Portfolio Selection by Metaheuristics

Dipartimento di Scienze
Settore scientifico disciplinare: INF/01

Dottorando:
Dott. Giacomo di Tollo

Coordinatore:
Prof. Giandomenico Boffi

Tutor:
Prof.ssa Maria Chiara Meo

Anno Accademico 2005-2008
## Contents

**Prologue**  
Outline of the thesis ........................................ III  
Acknowledgements ........................................ IV

1 Portfolio selection in general  
1.1 Historical perspective ..................................... 2  
1.1.1 The Markowitz model .................................. 2  
1.1.2 Other models ........................................ 4  
1.2 A classification of portfolio selection models ............. 7  
1.2.1 Classification of variables and domains ................. 7  
1.2.2 Classification of objective functions ................... 8  
1.2.3 Classification of constraints .......................... 11  
1.3 Conclusion ............................................. 17

2 Portfolio selection by metaheuristics  
2.1 Metaheuristics ........................................... 19  
2.2 Metaheuristic techniques for the portfolio selection problem 20  
2.2.1 Metaheuristic attributes ................................ 21  
2.2.2 Metaheuristic search components ....................... 25  
2.3 Comparative studies ..................................... 32  
2.4 Linear programming based related works .................... 34  
2.5 Conclusion ............................................. 35

3 A hybrid local search approach for portfolio selection  
3.1 Metaheuristic components ................................ 38  
3.1.1 Search space .......................................... 38  
3.1.2 Cost function ......................................... 38  
3.1.3 Neighborhood relations ................................ 38  
3.1.4 Initial solution ....................................... 39  
3.2 Solution techniques ..................................... 39  
3.2.1 Steepest and first descent ............................. 40  
3.2.2 Tabu search .......................................... 40  
3.2.3 Quadratic programming ................................ 40  
3.3 Experimental analysis and comparisons ...................... 40  
3.3.1 Benchmark instances ................................ 41  
3.3.2 Experimental setting of the solvers ................... 41  
3.4 Comparison with previous results .......................... 42  
3.4.1 Comparison with Schaerf and Moral-Escudero et al. .... 42  
3.4.2 Comparison with Crama and Schyns .................... 42
<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Pearson correlations amongst multi-objective measures</td>
<td>113</td>
</tr>
<tr>
<td>Bibliography</td>
<td>117</td>
</tr>
</tbody>
</table>
# List of Figures

1.1 Conceptual representation of the Markowitz model ............................ 3  
1.2 PSP model: variables and objective ............................................... 12  
1.3 PSP model: variables, objectives and constraints ............................... 17  

3.1 Running time: Comparison of SD+QP with exact solvers ......................... 45  
3.2 SD+QP: apl found varying $k$ ...................................................... 46  
3.3 UEF and pre-assignment constrained CEFs for extremal return values .......... 47  
3.4 Return/risk vs apl for one pre-assigned asset ................................... 48  
3.5 Hybrid basin of attraction ............................................................ 50  
3.6 Monolithic basin of attraction ....................................................... 52  

4.1 IT model: variables, objectives and constraints .................................... 63  

5.1 Instance 3, SD+QP, Trade-off between tracking error and return, continuous .. 86  
5.2 Instance 3, SD+QP, Trade-off between tracking error and return, discrete .... 87  
5.3 AUEF: SD+QP, Instance 3, Discretely computed returns .......................... 87  
5.4 AUEF: SD+QP, Instance 1, Continuously computed returns. .................... 88  
5.5 AUEF: SD+QP, Instance 2, Continuously computed returns. .................... 89  
5.6 AUEF: SD+QP, Instance 3, Continuously computed returns ........................ 90  
5.7 AUEF: SD+QP, Instance 4, Continuously computed returns ........................ 90  
5.8 SD+QP, Trade-off between tracking error and return, continuous .............. 91  
5.9 SD+QP, Trade-off between tracking error and variance, continuous ............ 91  

B.1 AUEF: SD+QP, Instance 1, Discretely computed returns ........................ 110  
B.2 AUEF: SD+QP, Instance 2, Discretely computed returns ........................ 110  
B.3 AUEF: SD+QP, Instance 3, Discretely computed returns ........................ 111  
B.4 AUEF: SD+QP, Instance 4, Discretely computed returns ........................ 111  
B.5 SD+QP, Trade-off between tracking error and return, discrete ................ 112  
B.6 SD+QP, Trade-off between tracking error and variance, discrete .............. 112
List of Tables

3.1 Hybrid approach: the benchmark instances ........................................... 41
3.2 Comparison of results with Schaefer and Moral-Escudero et al. .................. 43
3.3 Comparison of results with Crama and Schyns. ...................................... 43
3.4 Running times for the computation of the whole CEF on ORLib instances 1–4 ... 44
3.5 Pearson correlation between pre-assigned asset return/risk and the resulting apl .. 48
3.6 Number of local (non-global) minima for instance 4. .............................. 49
4.1 IT: The benchmark instances ................................................................. 70
5.1 Multi-Objective: the benchmark instances, continuously compounded returns ... 74
5.2 Multi-Objective: the benchmark instances, discretely compounded returns. ..... 74
5.3 Random portfolios: rank based correlation analysis. ................................. 77
5.4 Found versus Non-Dominated Portfolios, Continuously Compounded Returns. .. 77
5.5 Found portfolios versus Non-Dominated Portfolios, Discretely Compounded Re-
turns ........................................................................................................... 78
5.6 Computational time. .................................................................................. 78
5.7 Mean Equal Points, continuous .................................................................. 79
5.8 Standard Deviation Equal Points, continuous .......................................... 80
5.9 Mean Equal Points, discrete ..................................................................... 80
5.10 Standard Deviation Equal Points, discrete ............................................ 81
5.11 Mean Percentage Coverage Measure, continuous .................................... 81
5.12 STD Percentage Coverage Measure, continuous ..................................... 82
5.13 Mean Percentage Coverage Measure, discrete ........................................ 83
5.14 STD Percentage Coverage Measure, discrete ......................................... 83
5.15 Optimised portfolios: rank based correlation between TE and $\sigma^2$ .......... 84
5.16 Optimised portfolios: rank based correlation between TE and $r_p$ ............. 85
5.17 Optimised portfolios: rank based correlation between $r_p$ and $\sigma^2$ .......... 85
5.18 Percentiles of a distribution .................................................................... 96
C.1 Random portfolios: Pearson correlation analysis. ................................. 114
C.2 Optimised portfolios: Pearson correlation between Index Tracking Error and Return .......................................................... 114
C.3 Optimised portfolios: Pearson correlation between Return and Variance ....... 115
C.4 Optimised portfolios: Pearson correlation between Index Tracking Error and Variance .......................................................... 115
Prologue

Many financial concerns can be formulated as optimisation problems, and these problems are of critical real-life importance: asset allocation, risk management, derivative pricing, and model fitting belong to this class. The main specimen of these problems is the Portfolio Selection Problem (referred to as PSP), originally formulated in the fifties by Harry M. Markowitz [111]: this problem’s goal is to compute, out of a given set of assets, the portfolio which minimises the risk (assessed by portfolio’s variance) for a given level of minimum required return. The importance of this approach is in that it suggests and analytically justifies the portfolio diversification as an investment criterion, rather than focusing on maximising return as the only parameter. This work is nowadays recognised as the milestone of modern portfolio theory, and it is regarded as essential criterion by investors and fund-managers: its value has led the author to be awarded the Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel in 1990 (jointly with Merton H. Miller and William F. Sharpe, for their pioneering work in the theory of financial economics). Indeed, the basic formulation of this problem is a quadratic program and can be analytically solved by standard tools. Nevertheless, the complexity of management, the law-system and the policies of asset management companies, lead this basic model to be enriched by adding further constraints, having the effect to make the problem computationally demanding. Several approaches have been introduced so far but, as stated by Jobst, Horniman, Lucas and Mitra [77], solving these problems remains a computationally intractable task: hence the need to resort to approximated and heuristic algorithms in order to obtain an optimal solution to be dealt with in a reasonable amount of computational time.

In our work, we are using metaheuristic algorithms to solve diverse constrained Portfolio Selection Problems: metaheuristics can be described as high-level strategies used to guide subordinate heuristics to find an optimal solution over the search landscape. In a nutshell, they are high level criteria to drive algorithms for solving hard combinatorial problems from various Artificial Intelligence (AI) and Operational Research (OR) areas. Many amongst the most powerful algorithms to solve benchmark problems (SAT, TSP, etc.) are to be re-conducted to this class of strategies. Hence these methods, due to their intrinsic generality capabilities, can be (and have been) applied to a huge range of problems: amongst them, problems arising from the economic and financial areas are of a peculiar interest. A clarification must be done in this preliminary introduction: metaheuristics do not provide us with the solution proof of optimality, but this does not mean that the provided solution is sub-optimal. In our work, on the contrary, we are comparing metaheuristic and exact solvers, showing that they provide the very same solutions to the problem at hand. This is a key point to be made clear from the beginning: differently from approximated algorithms, proven to provide sub-optimal solutions, metaheuristics can provide the global optimum of the problem, meaning that reaching the global optimum is not hindered by formal properties, depending on the problem features, but they are used in problems for which the proof of optimality is not required. Consequently, the user is not meant to verify if the solution is the global optimum: what he cares for most is to have the best solution he can in a reasonable amount of time. This means that only other methods can
(perhaps) state if the solution at hand is the optimum or not. There even exist metaheuristics
providing a range of approximation for the found solution (say, an upper bound of the distance
from the global solution) but it is out of the scope of this work to discuss about these aspects.

In this work, different tipologies of portfolio selection will be dealt with. The milestone
the thesis starts from is the Mean-Variance analysis (another term referring to the Markowitz
model), in which the investor has to make an assessment (or prediction) about assets’ and
portfolio’s future return: this assessment is done by means of using historical mean returns as
estems for the future. The conceptual path of the thesis is to tackle Mean Variance analysis,
then to focus on a second strategy, relying on the same portfolio representation but implementing
other (and somewhat opposite) criterion, before combining the two approaches in a multi-criteria
framework. This latter strategy is given by the Index Tracking, a different approach that does
not need the investor to predict assets’ future returns, since his only goal is portfolio return to
stay as close as possible to a financial Stock Exchange Index return, such as, for instance, FTSI.
These features are important, as they lead us to classify portfolio strategies in active strategies
and passive strategies. In the first group, whose best representative is the Mean Variance
approach, decision makers build portfolios based on their expectations about the assets’ return
and risk. Objections to this approach are in that it is deemed difficult to produce reliable
estimates about future return and risk; furthermore, strategies based on these estimates are not
effective as the market on the long run: hence, passive strategies try to mimic a given market
index by investing in a portfolio whose behaviour is as similar to the benchmark’s as possible.

Indeed, in the last years several authors suggested to combine passive and active strategies to
embed preferences and to give more explanation power to the resulting model. Combining active
portfolio selection with Index Tracking has been suggested by Steuer, Qi and Hirschberger [147]
and Jorion [78], and experimental results have been provided by Burns [22] and Yu, Zhang
and Zhou [168]. Nevertheless, these works lack giving us a wide perspective on the motivation
beneath their approaches, and their experimental setting is not built in a consistent way about
strategies used, parameter tunings, and descriptions of their outcomes. This thesis wants to fill
these gaps, providing a better understanding of the models used, the strategies at hand and the
real significance of both.

Since the nineties, metaheuristics have been applied to solve such problems. The use of
these strategies led to a wider and wider corpus of research, as witnessed by the growing
interest shown by such communities as Computational Management Science, Computational
Econometrics, Numerical Methods, in top of less finance-oriented ones such as CPAIOR. Again,
a great number of papers on this topic has appeared in Journals from diverse areas, and also
from members belonging to the Operations Research and Artificial Intelligence communities
who are more and more attracted to financial and economics problems.

Indeed, after a first enthusiastic phase, in which often a naive proposal has also been ac-
cepted, a new rigorous and methodological approach begins to raise attention amongst both
academics and practitioners: model comparison, understanding of different formulations’ fea-
tures, choice of the appropriate strategy, search space analysis, effective parameter tuning, and
robustness of the approach seem to attract more than creating new strategies and methods.
This is also according to trends pursued by more technical communities such as Hybrid Meta-
heuristics. This thesis intends to place itself in this trend.

The work whose output is given by this thesis has been done during my Doctoral stud-
ies at the Dipartimento di Scienze–Università G.D’Annunzio–Pescara-Chieti (I) between fall
2005 and winter 2008. The main purpose of this work has been to integrate knowledge I have
acquired during my Bachelor (Laurea) and M.Sc (Laurea Specialistica), both in Economia In-
formatica (Computer Science Economics). All this work has been supervised by Dott. Ing.
Andrea Roli, since I started my PhD path under his supervision in Pescara, before his transfer to University of Bologna. Part of this work has been done during visiting research periods at the Econometric Department, coordinated by Professor Manfred Gilli (Geneva University, Geneva, CH), at the Center for Computational Finance and Economic Agents, coordinated by Professor Dietmar Maringer (Essex University, Colchester, UK), and at the Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle, coordinated by Professors Thomas Stuetzle and Mauro Birattari (Université Libre de Bruxelles, Bruxelles, B), which were made possible by a scholarship from POR Abruzzo2000-2006 C3/IC4E.

Results of this thesis have been presented at National and International Conferences, Workshops and Research Seminars. Furthermore the outcome was conveyed in joint papers with Andrea Roli, Luca Di Gaspero, Andrea Schaerf, Dietmar Maringer, Manfred Gilli, Enrico Schumann, Gerda Cabej, Evis Kellezi. More in detail, section 1.2 and chapter 2 consist of a paper published by International Journal of Operations Research [40]; chapter 3 was covered in a paper presented at CPAIOR 2007 [37] and currently in progress being prepared for a Journal submission; chapter 4 has been accepted as a book chapter in the volume Metaheuristics for Service Industry [39].

The thesis is written in English in order to be evaluated by an External Committee for institutional goals. Indeed, this is done, above all, to allow the results to be made directly available to the international scientific community.

Outline of thesis

This thesis is organised in seven main parts: this introductory part, five chapters consisting of the main corpus of my research, and a final chapter, prominently named Epilogue, which summarises the main results of this work, and outlines the extensions already investigated in top of this thesis, in addition to ideas, further directions, suggestions and visions to be pursued in future. Each chapter of this thesis has been extracted from autonomous research papers.\footnote{Chapter 1 and 2 consist of a new organisation, with further extension proposed in order to give a satisfactory introductory style.} This feature makes each chapter self-contained, and this will be of great value to readers who would like to browse certain parts of the thesis which are of interest to them: in this way, the reader interested in just a few subtopics covered by some chapter will be satisfied without the effort of dealing with the remainder of the work. Each chapter is enriched with an Introduction and a Conclusion.

Chapter 1 introduces the reader to the basis of the Portfolio Selection Problem: the main formulations and representations of the problem are given. Above all, the basic classification between active and passive strategies is outlined. This distinction is of the utmost importance for the remainder of the work, since chapters 1–3 will only deal with active strategies, while chapter 4 will be focused only on passive strategies (eventually, in chapter 5, an integration of the two approaches will be attempted). Furthermore, in the first chapter an analysis and survey of different (active) portfolio selection models will be outlined. We urge the reader to understand that in this thesis we will focus on aspects of the Portfolio Selection Problem as dealt with by literature employing metaheuristic approaches, rather than specific financial and management literature. It is out of the scope of this work to give comprehensive foundations about Portfolio Selection, and interested readers are prompted to more specialised works.

Chapter 2 surveys metaheuristics approaches for the Portfolio Selection Problem. Firstly a high level outline of metaheuristics is given, then metaheuristic components are explained in the way they are exploited by the literature we take into account. Last, metaheuristic...
strategies for the PSP are outlined. In the survey, only methods that have been used for solving portfolio choices are taken into account, and before explaining their use, a few words are used to understand their behaviour: the strategies are classified using the standpoint provided by MAGMA, a general framework for metaheuristics.

Chapter 3 proposes a hybrid metaheuristic approach for the PSP: this approach relies on a master-slave decomposition of the problem taken into account, and a dedicated strategy is given for each of these subproblems: a local search approach is used to determine assets to be present in the portfolio, while Quadratic Programming is used to determine the optimal shares. Different constraint setting are tackled and three local search approaches are used. After noticing that the three strategies produce comparable performances, a search landscape analysis is performed to understand the reason of this behaviour.

After having spent three chapters in dealing with active strategies only, in chapter 4 we survey metaheuristic approaches for passive strategies (Index Tracking). The framework to be used is the same already introduced in chapters 1 and 2 to deal with, respectively, models to be taken into account and strategies to solve the problems formulated in that way. To this extent, a conceptual representation of all the Index-Tracking-Problem model attributes (variables, objectives, constraints and time-horizon) will be given, along with the same formalism used to classify metaheuristics in chapter 2 (MAGMA).

