of the three approaches in terms of expected frequency of violations applied to the third sub-period.

MAGNITUDE OF VIOLATIONS

	PSO-VaR			PSO-ES		
	Static	**EWMA**	**GARCH(1,1)**	**Static**	**EWMA**	**GARCH(1,1)**
VaR	8.15%	6.28%	7.43%	7.11%	5.88%	5.39%
ES	1.80%	1.82%	1.36%	1.82%	0.98%	0.92%

Table 17: : Results of the three approaches in terms of average magnitude of violations applied to the third sub-period,

With respect to the cluster analysis, Table 18 and Appendix D (third sub-period) show the results obtained. It can be noticed, especially from the graphical analysis in Appendix D, that the results are more similar to the ones of the first sub-period, where the epicentre of the crisis,

like in the third sub-period, is out-of-sample. For instance, it can be seen that the biggest clusters are on the far right of the charts, around observations 300 and 350, highlighting the fact that during the crisis outbreak there were clustering violations for both risk measures' forecasts.

VIOLATIONS CLUSTERS

	PSO-VaR			PSO-ES		
	Static	**EWMA**	**GARCH(1,1)**	**Static**	**EWMA**	**GARCH(1,1)**
VaR	(301.75; 0.084)	(34.800; 0.041)	(107.63; 0.049)	(300.67; 0.075)	(22.500; 0.040)	(316.50; 0.074)
	(111.00; 0.073)	(301.62; 0.078)	(302.43; 0.113)	(108.00; 0.057)	(303.11; 0.079)	(26.800; 0.040)
	(368.50; 0.079)	(135.50; 0.049)	(360.25; 0.058)	(349.00; 0.075)	(140.33; 0.041)	(137.43; 0.041)
ES	(301.56; 0.022)	(62.000; 0.026)	(369.67; 0.014)	(297.29; 0.021)	(210.57; 0.013)	(7.0000; 0.012)
	(92.500; 0.009)	(288.00; 0.016)	(99.200; 0.015)	(21.000; 0.000)	(22.500; 0.007)	(308.29; 0.010)
	(368.50; 0.009)	(368.50; 0.014)	(291.33; 0.011)	(349.00; 0.016)	(131.80; 0.007)	(126.00; 0.005)

Table 18: Coordinates of the centres of the three centroids for the cluster analysis in the third sub-period.

Conclusions

The aim of this dissertation is to assess Model Risk deriving from the application of two selected risk measures to a realistic financial problem, so that I can compare their results in terms of Model Risk. For doing this I decided to follow the idea of Boucher et al. (2014) presented in Chapter 1, considering the three properties that a good risk forecast should possess: expected frequency of violations, absence of violations clustering and expected magnitude of violations.

Then, in the proposed experiment, I applied the two selected risk measures, VaR and ES, to a more realistic version of the original portfolio selection problem, where cardinality constraints were introduced. By introducing these constraints, the portfolio selection problem became harder to be solved through exact methods, and approximated algorithms were required. Thus, I selected two of these algorithms, the well known PSO and the more recent GWO, in order to compare their performances in terms of Model Risk. In fact, since they are approximated techniques, another component of Model Risk rose. To take this new risk component into consideration I considered three drivers: the best fitness value obtained, the number and the magnitude of the violations of the constraints and the computational time required by the algorithm to fully execute the code.

For the application I selected eleven securities from the S&P 500 and three different scenarios, two comprising crisis periods (to stress the risk measures), and one focused on a growth period of the US economy.

Thus, in Chapter 4, I applied the two metaheuristics considering both VaR and ES as objective functions to be minimized, building four different portfolios in each of the three scenarios. From the data reported in *subparagraphs 4.3.1*, *4.3.2* and *4.3.3* emerged that PSO is a more reliable algorithm for solving this complex optimization problem: it was able to achieve better fitness values, it violated less constraints and with lower magnitude and it required less computational time to be executed in every sub-period. For these reasons, I decided to pick only the two PSO's portfolios for a further backtest analysis. This analysis was presented in paragraph 4.4, and aimed to assess Model Risk for risk measures, focusing on the three properties of a good risk forecast, by historically modelling the risk forecasts made by the two risk measures on the data of each sub-period. I have repeated these backtests for each of the three sub-periods and I compared the results obtained by VaR with the ones

obtained by ES. For each sub-period it emerged that the ES forecasts were violated less than VaRs, being always under the 5% confidence level. Furthermore, when the forecasts were violated, the ES violations were on average far lower than the VaRs. Thus ES was able to capture the higher riskiness of that specific period better than VaR, providing a more negative forecast, which was more in line with the actual negative return. In addition, from the cluster analysis emerged that the ES violations are less clustered than the VaR, highlighting less consecutive violations on close days, and this is particularly important during crisis periods.
However, it could be argued that since the comparison between the two metaheuristics have been done imposing the same number of iterations and searching agents for PSO and GWO, and GWO's portfolios have in all cases worse fitness values, it could be that a higher number of iterations or searching agents might be needed by this algorithm to ensure a better convergence towards the optimum. In fact, as previously shown by the comparison between Figure 26 and Figure 27, GWO has a far lower convergence speed than PSO. A possible explanation might be that GWO is better exploring the searching space, devoting more effort and more iterations to find promising areas, as a consequence resulting in a slower exploiting phase, i.e. in a slower convergence.
The two parameters that can be modified in order to try to ensure a better convergence, and thus a better solution, are the number of iterations and the number of searching agents. However, as previously pointed out, increasing these two components means to increase the computational effort required, resulting in a higher computational time. So, there is a trade off between increasing these parameters and the time required by the algorithm.
For this reason I decided to implement two further tests considering only the two GWO's versions (GWO-VaR and GWO-ES) in the third sub-period (where they achieve better results in terms of fitness value) in order to see if by increasing one of the two parameters (while keeping the other constant) the solutions improve and if they are at least in line (in terms of quality) with the ones obtained by PSO. More in detail, for the first test I decided to run the code by doubling the number of iterations (now equal to 6000) while keeping the number of searching agents constant (equal to 70). On the other hand, the second test is done by keeping the number of iterations constant (equal to 3000) and by doubling the number of searching agents (now equal to 140). I am not going to report the tables with the results, as previously done, but I am just going to present the comments on the results. Recalling that

on average GWO takes 30 seconds to be executed[68] with 3000 iterations and 70 searching agents, in summary the results of both tests showed just a small improvement in the fitness value obtained but a far higher time required. The results of the first test highlighted a small improvement in the quality of the solution for both portfolios with respect to GWO's versions with 3000 iterations and 70 wolves. The new fitness values found are respectively of 8.93E+02 for the GWO-VaR portfolio and 1.27E+03 for the GWO-ES, that are slightly better than the ones presented in Table 8. However, they still remain far from the ones of PSO. Furthermore, these results come with a big cost, in fact the computational time is more than doubled, with an average of 65 seconds. Also the second test showed a small improvement in the convergence of the algorithms, thus better quality solutions are found. The new fitness values are respectively of 6.73E+02 for the GWO-VaR portfolio and 8.46E+02 for the GWO-ES, that are slightly better than the ones presented in Table 8. As we can see, also in these cases their quality still remains distant from PSO's results. Nonetheless, they come with the huge cost of increasing the computational time more than four times, with an average of 133 seconds. To sum it up, by doubling the number of iterations, the time required more than doubled; by doubling the number of searching agents the time required is four times more for both versions. In conclusion to these tests, by increasing these two parameters there is small improvement in the quality of the solutions, that however still remain far from the ones obtained by PSO. Moreover, there is an unreasonable trade-off between this small improvement and the resulting computational time.

Concluding my work, having collected the results of this further analysis as another clue, I shall advise in favour of PSO algorithm for solving this version of the portfolio selection problem, because it is able to provide a more reliable output that better approximates the real optimum solution. Furthermore, the more reliable financial risk measure in terms of Model Risk, that obtained better results in the backtesting analysis for every sub-period, is the Expected Shortfall. Being a coherent risk measure (Artzner et al., 1999), it is able to overcome the performance of the Value at Risk, which can however still be considered an important step forward from the variance, adopted in the Markowitz model.

[68] On MATLAB R2020b in a Windows 10 pc with Intel Core i5-1035G4 processor and 8GB of RAM.