Chapter 5 defines, moving from theoretical results, a multi-objective approach combining Portfolio Selection and Index Tracking. Three formulations are said to be possible, but just one is investigated because of computational issues. The same hybrid approach already proposed in chapter 3 is used and experimental results are explained and justified.

The thesis ends with a chapter summarising the main results of the thesis, pointing out open questions and ideas already pursued in joint work in progress.

Acknowledgements

As my mentor Andrea Roli points out in his PhD thesis, Scientific Research is a collaborative activity: most of the work in this thesis would not have been possible without the help, the support and the ideas suggested by colleagues, professors, friends and even non-adepts. My first great thank must be given to Andrea Roli, who supervised my activity providing me with the methodology and knowledge he thought to be necessary for my research and, on several occasions, brought me back on the proper route where appropriate. Without him, the outcome of my research would be not available at the moment. Andrea Schaerf and Luca Di Gaspero, to whom I will always be grateful for their patience and availability, have provided invaluable help.

I would like to acknowledge the help provided by Manfred Gilli, the first professor I have met in my research visiting periods, who introduced me into a community with whom I have been much in contact ever since, and helped to better understand the point of view of its members, which at the time were slightly different to my own. For the same reasons, I want to thank Dietmar Maringer for his support and for the discussions we had together during my visiting period at Essex and afterwards. Then, I would like to thank Thomas Stuetzle and Mauro Birattari for their rigorous and methodological advice which opened to me new perspectives over the problems at hand: their comments have always been detailed and a large section of this thesis would have been discussed in a very different way without their invaluable skills which they chose to share with me.

I want furthermore to thank the following: Joseph Andria, Prasanna Balaprakash, Arne Brutschy, Alexandre Campo, Antal Decugniere, Antonietta Di Salvatore, Eliseo Ferrante, Simone Giansante, Hilda Hysi, Evis Kellezi, Pranvera Kellezi, Anil Khuman, Max Manfrin, Re-
nata Mansini, Nithin Mathews, Marco Montes De Oca, Carlo Pincirol, Giovanni Pini, Ilir Roko, Mohammed Saifullah bin Hussin, Francisco Santos, Enrico Schumann, Peter Winker, Eric Yuan, Jin Zhang: each of them helped me open my mind, to exchange ideas, to hope for a better world and to feel at ease in new and challenging environments I came across during my activity.

Other than to be a collaborative task, scientific research is every now and then made while seated at a desk, in front of a laptop or with a paper at hand. I want to thank Pamela Peretti for having been seated in front of the desk next to me, often in front of the same laptop and sometimes with the same paper in hand. For sure, my feelings about the time spent in our LAB would not have been the same without her.

Another sincere thanks must be addressed to Denver Martin Smith, Thomas Wagner and Anil Khuman: they have been my English mentors, and they tried to help me in understanding this strange idiom. Please feel free to complain to them if you do not deem language used in this work to be reasonable.

But my most sincere thoughts must be addressed to others: firstly, to my family, especially my mother, my father and my sister, who provided me (far longer than these three years) a safe, warm and critical environment in which I had the chance to express myself as I did, always trying to make me think with a conscience and helping me to understand my peculiar feelings and wishes, at each moment and in every field; then, to my loving and caring companion Claudia, with whom I began to walk hand in hand in the very same moment I began my PhD studies, and with whom I shared the best part of these three years.

Thanks to you all!
Chapter 1

Portfolio selection in general

A common hypothesis in financial theory is that information about future asset prices is contained in their current (and historical) prices, so that returns can be treated as stochastic variables. In the most common (and simplest) approach, returns are treated as normal distributions, and all information about return realisation and deviation risk are to be described by the return’s expected value and variance. Let’s take an asset $i$ and $r_i$ be its return distribution: asset realization will be given by mean return $E(r_i)$ (from now on, for ease of use, referred to as $r_i$) and its risk by return variance $\sigma_i^2$ (or its standard deviation $\sigma_i$, generally referred to as volatility). It is worthwhile to notice that this notion of risk includes both negative (as intuitively valid) and positive deviations from the expected return.

Different assets can be held together forming a portfolio, conceived as a vector $X$ whose $i$th element represents the amount of money invested in asset $i$ $(1 \leq i \leq n)$. Assets are sorted out of a Universe $U$ composed of $n$ assets, and the same representation of risk and return still applies: for each asset the rate of return is represented by a random variable $R_i$, whose mean is given by $r_i$ and represents the expected return of asset $i$. Note that returns are to be drawn from prices, and there exist a great deal of methods for computing them. Let’s introduce $r_{i,t}$ to be asset $i$ return at time $t$ and $S_{i,t}$ to represent asset $i$ price at time $t$: we will introduce here returns to be either continuously compounded, so that $r_{i,t} = \ln \left( \frac{S_{i,t}}{S_{i,t-1}} \right)$, or discretely compounded, so that $r_{i,t} = \frac{S_{i,t} - S_{i,t-1}}{S_{i,t-1}}$. Furthermore, $\sigma_{ij}$ is the real-valued covariance of expected returns on assets $i$ and $j$, computed as $\sigma_i \cdot \sigma_j \cdot \rho_{ij}$ where $\rho_{ij}$ is the correlation coefficient ($-1 \leq \rho \leq 1$). Portfolio return variance $\sigma_p^2$ is given by $\sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij}x_ix_j$, whilst its expected return is given by $\sum_{i=1}^{n} r_ix_i$.

We stated that in this framework portfolio risk is given by its return variance, this has been the most common approach from the fifties, even if the first basis for the Modern Portfolio Theory were laid down by Hicks [70] two decades before, who first advocated the advantage offered by risk-reduction through diversification. Indeed, he thought that investor could not utterly diversify away risk because of assets indivisibility, ignoring that asset return correlation does not allow portfolio variance (risk) to be null. Furthermore, this work is important because it was the first to introduce the Separation Theorem, stating that, when a riskless asset is present in the market, investors can adjust their preferences investing a portion of their wealth in risky assets and the remainder in risk-free ones. Separation theorem will be dealt with in section 1.1.2.

The work by Markowitz [111], the seminal paper on which Modern Portfolio Theory is built upon, rejects the hypothesis that investors wish to maximise expected returns because this criterion does not imply that a diversified portfolio is preferable to a non-diversified one. Thus, he states, the goal is to select a portfolio with minimum risk at given minimal returns.
Furthermore he illustrates the set of feasible portfolios for his analysis to be piecewise linear. Above all, in this work variance is introduced as risk measure\(^1\). In the following section the Markowitz model, along with other well-established ones, will be explained in more detail.

### 1.1 Historical perspective

The literature has provided us with plenty of models for Portfolio choice. The first distinction amongst such models is borrowed from the practice, suggesting that strategies can be partitioned between *active* and *passive* strategies. *Active management strategies* are defined in a way such that decision makers are to build their portfolios based upon their expectations about assets’ risk and return, creating their own portfolios also based on their risk-aversion and other parameters. It has been shown that these strategies do not perform better than the market in the long run, and for this reason portfolio managers have begun to build portfolios following the market, so trying to replicate the performance of a financial index over time, w.r.t. a given measure, so defining a *passive management strategy*. This strategy does not require predictions about assets’ future returns, nor assumes returns to be following a given distribution: the only information to be held are the time series representing index and asset returns, without any further risk estimate. For this reason, it seems easier to be implemented than an active strategy such as the one dictated by Markowitz, that instead relies on return and risk estimates. We first illustrate over the most accredited active and passive models, then we classify portfolio models in section 1.2 and 4.2 using a general framework composed of the following attributes: variables, objectives and constraints.

We first present the Markowitz model along with the Black and Tobin model, the most important active PSP models that constitute the basis upon which the other models are obtained as variations and extensions; then, we will introduce the Index Tracking Problem, the most common passive strategy, to be conceived as a generalisation of all passive approaches. In the following we will use the acronym *PSP* to refer to active strategies only, while *IT* will be referring to the Index Tracking Problem, the only specimen of passive strategies we will be taking into account.

#### 1.1.1 The Markowitz model

Assuming normal return distribution, a perfect market without taxes (or transaction costs), with short-selling prohibition (so that variables are to be non-negative, see section 1.2.3) and with assets being infinitely divisible, the Markowitz model can be stated as follows:

\[
\begin{align*}
\min & \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j \\
\text{subject to} & \\
\sum_{i=1}^{n} r_i x_i & \geq r_e \\
\sum_{i=1}^{n} x_i & = 1 \\
x_i & \geq 0 \quad i = 1 \ldots n
\end{align*}
\] (1.1)

\(^1\)For sake of historical correctness, the use of variance was first proposed in [91].
1.1 Historical perspective

where we use the same notation as expressed before: \( n \) is the number of assets, \( x_i \) is the proportion of money invested in asset \( i \), \( r_i \) represents the expected return of asset \( i \); \( \sigma_{ij} \) is the covariance of expected returns on assets \( i \) and \( j \). The objective function is the variance (herein called risk-measure) \( \sigma_p^2 \), given by \( \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij}x_i x_j \). The portfolio return is represented by a random variable and the expected return is given by \( \sum_{i=1}^{n} r_i x_i \), whilst \( r_e \) represents the minimum required portfolio return. Constraint (1.3) ensures that asset weights sum up to one, as they are considered as fractions of the whole amount of money to be invested\(^2\). In this work, we do not take into account possible adjustments between estimated and actual returns and the problem we consider is a ‘single-period’ (i.e., single-stage) problem, even though some multi-period models will be dealt with in chapter 4 as extension of the basic Index Tracking model. Moreover, the PSP formulations we discuss are deterministic.

By resolving the problem for a set of values of \( r_e \) it is possible to estimate the efficient frontier for the Markowitz unconstrained problem (referred to as UEF). The investor can then choose the portfolio depending on specific risk/return requirements. The UEF is composed of Pareto optimal solutions, i.e., portfolios such that no criterion can be improved without deteriorating any other criterion: in our example, a portfolio \( s \) is said to be efficient (Pareto-optimal) if there is no other portfolio \( s_1 \) such that \( r_{s_1} > r_s \) and \( \sigma_{s_1} \leq \sigma_s \) or \( r_{s_1} \geq r_s \) and \( \sigma_{s_1} < \sigma_s \).

Markowitz himself stated that just optimising \( r \) would lead to no diversification, while using a bi-criterion such as mean and variance would lead to the right kind of diversification for the right motive. Indeed, it has been later argued that investors evaluate the future of their investment only in term of probability distributions with two parameters, such as normal distribution\(^{156} \) and that return and variance contain all the information for any arbitrary distribution (so, normal one being a specimen) when the investor utility is given by a quadratic function\(^{106} \); more in detail, we can say that the mean-variance framework approximately holds for any utility function embedding non-satiation and risk aversion. Note that this brief explanation here is needed in order to link the Markowitz model to investor behaviours. This work will not go into utility theory, and interested readers are forwarded to \(^{114} \).

\(^2\)In accordance to the original Markowitz Model we are using here the continuous model, whose variables are expressed as continuous quantities.

---

Figure 1.1: Conceptual representation of the Markowitz model. Rectangles represent the instantiation of a qualification (ellipse). In gray, qualifications and instantiations not present in the model.

The Markowitz model can be considered as the most simple formulation of the PSP. Its conceptual representation is depicted in Fig. 1.1. Note that the three attributes, variables,
objectives and constraints, can be directly instantiated, as in the case of constraints, or further
detailed through qualifications. This basic model can be varied and extended in many ways.
Every modification can be viewed as the result of the combination of simple variations, each of
which affecting only one attribute: the following models (Black and Tobin) are obtained just
applying such small extensions to the Markowitz model.

The extraordinary amount of works concerned with Mean-Variance analysis appeared after
its first introduction in 1952 fostered two different (but complementary) approaches:

1. Mean Variance has been used for investors to select portfolios, in order to support opti-
mised portfolio choice and to allocate the wealth amongst assets.

2. Mean Variance has been used to analyse the economic system, assuming that actors
(investors) look for mean-variance efficiency, generally using simplified models in order to
show relationships amongst economic magnitudes.

Our further work will rely on the first way of dealing with Markowitz model. It is eventually
worthwhile to mention that notwithstanding its potential in capturing the basic properties
of the problem, the Markowitz model suffers from several drawbacks. First, it might be difficult
to gather enough data and information for estimating risk and returns. Second, the estimation
of return and covariance (used for defining the risk) from historical data is very sensitive to
measurement errors [26]. Finally, it is nowadays considered too simplistic for practical purposes,
because it does not incorporate non-negligible aspects of real-world trading, such as maximum
size of portfolio, minimum lots, transaction costs, preferences over assets, management costs,
etc. These aspects can be modeled by adding constraints to the original formulation, leading
to the constrained PSP. Alternative objective function, also defined to take into account these
aspects, will be outlined in section 1.2.2.

1.1.2 Other models

For introductory purposes, in this section we are introducing other Portfolio Selection Models:
this review is far from being exhaustive, and will be covered in more details in section 1.2.
Markowitz’s is the most common model for active portfolio management, and in the following
we will introduce extensions of its basic properties through the Black, Tobin and Brennan
models, before enlightening the reader with the most common passive portfolio management
model: the Index Tracking.

Allowing short selling

The Markowitz model imposes asset weights to be non-negative. This can be removed from
the original formulation replacing equation (1.4) with

\[ x_i \in \mathbb{R} \quad i = 1 \ldots n \]  \hspace{1cm} (1.5)

This model is referred to as the Black model [15] and it is used in order to model Short Selling,
to be explained as follows: suppose at time \( t \) an investor thinks the price of asset \( i \) will rise. In
order to gain profits he must buy the asset \( i \) at market price \( S_{i,t} \) and, if \( S_{i,u} > S_{i,t} \), sell it at
time \( u \) \( (u > t) \), gaining the difference \( S_{i,u} - S_{i,t} \). Conversely, if at time \( t \) investor thinks \( i \) price
will fall, he has the following option: borrowing \( i \) from a broker, selling it to the market, buying
it back at time \( u \) and giving it back to the broker. In this case he will have gained \( S_{i,t} - S_{i,u} \)
(minus commission). The Black model can be subject to several extensions. For example, Black
himself investigated a model in which the following constraint is added to constraint (1.5):

\[ x_{n+1} \geq 0 \]  \hspace{1cm} (1.6)
1.1 Historical perspective

so introducing an asset forbidden to be sold short, whilst other ones are allowed to. Noticeably, short selling prohibition holds for the risk-free asset, and this model is referred to as Black second model.

It has to be noticed that a collateral is generally required in order to have the investor cover the position in case the price will rise instead: this makes the previously introduced model unrealistic. The first realistic model allowing short selling has been introduced in [96], where it is said that the short seller will receive interest at riskfree rate $r_c$ on the sales price placed in escrow, and may or may not also receive interest at the same rate on his cash remittance to the lender of the stock. Furthermore, additional constraints can be (and usually are) imposed by law: for instance, USA regulation $T$ imposes

$$
\sum_{i=1}^{n} |x_i| \leq 2
$$

(1.7)

There are also trading aspects difficult to be formalised: for example the investor can borrow an asset, but the lender can request it at any moment. An investor can also borrow from himself, but this phenomena is generally not taken into account. The investor borrows stocks and sells them over the market, investing the proceeds in cash equivalent [73] as collateral, even if the amount of money to be invested in cash-equivalent (say $c$) has to be generally greater than the short proceeds[73]. The borrower doesn’t get interest on the short selling proceeds if it is a small-investor, otherwise he generally gets only a part of interests (short-rebate $h$). Taking into account all these phenomena, firstly the return of a portfolio containing assets short sold must be defined

$$
r_p = \sum_{i=1}^{n} \left( r_i - h_i \cdot r_c \right) x_i
$$

(1.8)

$$
h_i = 0 \text{ if } x > 0 \quad 0 \leq h_i \leq 1 \text{ otherwise}
$$

(1.9)

where $h$ is the short-rebate, and is 0 if the asset is long, positive if is sold short [73, 74] Then a general framework (also embedding Reg $T$) can be introduced as

$$
\text{min} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j
$$

s.t.

$$
\sum_{i=1}^{n} \left( r_i - h_i \cdot r_c \right) x_i \geq r_e
$$

(1.10)

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

(1.11)

$$
-1 \leq x_i \leq 1 \quad i = 1 \ldots n
$$

(1.12)

$$
\sum_{i=1}^{n} |x_i| \leq 2
$$

(1.13)

$$
x_{\text{cash}} \geq c \cdot \left( \sum_{i=1}^{n} \min(0, x_i) \right)
$$

(1.14)

$$
h_i = 0 \text{ if } x > 0 \quad 0 \leq h_i \leq 1 \text{ otherwise}
$$

(1.15)

where $x_{\text{cash}}$ represents the amount to be invested as collateral. Note that several models have been introduced for dealing with short selling and a unique modelling is still far from being accepted, also due to different practices followed in different countries.
Introducing the risk-free asset

In the Markowitz framework (and in most works we are going to survey) only portfolios composed of stocks are taken into account. This is a simplification of real-world markets, but the proposed approaches can be easily extended to other kinds of securities like options, future contracts and commodities, as well as to features as adjusted returns, dividends, spreads. The case of debentures (conceived as risk-free assets, as their return is fixed and their variance is null) deserves instead more attention: they were investigated by Tobin [156, 157], removing the condition that each asset must be risky. Any combination of a risk-free asset \( c \) and a portfolio \( P \) composed of risky assets will show a relation between the expected return and the variance:

\[
    r = r_c + \frac{r_P - r_c}{\sigma_P} \cdot \sigma
\]

(1.16)

where \( r \) and \( \sigma \) are, respectively, the expected return and the variance of the combination of risk free asset and risky portfolio, \( r_c \) is the return of risk-free asset and \( r_P \) and \( \sigma_P \) are, respectively, the expected return and the variance of the risky portfolio. The ratio \( \frac{r_P - r_c}{\sigma_P} \) is referred to as reward-to-variability ratio (or Sharpe Ratio) and used is some works, see [106, 104] and section 2.2.2. In this framework the Markowitz assumption (all assets must be risky) can be modified allowing wealth to be invested in riskless assets (bank-accounts, treasury bills etc) introducing the separation theorem, as the efficient frontier in this approach is composed of linear combinations between the only optimal risky portfolio and the risk-free asset: the only optimal portfolio is given by the point in which the efficient Mean-Variance point is tangent to the straight line of risk-free asset only. It is only after defining this portfolio that it is possible to decide how to allocate wealth between risky portfolio and safe asset, depending on investor preferences.