Appendix A

MATLAB Code of the PSO – Expected Shortfall:

```
clc
clear
format long

%% Input of the problem
% Uploading historical data and computing logarithmic returns
[prices] = importdata("Data.xlsx");
[t,n_asset] = size(prices); % t is the period of time, n_asset is the number of assets considered
returns = log(prices(2:end,:)./prices(1:end-1,:)); % log returns
C = 100; % capital invested

% Differentiation between in-sample and out-of-sample
oos = 127; % number of returns out-of-sample +1
ris = returns(1:t-oos,:); %returns in-sample
ros = returns(t-oos+1:(t-1),:); %returns out-of-sample
TT = (t-oos);
rm_is = mean(ris); % In-sample mean returns
rm_oos = mean(ros); % Out-of-sample mean returns

% Data input
pi = mean(rm_is); % desired daily minimum return, set equal to the portfolio mean return over the in-sample period
alpha = 0.05; % significance level of ES
Kl = 5; % minimum number of assets
Ku = 9; % maximum number of assets
l = ones(1,n_asset)*0.03; % minimum percentage of investment in each asset
u = ones(1,n_asset)*0.20; % maximum percentage of investment in each asset

% PSO parameters initialization
P = 70; % particles number
niter = 3000; % iterations number
c1 = 1.49618; % individual acceleration coefficient
c2 = 1.49618; % social acceleration coefficient
iw = 0.7298 ; % inertia weight
vmaxx = zeros(1,n_asset);
vmaxz = zeros(1,n_asset);
epsilon = 1e-05; % parameter that penalizes violations of constraints

% Creation of vectors useful for objective function
% Risk measure
ES_port = zeros(P,1);
R_is = zeros(TT,P);
```

```
sorted_R_is = zeros(TT,P);
VaR = zeros(P,1);
% Constraints
constr_1 = zeros(P,1); % budget constraint
constr_2 = zeros(P,1); % desired minimum return constraint
constr_3 = zeros(P,1); % minimum number of asset constraint (z>=Kl)
constr_4 = zeros(P,1); % maximum number of asset constraint (z<=Ku)
app_5 = zeros(P,n_asset);
constr_5 = zeros(P,1); % minimum percentage of investment constraint (x>=l)
app_6 = zeros(P,n_asset);
constr_6 = zeros(P,1); % maximum percentage of investment constraint (x<=u)
app_7 = zeros(P,n_asset);
constr_7 = zeros(P,1); % z is either 0 or 1

%% Computation

% 1-Generation of position and velocity vectors and setting of fitness function
x = rand(P,n_asset);
vx = rand(P,n_asset);
z = rand(P,n_asset);
vz = rand(P,n_asset);
f = ones(P,1)*1.0e+255; % fitness function
x1 = zeros(P,n_asset); % matrix which state if the asset is in the portfolio (x*z)
% pb=pbest: vector with the best position of particles in previous iterations
pbx = [x f];
pbz = z;

% g=gbest: vector with the best global position and the associated objective
function's value
gx = zeros(1,n_asset+1);
gz = zeros(1,n_asset);

% Beginning of the loop
tic; % measuring time spent in the computation
for k=1:niter % Identifying dynamic range for maximum velocity
    for i=1:n_asset
    vmaxx(i) = abs(max(x(:,i))-min(x(:,i)));
    vmaxz(i) = abs(max(z(:,i))-min(z(:,i)));
    end
% 2-Objective function computation
     for p=1:P
         for i=1:n_asset
         x1(p,i) = x(p,i)*z(p,i);
         app_5(p,i) = max(0,l(i)*z(p,i)-x(p,i));
         app_6(p,i) = max(0,x(p,i)-u(i)*z(p,i));
         app_7(p,i) = abs(z(p,i)*(1-z(p,i)));
         end
    % Calculate portfolio returns for each particle (at its position)
    R_is(:,p)=ris*x(p,:)'; % TTxP matrix
    % Sort portfolio returns
```

```
    sorted_R_is = sort(R_is); % TTxP matrix
    % Store the number of returns
    num_returns_is = numel(R_is(:,1));
    % Calculate the index of the sorted return that will be VaR
    VaR_index_is = ceil((alpha)*num_returns_is);
    % Use the index to extract VaR from sorted returns
    VaR(p) = sorted_R_is(VaR_index_is,p);
    % Calculate historical ES
    ES_port(p) = -mean(sorted_R_is(1:VaR_index_is,p));


    % Sum of investment percentages equal to 1
    constr_1(p) = abs(sum(x1(p,:))-1);
    % Expected return at least equal to pi
    constr_2(p) = max(0,(pi-sum(x1(p,:)*rm_is')));
    % Minimum number of assets Kl
    constr_3(p) = max(0,Kl-sum(z(p,:)));
    % Maximum number of assets Ku
    constr_4(p) = max(0,sum(z(p,:))-Ku);
    % Minimum percentage l
    constr_5(p) = sum(app_5(p,:));
    % Maximum percentage u
    constr_6(p) = sum(app_6(p,:));
    % z is either 0 or 1
    constr_7(p) = sum(app_7(p,:));
    end

% Objective function
f =
(ES_port+(1/epsilon)*(constr_1+constr_2+constr_3+constr_4+constr_5+constr_6
+constr_7));

% 3-Comparing objective function's value with pbest
    for p=1:P
        if f(p) <pbx(p,n_asset+1)
        pbx(p,n_asset+1) = f(p);
        for i=1:n_asset
        pbx(p,i) = x1(p,i);
        pbz(p,i) = z(p,i);
        end
    end
end

% 4-Identifying the particle with the best position (g)
[minimum,position] = min(pbx(:,n_asset+1));
gx(n_asset+1) = minimum;
    for i=1:n_asset
    gx(i) = pbx(position,i);
    gz(i) = pbz(position,i);
    end
```

```
% 5-Updating velocity and position
    for p=1:P
        for i=1:n_asset
        vx(p,i) = iw*vx(p,i)+c1*rand*(pbx(p,i)-x(p,i))+c2*rand*(gx(i)- x(p,i));
        vz(p,i) = iw*vz(p,i)+c1*rand*(pbz(p,i)-z(p,i))+c2*rand*(gz(i)- z(p,i));
                if vx(p,i)>vmaxx(i) % maximum velocity limitation
                vx(p,i) = vmaxx(i);
                end
                if vz(p,i)>vmaxz(i) % maximum velocity limitation
                vz(p,i) = vmaxz(i);
                end
         x(p,i) = x(p,i)+vx(p,i);
         z(p,i) = z(p,i)+vz(p,i);
         end
    end
converg(k,:) = gx(:,end);

% 6-Go back to Step 2 until stop condition
end
% End of loop

toc;

%% Output of the problem
% Results
optimum_shares = gx(1,1:n_asset)'
best_fitness = gx(1,n_asset+1)
n_selected_assets = sum(z(position,:))
wealth_invested_percent = sum(optimum_shares)
constr_budget = constr_1(position)
constr_return = constr_2(position)
constr_Kl = constr_3(position)
constr_Ku = constr_4(position)
constr_min_share = constr_5(position)
constr_max_share = constr_6(position)
constr_z = constr_7(position)

% In sample analysis
ES_is = ES_port*C;
ES_in_sample = ES_is(position)
rm_in_sample = gx(1,1:n_asset)*rm_is'

% Out of sample analysis
R_oos=ros*optimum_shares;
sorted_R_oos = sort(R_oos); % TTxP matrix
num_returns_oos = numel(R_oos);
VaR_index_oos = ceil((alpha)*num_returns_oos);
ES_oos = -mean(sorted_R_oos(1:VaR_index_oos));
ES_out_of_sample = ES_oos*C
```

```
rm_out_of_sample= gx(1,1:n_asset)*rm_oos'

% Graph comparing the behaviour of the fitness function (Y axis) in relation to
the number of iterations made (X axis)
plot(converg,'Color','r','LineWidth',1.25)
title('Objective space')
xlabel('Iteration');
ylabel('Best score obtained so far');
axis tight
grid on
box on
legend('PSO')
```

MATLAB Code of the GWO – Expected Shortfall:

Main:

```
clear
clc
format long

% Uploading historical data and computing logarithmic returns
[prices] = importdata("Data.xlsx");

SearchAgents_no=70; % Number of search agents
Max_iteration=3000; % Maximum number of iterations
C = 100; % capital invested

% Load details of the selected benchmark function
lb=-1e18;
ub=1e18;
dim=11;

% dividing in sample and out of sample periods
[t,n_asset] = size(prices); % t is the period of time, n_asset is the number of assets
considered
returns = log(prices(2:end,:)./prices(1:end-1,:)); % computing log returns
oos = 127; % number of returns out-of-sample +1
ris = returns(1:t-oos,:); %returns in-sample
ros = returns(t-oos+1:(t-1),:); %returns out-of-sample
TT = (t-oos);
rm_is = mean(ris); % In-sample mean returns
rm_oos = mean(ros); % Out-of-sample mean returns


[Best_constraints,Best_score,Best_pos,Best_pos_z,GWO_cg_curve,alpha,ES_por
t]=GWO_ES(SearchAgents_no,Max_iteration,lb,ub,dim,prices);

%****Output of the problem****
```

```
% Results of the optimization problem
optimum_shares = Best_pos(1,1:dim)'
best_fitness = Best_score
n_selected_assets = sum(Best_pos_z)
wealth_invested_percent = sum(optimum_shares)
display(Best_constraints')

% In sample analysis
ES_is = mean(ES_port);
ES_in_sample = mean(ES_port).*C
rm_in_sample = Best_pos(1,1:dim)*rm_is'

% Out of sample analysis
R_oos=ros*optimum_shares;
sorted_R_oos = sort(R_oos); % TTxP matrix
num_returns_oos = numel(R_oos);
VaR_index_oos = ceil((alpha)*num_returns_oos);
ES_oos = -mean(sorted_R_oos(1:VaR_index_oos));
ES_out_of_sample = ES_oos*C
rm_out_of_sample= Best_pos(1,1:dim)*rm_oos'

%Draw objective space
plot(GWO_cg_curve,'Color','r','LineWidth',1.25)
title('Objective space')
xlabel('Iteration');
ylabel('Best score obtained so far');
axis tight
grid on
box on
legend('GWO')
```

Initialization function:

```
% This function initialize the first population of search agents
function [Positions_x,
Positions_z]=initialization_ES(SearchAgents_no,dim,ub,lb)

Boundary_no= size(ub,2); % number of boundaries

if Boundary_no==1
        Positions_x = rand(SearchAgents_no,dim);
    Positions_z = rand(SearchAgents_no,dim);
end

% If each variable has a different lb and ub
if Boundary_no>1
    for i=1:dim
        ub_i=ub(i);
        lb_i=lb(i);
```

```
        Positions_x(:,i)=rand(SearchAgents_no,1);
        Positions_z(:,i)=rand(SearchAgents_no,1);
    end
end
```