The Tobin efficient frontier will be described by a linear relationship between efficient portfolios risk (\( \sigma_p \)) and return (\( r_p \))

\[
    r_p = r_c + \frac{r_T - r_s}{\sigma_T} \cdot \sigma_p
\]

(1.17)

where \( r_c \) is the risk-free return, and \( r_t, \sigma_t \) are respectively return and risk of the tangency portfolio. This finding has strong implications as in that way the model can be solved analytically and described by a single equation, differently from the Mean Variance model in which there is a method to compute efficient portfolios but no way to determine such an equation. There exist several extensions of the separation theorem. As already stated, Tobin shows that efficient portfolios are linear combinations between the optimal risky portfolio and the risk-free asset, while Sharpe [140] shows that if the model has the only budget constraint \( \sum_{i=1}^{n} x_i = 1 \) efficient portfolios are linear combinations between the optimal risky portfolio and the minimum variance portfolio. Brennan [19] introduces borrowing and lending, considering two different risk-free rates for the two cases (respectively \( r_b \) and \( r_l \), with \( r_b \geq r_l \)). In this model, after determining the two tangency portfolios for the two cases (\( t_b \) and \( t_l \)) we see that the efficient set is no longer a straight line, but a piecewise line composed of the following objects: 1) the straight line connecting the risk-free asset \( l \) and \( t_l \), for \( r \in [0, r_l] \); 2) the Markowitz efficient frontier, for \( r \in [r_b, r_l] \); 3) the straight line connecting the risk-free asset \( b \) and \( t_b \), for \( r > r_l \). Note that this model is equivalent to the Tobin one when \( r_l = r_b \).

In a nutshell, in the Tobin framework, the investment decision is split in two phases: first determine the risky portfolio \( P \) optimising the reward-to-variability ratio and then combine it with the risk-free asset (separation theorem). In this way, the risk-aversion of investors does

---

\( ^3 \)Examples are given by corporate bonds, mortgage bonds, gold bonds, common bonds.
not affect the choice of the risky portfolio, but only the proportions to be assigned to the risky portfolio $T$ and to the risk-free asset $s$: a very risk-averse investor will allocate a huge proportion of his endowment to the risk-free asset, while a very risky seeking will prefer to invest more in the risky portfolio.

**Index tracking**

While the aforementioned models belong to the class of *active management strategies*, now we are going to introduce the *Index Tracking Model*. As already pointed out, this strategy doesn’t require predictions about assets’ future returns so in principle it seems easier to be implemented than an active strategy such as the one dictated by Markowitz. As this model is concerned with finding a portfolio where the returns are as similar to the benchmark as possible, the manager has to find asset weights so that the portfolio’s behaviour differs from the benchmark’s as little as possible. Hence, in its basic version, the IT problem is to minimise a distance measure (*tracking error*) between the portfolio and benchmark returns over time; no requirements are given on the expected return, but the constraints are usually similar to those in the Markowitz’s (and related models), as we will detail in sections 4.2.4. As a contribution of this thesis’ is given by combining Passive Management and Mean Variance analysis, the Index Tracking Problem, along with its formalisation, will be covered in more detail in the following sections.

### 1.2 A classification of portfolio selection models

Constrained optimisation problems can be defined by specifying variables, along with their domains, objectives and constraints among variables. These entities can also play the role of model attributes and serve as the basis for a classification of the different models. Attributes may have several qualifications, that, in turn, may be subdivided in more detailed categories, till reaching the specification of the actual attribute instantiation. For instance, objectives (an attribute) can either be single or multi-criteria (qualifications); each qualification can be specified by instantiating the actual objective function, for example the minimisation of a given risk measure.

As already pointed out, the Markowitz model can be considered as the most simple formulation of the PSP. In the following, we will detail the most important extensions of the basic model(s), by keeping in the background the conceptual model scheme. Our description has not the goal of providing an overview of all the formulations of the PSP, but rather of illustrating, in a unifying view, the diverse models of the problem as described in the specific literature on metaheuristics.

#### 1.2.1 Classification of variables and domains

We first briefly discuss the possible choices for variable domains in a PSP model. In the Mean-Variance model, variables are real and they range between zero and one, as they represent the fraction of available money to invest in an asset\(^4\). This choice is quite ‘natural’ and has the advantage of being independent of the actual budget. Conversely, another possibility is to choose integer values for variables and make them range between zero and the maximum available budget.\(^5\) When variables are integer, it is possible to add to the model constraints

\(^4\)For the sake of simplicity, we are not taking into account short sales yet. They will be introduced in section 1.2.3.

\(^5\)For sake of simplicity we think of integer variables as an amount of money, but the reader must be aware that discrete values are often introduced their quantity being the actual amount of asset units; furthermore, in the presence of the so-called rounds, the domain values correspond to the number of rounds. This extension will
that involve actual budget values, such as minimum trading lots and also introduce more realistic objective functions. The integer formulation better explains real-world situations: for instance, it turns out that small investors are more sensitive to integer constraint on variables, as the resulting portfolios are less diversified[106]. In the following we are using the letter $x$ when referring to variables expressed as discrete values, whereas $w$ will refer to variables consisting of continuous values. Other advantages and disadvantages of the two approaches will be discussed in the following sections, in which variations of the basic model are presented.

1.2.2 Classification of objective functions

In the most common PSP formulations, the objective can be either to minimise the risk (satisfying a given return), or maximise the return (not exceeding a given maximum risk), or both. In the former cases the problem is single-criterion, while in the latter case it is multi-objective. Metaheuristics have mostly been applied to single-criterion models, but there are some notable works which deal with multi-criteria models, such as [153][47][124]. The applications on the single-objective formulation (in which the risk has to be minimised) very often solve a PSP instance as a function of the desired expected return $r_e$ that is then seen as an instance parameter. Solving the instance for $r_e$ ranging over values from a finite set, gives an estimation of the efficient frontier that could be drawn by directly solving the bi-objective formulation. In these cases, it is common to use the expression ‘efficient frontier’ also for that set of solution points, even if it is just an approximation of such a frontier.

In our classification, we consider these two qualifications for the attribute objectives, as shown in Fig. 1.2.

Single-criterion objectives

Although metaheuristics have been successfully applied to tackle both single and multi-criteria optimisation problems, the PSP has been mostly modeled as a single-criterion optimisation problem.

A way of modelling the problem in a single-criterion framework to be tackled by metaheuristics consists in including constraint (1.2) in the objective function in a Lagrangean relaxation fashion [25] [164] [83]:

$$\text{max } (1 - \lambda) \sum_{i=1}^{n} r_i w_i - \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} w_i w_j$$

subject to constraints (1.3) and (1.4), where $\lambda$ is a trade-off coefficient ranging in $[0, 1]$. If $\lambda = 0$ the investor completely disregards risk and aims to maximise returns; conversely, when $\lambda = 1$, the investor is risk-averse and only wants to minimise risk. By resolving the problem for a set of values of $\lambda$ it is possible to estimate the efficient frontier for the Markowitz unconstrained problem (UEF). The investor can then choose the portfolio depending on specific risk/return requirements. Obviously, metaheuristics cannot guarantee the optimality of solutions, and are aimed in providing us with an approximation of the actual Pareto frontier. In the following we will distinguish between the actual efficient frontier (UEF) and the approximated one (AUEF). Moreover, since we are going to introduce other classes of constraints in our discussion, we will refer to the constrained efficient frontier as CEF, whilst its approximation will be referred to as ACEF. We notice here that the unconstrained frontier dominates the constrained one and the goal of most works introducing metaheuristics tackling the PSP is to draw out the CEF for the problem at hand, so we assume that the algorithms we will discuss are aimed at drawing out

be detailed in section 1.2.3.
1.2 A classification of portfolio selection models

the CEF instead of single portfolios, if not differently explicitly stated. The class of constraints introduced varies amongst works and will be explained for each implementation.

The single-criterion problem (equations (1.1) to (1.4)) can be solved for a set of equally distributed values for the minimum required return \( r_e \), so solving several instances of the problem introducing different values of \( r_e \) in constraint (1.2) in order to obtain a distribution of equally distanced points able to provide us with a range of solutions, exploiting an approach similar to multi-objective problem solving [137][119][30]. In this way the efficient frontier is bounded by the minimum risk portfolio (MRP), defined as the solution of the problem without lower bound on return \( (r_e = 0) \). MRP has its own return \( r_{mrp} \): it follows that solving instances whose \( r_e \leq r_{mrp} \) the solution will be the MRP itself, whilst for larger values of \( r_e \) the solution will consist of portfolios so that \( \sum_{i=1}^{n} w_i r_i = r_e \).

It is worth noticing here that the Mean-Variance formulation presents its main drawbacks in being incompatible with the axiomatic models of preference for choice under risk [161] and lacking in coherence [6]. This consideration, together with the ones previously mentioned (see section 1.1.1) motivated researchers to define other risk measures: different measures can be taken, thus defining different objective functions. Markowitz himself suggested the use of semi-variance instead of variance in order to assess portfolio risk. Semi-variance can be defined as

\[
\text{semivar} = \sum_{j: r_j \leq E[R]} p_j (r_j - E[R])^2 \tag{1.19}
\]

where \( R \) is a distribution of returns, often statistically computed by enumerating the most probable scenarios, \( r_j \) is the return of the \( j \)-th element of the distribution, \( p_j \) its probability and \( E[R] \) the mean of the distribution. This measure is equivalent to variance if return distribution is symmetric around the mean and captures the essence of risk as perceived by investors, characterised by the likelihood of incurring a loss. Its drawback is that an investor can perceive the loss not necessarily when returns are below the mean, but below some other subjective thresholds \( \tau \). This idea refers to the part of distribution below a certain target of return, and for this reason the corresponding measures are referred to as target downside risk measures:

\[
\text{DSR}(\tau) = \sum_{r_i \leq \tau} p_i (\tau - r_i)^q \tag{1.20}
\]

When \( q = 2 \) the formula is referred to as target semi-variance expression; in this case if \( \tau = E[R] \) the formula is equivalent to semi-variance.

The threshold \( \tau \) is referred to as Value-at-Risk (VaR) and can be conceived as a measure of the portfolio catastrophic risk, since investors are concerned with the chance of losing their wealth because of a low-probability-high-impact-event[153]. \( \tau \) has been used as the threshold below which the investor perceives a loss[59, 61, 110, 107] and in their context VaR is bounded in the constraints and the objective to maximise is the expected return of the portfolio. The probability that portfolio returns fall below the VaR level is called Shortfall Probability:

\[
SP = p(r < VaR) \tag{1.21}
\]

where \( r \) stands for \( \sum_{i=1}^{n} r_i w_i \). Furthermore, the Expected Shortfall is defined as the expected portfolio return given that its value has fallen below VaR:

\[\text{A risk measure is said to be coherent when it fulfills properties of translation invariance, subadditivity, positive homogeneity and monotonicity, see Artzner et al.[6] for details.}\]

\[\text{The definitions of measures such Var and CVaR were originally based on prices rather than returns. Indeed they are also modeled with returns in some works[2, 101] For homogeneity we decided to present them as based on returns: this could lead to different optimisation results when using continuous rates of returns. It is out of the scope of this work giving details of this issue.}\]
Similarly to variance, VaR lacks in coherence and it does not stress the importance of portfolio-
diversification in order to reduce the risk[6]. Anyway, it is nowadays one of the most used risk
measures, as it is imposed by the Basel agreement, which allows banks to use their own VaR
models in order to assess credit risk[8].

Amongst other approaches it is worth mentioning the Mean-Absolute-Deviation model (MAD)
[89], in which the risk is defined as the mean absolute deviation of the portfolio rate of re-
turn. This model does not rely on probabilistic assumptions on returns (it is equivalent to the
Markowitz model if returns are considered as normally distributed) and it is easier to handle
because it does not require the covariance matrix:

\[
\min E \left[ \left| \sum_{i=1}^{n} r_i w_i - E \left[ \sum_{i=1}^{n} r_i w_i \right] \right| \right]
\]  

(1.23)

Assuming \( r_i = \frac{\sum_{i=1}^{T} r_{it}}{T} \), where \( T \) represents the set of observations, this equation can be re-
formulated as follows:

\[
\min \sum_{t=1}^{T} \left| \sum_{i=1}^{n} (r_{it} - r_i) w_i \right|
\]  

(1.24)

Following the same ideas, in [145] risk is measured as the mean semi-absolute deviation of the
rate of return below the average:

\[
\sum_{t=1}^{T} \left| \min \left(0, \sum_{i=1}^{n} (r_{it} - r_i) w_i \right) \right| \]  

(1.25)

This function is shown to be equivalent to MAD, as semi-deviation is equal to half of absolute
deviation.

Furthermore, since the Mean-Variance formulation is non-linear, efforts have been made to
model the problem as a linear programming model. Amongst them, besides the above cited
ones, the approach proposed by Young[167] defines the risk as the minimum return achieved
by a portfolio over a set of scenarios (worst case realisation): the optimisation problem, in
this framework, consists in finding the portfolio whose worst case realisation is maximum. A
generalization of this approach is the Conditional Value at Risk (CVaR), obtained measuring
the mean of a specified quantile of worst realization distribution[131].

Up to now we described only risk measures to be minimised. Indeed the PSP can be
formalised as a maximisation problem in which a safety measure must be optimised. It has been
shown[101] that for each risk measure there exists a corresponding safety measure (obtained
combining return and risk measures) and vice-versa, but minimising a risk measure is not always
equivalent to maximise the corresponding safety measure: equality holds only if measures are
independent from distribution-specific parameters (e.g. minimising semi-variance (formula 1.19
is not equivalent to maximise its corresponding safety-measure, as its formulation is function of
the mean of return distribution).

Multi-criteria objectives

In the multi-criteria variant of the PSP model, the objectives are usually the following [151][3]:
1.2 A classification of portfolio selection models

\begin{align*}
\min & \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij}w_i w_j \\
\max & \sum_{i=1}^{n} r_i w_i
\end{align*}

subject to constraints (1.3)(1.4).

Moreover, it is possible to have several functions to optimise: Subbu et al. [153], for instance, propose the following:

\[
\begin{cases}
\max & \text{Portfolio expected return} \\
\min & \text{Variance} \\
\min & \text{Portfolio Value at Risk}
\end{cases}
\]

This model can also handle preferences, by introducing other three metrics: Market-yield, Dollar duration weighted Market-yield and Transaction costs. These metrics are used to describe and structure ordinal preferences.

The approach consisting in weighting multi-objective criteria is common when the model is aimed to support decision processes. For example, in Ehrgott et al. [47], the objective is to maximise a weighted sum of five measures (annual price-performance, annual dividend, three year price-performance, S&P rating and volatility) and weights are to be defined by users in order to specify their preferences.

A different multi-objective formulation is given in Ong et al. [124]. According to existing models, they assume portfolio risk being divided in the uncertainty risk and the relation risk. The uncertainty risk measures the uncertainty on future return rates, whilst relation risk measures the trending degree of the sequence. In this framework the objective is given by

\[
\begin{cases}
\max & \text{Portfolio expected return} \\
\min & \text{Uncertainty Risk} \\
\min & \text{Relation risk}
\end{cases}
\]

Many other objective functions and utility measures have been proposed, an overview of which can be found in [80].

A visual conceptual overview of the different kinds of objectives is depicted in Fig. 1.2, along with the possible choices for variable domains.

Before discussing the third attribute of the model, i.e., constraints, we have to note that the estimation of returns from real-world data raises statistical and practical issues that have to be taken into account when the PSP is tackled. A discussion on this topic is out of the scope of this paper and we forward the interested reader to the specific literature on the subject [121, 10, 9, 59, 61, 45, 160].

1.2.3 Classification of constraints

The set of feasible portfolios, over which a risk–return measure is to be optimised, is defined by the constraints of the model.

Constraints can be first distinguished into two classes: theoretical and practical. The first class includes budget and return constraints, while practical constraints are motivated by actual problem requirements (for example, minimum lots imposed by law). In most portfolio models there are a small number of constraints, but this is not required by theoretical assumptions,
1. Portfolio selection in general

Figure 1.2: Conceptual representation of the PSP model attributes *variables* and *objectives*.

nor by computational issues [114]. Note that here we are dealing with constraints introduced for active portfolio management (PSP), but the same constraints can be applied to passive portfolio management approaches. It is worthwhile to notice that the introduction of constraints (i.e., just maximum cardinality constraints) have the PSP become NP-complete, as proven by [13].

**The basic constraints: return and budget**

Return and Budget constraints are the most important ones, because they characterise the essential part of the problem. These constraints are included in the unconstrained Markowitz model and are used to theoretically define the feasibility of a solution:

\[
\sum_{i=1}^{n} r_i w_i \geq r_e \quad (1.30)
\]

\[
\sum_{i=1}^{n} w_i = 1 \quad (1.31)
\]

Return constraint (1.30) is very important as returns represent one of the two main aspects of the problem; constraint (1.31) means that all the capital must be invested. If an integer formulation is used, in which assets are represented by their actual value rather than their ratio to the whole portfolio, it can be expressed in two ways: either introducing upper and lower bounds to the capital to be invested [95, 102, 145]:

\[
B_{\min} \leq \sum_{i=1}^{n} x_i \leq B_{\max} \quad (1.32)
\]

where \(B_{\min}\) and \(B_{\max}\) are respectively the lower and upper bound on the budget expressed by their currency value; or imposing assets value to sum up to the available budget

\[
\sum_{i=1}^{n} x_i = B \quad (1.33)
\]

where \(B\) is the available budget; constraint (1.32) is less tight than constraint (1.33) and it is imposed when minimum lots are added to the formulation, as introducing them (in an integer formulation) makes it more difficult to find a solution w.r.t the former budget constraint (see section 1.2.3).

---

\(^8\)Section 4.2.4 will deal with constraints as introduced explicitly for Index Tracking Problems.
1.2 A classification of portfolio selection models

The importance of budget constraint is due because it introduces us to a special case of the
constraint set, whose elements are all defined in equality form: the affine constraint set. This
case is of utmost importance for two reasons:

- It allows the two-fund separation meaning that each efficient portfolio can be expressed
  as a combination of efficient portfolios\(^9\).

- In this form, the mathematical program is in principle solvable with standard methods (i.e.
  Matrix Inversion), while inequality constraints need, at least, a Quadratic Programming
  solver that can turn to be very cumbersome to implement and run. This aspect will
  turn to be essential in Appendix A.2, where it will be outlined how the main analytical
  procedure core for computing efficient frontiers is given by solving related problems with
  affine constraint sets.

As stated above, a shortcoming of the original Markowitz formulation is that it does not incor-
porate many aspects of real-world trading, such as maximum size of portfolio, minimum lots,
transaction costs, preferences of which assets to include in the portfolio, management costs,

Cardinality constraints

The number of assets in the portfolio is often either set to a given value or it is bounded.
Introducing a binary variable \(z_i\) equal to 1 if asset \(i\) is in the portfolio and 0 otherwise, the
constraint can be expressed as follows:

\[
\sum_{i=1}^{n} z_i \leq k
\]

(1.34)

This constraint is imposed to facilitate the portfolio management and to reduce its man-
agement costs. When the model contains this constraint, it can be named “The asset par-
ing problem”[97]. Accordingly to financial and OR literature, it has been experimentally
shown that, when the cardinality constraint is imposed, the ACEF tends to tightly approx-
imate the UEF for high values of \(k\) [54][25]. The inequality form is quite common (see, for
instance, [137][30][83]), and also a lower bound can be introduced:[37]

\[
k_{\min} \leq \sum_{i=1}^{n} z_i \leq k_{\max}
\]

(1.35)

however the constraint can also be expressed in the equality form [3], i.e., \(\sum_{i=1}^{n} z_i = k\).

Note that cardinality constraints can be implicitly defined by floor and ceiling constraints
(see section 1.2.3), as if a lower bound \(\varepsilon\) for each asset is introduced, then a \(k_{\min}\) minimum
Cardinality \(\lceil \frac{1}{\varepsilon} \rceil\) is required; conversely if an upper bound \(\delta\) for each asset is introduced, then a
\(k_{\max}\) maximum cardinality \(\lfloor \frac{1}{\delta} \rfloor\) is needed.

\(^9\)The main consequence of this assertion is that the market portfolio, composed of the sum of each investors’
holding, is efficient.
Floor and ceiling constraints

With these constraints we impose a minimum and maximum proportion (\(\varepsilon_i\) and \(\delta_i\) respectively) allowed to be held for each asset in a portfolio, so that \(x_i = 0 \lor \varepsilon_i \leq x_i \leq \delta_i\) (\(i = 1\ldots n\));
in other words, the portion of the portfolio for a specific asset must range in a given interval:

\[\varepsilon_i z_i \leq x_i \leq \delta_i z_i\]  \hspace{1cm} (1.36)

Ceiling constraints (i.e., upper bound constraints) are introduced to avoid excessive exposure to a specific asset and in some cases are imposed by law. Floor constraint (i.e., lower bound) is used to avoid the cost of administrating very small portions of assets and may be implied by transaction costs (see section 1.2.3).

It is also possible to impose different upper and lower bounds for each asset, but this opportunity has not yet been explored in the literature for the PSP.

Short sales

In the model we discussed in section 1.1.1 asset weights are non-negative (constraint (1.4)): this constraint means that no short sales are allowed and it is imposed in almost all the works we are analysing (a notable exception is given by [133], whilst in [30] short sales are allowed in the initial formulation). Indeed, as already pointed out, it is a common practice to sell assets that are not yet owned by the investor at the time, in expectation of a price falling: this can be formulated replacing constraint (1.4) with

\[x_i \in \mathbb{R} \quad \forall i\] \hspace{1cm} (1.37)
as introduced in [15] (see section (1.1.2)).

Note that, if short sales are to be forbidden, constraint (1.4) becomes redundant when imposing floor constraints.

Rounds

The unconstrained Markowitz model considers investments as perfectly divisible, so as to be represented by real variables, whilst in several markets (such as the Japanese and most of European ones) securities are negotiated as multiples of minimum lots. For each asset there exists a minimum tradable lot, referred to as round. Rounds are usually measured in units of money, so this constraint is encountered in the PSP integer formulation [95][82][102]. If \(S_j\) is the price of asset \(j\) and \(\rho_j\) its minimum tradable quantity, the minimum lot \(u_{\text{min}}(j)\) of asset \(j\), measured in units of money, is given by \(u_{\text{min}}(j) = \rho_j S_j\). If rounds are introduced in the formulation, they becomes the integer decision variables [82, 102].

When using the continuous formulation its application consists in imposing that each weight must be a multiple of a given fraction [151], and, obviously, its meaning is different from imposing rounds in integer formulation.

Although rounds are imposed by the Exchange Market independently of the investor size, their effects (measured in terms of deviation from the unconstrained results) seem to be relevant for small investors but negligible for big ones and their introduction has the effect of reducing the number of different assets in the optimal portfolio.

Class constraints

In the real world of finance it may happen that investors ideally partition the assets in mutually exclusive sets (classes). Each set consists of assets with common characteristics (insurance
assets, naval assets, etc.), and investors want to limit the proportion of each class. Let $M$ be the set of classes $\Gamma_1, \ldots, \Gamma_M$, $L_m$ and $U_m$ the lower and upper proportion limit (respectively) for class $m$, the class constraint can be defined as\(^{10}\)

$$L_m \leq \sum_{i \in \Gamma_m} w_i \leq U_m \quad m = 1 \ldots M$$ (1.38)

Pre-assignment

An investor may wish that some specific assets be included in the portfolio, in proportion fixed or to be determined. This constraint can be imposed by setting $z_i = 1$ for the corresponding assets and imposing more or less restrictive upper and lower bounds\(^{11}\). It has been discussed informally by Chang et al. in [25, 37], but is not addressed in the experimental setting.

Transaction costs

Transaction costs consist of the amount of money to be paid in order to buy assets, and can be due to brokerage costs, taxes, fund loads, bid-ask spreads etc., resulting in a reduction of the expected gain.

They cannot be considered properly as constraints, as they rather represent extensions of the model allowing to take into account additional real-world features. As stated in Konno and Wijayanayake\(^8\) the total costs follow a non-convex function on the size of the transaction: at the beginning it is concave up to a certain point (unit-transaction cost gradually decreases as size increases), then it increases linearly up to another point (unit-transaction costs are here constant) and then becomes convex due to the illiquidity premium (unit price increases due to the shortage of supply). Thus, transaction costs can be plotted as a V-Shaped function\(^{164}\).

It has been proven that ignoring transaction costs leads to inefficient portfolios\(^5\). Nevertheless metaheuristics lack in including transaction costs and just a few authors considered them in their formulation of the PSP\(^{[95][159][164][105][106]}\). This is because most works use the Mean-Variance formulation (see section 1.1.1) that fails in including fixed transaction costs (but it can handle proportional costs). Indeed, even if Modern Portfolio Theory states that diversified portfolio are preferable to non-diversified ones\(^{[113]}\), there is evidence that investors choose non-diversified portfolios\(^{[17, 67, 75]}\). This is due to the influence of transaction costs, since they are not included in the original model.

Transaction costs are instead taken into account by works exploiting other approaches, such as LP-based heuristics approaches (see section 2.4). In these works, exploiting an integer formulation, both fixed and proportional transaction costs are taken into account.\(^{12}\)

Fixed transaction costs can also be applied if the sum of money invested in the individual asset exceeds a given threshold\(^{[82]}\); this is made in order to facilitate small investors exclude taxes and other fixed costs when the amount invested in an individual security is small.

Maringer\(^{[106]}\) investigates fixed only, proportional only, proportional with lower bound and proportional plus fixed costs, using an integer formulation in which the invested amount varies from 500 up to 10,000,000 euros. It is shown that the higher the fixed costs, the smaller the cardinality of the portfolios and the portfolio performances: this effect is more evident for small

\(^{10}\)It is worthwhile to notice that a class constraint could be in principle conceived as any weighted sum (or, more generally, any combination) of variables (see [114]). In this form, even the budget constraint can be thought of as a class constraint.

\(^{11}\)In truth, also a constraint relating $z$ and $w$ values should be imposed, i.e., $w_i \leq z_i$.

\(^{12}\)Taxes can be considered as additional proportional costs whose amount is determined w.r.t the type of trader and the kind of operation performed, see [102].
investors. Proportional costs instead, depending exclusively on the transaction volume, cannot be avoided by substituting securities by adding shares to already included ones. Nevertheless, even in this case the higher the cost, the smaller the cardinality of the portfolios and the portfolio performances. The case of compound costs is even more interesting and has been tackled by [2] too, where the proportional cost to be paid for a given asset cannot be lower than a threshold $P_{min_i}$. This means that for each asset $i$, assuming that $pc_i$ represents its associated proportional cost, the investor will always buy a number of rounds (as they represent decision variables, see section 1.2.3) $x_i$ such that $pc_i q_i x_i \geq P_{min_i}$, where $q_i$ represents asset price. This implicitly defines a lower bound $\varepsilon_i$ for each asset:

$$\varepsilon_i = \left\lfloor \frac{P_{min_i}}{pc_i q_i} \right\rfloor$$ (1.39)

The higher $P_{min_i}$, the higher the lower bound, while the higher the proportional costs, the smaller the lower bound. This lead to the phenomenon that for a given $P_{min_i}$, the portfolio will be more diversified the higher the transaction cost is. Imposing proportional plus fixed combines the effect of the two costs, as increasing them will reduce portfolio diversification.

So, considering all typologies, global transaction costs tend to reduce portfolio diversification, but this assertion must be taken *cum grano salis* as investor behaviour depends on subjective factors too: in Glover et al.[63] it is explicitly stated that if the investor is risk-averse the portfolio held is more diversified with taxes and transaction costs, whilst diversification is not requested by investors with low risk-aversion if taxes an transaction costs are included in the model. It is clear however that only proportional costs are suitable to be included in the continuous model, as the remainder is sensitive to the invested amount.

**Turnover and trading constraints**

For the sake of completeness, we also mention a class of constraints that arise in the multi-period formulation of the problem. These constraints define upper and lower bounds, respectively in case of buying and selling, for the variation of asset values from one period to the next one. Moreover, they are usually combined with transaction costs and taxes. These constraints have been introduced by Crama and Schyns in [30] in a variant of the single-period formulation.

The complete classification of the PSP model variants that can be found in the main literature on the subject is depicted in Fig. 1.3.
Figure 1.3: Conceptual representation of all the PSP model attributes (variables, objectives and constraints).

1.3 Conclusion

In this chapter the main Portfolio Models have been outlined. The basic distinction between active and passive portfolio strategies has been introduced, and this will be central to the remainder of the work. A conceptual representation of active portfolio selection models has been introduced and this will be helpful in the next chapter, where metaheuristic approaches for portfolio choice will be surveyed.
Chapter 2

Portfolio selection by metaheuristics

In chapter 1 we outlined the basic models for the PSP, stating the difference occurring between active and passive portfolio management and introducing a conceptual representation of active portfolio strategies (the same representation will be tailored to passive strategies in chapter 4). In this chapter we will introduce the concept of metaheuristics, together with a brief classification, and we will classify metaheuristic approaches for the PSP by means of MAGMA, a general framework for metaheuristics.

2.1 Metaheuristics

Metaheuristics are solution finding strategies, based on which approximate algorithms for combinatorial optimisation problems can be designed and implemented. In general, metaheuristics can be defined as incomplete algorithms that try to combine basic heuristic methods in higher level frameworks aimed at efficiently and effectively exploring a search space\textsuperscript{1}. Furthermore, there is still plenty of confusion, in the financial field, between the terms heuristics and metaheuristics\textsuperscript{[62, 106]}. The term metaheuristic comes from joining two ancient Greek words: heuriskein which means to find, and the suffix meta, that means beyond, in an upper level.

Without entering into details, we can highlight the main metaheuristics features as follows:

- Metaheuristics are strategies that guide the search process.
- The goal is to efficiently explore the search space in order to find (near)optimal solutions.
- Techniques which constitute metaheuristic algorithms range from simple local search procedures to complex learning processes.
- Metaheuristic algorithms are usually non-deterministic.
- They may incorporate mechanisms to avoid getting trapped in confined areas of the search space.
- The basic concepts of metaheuristics permit an abstract level description.
- Metaheuristics are not problem-specific.
- Metaheuristics may make use of domain-specific knowledge in the form of heuristics that are controlled by the upper level strategy.

\textsuperscript{1}Although the term metaheuristic is widely accepted, these methods are sometimes referred to as modern heuristics. [77]
In general, metaheuristic-based algorithms cannot prove the optimality of the returned solution, but they are usually very efficient in finding near-optimal solutions. Some techniques, such as tabu search, iterated local search, variable neighborhood search, ant colony optimisation and evolutionary algorithms have proved to be very successful in tackling real-world problems. Although these techniques show much differences amongst them, their goal is to perform a guided local search using some high-level criteria, while avoiding getting stuck in local minima through the use of an escape mechanism. These high level criteria are generally aimed to exploit a dynamic balance between diversification and intensification. Although used to convey several meanings, the term diversification generally refers to the exploration of different portions of the search space, whereas the term intensification refers to the exploitation of the accumulated search experience, through visiting search space regions that appear to be more promising. This means to identify regions in the search space with high quality solutions and to discard regions of the search space which are either already explored or which do not provide high quality solutions. This dualism between intensification and diversification has been introduced in some of the works we are going to outline and will be highlighted when necessary.