GWO function:

```
% Grey Wolf Optimizer
function
[Alpha_constraints,Alpha_score_x,Alpha_pos_x,Alpha_pos_z,Convergence_curv
e,alpha,ES_port]=GWO_ES(SearchAgents_no,Max_iter,lb,ub,dim,prices)

%****initialize alpha, beta, and delta_pos****
Alpha_constraints= zeros(1,dim)';
Alpha_pos_x=zeros(1,dim);
Alpha_pos_z=zeros(1,dim);
Alpha_score_x=inf; %change this to -inf for maximization problems
Alpha_score_z=inf; %change this to -inf for maximization problems
Beta_pos_x=zeros(1,dim);
Beta_pos_z=zeros(1,dim);
Beta_score_x=inf; %change this to -inf for maximization problems
Beta_score_z=inf; %change this to -inf for maximization problems
Delta_pos_x=zeros(1,dim);
Delta_pos_z=zeros(1,dim);
Delta_score_x=inf; %change this to -inf for maximization problems
Delta_score_z=inf; %change this to -inf for maximization problems

% Initialize prices
[t,n_asset] = size(prices); % t is the period of time, n_asset is the number of assets
considered
returns = log(prices(2:end,:)./prices(1:end-1,:)); % compute the log returns

% Differentiation between in-sample and out-of-sample
oos = 127; % number of returns out-of-sample +1
ris = returns(1:t-oos,:); %returns in-sample
ros = returns(t-oos+1:(t-1),:); %returns out-of-sample
TT = (t-oos);
rm_is = mean(ris); % In-sample mean returns
rm_oos = mean(ros); % Out-of-sample mean returns

% Data input
pi = mean(rm_is); % desired daily minimum return, set equal to the portfolio
mean return over the in-sample period
alpha = 0.05; % significance level of ES
Kl = 5; % minimum number of assets
Ku = 9; % maximum number of assets
l = ones(1,n_asset)*0.03; % minimum percentage of investment in each asset
u = ones(1,n_asset)*0.20; % maximum percentage of investment in each asset
```

```
epsilon = 1e-05; % parameter that penalizes violations of constraints

%****Creation of vectors useful for objective function****
% Risk measure
R_is = zeros(TT,SearchAgents_no);
sorted_R_is = zeros(TT,SearchAgents_no);
VaR = zeros(SearchAgents_no,1);
ES_port = zeros(SearchAgents_no,1);

% Constraints
constr_1 = zeros(SearchAgents_no,1); % budget constraint
constr_2 = zeros(SearchAgents_no,1); % desired minimum return constraint
constr_3 = zeros(SearchAgents_no,1); % minimum number of asset constraint
(z>=Kl)
constr_4 = zeros(SearchAgents_no,1); % maximum number of asset constraint
(z<=Ku)
app_5 = zeros(SearchAgents_no,dim);
constr_5 = zeros(SearchAgents_no,1); % minimum percentage of investment
constraint (x>=l)
app_6 = zeros(SearchAgents_no,dim);
constr_6 = zeros(SearchAgents_no,1); % maximum percentage of investment
constraint (x<=u)
app_7 = zeros(SearchAgents_no,dim);
constr_7 = zeros(SearchAgents_no,1); % z is either 0 or 1

%Initialize the positions of search agents
[Positions_x, Positions_z]=initialization_ES(SearchAgents_no,dim,ub,lb);
Convergence_curve=zeros(1,Max_iter);
Positions_x1 = zeros(SearchAgents_no,dim); % matrix which state if the asset is
in the portfolio (x*z)

%****Main loop****
loop_counter=0;% Loop counter
tic;

while loop_counter<Max_iter
   for p=1:SearchAgents_no  % iterate particles

      %Return back the search agents that go beyond the boundaries of the search
space
       Flag4ub=Positions_x(p,:)>ub;
       Flag4lb=Positions_x(p,:)<lb;

Positions_x(p,:)=(Positions_x(p,:).*(~(Flag4ub+Flag4lb)))+ub.*Flag4ub+lb.*Flag
4lb;

Positions_z(p,:)=(Positions_z(p,:).*(~(Flag4ub+Flag4lb)))+ub.*Flag4ub+lb.*Flag
4lb;

       % **** APP ****
```

```
        for j=1:dim          % iterate titles
            Positions_x1(p,j) = Positions_x(p,j)*Positions_z(p,j);
            app_5(p,j) = max(0,l(j)*Positions_z(p,j)-Positions_x(p,j));
            app_6(p,j) = max(0,Positions_x(p,j)-u(j)*Positions_z(p,j));
            app_7(p,j) = abs(Positions_z(p,j)*(1-Positions_z(p,j)));
        end

        % **** VAR *****
        % Calculate portfolio returns for each particle (at its position)
        R_is(:,p)=ris*Positions_x(p,:)'; % TTxP matrix
        % Sort portfolio returns
        sorted_R_is = sort(R_is); % TTxP matrix
        % Store the number of returns
        num_returns_is = numel(R_is(:,1));
        % Calculate the index of the sorted return that will be VaR
        VaR_index_is = ceil((alpha)*num_returns_is);
        % Use the index to extract VaR from sorted returns
        VaR(p) = -sorted_R_is(VaR_index_is,p);
        % Calculate historical ES
        ES_port(p) = -mean(sorted_R_is(1:VaR_index_is,p));

        % **** CONSTR ****
        % Sum of investment percentages equal to 1
        constr_1(p) = abs(sum(Positions_x1(p,:))-1);
        % Expected return at least equal to pi
        constr_2(p) = max(0,(pi-sum(Positions_x1(p,:)*rm_is')));
        % Minimum number of assets Kl
        constr_3(p) = max(0,Kl-sum(Positions_z(p,:)));
        % Maximum number of assets Ku
        constr_4(p) = max(0,sum(Positions_z(p,:))-Ku);
        % Minimum percentage l
        constr_5(p) = sum(app_5(p,:));
        % Maximum percentage u
        constr_6(p) = sum(app_6(p,:));
        % z is either 0 or 1
        constr_7(p) = sum(app_7(p,:));

        % % Calculate objective function for each search agent
        fobj =
(ES_port+(1/epsilon)*(constr_1+constr_2+constr_3+constr_4+constr_5+constr_6
+constr_7));

        fitness=fobj(p);

        % Update Alpha_x, Beta_x, and Delta_x
        if fitness<Alpha_score_x
            Alpha_score_x=fitness; % Update alpha
            Alpha_pos_x=Positions_x1(p,:);
```

```
            Alpha_constraints= [constr_1(p) constr_2(p) constr_3(p) constr_4(p)
constr_5(p) constr_6(p) constr_7(p)];
        end

        if fitness>Alpha_score_x && fitness<Beta_score_x
            Beta_score_x=fitness; % Update beta
            Beta_pos_x=Positions_x1(p,:);
        end

        if fitness>Alpha_score_x && fitness>Beta_score_x &&
fitness<Delta_score_x
            Delta_score_x=fitness; % Update delta
            Delta_pos_x=Positions_x1(p,:);
        end

        % Update Alpha_z, Beta_z, and Delta_z
        if fitness<Alpha_score_z
           Alpha_score_z=fitness;
           Alpha_pos_z=Positions_z(p,:);
        end

        if fitness>Alpha_score_z && fitness<Beta_score_z
           Beta_score_z=fitness; % Update beta
           Beta_pos_z=Positions_z(p,:);
        end

        if fitness>Alpha_score_z && fitness>Beta_score_z &&
fitness<Delta_score_z
            Delta_score_z=fitness; % Update delta
            Delta_pos_z=Positions_z(p,:);
        end
    end

    a=2-loop_counter*((2)/Max_iter); % a decreases linearly from 2 to 0

    % Update the Position of search agents including omegas
    for p=1:SearchAgents_no
        for j=1:dim

            %***Alpha_x***
            r1=rand(); % r1 is a random number in [0,1]
            r2=rand(); % r2 is a random number in [0,1]

            A1=2*a*r1-a;
            C1=2*r2;

            D_alpha_x=abs(C1*Alpha_pos_x(j)-Positions_x(p,j));
            X1_x=Alpha_pos_x(j)-A1*D_alpha_x;

            %***Alpha_z***
```

```
r1=rand(); % r1 is a random number in [0,1]
r2=rand(); % r2 is a random number in [0,1]

A1=2*a*r1-a;
C1=2*r2;

D_alpha_z=abs(C1*Alpha_pos_z(j)-Positions_z(p,j));
X1_z=Alpha_pos_z(j)-A1*D_alpha_z;

%***Beta_x***
r1=rand();
r2=rand();

A2=2*a*r1-a;
C2=2*r2;

D_beta_x=abs(C2*Beta_pos_x(j)-Positions_x(p,j));
X2_x=Beta_pos_x(j)-A2*D_beta_x;

%***Beta_z***
r1=rand(); % r1 is a random number in [0,1]
r2=rand(); % r2 is a random number in [0,1]

A1=2*a*r1-a;
C1=2*r2;

D_beta_z=abs(C2*Beta_pos_z(j)-Positions_z(p,j));
X2_z=Beta_pos_z(j)-A2*D_beta_z;

%***Delta_x***
r1=rand();
r2=rand();

A3=2*a*r1-a;
C3=2*r2;

D_delta_x=abs(C3*Delta_pos_x(j)-Positions_x(p,j));
X3_x=Delta_pos_x(j)-A3*D_delta_x;

%***Delta_z***
r1=rand(); % r1 is a random number in [0,1]
r2=rand(); % r2 is a random number in [0,1]

A1=2*a*r1-a;
C1=2*r2;

D_delta_z=abs(C3*Delta_pos_z(j)-Positions_z(p,j));
X3_z=Delta_pos_z(j)-A3*D_delta_z;

%***Omega_x and Omega_z***
```

```
            Positions_x(p,j)=(X1_x+X2_x+X3_x)/3;
            Positions_z(p,j)=(X1_z+X2_z+X3_z)/3 ;

        end
    end
    loop_counter=loop_counter+1;
    Convergence_curve(loop_counter)=Alpha_score_x;

end

toc;
```