There are several ways to classify and describe metaheuristic algorithms. Depending on the features selected to differentiate amongst them, several classifications are possible, each of them being the result of a specific viewpoint. Amongst these several ways, we want to introduce the following:

**Trajectory Methods versus Discontinuous Methods** When using trajectory methods, the next solution is obtained by searching within the neighborhood of the current solution (set of solutions obtained by slightly modifying the current solution). Conversely, in discontinuous methods, the new solution can also be far from the current solution, as there are operators (e.g., genetic operators), charged to generate the new solutions by jumps in the search space.

**Single Agent Methods versus Multi-Agent Methods** When exploiting a single agent method, only one solution at a time is processed, while multi-agents (populations) are searching agents who contribute to the search process independently and in parallel.

**Guided Search versus Unguided Search** Guided search, also referred to as search with memory storage, incorporates some additional rules and hints about where to search. For instance, in ant colonies this is attained by pheromone traces, representing an adaptive memory of previously visited solutions, and the same can be seen with Tabu Search, which implements the Tabu List, a memory that stores recently visited states. On the contrary, unguided search (memoryless) methods rely only on the behaviour of the search process, without additional hints and help.

For further details on metaheuristics we forward the reader to [16] and [71]. In the following we will not go into a detailed description of metaheuristics, rather we will introduce metaheuristics used in Portfolio optimisation, giving a brief conceptual description before looking at how they are implemented. Other short details will be given in section 4.3, when introducing metaheuristics approaches for the Index Tracking Problem.

### 2.2 Metaheuristic techniques for the portfolio selection problem

In this section, we provide a review of the most relevant metaheuristic approaches to the PSP. To this purpose, we will adopt the standpoint provided by MAGMA, a general framework for metaheuristics [118]. MAGMA (MultiAGent Metaheuristics Architecture) provides a framework
for classifying and designing metaheuristics as multi-agent systems. Metaheuristics can be seen as the result of the interaction among different kinds of agents: the basic architecture contains three levels, each hosting one or more agents. At each level there are one or more specialised agents, each implementing an algorithm. LEVEL–0 provides a feasible solution (or a set of feasible solutions) for the upper level, therefore it can be considered as the (initial) solution construction level. LEVEL–1 deals with solution improvement and agents perform a trajectory in the search space until a termination condition is verified (basic local search level). LEVEL–2 agents have a global view of the space, or, at least, their task is to guide the search toward promising regions and to provide mechanisms for escaping from local optima (long term strategy level). Classical metaheuristic techniques, such as tabu search, can be easily described via these three levels. This basic three level architecture can be enhanced with the introduction of a fourth level of agents, LEVEL–3 agents, coordinating lower level agents. With this fourth level, the framework can also describe hybrid techniques such as large neighborhood search, in which complete solvers are integrated into metaheuristics [32], [127]. We first survey the basic concepts metaheuristics for PSP are based upon, i.e., the various choices for defining the set of feasible solutions, the cost function, and the neighborhood structure(s). Then, we give an overview of the techniques level by level, starting from the solution construction till the most general search strategies.

2.2.1 Metaheuristic attributes

We can conceive a metaheuristic as an abstract class whose attributes are the search space, the cost function and the neighborhood structure(s), that represent the basic components of the search strategy. Once these attributes are instantiated, the search strategy can be designed by instantiating the algorithm for each of the search levels, i.e., solution construction, solution improvement, search strategy and coordination strategy.

Search space

Usually, a solution to the PSP is represented by an array of n variables $w_1 \ldots w_n$, where $w_i$ represents the fraction invested in asset i (or the actual amount of money in the integer variables model). Besides those variables, auxiliary variables and data structures can be added for improving algorithm efficiency. An important distinction has to be made in the way the different approaches deal with constraint violations. Indeed, some works define the search space explored by the algorithm as consisting of only feasible portfolios (i.e., satisfying every constraint in the model), while in other works the search process is allowed to also explore unfeasible solutions.

We therefore can classify the search processes depending on how they handle infeasibility:

- **All feasible** approach: each candidate solution $s$ must satisfy the constraints at any step of the search process (e.g. Chang et al. [25]).

- **Repair** approach, in which if an unfeasible solution is found, this is immediately forced to satisfy the constraints by means of an embedded repair mechanism (e.g. Streichert et al. [151]).

- **Penalty** approach: it is allowed to visit unfeasible solutions, but those will be assigned a penalty in the cost function, depending on the amount of violation (e.g. Schaerf [137]).

Repair mechanisms provide a trade-off between diversification and intensification. A basic repair approach is presented in [41] and [133]; these works tackle an unconstrained formulation, so constraints likely to be violated are the budget and return constraints. In this case the
only action to be performed is to normalise weights so as to sum to one, but in [41] weights are normalised at each step, whilst [133] repairs solutions only after finding five consecutive unfeasible moves. This mechanism repairs asset weights in the following way:

\[ w'_i = \frac{w_i}{\sum_j w_j} \]  

(2.1)

where \( w_i \) represents the actual weight of asset \( i \) and \( w'_i \) the repaired weight.

Extended versions of this mechanism are given by [137](idR and idID), where the repair mechanism must satisfy the floor and ceiling constraints too. This is done by normalising, for each asset \( i \), values \( w_i - \varepsilon_i \) rather than \( w_i \), as previously introduced, in order to ensure that no asset can fall below the minimum allowed \( \varepsilon_i \); the repaired asset weights (accordingly with the previously introduced notation) will be:

\[ w'_i = \varepsilon_i + \frac{w_i - \varepsilon_i}{\sum_j w_j - \varepsilon_j} \]  

(2.2)

A more complex typical repair mechanism is explained in Streichert et al.[152], referring to a formulation with cardinality and minimum lots constraints. This mechanism takes as input a non-normalised solution vector and repairs it through the following deterministic procedure:

1. A vector \( w' \) is generated: if a cardinality constraint is imposed, it will contain only the \( k \) asset with highest weights whilst the surplus variables are set to 0; otherwise \( w' \) will be composed of all assets;

2. All weights are normalised so as to sum to one. This is done by setting weights \( w''_i = \frac{w'_i}{\sum_j w'_j} \);

3. A further modification is required to meet minimum lots constraint: asset weights are forced to the largest roundlot level less or equal than the current asset weight, i.e., \( w''_i = w''_i - (w''_i \mod u_{\text{min}}(i)) \). The residual amount of budget is redistributed so as to meet minimum lots constraint by buying quantities \( c_i \) of assets with the largest \( (w''_i \mod u_{\text{min}}(i)) \) until all the budget is spent.

This repair mechanism can fail in finding feasible solutions w.r.t return constraint. In this case, as authors exploit a genetic algorithm, the fitness of the portfolio will be assigned the worst possible value.

Indeed, these three ways of handling constraints are mostly used together, deciding, for each constraint, which is the most suitable way for handling them. For instance, a budget constraint is used to norm the solution, so it is preferred having solutions strictly satisfying them, exploiting either a feasible approach[30] or a repair mechanism[83]. Also return constraint is used to norm the solution, but it can be used in both feasible or penalty approaches. The choice between these two strategies depends on the cost function used: the penalty approach is used if the cost function consists of a combination of the objective of the problem and the violation of the return constraint, as it allows moving toward an unfeasible state, assigning a penalty for the violation of constraints (the main examples of this approach will be discussed in section 2.2.1). If other cost functions are used, the feasible approach is preferred[30].

There is, instead, no reason for preferring one of these strategies when dealing with other constraints. For instance, at point (1) we explicitly referred to the case of cardinality constraint as being repaired by a useful mechanism, but ensuring that solutions are always feasible w.r.t. this constraint has been exploited in [30, 83, 104, 3].
We mention that it turns out to be difficult determining which class a search method belongs to, as it can be difficult to determine if a search trajectory moves only in feasible areas because of its formulation or because an implicit repair mechanism is embedded. For this reason, the pure all-feasible approach can hardly be found in literature, and the few examples are to be reformulated as unconstrained PSP\[24, 133\].

Cost function

When the PSP is attacked by metaheuristic algorithms, it is important to distinguish between objective function and cost function. The former represents the function to be optimised to solve the problem, while the latter represents the function guiding the search process over the search space. Often the objective of the problem is used as the cost function, but sometimes different cost functions can better guide the search toward promising solutions.

An example of cost function for the PSP is provided by Schaerf\[137\] who defines a cost function in which the cost associated to the violation of return constraint \(f_1(W)\) is combined with the original objective function \(f_2(W)\). The overall cost function to be minimised is a weighted sum of the two components \(a_1 f_1(W) + a_2 f_2(W)\), where \(a_1\) and \(a_2\) vary during search according to a shifting penalties mechanism.

\[
\begin{align*}
\min \ a_1 f_1(W) + a_2 f_2(W) \\
f_1(W) &= \max \left(0, \sum_{i=1}^{n} r_i w_i - r_e\right) \\
f_2(W) &= \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} w_i w_j
\end{align*}
\]

A similar approach is followed by Gilli and Kellezi in \[59\]. They choose the following cost function:

\[
\begin{align*}
\min \left( \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} w_i w_j + \overline{p} \left( r_e - \sum_{i=1}^{n} r_i w_i \right) \right) \\
\overline{p} = \begin{cases} 
\overline{c} & \text{if returns are smaller than } r_e \\
0 & \text{otherwise}
\end{cases}
\end{align*}
\]

where \(\overline{p}\) is the penalty term and \(\overline{c}\) is greater than 0.

Neighborhood relations

The neighborhood relation defines the states of the search space that are reachable from the current one. The definition of neighborhood structures to be used during the search is one of the key components of metaheuristic algorithm design.

In general, neighborhood relations can be divided in two classes:

1. Neighbors are generated by modifying the weights of a subset of the assets of the current portfolio.

2. Neighbors are generated by modifying all the assets in the current portfolio.
We can ideally define a neighbor of a solution by selecting one asset to be modified, specifying the amount of variation and performing the change. This asset is referred to as pivot\cite{3}. Then, this modification is counterbalanced by changing the weights of some other assets. If only a pre-determined subset of assets is selected to be modified the neighborhood is said to belong to class 1, otherwise the neighborhood is said to belong to class 2.

The neighborhood structures of class 1 can either consist only of feasible solutions (e.g., \cite{137}, structure TID) or allowing unfeasible moves too (e.g., \cite{133}). The simplest neighborhoods in this group are generated by modifying the pivot weight and counterbalancing this change by modifying the weight of only one other asset (\cite{133} TID; \cite{59}). This structure can be generalised by introducing an integer $pv$ representing the number of assets to be modified in order to counterbalance the pivot variation. Crama and Schyns\cite{30} use $pv = 2$, but it is possible to set $pv$ at any number, even 0, thus allowing unfeasible moves\footnote{In this case a repair mechanism should be included.}. In the previously described neighborhoods, the step size is set before choosing the assets involved in the modification; however, neighbors can also be generated by varying this value. For example, in \cite{3} neighbors are generated by varying step from a minimum of $\frac{w_{pivot}}{n}$ to a maximum of $w_{pivot}$, being forced to assume all multiples of $\frac{w_{pivot}}{n}$.

Neighborhoods of class 2 are generally used in population-based algorithms\cite{41, 99, 151, 95}, especially in genetic algorithms, in which crossover and mutation operators could return unfeasible solutions. In this case, it is often impossible to determine which asset plays the role of pivot. Anyway, there are some representative cases in which the pivot is used, such as in \cite{25} and \cite{24}.

The first examples of neighborhood relations in local search for the PSP were introduced by Rolland\cite{133}. These neighborhoods are defined for the unconstrained model, i.e., the one with only theoretical constraints as explained in section 1.1.1, and can be considered the basic structures upon which further developments have been designed. In the first structure (referred to as RollandI) the neighbor of a solution is defined as a solution in which the weight of only one asset is increased or decreased of a given quantity, called step. The second neighborhood (referred to as RollandII) is defined so that the weight of an asset is increased or decreased of a given step and the value of another asset is respectively decreased or increased of the same value.

With these two neighborhood structures, assets contained in the final solution are a subset of the starting portfolio, since the assets to be modified are chosen amongst the ones in the portfolio. Anyway, this does not prevent the search from being able to explore all the possible asset combinations, because the model is unconstrained and the portfolio is initialised with $w_i = \frac{1}{n}$, for each asset $i = 1 \ldots n$. We also observe that RollandI might move the search to unfeasible solutions. These neighborhoods are well suited for the unconstrained model, but have to be modified for the constrained models because assets cannot be present in the portfolio in every possible quantity. Hence, these neighborhood structures are modified by embedding asset insertion and deletion operations.

RollandII can be extended by transferring a quantity from one assets $i$ to another asset $j$ even if the latter does not belong to the portfolio. In this case, asset $j$ will be inserted in the portfolio (see the neighborhood called TID in \cite{137} and \cite{59}). This approach should also include some mechanisms to handle upper and lower bounds, in case they are included in the formulation. RollandI can be modified by enforcing the satisfaction of the budget constraint and by allowing insertions and deletions of assets. Feasibility w.r.t. the budget constraint can be enforced by increasing the weight of one asset and decreasing other assets weights \cite{24}. More precisely, if a solution is given by a weight vector $(w_1 \ldots w_n)$, a neighboring one is $(\frac{w_i}{1+\text{step}}, \ldots \frac{w_i+\text{step}}{1+\text{step}}, \ldots \frac{w_n}{1+\text{step}})$, for only one $i$, $1 \leq i \leq n$. This neighborhood is proven to be complete, i.e., for a long enough
sequence of moves, each solution can in principle be reached. Completeness does not depend on the initial solution and holds iff \( \text{step} \leq \frac{1}{n} \). The possibility of having asset insertions and deletions lead to neighborhoods defined in [137] (called \( \text{idR} \)), and [25]. This neighborhood takes into account the case that an asset \( i \) is decreased so that its value falls below its lower bound \( \epsilon_i \); hence, asset \( i \) is deleted and another asset \( j \) is inserted in the portfolio. Conversely, if asset \( i \) is increased so that its value exceeds its upper bound \( \delta_i \), then its weight is set to \( \delta_i \) and all other asset weights are normalised. Note that all these variants do not change the number of assets in the portfolio. A further improvement is thus possible by allowing neighbor solutions to have different number of assets (see [137], \( \text{idID} \)), defined by allowing three kinds of operations on the selected asset \( i \):

- If asset \( i \) is already in the portfolio, increase its weight of a given quantity. If the resulting value exceeds the upper bound \( \delta_i \), then set the value to \( \delta_i \).
- If asset \( i \) is already in the portfolio, decrease its value of a given quantity. If the value falls below the lower bound \( \epsilon_i \), asset \( i \) is deleted and not replaced by any asset.
- If asset \( i \) is not in the portfolio, it is inserted in the portfolio with weight equal to its lower bound.

After these operations, asset weights are normalised.

2.2.2 Metaheuristic search components

In this section, we describe the search methods composing the metaheuristics for the PSP. We first present single-agent (trajectory) based strategies, such as simulated annealing and tabu search, then we introduce population-based metaheuristics, such as evolutionary algorithms and ant colony optimisation.

Initial solution

It has been empirically observed that metaheuristics for the PSP are usually quite robust with respect to the initial solution choice. This assertion has been formally proven by Catanas in [24], subject to the specific neighborhood structure defined therein. For this reason, most works assume as starting solution a randomly generated one or a solution constructed by means of a simple heuristic procedure [47], possibly embedding a mechanism to ensure feasibility [30].

Iterative improvement

Iterative improvement can be considered as the simplest local search, as it performs a path in the search space by moving from a solution to a neighboring one with a lower cost. This search can be named \( \text{best improvement} \) (alternatively referred to as \( \text{steepest descent} \)), if the neighbor chosen is the best among feasible neighbors, or \( \text{first improvement} \) (alternatively referred to as \( \text{first descent} \)), if the chosen neighbor is the first state found during the neighborhood enumeration that is better than the current one. Iterative improvement is usually incorporated into a more complex strategy, rather than being used as a stand-alone local search. For instance, in Glover et al. [63] iterative improvement is the local search component of a variable neighborhood search technique. As another example, we mention Armañanzas and Lozano [3] who use a greedy search to refine solutions found by an ant algorithm.
Simulated annealing

The possibility of moving to solutions with a higher cost (i.e., performing degrading moves) characterises Simulated annealing (SA). The probability of moving toward solutions worse than the current one depends on the cost difference between the two solutions and it also decreases during the search. This probabilistic acceptance criterion enables the search to escape from local optima. Crama and Schyns [30] apply SA to various PSP models by first considering only one constraint class at a time (floor, ceiling and turnover first, then trading and cardinality), then they include all constraints in the model. The authors experiment with three strategies:

- Independent runs, starting from the same initial solution.
- Subsequent runs, using as initial solution for the current run the best one found in the previous one.
- Run the algorithm a number of times such that a list \( P \) of promising solutions is created, then perform \(|P|\) independent runs, using as initial solutions the ones stored in \( P \).