Appendix B

MATLAB Code for the Backtesting Analysis

```
clear
clc
close all

%% Upload dataset
DataSet = xlsread('Data_3');

DateTime=readtable('Calendar_3');
DateTime.Var1=datetime(DateTime.Var1,'InputFormat','MMM-yy');

%% Portfolios

returns_AA = diff(log(DataSet(:,26)));
%25->PSO VaR
%26->PSO ES
%27->GWO VaR
%28->GWO ES

a = 0.05;
N_crit_value = norminv(a);
ES_crit_value = normpdf(N_crit_value);
%% 4 different models

% ****Model 1 - no time-varying****
% r_t = eps_t , eps_t - iid N(0, sigma2)
% N(0, sigma2) = sigma*N(0,1)
% histfit(returns_AA)

standard_dev_ex1 = std(returns_AA);
VaR_ex1 = N_crit_value*standard_dev_ex1;
ES_ex1 = -standard_dev_ex1*(ES_crit_value/a);


% ****Model 3 - EWMA****
% r_t = sigma_t * eps_t , eps_t - N(0,1)
% sigma_t^2 = (1-\lamba)*r_{t-1}^2 + lambda*sigma_{t-1}^2
% r_t | I_{t-1} - N(0, sigma_t^2)

T = size(returns_AA,1);
lambda = 0.94;
sigma2_ex3 = zeros(T,1);

for t = 2 : T
```

```
sigma2_ex3(t,:) = (1-lambda)*returns_AA(t-1,:)^2+lambda*sigma2_ex3(t-1,:);
VaR_ex3(t,:) = N_crit_value*(sigma2_ex3(t,:)^0.5);
ES_ex3(t,:) = -(sigma2_ex3(t,:)^0.5)*(ES_crit_value/a);
end

%%
% ****Model 4 - GARCH(1,1) - ARCH(2)****
% r_t = sigma_t * eps_t , eps_t - N(0,1)
% sigma_t^2 = omega + alpha*r_{t-1}^2 + beta*sigma_{t-1}^2 (GARCH(1,1))
% sigma_t^2 = omega + alpha1*r_{t-1}^2 + alpha2*r_{t-2}^^2 (ARCH(2))
% r_t | I_{t-1} - N(0, sigma_t^2)

% ARCH(2)
model_ARCH_2 = garch(0,2);
[est_ARCH2, est_SE_ARCH2, logL_ARCH2] =
estimate(model_ARCH_2,returns_AA);
% GARCH(1,1)
model_GARCH_11 = garch(1,1);
[est_GARCH11, est_SE_GARCH11, logL_GARCH11] =
estimate(model_GARCH_11,returns_AA);

% Akaike: -2*loglk + 2*k
Akaike_ARCH2 = -2*logL_ARCH2+2*3;
Akaike_GARCH11 = -2*logL_GARCH11+2*3;
% BIC: -2*loglk + k*log(T)
BIC_ARCH2 = -2*logL_ARCH2+3*log(T);
BIC_GARCH11 = -2*logL_GARCH11+3*log(T);

% we have selected GARCH(1,1)
omega_hat = est_GARCH11.Constant;
alpha_hat = est_GARCH11.ARCH{1,1};
beta_hat = est_GARCH11.GARCH{1,1};

sigma2_ex4 = zeros(T,1);

for t = 2 : T
sigma2_ex4(t,:) = omega_hat + alpha_hat*returns_AA(t-
1,:)^2+beta_hat*sigma2_ex4(t-1,:);
VaR_ex4(t,:) = N_crit_value*(sigma2_ex4(t,:)^0.5);
ES_ex4(t,:) = -(sigma2_ex4(t,:)^0.5)*(ES_crit_value/a);
end

%% Figure VaR and ES

figure(1)
plot(DateTime.Var1,[returns_AA ES_ex1*ones(T,1) ES_ex3
ES_ex4],'LineWidth',1.25)
title('Expected Shortfall Predictions')
legend({'Returns','ES - ex1','ES - ex3','ES - ex4'})
xlabel('Date')
```

```
figure(2)
plot(DateTime.Var1,[returns_AA VaR_ex1*ones(T,1) VaR_ex3
VaR_ex4],'LineWidth',1.25)
title('Value at Risk Predictions')
legend({'Returns','VaR - ex1','VaR - ex3','VaR - ex4'})
xlabel('Date')


figure(3)
plot(DateTime.Var1,[returns_AA VaR_ex3 ES_ex3])
legend({'Returns','VaR - ex3','ES - ex3'})
xlabel('Date')


figure(4)
plot(DateTime.Var1,[returns_AA VaR_ex4 ES_ex4])
legend({'Returns','VaR - ex4','ES - ex4'})
xlabel('Date')

%% Frequency of Violations
%When the returns are lower than the predicted VaR

VaR_cond_1 = sum(returns_AA<VaR_ex1*ones(T,1));
VaR_cond_3 = sum(returns_AA<VaR_ex3);
VaR_cond_4 = sum(returns_AA<VaR_ex4);

ES_cond_1 = sum(returns_AA<ES_ex1*ones(T,1));
ES_cond_3 = sum(returns_AA<ES_ex3);
ES_cond_4 = sum(returns_AA<ES_ex4);


summary_cond_VaR = [VaR_cond_1 VaR_cond_3 VaR_cond_4]
summary_cond_percent_VaR = [VaR_cond_1 VaR_cond_3 VaR_cond_4]./T

summary_cond_ES = [ES_cond_1 ES_cond_3 ES_cond_4]
summary_cond_percent_ES = [ES_cond_1 ES_cond_3 ES_cond_4]./T

%% Magnitude of Violations

VaR_magnitude1=zeros(T,1);
ES_magnitude1=zeros(T,1);
VaR_magnitude3=zeros(T,1);
ES_magnitude3=zeros(T,1);
VaR_magnitude4=zeros(T,1);
ES_magnitude4=zeros(T,1);

%****Model 1 non-time-varying MA****
for t = 1 : T
```

```
    if returns_AA(t,:)<VaR_ex1*ones(T,1)
    VaR_mag1(t,:) = -returns_AA(t,:) - N_crit_value*standard_dev_ex1;
    end

    if returns_AA(t,:)<ES_ex1*ones(T,1)
    ES_mag1(t,:) = -returns_AA(t,:) - standard_dev_ex1*(ES_crit_value/a);
    end
end

VaR_magnitude1 = VaR_mag1(VaR_mag1~=0);
VaR_magnitude1_mean= mean(VaR_magnitude1)
ES_magnitude1 = ES_mag1(ES_mag1~=0);
ES_magnitude1_mean= mean(ES_magnitude1)


%****Model 3 - EWMA****
for t = 1 : T
    if returns_AA(t,:)<VaR_ex3(t,:)
    VaR_mag3(t,:) = -returns_AA(t,:) - N_crit_value*(sigma2_ex3(t,:)^0.5);
    end

    if returns_AA(t,:)<ES_ex3(t,:)
    ES_mag3(t,:) = -returns_AA(t,:) - (sigma2_ex3(t,:)^0.5)*(ES_crit_value/a);
    end
end

VaR_magnitude3 = VaR_mag3(VaR_mag3~=0);
VaR_magnitude3_mean= mean(VaR_magnitude3)
ES_magnitude3 = ES_mag3(ES_mag3~=0);
ES_magnitude3_mean= mean(ES_magnitude3)


%****Model 4 - Garch(1,1)****
for t = 1 : T
    if returns_AA(t,:)<VaR_ex4(t,:)
    VaR_mag4(t,:) = -returns_AA(t,:) - N_crit_value*(sigma2_ex4(t,:)^0.5);
    end

    if returns_AA(t,:)<ES_ex4(t,:)
    ES_mag4(t,:) = -returns_AA(t,:) - (sigma2_ex4(t,:)^0.5)*(ES_crit_value/a);
    end
end


VaR_magnitude4 = VaR_mag4(VaR_mag4~=0);
VaR_magnitude4_mean= mean(VaR_magnitude4)
ES_magnitude4 = ES_mag4(ES_mag4~=0);
ES_magnitude4_mean= mean(ES_magnitude4)

%% Violation Clusters
```

```
%****Model 1 non-time-varying MA****
%VaR
Time1=[1:1:size(VaR_mag1)]';
VaR_mag1(VaR_mag1==0)=NaN;
AA1_VaR = [VaR_mag1, Time1];
[idx,C1_VaR,sumd1_VaR] = kmeans(AA1_VaR,3);

C1_VaR
sumd1_VaR

figure
gscatter(AA1_VaR(:,2),AA1_VaR(:,1),idx,'bgm')
hold on
plot(C1_VaR(:,2),C1_VaR(:,1),'kx')
legend('Cluster 1','Cluster 2','Cluster 3','Cluster Centroid')
title('Value at Risk Violations Clusters')
xlabel('Date')
ylabel('Magnitude of Violations')

%ES

Time1=[1:1:size(ES_mag1)]';
ES_mag1(ES_mag1==0)=NaN;
AA1_ES = [ES_mag1, Time1];
[idx,C1_ES,sumd1_ES] = kmeans(AA1_ES,3);

C1_ES
sumd1_ES

figure
gscatter(AA1_ES(:,2),AA1_ES(:,1),idx,'bgm')
hold on
plot(C1_ES(:,2),C1_ES(:,1),'kx')
legend('Cluster 1','Cluster 2','Cluster 3','Cluster Centroid')
title('Expected Shortfall Violations Clusters')
xlabel('Date')
ylabel('Magnitude of Violations')


%****Model 3 - EWMA****
%VaR
Time3=[1:1:size(VaR_mag3)]';
VaR_mag3(VaR_mag3==0)=NaN;
AA = [VaR_mag3, Time3];
[idx,C3_VaR,sumd3_VaR] = kmeans(AA,3);

C3_VaR
sumd3_VaR

figure
```

```
gscatter(AA(:,2),AA(:,1),idx,'bgm')
hold on
plot(C3_VaR(:,2),C3_VaR(:,1),'kx')
legend('Cluster 1','Cluster 2','Cluster 3','Cluster Centroid')
title('Value at Risk Violations Clusters')
xlabel('Date')
ylabel('Magnitude of Violations')

%ES
Time3=[1:1:size(ES_mag3)]';
ES_mag3(ES_mag3==0)=NaN;
AA3_ES = [ES_mag3, Time3];
[idx,C3_ES,sumd3_ES] = kmeans(AA3_ES,3);

C3_ES
sumd3_ES

figure
gscatter(AA3_ES(:,2),AA3_ES(:,1),idx,'bgm')
hold on
plot(C3_ES(:,2),C3_ES(:,1),'kx')
legend('Cluster 1','Cluster 2','Cluster 3','Cluster Centroid')
title('Expected Shortfall Violations Clusters')
xlabel('Date')
ylabel('Magnitude of Violations')


%****Model 4 - Garch(1,1)****
%VaR
Time4=[1:1:size(VaR_mag4)]';
VaR_mag4(VaR_mag4==0)=NaN;
AA4_VaR = [VaR_mag4, Time4];
[idx,C4_VaR,sumd4_VaR] = kmeans(AA4_VaR,3);

C4_VaR
sumd4_VaR

figure
gscatter(AA4_VaR(:,2),AA4_VaR(:,1),idx,'bgm')
hold on
plot(C4_VaR(:,2),C4_VaR(:,1),'kx')
legend('Cluster 1','Cluster 2','Cluster 3','Cluster Centroid')
title('Value at Risk Violations Clusters')
xlabel('Date')
ylabel('Magnitude of Violations')

%ES
Time4=[1:1:size(ES_mag4)]';
ES_mag4(ES_mag4==0)=NaN;
AA4_ES = [ES_mag4, Time4];
```

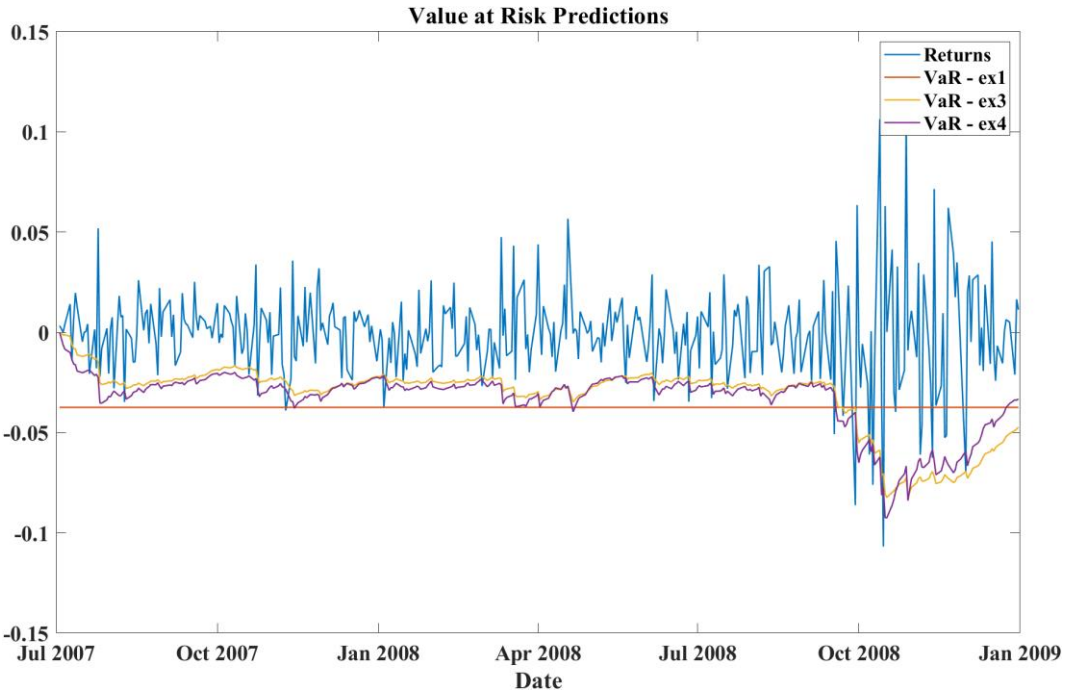
```
[idx,C4_ES,sumd4_ES] = kmeans(AA4_ES,3);

C4_ES
sumd4_ES

figure
gscatter(AA4_ES(:,2),AA4_ES(:,1),idx,'bgm')
hold on
plot(C4_ES(:,2),C4_ES(:,1),'kx')
legend('Cluster 1','Cluster 2','Cluster 3','Cluster Centroid')
title('Expected Shortfall Violations Clusters')
xlabel('Date')
ylabel('Magnitude of Violations')
```