The fact that there is no clear dominance among these strategies gives support to the statement that such search processes are insensitive to the initial solution. SA by Crama and Schyns is able to plot the UEF exactly and achieves good performances in the model with floor, ceiling and turnover constraints. Nevertheless, the ACEF returned in the model with trading constraints appears to be quite rugged. Anyway, in all the cases this technique is able to approximate the CEF in reasonable runtimes for medium-sized instances.

The concepts of SA can also be effectively used inside population-based algorithms, as done in [83] [65]. In the approach proposed in [83], an initial population of random portfolios is generated. Then, for each portfolio \( p_o \) in the initial population, a new portfolio \( p_n \) is created by selecting some assets \( i \) and modifying them according to the following rule:

\[
\omega_{i_n} = \max(\omega_{i_o} + s, 0)
\]

where \( s \) is randomly chosen in the range \([-U_t, U_t]\) and this range decreases over time. Weights are then normalised and \( p_n \) is evaluated and accepted or not depending on the Metropolis criterion. Once the new population is created, it is further refined by replacing worst portfolios either by a clone of a probabilistically selected portfolio with higher fitness with probability \( r \) or, with probability \( 1 - r \), by a portfolio composed of assets with average weights over best portfolios.

An interesting application of SA for a multi-objective formulation is presented in [3], in which moves are always accepted if at least one criterion is improved, while deteriorating moves are subject to the SA usual probabilistic acceptance criterion. This approach, applied in a formulation with floor, ceiling and cardinality constraints, finds good solutions in the lower part of the frontier, where risk and profits are small.

Threshold accepting

Threshold accepting (TA) shares some analogies with SA, as a degrading move can be accepted if the cost difference between the current and the new solution is within a given threshold, that is progressively decreased to zero. The threshold decreasing schedule is defined by estimating the distribution of distances between objective values of neighboring positions (an analogous parameter tuning procedure is undertaken also for SA). TA has been applied to the PSP by Dueck and Winker [45] and Gilli et al. [59] [61]. These works are primarily aimed at comparing risk measures, so the algorithm represents the technical mean to investigate financial aspects.
2.2 Metaheuristic techniques for the portfolio selection problem

For example, in [45] different risk measures are compared. Experimental results show that the solutions found optimising a risk measure are generally not efficient w.r.t. other measures. In this way it is possible to directly compare risk measures. In [45] it is stated that the ACEF is not smooth, since it turns out to be composed of linear fragments, and the curve switches from a segment to another one when the fraction held in a particular asset changes sharply.

Besides solving a floor and ceiling constrained mean-variance PSP, Gilli and Kellezi [59] tackle a more realistic problem in a downside-risk framework in which decision variables are integers. The problem is formulated as a maximisation of future returns, while Value-at-Risk (VaR) and Expected-Shortfall (ES) are compared as risk measures constraining the shortfall probability for a given level of ES and VaR. In a further work [61] TA is used to compare three different risk measures: VaR, ES and Omega (Ω) measure (defined as the ratio of the weighted conditional expectation of losses over the weighted conditional expectation of gains) in a formulation with cardinality and upper/lower bounds constraints. Results show that Mean-VaR portfolios are more diversified than those obtained with ES criteria, while ES frontier dominates the other two.

These works stress the importance to be paid to the choice of appropriate risk measures. Indeed, efficient portfolios w.r.t. a risk measure are usually not efficient with respect to other measures and efficient portfolios are very different from each other with respect to different utility functions.

Tabu search

The Tabu search metaheuristic (TS) moves away from local optima by forbidding the search to execute the inverse of the last $l$ recently performed moves. This simple mechanism, enhanced with the exploitation of the search history for intensifying and diversifying the search, makes TS one of the best performing local search strategies. The application of TS to the PSP has its milestones in the works by Rolland [133] and Glover et al.[63]. These works refer to different formulations of the problem and moreover Glover tackles a multi-period formulation. Nevertheless, both deserve to be analyzed for the richness of concepts presented.

Rolland uses a TS for the unconstrained problem. The author tackles two problems of minimising variance and minimising variance given an expected level of returns. That work is more oriented in finding a single point (describing the trajectory followed by the algorithm over time to reach it) rather than drawing out the whole UEF.

The approaches designed for tackling these two problem formulations differ in the repair mechanism. In the minimum variance formulation, after having executed five steps in the unfeasible search space area, the algorithm repairs the incumbent solution as follows:

- If the investment exceeds the budget (i.e., if $\sum_i w_i > 1$), find the asset $i$ with maximum sum of covariance referring to other assets ($i$ such that $\sum_j \sigma_{ij} w_i w_j$ is maximal) and decrease $w_i$ in order to ensure feasibility.

- If the investment is less than the budget (i.e., if $\sum_i w_i < 1$), find the asset $i$ with minimum sum of covariance referring to other assets ($i$ such that $\sum_j \sigma_{ij} w_i w_j$ is minimal) and increase $w_i$ in order to ensure feasibility.

The algorithm for the minimum-variance-given-return formulation initially tries to reach the desired level of returns, repairing the solution as follows (after having visited consecutively five unfeasible solutions):
• find

\[ i = \text{argmin} \left| \left( 1 - \sum_j w_j r_i \right) - \left( \sum_j w_j r_j - r_e \right) \right| \]  

(2.9)

• If the investment exceeds the budget (i.e., if \( \sum_i w_i > 1 \)), decrease \( w_i \) in order to make the solution feasible.

• If the total investment is less than budget (i.e., if \( \sum_i w_i < 1 \)), increase \( w_i \) so as to make the solution feasible.

When the return level of the best solution found is within the 0.005% of the desired level, the repair mechanism invoked is the one described for the minimum variance problem, so that the solution is feasible w.r.t. the requested minimum-variance point after the requested return level has been reached.

Even if the proposed TS is said to attain good performances, it is useful only to find single point instead of the whole UEF, therefore this implementation does not represent the most powerful solution for real-world problems; however, it can be useful when only one desired level of return is given.

Glover et al. tackle the asset-allocation with fixed-mix, a problem similar to the PSP. This is a multi-period problem in which we want, for each period, to respect the proportions of asset classes (in this case assets, bonds and treasury bills) out of the whole portfolio, in order to attain the same risk profile for each period, taking into account cash-flows generated by the portfolio management. At the beginning of each period, the portfolio must be re-balanced in order to ensure feasibility, as assets generate dividends to be re-invested, transaction costs must be taken into account and constraints on proportions held can be considered. The simplest strategy is given by selling a portion of asset classes with returns higher than the average return and buying a portion of asset classes with returns below average.

Both cases with and without transaction costs are investigated and the search strategy is implemented by interleaving TS with variable scaling. With this term we indicate a strategy in which, like a basic Variable Neighborhood Search (see afterwards) the neighborhood changes over iterations due to a change of the step length of moves (the biggest step length is 5% and the smallest is 1%). Step lengths are defined and ranked in decreasing order, and an Iterative improvement search is performed with the first step length. When no improvements are obtained, the step length changes to the next value and the Iterative improvement procedure is repeated starting from the last solution found. This process is iteratively repeated until the last step value of the list is reached. At this point, if improvements were reached over the list, the process restarts from the first value, otherwise the procedure stops. At the end of this phase, a TS run is performed; in case of improvements, the search switches back to variable scaling and the process continues until no improvements are achieved. The step size is crucial for the effectiveness of the algorithm and in TS it is set at a higher value than in Variable Scaling so as to diversify the search. The ACEF is compared with the frontier obtained with exact global optimisation, and it is shown that they are almost identical, in both the cases with or without transaction costs.

Tabu Search has been widely applied to solve the PSP. It is easy to find it in works aimed at comparing the performance of different algorithms on the same instance (see sec. 2.3). A very successful application of TS can be found in Schaerf [137], in which TS is improved by dynamically changing the neighborhood structures.
2.2 Metaheuristic techniques for the portfolio selection problem

Variable neighborhood search

Variable Neighborhood Search\[68\] (VNS) is a metaheuristic that dynamically changes neighborhood structures during search, so that a neighborhood is substituted by another one when the current solution cannot be further improved. There is no explicit application of VNS to the PSP, however, as this strategy is very general, its principles can be found in some important works in the literature. This is the case of the work by Glover et al.\[63\], in which the implementation of variable scaling can be considered a VNS, as a new neighborhood is introduced by changing the step when no further improvement is possible. A similar technique can be found in \[47\], in which the search switches between two neighborhoods. Moreover, similar ideas can be found in \[137\], in which TS is implemented in a token ring sequence, in which runs using a different neighborhood structures are interleaved.

It is worth mentioning also the work by Speranza\[145\], in which a heuristic algorithm is defined and applied to the Milan Stock Market using an integer formulation enriched by introducing proportional transaction costs (floor, ceiling and cardinality constraints are discussed but not addressed in the computational analysis). Here, in order to satisfy the constraints on capital, assets are ordered and re-numbered in nondecreasing order of \(x_i\) in the portfolio; then \(x_1\) is increased (and, if this move is unsuccessful, decreased) by one unit. If the new solution is feasible, the algorithm stops, otherwise the procedure is repeated over \(x_2 \ldots x_n\). At the end of this phase, if no feasible solution is found, the cycle is repeated increasing assets by two units, then three and so on. This mechanism can be considered as a sort of VNS, even if the neighborhood cardinality is constant over the whole process and the neighbor selection process is deterministic.

Evolutionary algorithms

Evolutionary algorithms (EA) are population-based metaheuristics whose inspiring principle is the Darwin theory of natural evolution and selection. These search strategies maintain and manipulate a set of solutions at each iteration, combining the best solutions of the current set to generate the solutions of the new set. Often EA-based metaheuristics are enhanced by hybridising EAs with advanced constructive procedures and local search strategies. The strategies presented in these works can be better labelled as memetic algorithms, as local search runs are executed to improve the quality of the solutions constructed by the EA.

The first applications of EAs to the unconstrained PSP are presented by Tettamanzi et al. in \[4\][99][100]. In \[4\] a genetic algorithm (GA) is implemented for the PSP with down-side measure of risk. In the algorithm, one population is handled and individuals are generated according to investor preferences: a specie is defined for each \(\lambda\) (where \(\lambda\) is the trade-off coefficient between return and risk, as discussed in sec. 1.2.2). Individuals are generated such as their probability of belonging to a specie is proportional to the investor’s interest in that specie. At each generation, a new individual replaces the worst one in the previous population.

In a further work \[100\], a distributed genetic algorithm is applied in which each \(\lambda\) value is associated to a subpopulation. As the AUEF (the unconstrained PSP is tackled) is composed by plotting a point for each \(\lambda\), the greater the number of populations, the finer the resolution of the frontier. Migrations of individuals between populations corresponding to neighboring values of \(\lambda\) are permitted, in order to avoid premature convergence of the algorithm. Individuals are allowed to mate only with individuals of the same population or of adjacent ones. This implementation outperforms the previous sequential version, and in \[99\] a detailed description of the implementation and risk measures is provided. In parallel implementation of GA, if the cardinality constraint is imposed it is possible to search in parallel several ACEF corresponding to each value of \(k\), using information from each of these to improve the search process of others.
With this approach, the ACEF approximates the UEF with increasing precision, as \( k \) increases and constrained optimal portfolios are shown to be not significantly different from unconstrained ones, except for very small number of assets and very low risk levels.

Liu and Stefek [97] tackle the PSP with cardinality and ceilings constraints, comparing GA with a heuristic proprietary method and they investigate crossover rates, population size and elitist strategy showing that GA can achieve good performances, even if worse than the heuristic, especially concerning execution time.

Memetic algorithms for the PSP are presented in [110], in which the use of SA and TA inside the EA framework is compared to tackle the unconstrained PSP. The results discussed indicates that TA is more suitable when \( \text{VaR} \) is used as risk measure, while SA makes the algorithm perform better when \( \text{ES} \) is chosen. An explanation of this result is given by observing that \( \text{VaR} \) induces a rugged search space, while \( \text{ES} \) induces a smoother landscape. In that work also the use of a kind elitist strategy is investigated, that implement a sort of intensification of the search around the best found solutions. This strategy improves the performance of the algorithm when the search space is smooth, while it does not payoff when the search space is rugged, as it reinforces the local optimum we want escape from. Moreover, the introduction of this kind of intensification makes the algorithms more robust against changes in parameter values.

The previously discussed works (together with [160, 47, 164, 25]) tackle the inherently multi-criterion PSP using single-objective formulations and techniques (see section 1.2.2). Indeed GA, by being inherently effective in diversifying the search, show good performances especially in multi-objective formulations of PSP, as shown by the applications of MOEAs (Multi-Objective Evolutionary Algorithms) [150][151][152][124]: for instance, \textit{NSGA II} [33] represents one of most powerful multi-objective metaheuristics and has been applied to PSP in [95, 41]. Furthermore it has been argued that handling PSP in single-objective fashion make strategies less flexible to decision makers preferences[153].

In this multi-objective framework, PSP is tackled by Streichert et al. in [150, 151, 152], using a bi-objective optimisation model, enriching their implementation by adding an archive in order to store the frontier obtained so far. In their work, they introduce the knapsack representation of portfolios, comparing it with the standard one. The authors also investigate the use of \textit{Lamarckism}. In fact, these algorithms embed a repair mechanism that prevents the search from rejecting unfeasible solutions. In the GA version without Lamarckism, only the phenotype of an individual (i.e., the normalised vector of assets) is altered by the repair mechanism, while the genotype (i.e., the non-normalised vector of assets) remains unaltered. Conversely, in the version with Lamarckism, the repair mechanism modifies the genotype too, according to the phenotype. In each case, this solution representation leads to a better performance than the standard one. Moreover, Lamarckism also helps to improve performances. Furthermore, different variable representations (binary and real-valued) are also compared and different coding [151] and crossover operators [152] are examined.

We should observe that the model with floor, ceiling and cardinality constraints is the most commonly used in literature when GA are applied [25][47][53]. GA have also been used in conjunction with formulations differing from the canonical Mean-Variance one, in order to define more realistic customer-oriented frameworks. An interesting example is represented by [164], in which the objective function to maximise is given by the usual weighted objective function (eq. 1.18), but they solve this model for different isolated values of \( \lambda \) rather than trying to plot the whole frontier. They show that in the obtained portfolios return is higher than the best one provided by optimisation software for Mean-Variance (LINGO[81]) even if they are more risky.
One of the main contributions of that work is that the expected return is considered as a variable, rather than an instance data. The return ranges in an interval in which arithmetical mean represents lower bound $a$ if its recent history trend has been increasing, the upper bound $b$ if its trend has been decreasing. No additional constraints are added to the formulation. V-shaped transaction costs are also investigated for portfolio revision, but they are only considered as proportional\(^3\). Transaction costs (embedded in a MAD objective function) and single $\lambda$ values analysis are considered in Wang et al.\cite{160} in which a sample procedure for stochastic returns is introduced instead of the classical scenario analysis.

More complex approaches are proposed aimed at helping decision making by introducing other measures either to define an ordinal-preference framework in which other measures are added to the formulation (see \cite{153}\cite{47}), or to predict the future return rate and to estimate the uncertainty risk of the future return rate when the sample is small \cite{124}.

**Particle swarm**

The nature-inspired paradigm referred to as *Particle swarm* is a promising search paradigm, especially when continuous optimisation problems are tackled. Nevertheless, its application to the PSP is still limited, and the works on this topic do not tackle the standard formulation, being aimed at finding one portfolio optimal with respect to a measure such as the reward-to-variability ratio out of a given set of assets, rather than drawing out the whole efficient frontier\cite{84}\cite{120}.

**Ant colony optimisation**

Ant colony optimisation (ACO) is a population-based metaheuristic inspired by the foraging behaviour of ants. Solutions are built component by component, according to a probabilistic procedure that bias the choice of the next solution component on the basis of the previous constructed solutions quality. Usually, ACO also incorporates some local search algorithms to improve the quality of the solutions built. Initially conceived for discrete spaces, ACO has been adapted also for continuous spaces (see, for instance, \cite{144}). Nevertheless, the potential of ACO for tackling the PSP appears still not completely exploited.

A successful application of ACO can be found in a PSP modeled with the cardinality constraint \cite{3}\cite{104}. The approach consists in defining a population of $n$ ants that explore a completely connected graph composed of $n$ nodes. Assets and nodes are in one-to-one mapping and the path traversed by an ant corresponds to the assets to be chosen for the portfolio. Path lengths are of exactly $k$ steps, where $k$ is the portfolio cardinality. In the case of multi-criteria optimisation, ants are divided in populations such that each population solves a problem corresponding to one objective function \cite{3}. When ants terminate the exploration phase, a greedy search refines the solutions. This method finds better solutions than SA and iterative improvement and results are particularly striking in the upper part of the frontier, where risk and profits are high.