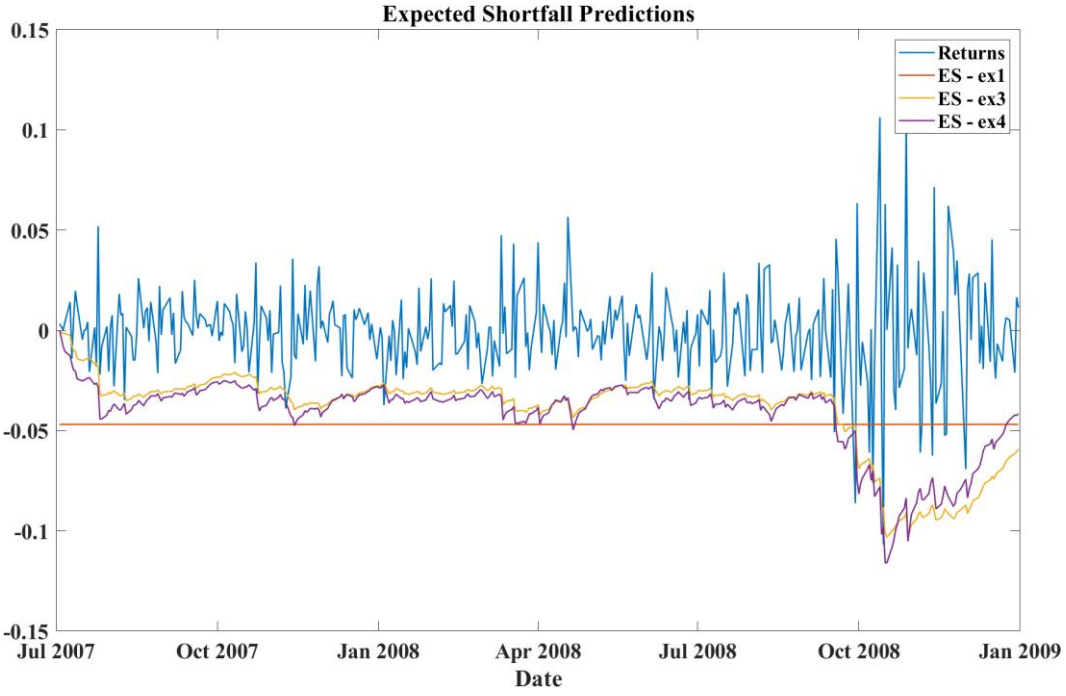
Appendix C

Application to the First Sub-period

PSO-ES portfolio

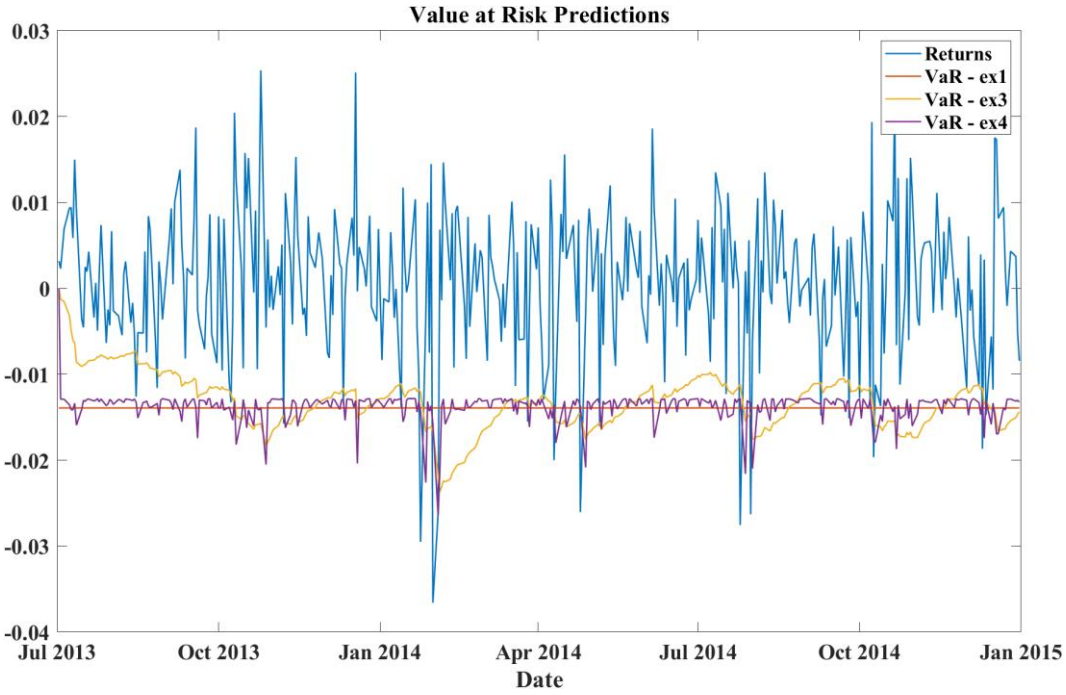
Notice that ex1 is the static approach, ex3 is the EWMA and ex4 is the GARCH (1,1).

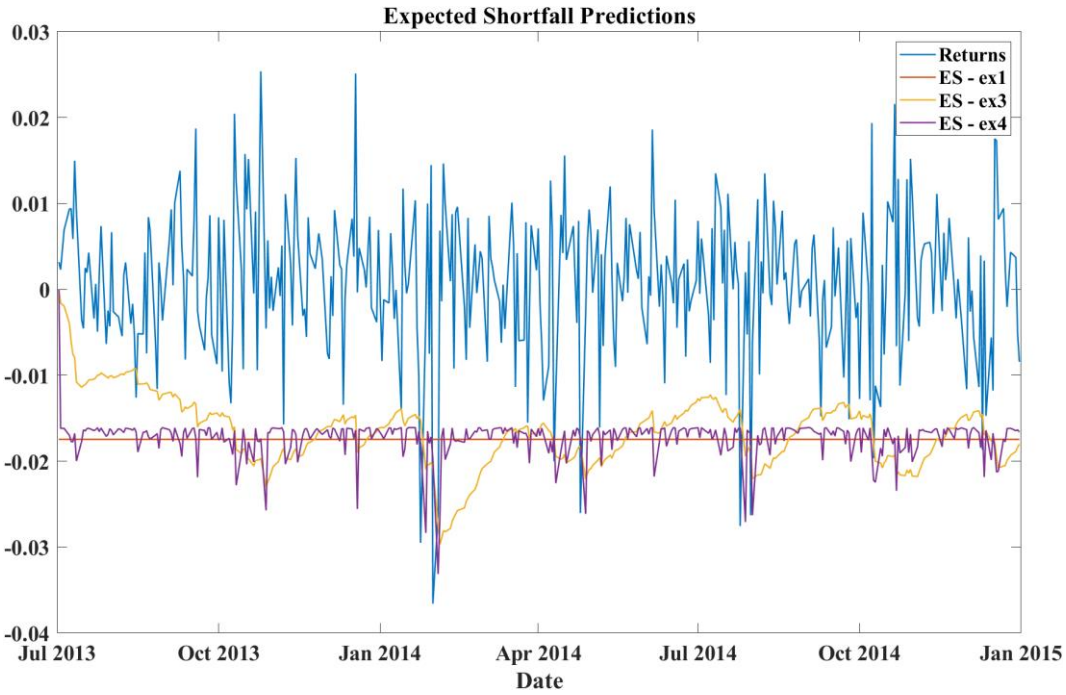




Application to the Second Sub-period

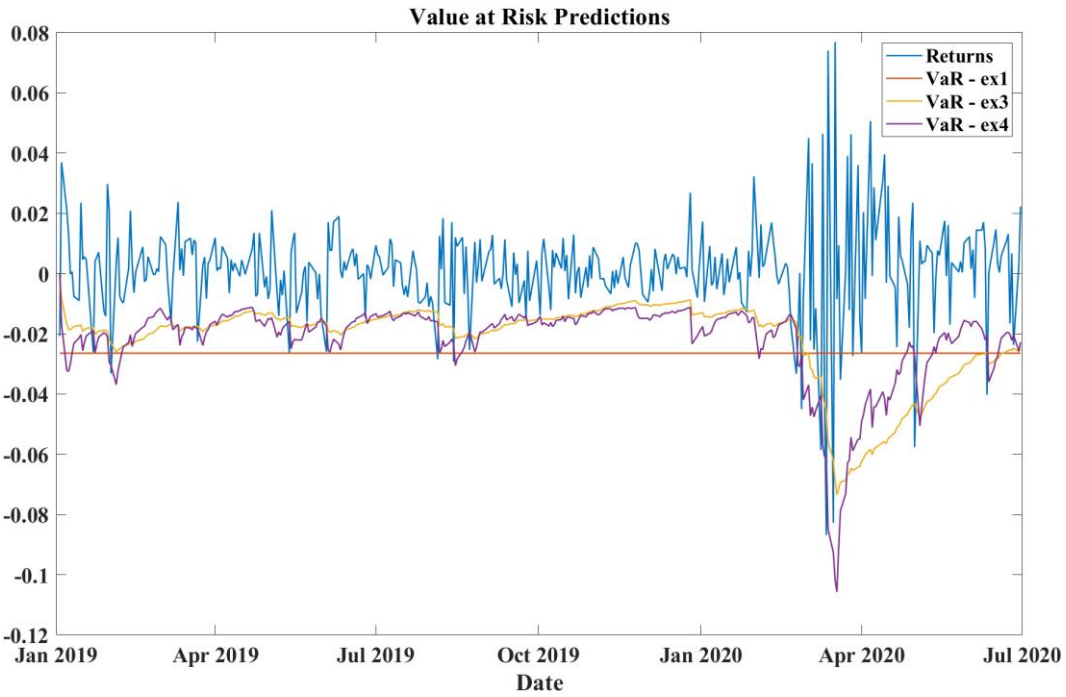
PSO-ES portfolio

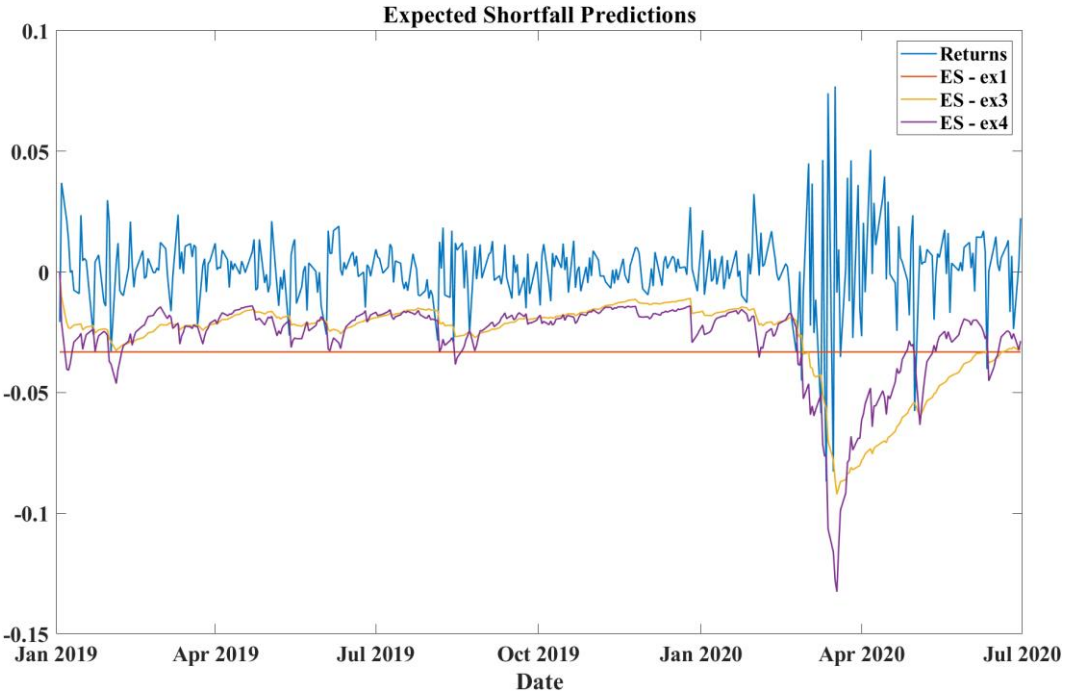




Application to the Third Sub-period

PSO-ES portfolio

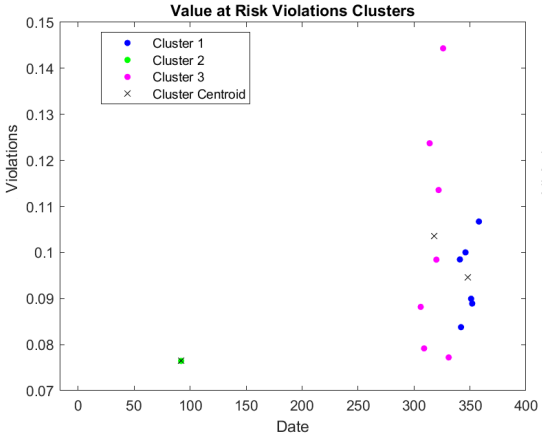


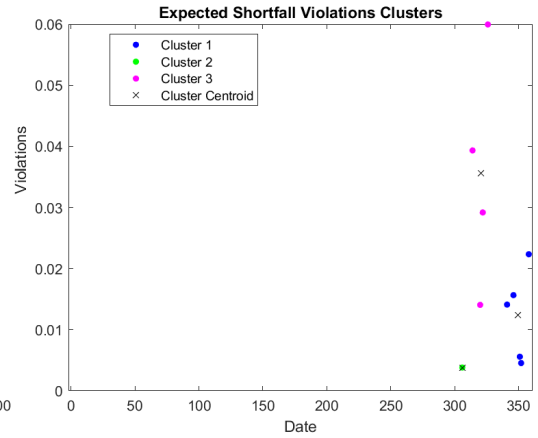


Appendix D

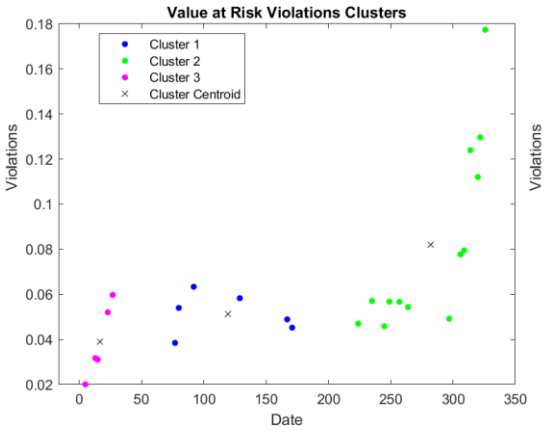
Application to the First Sub-period

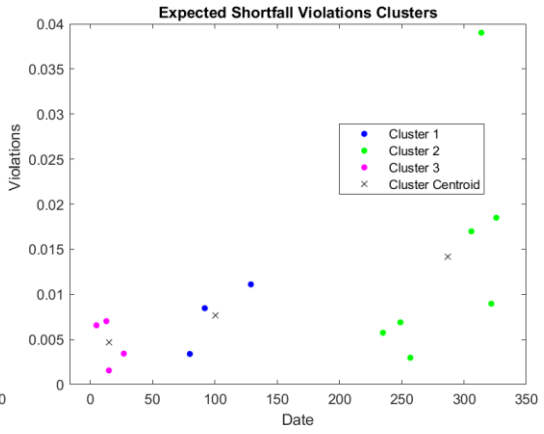
PSO-ES portfolio, Static approach



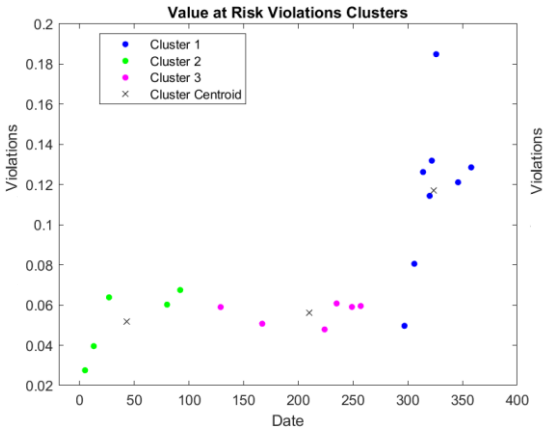


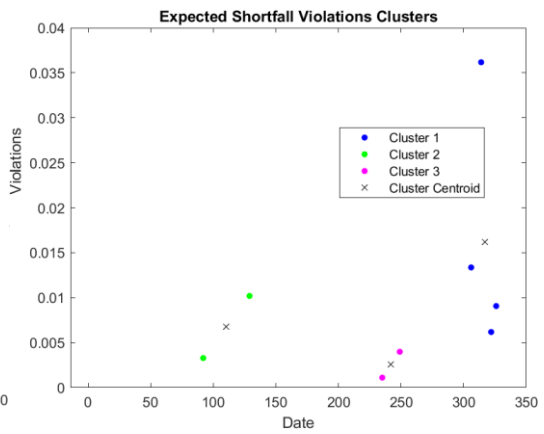
PSO-ES portfolio, EWMA approach





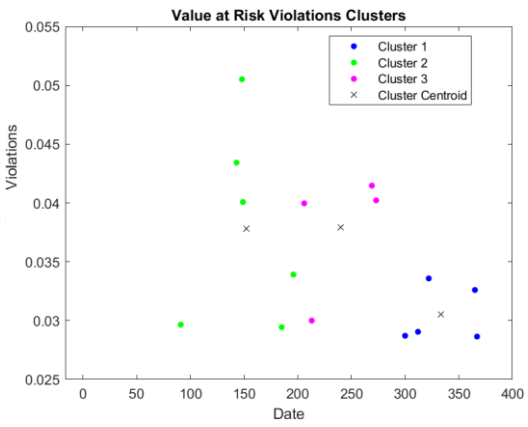
PSO-ES portfolio, GARCH(1,1) approach

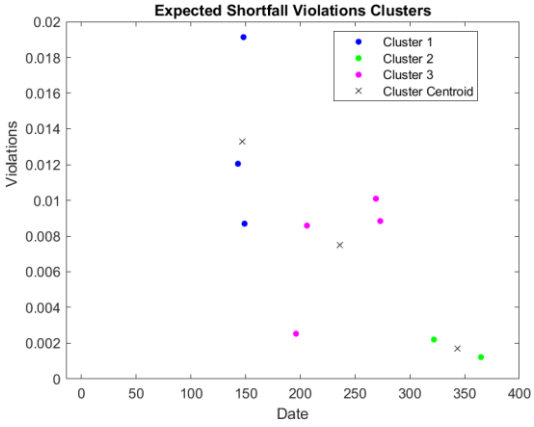




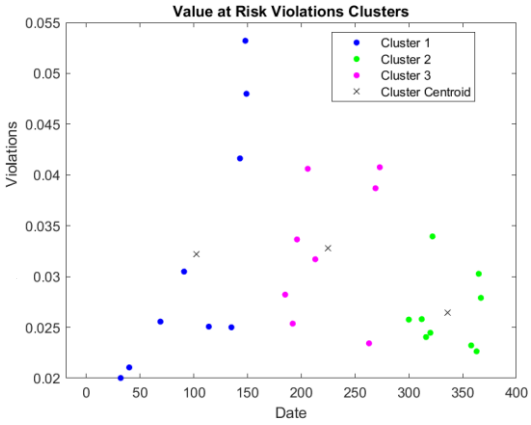
Application to the Second Sub-period

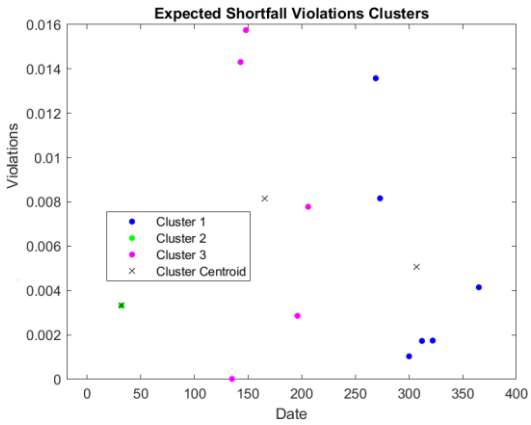
PSO-ES portfolio, Static approach



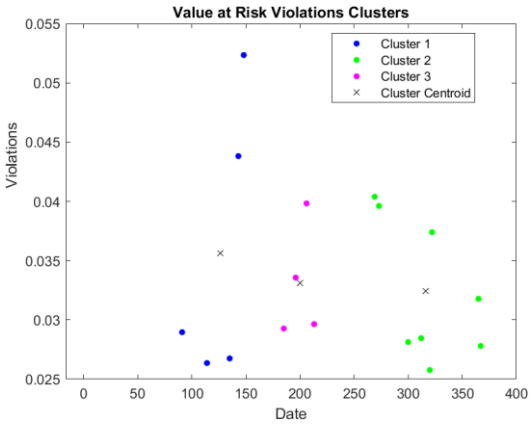


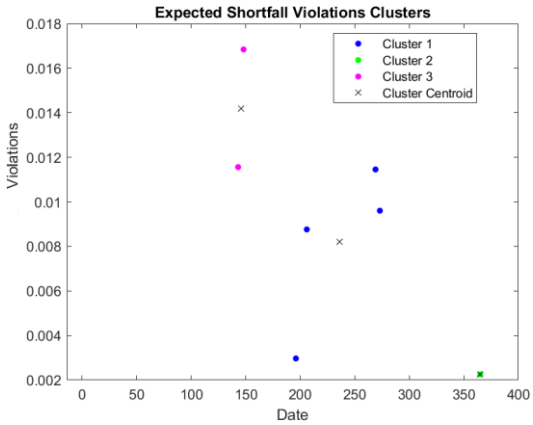
PSO-ES portfolio, EWMA approach





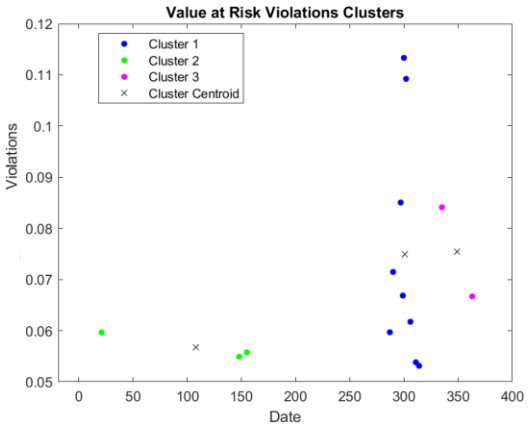
PSO-ES portfolio, GARCH(1,1) approach

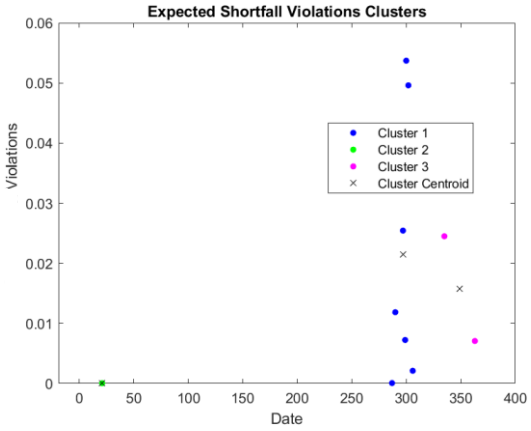




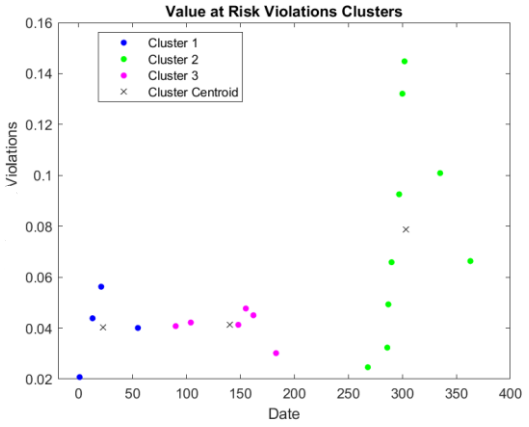
Application to the Third Sub-period

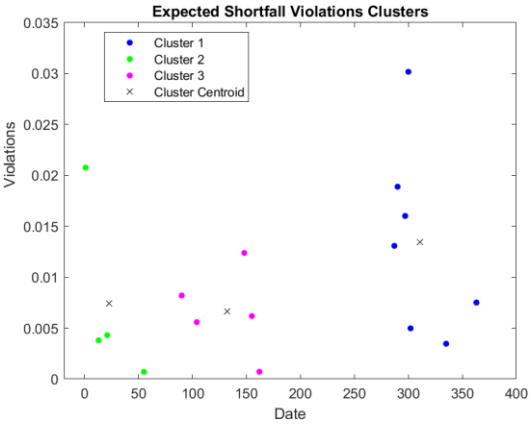
PSO-ES portfolio, Static approach



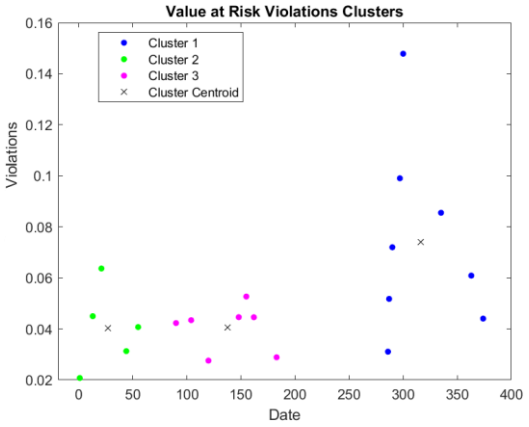


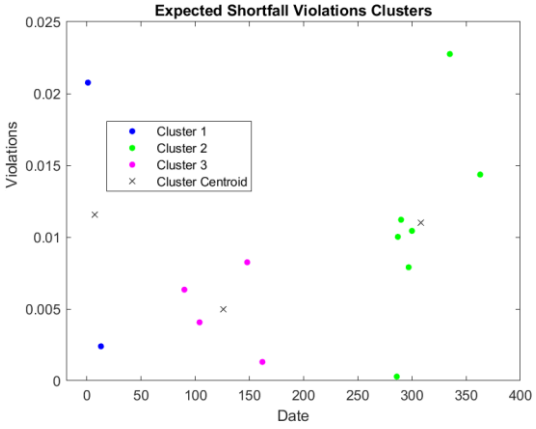
PSO-ES portfolio, EWMA approach





PSO-ES portfolio, GARCH(1,1) approach





Bibliography

[1] Acerbi, C., Tasche, D., 2002. *On the coherence of expected shortfall*, The Journal of Banking and Finance *26*(7), pp. 1487–1503.

[2] Acerbi, C., Szekely, B., 2014. *Backtesting Expected Shortfall*, The Journal of Risk *27*, pp. 76-81.

[3] Alexander, C., Sarabia, J. M., 2012. *Quantile uncertainty and value-at-risk model risk*, Risk Analysis *32*(8), pp. 1293–1308.

[4] Andramonov, M. Y., e Corazza, M., 2003. *Mixed-Integer Non-Linear Programiming Methods for Mean-Variance Portfolio Selection*, Rendiconti per gli Studi Economici Quantitativi, pp. 21-34.

[5] Arthur, D., Vassilvitskii, S., 2007. *K-Means++: The Advantages of Careful Seeding*, Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 7-9.

[6] Artzner, P., Delbaen, F., Eber, J.M., Heath, D., 1999. *Coherent measures of risk,* Mathematical Finance 9, pp. 203-228.