ACO has found application in problems similar to the PSP such as the so-called multi-objective project portfolio selection \cite{42}\cite{43}, a generalisation of the bin-packing problem in which the goal is to choose a portfolio of project proposals (e.g. research and development projects) constraining the problem so as to ensure that the portfolio will contain no more than a given maximum number of projects out of a certain subset (e.g. projects pursuing the same goal) and imposing resource limitations and minimum benefit requirements.

\(^3\)In a further work\cite{165} risk-free asset are introduced and the formulation is based on a linear programming model.
2.3 Comparative studies

Comparing PSP techniques as described in the literature is an awkward task, primarily because data-sets are rarely the same, different algorithm implementations can lead to unfair comparisons, utility and performance measures are often different. Furthermore, comparisons can be driven by different criteria, such as efficiency, robustness, performance with respect to a given model, etc. For these reasons, the comparison amongst different works is not possible and we have to resort to papers describing and comparing different algorithms on the same instance set and model. Before overviewing the most relevant works on this subject, we briefly comment on the performance measures used for the comparison of the algorithms.

Performance measures are usually obtained by comparing constrained results (ACEF) with the ones obtained in the unconstrained case for each level of return (each point of the UEF) and computing statistical measures (mean and median percentage error, standard deviation etc.) for the overall frontier. There are however many ways to define an error measure. For instance, Chang et al. [25] consider the distance of the point from the UEF, defined as the minimum between the distance on the x-axis direction and the distance on the y-axis direction. Another measure can be found in Streichert et al. [150] [151] [152], where the algorithm performance is computed as the percent difference between the area below the UEF and the obtained ACEF.

The issue of comparing two frontiers is just an instance of the more general problem of comparing algorithms for multi-objective optimisation [125]. Often, also statistical tests are used [54] [41], especially to determine if the difference between UEF and CEF is significant, and some works introduce measures to determine the best portfolio in a frontier [45].

One of the first comparative works is due to Catanas [24]. That paper is focused on investigating properties of the proposed neighborhoods (see section 2.2.1). The author uses TS and SA, implemented in both robust and dynamic way to tackle the unconstrained PSP. In the robust implementation, the step is kept fixed during all iterations, while in the dynamic one it is decreased to zero during the execution. Furthermore, a schema for the variation of the step is defined such that its value is increased if solution quality worsens and decreased if solution quality improves. Moreover, a threshold on the minimum value of step is introduced, since too small values can make the search stagnate.

Chang et al. [25] introduce cardinality, floor and ceiling constraints and observe that the CEF becomes discontinuous. This is due to the fact that feasible proportions of assets are dominated (because of the existence of portfolios with lower variance and higher return); furthermore portions of frontier could not be reachable for a classical $\lambda$-weighting drawing approach, due to minimum proportion constraints. In [25], the authors implement GA, TS and SA to solve the problem. Results show that GA is able to approximate the UEF with the lowest average mean percentage error. Regarding the constrained problem, GA seems to perform better than SA and TS, but differences are not as clear as in the unconstrained case, so they use portfolios from the three metaheuristics to draw out the ACEF. Their approach is to store, for each heuristic, all the improving solutions found in the search process and, finally, deleting the dominated ones. The sets obtained by the three heuristics are then pooled to draw the ACEF. This approach shows that for the constrained problem the ACEF approximate the UEF when the asset cardinality is high (as already stated in [54]).

Jobst et al. [77] compare the results presented in [25] against two heuristic methods. The first is an integer-restart procedure that plots the CEF starting from the highest return and its corresponding risk to lower return and reduced risk. The result obtained at each stage is supplied as the starting point to the next (lower return) stage, considering it as first feasible
value (this heuristic is referred to as \textit{warm restart} heuristic). The second, inspired to an idea similar to [145], first solves a continuous relaxation without any constraint, then uses the \( k \) assets with highest weights as input for a problem in which constraints are imposed (this heuristic is referred to as \textit{re-optimisation heuristic}). Both heuristics are embedded in a branch-and-bound and they are said to outperform metaheuristics used in [25]. Anyway, we should note that re-optimisation heuristic is not able to draw the whole frontier when the continuous relaxation produces a portfolio with less than \( k \) assets.

Another important work that compares different techniques is the one by Schaerf[137], in which the model includes floor, ceiling and cardinality constraints. The author defines three neighborhood relations, specifying moves that satisfy the budget constraint, and defines a cost function that account for the violation of other constraints. The initial state is selected as the best amongst 100 randomly generated portfolios with \( k \) assets. A first phase of experiments with \textit{best} and \textit{first} Iterative improvement, SA and TS run as single solvers is performed. Then TS, the most promising solver in the preliminary experimental analysis, is chosen for an extensive experimental analysis, combining neighborhood relations in various token-ring strategies. In this case, the \textit{step} length is set to a higher value in the first used solvers to favor diversification, while it is set to a smaller value in the last used solvers for intensifying the search. Experimental results show that the best performances are achieved by token ring solvers with random steps, even if fixed steps seem to behave well too. Single solvers do not attain comparably good results.

Armañanzas and Lozano [3] compare iterative improvement, SA and ACO in a multi-objective formulation with cardinality, floor and ceiling constraints. The algorithms used are tailored to the multi-objective problem, and ACO outperforms the other techniques. The simple greedy search (iterative improvement) shows poor performances if used alone, but turns out to be effective when used to refine solutions provided by ACO. Interestingly, ACO and SA best performances are found in different areas of the frontier: the first in the upper part of the frontier, the latter in the lower part.

Also Ehrgott et al.[47] proposes a multi-objective framework with cardinality, floor and ceiling constrains in which utility functions are interpolated over utility values for a set of points. They use SA, TS, GA and a local search similar to a VNS embedding a random escaping mechanism to avoid stagnation at local minima. They test the algorithms over both random and real-world instances, showing that GA appears to be the best performing solver on both classes. The local search and SA achieve good results, while TS performances appear to be the worst ones.

A further interesting comparison is made by Fernandez and Gomez[53], in which metaheuristics by Chang et al. are compared with a neural net approach. A Hopfield network\(^5\) is used to plot the ACEF when cardinality constraint and bounds (lower and upper bounds) are imposed. Their results show that there is no significant difference between their neural network and metaheuristics such as GA, TS and SA. In order to improve the performance, portfolios from the four approaches are pooled and dominated solutions are deleted, so as to obtain an improved ACEF (the same approach pursued by Chang et al.). The solution quality returned is high, making this neural nets approach successful\(^6\). Nevertheless, the number of different portfolios returned by the neural net is lower than the number returned by other heuristics, therefore, even if the quality is high, stand-alone neural nets approaches are not suitable for solving the problem in the whole frontier.

\(^5\)Hopfield networks\([72]\) are neural network composed of a single layer of neurons fully connected and are widely applied in combinatorial optimisation[143].

\(^6\)Indeed, neural nets can capture non linear relations among variables and do not need model assumptions, therefore they are suited for forecasting future returns without relying on the stock returns normal distribution assumption. This idea has been also exploited in [146] and [170] in order to optimise portfolio management.
2.4 Linear programming based related works

For the sake of completeness, in this section we briefly review heuristic approaches based on linear programming, that can be very useful as components of more robust and complex meta-heuristic strategies. These works are also important because they provide experimental results for the mean semi-absolute deviation (see section 1.2.2) and they deal with integer formulations of the PSP in which assets are assigned integer values corresponding to the actual amount of money to be invested in each asset and variables are formalised as the number of rounds to be purchased for each asset.

Speranza[145] models the problem by including transaction costs, minimum lots, cardinality, floor and ceiling constraints and by introducing two auxiliary binary variables to indicate whether a security has fixed transaction costs and whether it belongs to the portfolio. The idea presented is to relax the integer constraint on quantities, transforming the problem into a linear programming one (to be solved efficiently even when the number of securities is high) and finding a solution to it. Fractional asset weights are then rounded to the closest integer and heuristics are applied to force the solution to satisfy capital and rate of return requirements. If the algorithm ends without solutions, less restrictive bounds on capital are iteratively set. This algorithm is tested on small instances and does not guarantee to find a feasible solution (it has been further tested in [102] and did not compare very favourably against competitor solvers); nevertheless it provides good performance when the total number of assets is low and it reaches a solution close to the optimal one when the invested capital is large.

In Mansini and Speranza[102], the formulation of the problem includes minimum lots and proportional taxes. The authors provide three heuristic algorithms based upon the idea of solving sub-problems of the original formulation, involving subsets of initial universe of assets: these subsets are composed of assets chosen exploiting the information obtained by solving the continuous relaxation, i.e., reduced costs. In the first heuristic (referred to as Basic-MILP-based-heuristic) they solve the continuous relaxation of the problem. Then, they use this solution to feed the mixed integer-linear programming solver. The second heuristic (referred to as Reduced-cost-MILP-heuristic) considers a vector $x_R$ with a number of assets greater than the vector of assets $i$ s.t. $x_i \neq 0$ as input of MILP-procedure, thus including also assets whose quantity in the solution of the relaxed problem is zero. The third method consists in an iterated routine: after solving the relaxed problem, the vector $x_R$ is used as input for a MILP procedure. After each step, half of the assets $i$ s.t. $x_i = 0$ is deleted and half is replaced in the solution. The process ends when a given number of securities has been considered. This third heuristic is the most effective, but it requires more computational time. These heuristics perform reasonably better than simpler problem specific heuristics proposed in [145]: they have the advantage of being more general and are also used in Kellerer et al.[82] in a formulation enriched by introducing fixed transaction costs and minimum lots. These heuristics are applied to solve four different models that include rounds and fixed costs, to be applied if the amount invested in a security exceeds a minimum threshold.

Even if adding constraints makes it intractable to solve real-world instances of the PSP with proof of optimality [see, e.g., 77], also exact method have been proposed for the constrained PSP: Konno and Wijayanayake[86, 87] use a piecewise linear convex underestimation strategy inside a branch and bound to model a continuous PSP with concave transaction costs, imposing ceiling constraints, no short sales, no cardinality and no floor constraints. The same approach is then modified to handle minimum lots, rounding off the solution found\(^7\). Mansini and Speranza [103] consider the semi-MAD related safety measure (downside underachievement) to be maximised,

\(^7\)Also portfolio rebalancing is considered.
using an integer formulation with rounds, ceiling constraint, fixed and proportional transaction costs and forbidding short sales. They divide the problem into two subproblems, solving the first and considering its solution as lower bound for the second subproblem. Experiments show that the first procedure alone can be effectively used as heuristic, indeed the authors show that, on the considered instances, this procedure can find an optimal solution, therefore it is very likely that it can achieve very good performances in general. Nevertheless, like [86, 87] this work is aimed at finding single points over the frontier instead of the whole frontier, so a comparison with metaheuristic techniques is not possible.

Exact methods have been also employed as components of hybrid metaheuristics for the constrained PSP. For example, Quadratic Programming (QP) has been used in [37, 119] in a problem decomposition in which a metaheuristic searches in the space of assets only (i.e., on the binary variables $z_i$) and at each step the QP solver determines the optimal allocation over them (cardinality, floor and ceiling constraints are introduced). The difference between these two works is the metaheuristic strategy they use: [37] use Iterative improvement (first and best) and Tabu Search, whilst [119] use a genetic algorithms. Promising results obtained by these works indicate that hybridization is able to improve performances in both terms of time and solution quality.

2.5 Conclusion

In this chapter we provided a survey about metaheuristic approaches for the Portfolio Selection Problem. First a high level outline of metaheuristics has been given, then metaheuristic components have been explained as they are exploited by the literature for the PSP. An outline of metaheuristic strategies for the PSP has been given, taking into account only methods that have been used for solving portfolio choices: these strategies have been classified using the standpoint provided by MAGMA, a general framework for metaheuristics. In the next chapter a hybrid metaheuristic approach will be implemented to solve Portfolio Selection Problem.
Chapter 3

A hybrid local search approach for portfolio selection

In this chapter we are defining a solver to tackle the Markowitz constrained PSP. We are tackling the minimisation of variance as a risk measure in a continuous formulation, introducing, in top of budget and return constraints, pre-assignement, cardinality (in inequality form), floor and ceiling constraints\(^1\).

\[
\min F = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} w_i w_j \tag{3.1}
\]

subject to

\[
\sum_{i=1}^{n} r_i w_i \geq r_e \tag{3.2}
\]

\[
\sum_{i=1}^{n} w_i = 1 \tag{3.3}
\]

\[
w_i \geq 0 \quad i = 1 \ldots n \tag{3.4}
\]

\[
k_{\text{min}} \leq \sum_{i=1}^{n} z_i \leq k_{\text{max}} \tag{3.5}
\]

\[
\varepsilon_i z_i \leq w_i \leq \delta_i z_i \tag{3.6}
\]

\[
x_i \leq z_i \quad i = 1 \ldots n \tag{3.7}
\]

It can be noticed that in this formulation decision variables are both integer (referring to \(Z\) vector) and continuous (referring to \(W\) vector), and for this reason a hybrid procedure composed of two approaches (one for each class of decision variables) can be devised. Furthermore, when having determined the values assumed by \(z\) variables, \(w\) values can be determined with proof of optimality using an exact methods. It is therefore straightforward to decompose the problem in two sub-problems and to use a different strategy for each subproblem.

Our solver is based in a problem decomposition that combines a local search metaheuristic, as \(\text{master}\) solver, with a Quadratic Programming procedure, as \(\text{slave}\) solver: a metaheuristic is used for selecting assets to be included in the portfolio, which at each step resorts to a Quadratic Programming (QP) solver for computing the best asset allocation, using as input assets only those

---

\(^1\)In the following, we will use symbols as already defined in previous sections.
such that \( z_i = 1 \). In other words, at each step a metaheuristic decides which assets are in the portfolio, then, the QP solver is invoked in order to compute the objective function of the current state. The QP procedure implements the Goldfarb-Idnani dual algorithm for strictly convex quadratic programs [64]. With such decomposition, we are also obtaining a more efficient way of exploring the search landscape as the local optima basin of attraction become wider (see section 3.7).

In this chapter we are first describing our metaheuristic components using the same grid introduced in the previous chapter (search space, cost function, neighborhood relations) before introducing the solver implementation. An experimental analysis will be carried out, assessing performances, comparing our results with the ones already appeared in the literature and with a family of exact solvers. Furthermore, a basin of attraction (BOA) analysis will be introduced in order to better understand algorithmic behaviour.

### 3.1 Metaheuristic components

In order to explain our metaheuristic approach, we are defining the search space, the cost function, and the neighborhood relations, following the same schema already adopted in section 2.2. In top of it, initial solution construction will be detailed in section 3.1.4.

#### 3.1.1 Search space

The search space \( S \) is composed of all configurations of \( Z \) that satisfy cardinality and pre-assignment constraints (the upper bound of the configurations is \( 2^n \)) so these constraints are handled directly by local search, while the others are dealt by the Quadratic Programming solver.

At each step, the QP solver receives as input the vector of assets determined by local search and computes the optimal assignment of the corresponding \( w_i \) variables satisfying the constraints of the formulation (of course, for all assets \( i \) that are not given as input to the QP, \( w_i \) value is set to 0).

#### 3.1.2 Cost function

Along with the optimal assignment, the QP also provides as output the computed risk \( F \) (corresponding to variance \( \sigma^2 \)) for the solution produced, which represents the cost of the state: in this case objective function and cost function coincide. Nevertheless, some special cases are taken into account: if the QP solver is unable to produce a feasible solution it returns the special value \( F = +\infty \), and the values for \( w_i \) produced are not meaningful. In this case, a greedy procedure is used to determine the configuration of assets included in \( s \) that provides the highest return, without violating other constraints: first, assets are sorted in decreasing order of return, then the maximum possible quantity\(^2\) is assigned to each asset till all the budget is spent. In this case, the cost is the degree of violation of constraint (3.2) multiplied by a suitably large constant (that ensures that return-related costs are always bigger than risk-related ones).

#### 3.1.3 Neighborhood relations

Three kinds of neighbor relations are introduced: addition, deletion and replacement of one asset. Note that these relations are far different from those with the same name introduced in section 2.2.1 because quantities are not taken into account, as handled separately in a more

\(^2\)I.e., the maximum between the asset upper bound quantity and the sum of previously assigned assets.
3.2 Solution techniques

efficient way by the QP solver. A move between portfolios is identified by a pair \( (i, j) (i \neq j) \), where \( i \) is the asset to be inserted and \( j \) the one to be deleted. To generate a proper neighbor, either \( i \) or \( j \) (or both) out of the pair must be other than 0, since having both \( i \) and \( j \) equal to 0 would mean to create a neighbor inserting no asset and deleting no asset, which means that the starting and the neighbor portfolios are exactly the same object. In detail

- \( 0 < i \leq n; \ j = 0 \) identifies addition (asset \( i \) is added to the portfolio).
- \( i = 0 \ 0 < j \leq n \) identifies deletion (asset \( j \) is removed from the portfolio).
- \( 0 < i \leq n 0 < j \leq n \) identifies replacement (asset \( i \) is added to the portfolio and asset \( j \) is removed).