[7] Basel Committee on Banking Supervision, 2011. *Basel III: A global regulatory framework for more resilient banks and banking systems*, Bank for International Settlements, http://www.bis.org/publ/bcbs189.pdf.

[8] Basel Committee on Banking Supervision, 2014. *Guidelines: Corporate governance principles for banks*, Bank for International Settlements, https://www.bis.org/publ/bcbs294.pdf.

[9] Bellof, T., Wehn, C. S., 2018. *On the treatment of Model Risk in the Internal Capital Adequacy Assessment Process*, Journal of Applied Finance & Banking *8*(4), pp. 1-15.

[10] Beni, G., & Wang, J., 1993. *Swarm intelligence in cellular robotic systems*, In *Robots and biological systems: towards a new bionics?*, Springer, Berlin, Heidelberg, pp. 703-712

[11] Berkowitz, J., 2001. *Testing density forecasts with applications to risk management*, Journal of Business and Economics Statistics *19*(4), pp. 465–474.

[12] Biratti, M., Dorigo, M., 2007. *Swarm Intelligence*, Scholarpedia *2*(9), pp. 1462.

[13] Board of Governors of the Federal Reserve System, Office of the Comptroller of the Currency, 2011. *Supervisory Guidance on Model Risk Management*, Supervisory Review Letter 2011-7.

[14] Boucher, C. M., Danìelsson, J., Kouontchou, P., Maillet, B. B., 2014. *Risk models-at-risk*, Journal of Banking and Finance *44*, pp. 72-92.

[15] Brock, W., Durlauf, S., West, K., 2003. *Policy Evaluation in Uncertain Economic Environments*, Brookings Papers of Economic Activity *1*, pp. 235–322.

[16] Brown, J., McGourty, B., Schuermann, T., 2015. *Model Risk and the Great Financial Crisis: The Rise of Modern Model Risk Management*, Wharton Financial Institutions Center Working Paper *15*(1).

[17] Christoffersen, P., 1998. *Evaluating interval forecasts*, International Economic Review *39*(4), pp. 841–862.

[18] Clerc, M., Cooren, Y., Siarry, P., 2009. *Performance evaluation of TRIBES, an adaptive particle swarm optimization algorithm*, Swarm Intelligence *3*(2), pp. 149-178.

[19] Clerc, M., J. Kennedy, J., 2002. *The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space*, IEEE Transactions on Evolutionary Computation *6*, pp. 58-73.

[20] Cont, R., 2001. *Empirical Properties of Asset Returns: Stylized Facts and Statistical Issues*, Quantitative Finance *1*, pp. 223-236.

[21] Corazza M., Favaretto D., 2007. *On the existence of solutions to the quadratic mixed-integer mean-variance portfolio selection problem*, European Journal of Operational Research *176*, pp. 1947-1960.

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

[23] Corazza, M., di Tollo, G., Fasano, G., Pesenti, R., 2021. *A novel hybrid PSO-based metaheuristic for costly portfolio selection problems*, Annals of Operations Research.

[24] Directive (EU) No 36/2013 of the European Parliament and of the Council of 26 June 2013 on *Access to the Activity of Credit Institutions and the Prudential Supervision of Credit Institutions and Investment Firms*, amending Directive (EU) 87/2008 and repealing Directives (EU) 48/2006 and 49/2006.

[25] Donnelly, C., Embrechts, P., 2010. *The devil is in the tails: actuarial mathematics and the subprime mortgage crisis*, Astin Bulletin, *40*(1), 1-33.

[26] Du, Z., Escanciano, J. C., 2016. *Backtesting expected shortfall: accounting for tail risk*, Management Science *63*(4), pp. 940–958.

[27] Dudani, A. R., Chudasama, K., 2016. *Partial discharge detection in transformer using adaptive grey wolf optimizer based acoustic emission technique*, Cogent Engineering *3*(1).

[28] Duffie, D., 2008. *Innovations in Credit Risk Transfer: Implications for Financial Stability*, Bank for International Settlements Working Paper *255*.

[29] Embrechts, P., 2000. *Extremes and Integrated Risk Management*, Risk.

[30] Escanciano, J., Olmo, J., 2009. *Specification tests in parametric value-at-risk models*, Gouriéroux, Jeanblanc (Eds.), *Financial Risks: New Developments in Structured Product & Credit Derivatives*. Proceedings of the 1st International Financial Research Paris, FR, Economica, pp. 49–62.

[31] Escanciano, J., Olmo, J., 2010. *Backtesting parametric value-at-risk with estimation risk*, Journal of Business and Economic Statistics *28*(1), pp. 36–51.

[32] Escanciano, J., Olmo, J., 2011. *Robust backtesting test for value-at-risk*, Journal of Financial Econometrics *9*(1), pp. 132–161.

[33] Faris, H., Al-Betar, M. A., Aljarah, I., Mirjalili, S., 2017. *Grey wolf optimizer: a review of recent variants and applications*, Neural Computing and Applications *30*(22).

[34] Gabaix, X., Laibson, D., 2008. *The Seven Properties of Good Models*, The foundations of positive and normative economics: A handbook, pp. 292-319.

[35] Glover, F., 1986. *Future paths for integer programming and links to artificial intelligence*, Computers and Operations Research *13*(5), pp. 533-549.

[36] Glover, F., 1986. *Future Paths for Integer Programming and Links to Artificial Intelligence*, Computers and Operations Research, *13*(5), pp. 533-549.

[37] Guideline (EU) 2017 of European Central Bank. *Guide for the Targeted Review of Internal Models (TRIM).*

[38] Guideline (EU) No 13/2014 of the European Banking Authority on *Common Procedures and Methodologies for the Supervisory Review and Evaluation Process (SREP)*.

[39] Hoeting, J. A., Madigan, D., Raftery, A. E., Volinsky, C. T., 1999. *Bayesian model averaging: a tutorial*, Statistical science, pp. 382-401.

[40] Hu, W., Kercheval, A. N., 2007. *Portfolio Optimization for t and Skewed t Returns*, Florida State University.

[41] J.P. Morgan/Reuters, 1996. *RiskMetricsTM - Technical Document*, Fourth Edition, 1996.

[42] Jason, S., Middendorf, M., 2004. *A hierarchical particle swarm optimizer*, Proc. of the IEEE Conference on Evolutionary Computation *2*.

[43] Jobst, N. J., Horniman, M. D., Lucas, C. A., Mitra, G., 2001. *Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints*, Quantitative finance *1*(5), pp. 489-501.

[44] Kennedy, J., Eberhart, R., 1995. *Particle Swarm Optimization*, Proc. IEEE International Conference on Neural Networks *4*, pp. 1942-1948.

[45] Kennedy, J., Eberhart, R., 2001. *Swarm Intelligence*, Morgan Kaufmann of Academic Press, New York.

[46] Kerkhof, J., Melenberg, B., Schumacher, H., 2010. *Model risk and capital reserves*, Journal of Banking & Finance, *34*(1), pp. 267-279.

[47] Köybasi, A., Yazici, I., 2020. *Solution of Test Problems with Grey Wolf Optimization Algorithm and Comparison with Particle Swarm Optimization*, Sakarya University Journal of Science *24*(6), pp. 1252-1264.

[48] Koziel, S., Michalewicz, Z., 1999. *Evolutionary Algorithms, Homomorphus Mappings, and Constrained Parameter Optimization*, Evolutionary Computation *7*(1), pp. 19-44.

[49] Kumar, G., Mohagheghi, S., Hernandez, J. C., Del Valle, Y., 2008. *Particle swarm optimization: Basic concepts, variants and applications in power systems*, IEEE Transactions on Evolutionary Computation *12*, pp. 171-195.

[50] Kupiec, P., 1995. *Techniques for verifying the accuracy of risk measurement models*, Journal of Derivatives *3*(2), pp. 73–84.

[51] Lazar, E., Zhang, N., 2019. *Model Risk of Expected Shortfall*, Journal of Banking and Finance *105*, pp. 74-93.

[52] Li, D. X., 1999. *On default correlation: A copula function approach*, Social Science Research Network Working Paper Series.

[53] Linag, B., Park, H., 2007. *Risk Measures for Hedge Funds: a Cross-sectional Approach*, European Financial Management *13*(2), pp. 333-370.

[54] Long, W., Liang, X., Cai, S., Jiao, J., Zhang, W., 2016. *A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems*, Neural Computing and Applications *28*, pp. 421–438.

[55] Luenberger, D. G., 1984. *Linear and Nonlinear Programming*, Second Edition, Addison Wesley, Inc., Reading, Massachusetts.

[56] MacKenzie, D., Spears, T., 2014. *'The formula that killed Wall Street': The Gaussian copula and modelling practices in investment banking*. Social Studies of Science *44*, pp. 393-417.

[57] Malik, M. R. S., Mohideen, E. R., Ali, L., 2015. *Weighted distance grey wolf optimizer for global optimization problems*, Proc. of IEEE international Conference on Computational Intelligence and Computing Research (ICCIC), 1–6.

[58] Markowitz, H.M., 1952. Portfolio selection, Journal of Finance 7(1), pp. 77-91.

[59] Mendes R., Kennedy J., 2002. *Population structure and particle swarm performance*, Proc. IEEE Congress on Evolutionary Computation ,pp. 1671–1676.

[60] Mendes R., Kennedy J., 2003. *Neighbourhood topologies in fullyinformed- and best-ofneighbourhood particle swarms*, Proc. IEEE International Workshop Soft Computing in Industrial, pp. 45–50.

[61] Merton, R. C., 1969. *Lifetime portfolio selection under uncertainty: The continuous-time case*, The review of Economics and Statistics, pp. 247-257.

[62] Mirjalili, S., Mirjalili, S. M., Lewis, A., 2014. *Grey wolf optimizer*, Advances in engineering software *69*, pp. 46-61.

[63] Mittal, N., Singh, U., Sohi, B. S., 2016. *Modified grey wolf optimizer for global engineering optimization*, Applied Computational Intelligence and Software Computing *8*.

[64] Muro, C., Escobedo, R., Spector, L., Coppinger, R. P., 2011. *Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations*, Behavioural processes *88*(3), pp. 192-197.

[65] Natarajan, K., Pachamanova, D., Sim, M., 2008. *Incorporating asymmetric distributional information in robust value-at-risk optimization*, Management Science *54*(3), pp. 573-585.

[66] Office of the Comptroller of the Currency, 2011. *Sound Practices for Model Risk Management: Supervisory Guidance on Model Risk Management*, Bulletin 2011-12.

[67] Ortobelli, S., Rachev, S. T., Stoyanov, S., Fabozzi, F. J., Biglova, A., 2005. *The proper use of risk measures in portfolio theory*, International Journal of Theoretical and Applied Finance *8*(8), pp. 1107-1133.

[68] Peram T., Veeramachaneni K., Mohan C., Osadciw L. A., 2003. *Optimization Using Particle Swarms with Near Neighbor Interactions*, Genetic and Evolutionary Computation *2723*.

[69] Poli, R., Kennedy, J., Blackwell, T. 2007. *Particle swarm optimization*, Swarm intelligence *1*(1), pp. 33-57.

[70] Rachev, S., Ortobelli, S., Stoyanov, S., Fabozzi, F. J., & Biglova, A., 2008. *Desirable properties of an ideal risk measure in portfolio theory*, International Journal of Theoretical and Applied Finance *11*(1), pp. 19-54.

[71] Regulation (EU) No 575/2013 of the European Parliament and of the Council of 26 June 2013 on *Prudential Requirements for Credit Institutions and Investment Firms*, amending Regulation (EU) No 648/2012 (1).

[72] Richards, A., How, J., 2005. *Mixed-integer programming for control*, in Proceedings of the 2005 American Control Conference, pp. 2676-2683.

[73] Sharpe, W., 1963. *A Simplified Model for Portfolio Analysis*, Management Science *9*(2), pp. 277-293.

[74] Shi Y., Eberhart R., 1988. *A modified particle swarm optimizer*, Proc. of the IEEE Congress on Evolutionary Computation, pp. 69-73.

[75] Singh, N., Singh, S. B., 2017. *Hybrid Algorithm of Particle Swarm Optimization and Grey Wolf Optimizer for Improving Convergence Performance*, Journal of Applied Mathematics *15*.

[76] Sörensen, K., & Glover, F., 2013. *Metaheuristics*, Encyclopedia of operations research and management *science 62*, pp. 960-970.

[77] Suganthan, P. N., 1999. *Particle swarm optimiser with neighbourhood operator*, Proc. of the IEEE Congress on Evolutionary Computation, pp. 1958—1962.

[78] Szegö, G., 2002. *Measures of risk*, Journal of Banking and Finance *26*(7), pp. 1253-1272.

[79] Von Neumann, J., Morgensten, O., 1947. *Theory of Games and Economic Behaviour*, Princeton University Press.

[80] Watts, S., 2016. *The Gaussian Copula and the Financial Crisis : A Recipe for Disaster or Cooking the Books ?*, University of Oxford.

[81] Yang, B., Zhang, X., Yu, T., Shu, H., Fang, Z., 2016. *Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine*, Energy Conversion and Management *133*, pp. 427–443.