Within these moves, unfeasible situations can be seen in the following cases:

- Adding an asset already present in the portfolio.
- Deleting an asset not present in the portfolio.
- Removing an asset when the cardinality is at its minimum allowed value.
- Adding an asset when the cardinality is at its maximum allowed value.
- Removing a pre-assigned asset.
- Replacing an asset with another already being hold in the portfolio.
- Replacing an asset not present in the portfolio.

Such unfeasible moves are not generated and not evaluated by the local search procedure.

3.1.4 Initial solution

Initial solutions are built so that to satisfy cardinality and pre-assignment constraints, but the procedure to build the state is different in different stages of the search process: three strategies are defined as follows:

RandomCard: we draw at random a number \( k \) (between \( k_{\min} \) and \( k_{\max} \)), and we insert into the portfolio \( k \) randomly selected assets.

MaxReturn: we build the portfolio that produces the maximum possible return, independently of the risk (using the greedy algorithm mentioned in section 3.1.2).

WarmRestart: we use the final solution of the previously computed point of the efficient frontier.

In order to improve the robustness of the solver, for all metaheuristics, we make two runs for each value of \( r_c \): one using the RandomCard initial solution construction, and the other using the WarmRestart one. For the very first point of the frontier (highest requested return and no previous point available) we use instead the MaxReturn construction.

3.2 Solution techniques

To implement our solver, we devised a combination of Local Search and Quadratic Programming, and in this section we outline briefly both. As local search techniques, we used Steepest Descent, First Descent and Tabu Search.
3.2.1 Steepest and first descent

A local search algorithm starts from an initial solution and enters into a loop that navigates over the search space, moving from the current state to one of its neighbors. In Steepest Descent all neighbors of the current solution are evaluated, and the best (w.r.t. cost function) is selected and compared to the current solution: if its cost function is better than those of the current state, the neighbor is accepted as new current solution, otherwise the search process stops, getting stuck in a local minimum. Instead, using First Descent, the first improving solution found in the neighborhood is selected rather than the best (ties are randomly broken). Also in this case, the search stops when a local minimum is reached.

3.2.2 Tabu search

Tabu search uses an additional memory to perform the search: at each step, only a subset of neighbors is explored, and the best (w.r.t. cost function) out of these neighbors is accepted as the new current solution, no matter if it is better or worse than the current solution (the best solution yet is stored). To enhance exploration capabilities, a tabu list is used, which determines forbidden states (or moves): in this list recently accepted states (or moves) are stored. This list can be static, if states (or moves) are forbidden for a pre-fixed time, or dynamic, if for each state a forbidden time belonging to an interval $[l_{\text{min}}, l_{\text{max}}]$ is randomly generated. For TS we use a dynamic-size tabu list and we search for the next state by exploring the full neighborhood at each iteration.

3.2.3 Quadratic programming

Quadratic Programming is used to determine the best wealth allocation amongst assets provided as input by the local search procedure. To this extent, we implemented the Goldfarb and Idnani dual set method [64], which works for positive definite programs. The method starts from an unconstrained solution for the QP, which is both a dual feasible point and a primal optimal point for a subset of constraints of the original problem (i.e., an empty set). The algorithm operates till primal feasibility (i.e., dual optimality) is achieved, while maintaining the primal optimality (i.e., dual feasibility) of intermediate subproblems. These subproblems are identified by an active-set containing constraints to be satisfied as equalities by the current solution estimate.

Though active-set methods are not deemed to be state-of-the-art techniques for Quadratic Programming, they are shown to achieve as good performances as more sophisticated techniques, especially when dealing with dense matrices such as the ones at hand. Furthermore, such a strategy can be used, without being coupled with a local search approach, as a stand-alone method for the Unconstrained Portfolio Selection Problem.

As already mentioned, the local search operates over the space of assets only, determining the actual state of the search process. The cost of such a state is instead determined by the QP solver. Again, the cost of that state is certified to be optimum w.r.t. assets in the portfolio. In the next move, a neighbor solution is generated, compared with the one at hand, and accepted (or not) depending on the local strategy in use at the moment.

3.3 Experimental analysis and comparisons

In this section, we first present the benchmark instances and the settings of our solvers. In the following sections, we show the comparison with previous works that use the same formulation
3.3 Experimental analysis and comparisons

Table 3.1: The benchmark instances.

<table>
<thead>
<tr>
<th>ORlib dataset</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Inst.</td>
<td>Origin</td>
<td>assets</td>
<td>UEF</td>
</tr>
<tr>
<td>1</td>
<td>Hong Kong (Hang Seng)</td>
<td>31</td>
<td>1.55936·10^{-3}</td>
</tr>
<tr>
<td>2</td>
<td>Germany (DAX 100)</td>
<td>85</td>
<td>0.412213·10^{-3}</td>
</tr>
<tr>
<td>3</td>
<td>UK (FTSE 100)</td>
<td>89</td>
<td>0.454259·10^{-3}</td>
</tr>
<tr>
<td>4</td>
<td>USA (S&amp;P 100)</td>
<td>98</td>
<td>0.502038·10^{-3}</td>
</tr>
<tr>
<td>5</td>
<td>Japan (NIKKEI)</td>
<td>225</td>
<td>0.458285·10^{-3}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Crama and Schyns dataset</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Inst.</td>
<td>Origin</td>
<td>assets</td>
<td>UEF</td>
</tr>
<tr>
<td>S1</td>
<td>USA (DataStream)</td>
<td>20</td>
<td>4.812528</td>
</tr>
<tr>
<td>S2</td>
<td>USA (DataStream)</td>
<td>30</td>
<td>8.892189</td>
</tr>
<tr>
<td>S3</td>
<td>USA (DataStream)</td>
<td>151</td>
<td>8.64933</td>
</tr>
</tbody>
</table>

or a simpler one (with less constraint types). We conclude showing a search space analysis that tries to explain the behaviour of our solvers on the proposed instances.

3.3.1 Benchmark instances

The instances we used in the experimental phase are drawn from real markets and have been already used in previous works. We used two sets of instances: the first is the group of five instances taken from the repository ORlib available at the URL http://people.brunel.ac.uk/~mastjjb/jeb/info.html. These instances have been proposed by [10] and used by [25] [137] [3] [119] [65]. The second group of three instances has been provided to us by M. Schyns and are used in [30].

For the first group, a discretised UEF composed of 100 equally distributed values for the minimum required return \( r_e \) is provided along with the data. For the second group, we use the QP solver to compute the discretised UEF over the benchmark. Table 3.1 illustrates for all instances the original market, and the average variance of the UEF (to be used as a coarse indicator of the benchmark riskiness).

Solution quality is defined using an aggregate indicator that measures the deviation of the CEF found by the algorithms w.r.t. the UEF. We call this measure average percentage loss (apl) and we define it as follows: let \( r_l \) be the minimum required return, \( V(r_l) \) and \( V_U(r_l) \) the values of the function \( F \) returned by the solver and the risk on the UEF, respectively, and \( l = 1 \ldots p \) where \( p \) is the number of points of the frontier; the \( \text{apl} \) is equal to \( \frac{100}{p} \sum_{l=1}^{p}(V(r_l) - V_U(r_l))/V_U(r_l) \).

3.3.2 Experimental setting of the solvers

Experiments were performed on a Pentium 4 (3.2 GHz) processor running Linux; the Steepest Descent, First Descent and Tabu Search (TS) metaheuristics have been coded in C++ exploiting the framework EASYLOCAL++ [38], the QP solver has also been coded in C++ and is made publicly available through the website http://www.diegm.uniud.it/digaspero/. The executables were obtained using the GNU C/C++ compiler (v. 4.0.1). All the codes have been implemented by Luca Di Gaspero and Andrea Schaerf and were kindly provided to us by the authors.

Steepest Descent and First Descent have no parameter to be set; for TS we tuned its parameters by means of a statistical technique called F-Race [14] and found that the algorithm is very robust with respect to parameter setting. We set the tabu list size in the range [3,10].
and we stopped the execution of TS when a maximum of 30 iterations without improvement is reached.

3.4 Comparison with previous results

Portfolio optimisation is nowadays amongst the most studied topics in finance, and a wide range of models have been introduced to take care of the problem: several variables definitions, objective functions, constraint sets, benchmarks and heuristic techniques have been proposed. For this reason, as already pointed out in section 2.3 a fair comparison amongst works cannot be performed, and, due to different formulations employed by the authors, the only papers we can compare with are those of Schaerf [137] and Moral-Escudero, Ruiz-Torrubiano and Suarez [119], who employ the same set of constraints on the ORlib instances, and with Crama and Schyns [30] who deal with a slightly different setting and with a novel set of instances.

Schaerf [137] uses a monolithic local search solver exploring a search space composed of both continuous and discrete variables; whereas in our approach the local search focuses on the discrete variables. Moral-Escudero, Ruiz-Torrubiano and Suarez [119], instead, use like us a hybrid solver, although they make use of genetic algorithms instead of local search for the determination of the discrete variables.

Armañanzas and Lozano [3] introduce the fixed cardinality constraint (i.e., \(k_{\text{min}} = k_{\text{max}} = k\)) on the ORlib instances. However, due to what we believe is an error in the implementation of their solution methods\(^3\) they obtain a set of points that are unfeasible w.r.t. constraint (3.3)\(^4\) and for this reason we cannot compare our solvers with them. Nevertheless, we are going to present in section 3.6.1 some results on the behaviour of one of our solvers on the formulation proposed in their paper.

3.4.1 Comparison with Schaerf and Moral-Escudero et al.

In order to compare our approach with the ones proposed by [137] and [119] we use their same formulation with the same parameter settings (\(k_{\text{min}} = 1; k_{\text{max}} = 10; \varepsilon_i = 0.01; \delta_i = 1\)). Running times of [137] have been obtained re-executing the experiments on our machine, while those of [119] were obtained from their work and taken as they come since obtained over a comparable machine. Furthermore, w.r.t. [119], we can just compare with their best results, as no further statistical indicators have been provided in that work.

Table 3.2 shows that our solver outperforms the monolithic approach by [137], in terms of both time and quality. Regarding [119], we think instead that they obtain the same results as we do, and the different representation is due to the different precision adopted. Nevertheless, we obtained those results in less time, even though a fair comparison should take into account the whole solution distribution rather than the best representative.

3.4.2 Comparison with Crama and Schyns.

Some considerations are necessary when comparing our approach with the one by Crama and Schyns [30]. First, they use Standard Deviation instead of Variance as a risk measure; then, they present their results graphically, without any analytical insight. Last, their instances are different from the most used in literature. For sake of a fair comparison we re-run their solver

\(^3\)We found the error in our analysis of the data kindly provided to us by J. Lozano.

\(^4\)In details, they assign to the assets \(i\) for which \(z_i = 1\) chosen by their ACO algorithm the quantity \(w_i = (\delta_i - \varepsilon_i)/k\), therefore since they set \(\varepsilon_i = 0.001, \delta_i = 1\) for all \(i = 1, \ldots, n\), they obtain \(\sum_{i=1}^{n} w_i = 0.999\) instead of 1.
3.5 Comparison with a family of exact solvers

As already pointed out, metaheuristics are used in many practical optimisation problems, especially when dealing with large sized instances and when proof of optimality is not required for the problem at hand. In order to evaluate the effectiveness of our techniques w.r.t. the state-of-the-art Mixed Integer Non-linear Programming (MINLP) solvers we encode in AMPL [56] the problem formulation. The AMPL model is then solved using CPLEX 11.0.1 and MOSEK 5. We experimented the MINLP solvers with two different settings: (i) cold restarts, i.e. the MINLP solver is started from an empty initial solution, and (ii) warm restarts, where, for each point over the frontier, the solution computed for the previous return point is provided as initial solution.

As it can be intuitively argued, the warm restart strategy performs better in terms of execution time, and in the following we are reporting results obtained using this setting.

We run the MINLP solver over ORLib instances, using the same constraint setting introduced in section 3.4.1 \((k_{min} = 1; k_{max} = 10; \epsilon_i = 0.01; \delta_i = 0.25)\). We operate 10 run of our SD + QP, reporting the averages of solution qualities (risk) and running time for each minimum required return, whereas we compute the best risk value with CPLEX recording the running

<table>
<thead>
<tr>
<th>Inst.</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
<th>TS(1)</th>
<th>GA+QP(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>apl min</td>
<td>time</td>
<td>apl min</td>
<td>time</td>
<td>apl min</td>
</tr>
<tr>
<td>1</td>
<td>0.00366</td>
<td>1.5s</td>
<td>0.00321</td>
<td>3.1s</td>
<td>0.00321</td>
</tr>
<tr>
<td>2</td>
<td>2.66104</td>
<td>9.6s</td>
<td>2.53139</td>
<td>14.1s</td>
<td>2.53139</td>
</tr>
<tr>
<td>3</td>
<td>2.00146</td>
<td>10.1s</td>
<td>1.92146</td>
<td>16.1s</td>
<td>1.92133</td>
</tr>
<tr>
<td>4</td>
<td>4.77157</td>
<td>11.2s</td>
<td>4.69371</td>
<td>18.8s</td>
<td>4.69816</td>
</tr>
<tr>
<td>5</td>
<td>0.24176</td>
<td>25.3s</td>
<td>0.20219</td>
<td>45.9s</td>
<td>0.20210</td>
</tr>
</tbody>
</table>

Table 3.2: Comparison of results with (1) Schaerf and (2) Moral-Escudero et al.

<table>
<thead>
<tr>
<th>Inst.</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
<th>TS(1)</th>
<th>GA+QP(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>apl</td>
<td>time</td>
<td>apl</td>
<td>time</td>
<td>apl</td>
</tr>
<tr>
<td>S1</td>
<td>0.72 (0.094)</td>
<td>0.3s</td>
<td>0.35 (0.0)</td>
<td>1.4s</td>
<td>0.35 (0.0)</td>
</tr>
<tr>
<td>S2</td>
<td>1.79 (0.22)</td>
<td>0.5s</td>
<td>1.48 (0.0)</td>
<td>3.1s</td>
<td>1.48 (0.0)</td>
</tr>
<tr>
<td>S3</td>
<td>10.50 (0.51)</td>
<td>10.2s</td>
<td>8.87 (0.003)</td>
<td>53.3s</td>
<td>8.87 (0.0003)</td>
</tr>
</tbody>
</table>

Table 3.3: Comparison of results with (3) Crama and Schyns.

on the three instances employed in their experimentation employing the same parameter setting reported in their paper: \(k_{min} = 1, k_{max} = 10, \epsilon_i = 0, \delta_i = 0.25\). Acting like this, we can make a more sound analysis of the results, running 30 times the algorithms over each instance, computing the whole CEF and analysing the average (and standard deviation in parenthesis) together with the average time spent (table 3.3).

It is clear that our hybrid approach outperform the SA by [30] in terms of solution quality. Nevertheless, this comes at cost of higher running times. This is due to two main factors: first, the strategy employed by our algorithms utterly explore the neighborhood at each candidate solution, while SA randomly select only some neighbors, thus saving time in the cost function evaluation; Then, the slave QP procedure is more time-consuming than the simple evaluation function used by Crama, contributing to the higher running time as stressed by table 3.3.

3.5 Comparison with a family of exact solvers

Table 3.2: Comparison of results with (1) Schaerf and (2) Moral-Escudero et al.

<table>
<thead>
<tr>
<th>Inst.</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
<th>TS(1)</th>
<th>GA+QP(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>apl min</td>
<td>time</td>
<td>apl min</td>
<td>time</td>
<td>apl min</td>
</tr>
<tr>
<td>1</td>
<td>0.00366</td>
<td>1.5s</td>
<td>0.00321</td>
<td>3.1s</td>
<td>0.00321</td>
</tr>
<tr>
<td>2</td>
<td>2.66104</td>
<td>9.6s</td>
<td>2.53139</td>
<td>14.1s</td>
<td>2.53139</td>
</tr>
<tr>
<td>3</td>
<td>2.00146</td>
<td>10.1s</td>
<td>1.92146</td>
<td>16.1s</td>
<td>1.92133</td>
</tr>
<tr>
<td>4</td>
<td>4.77157</td>
<td>11.2s</td>
<td>4.69371</td>
<td>18.8s</td>
<td>4.69816</td>
</tr>
<tr>
<td>5</td>
<td>0.24176</td>
<td>25.3s</td>
<td>0.20219</td>
<td>45.9s</td>
<td>0.20210</td>
</tr>
</tbody>
</table>
Table 3.4: Running times for the computation of the whole CEF on ORLib instances 1–4

<table>
<thead>
<tr>
<th>Instance</th>
<th>avg(SD + QP)</th>
<th>CPLEX 11</th>
<th>MOSEK 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.1s</td>
<td>2.1s</td>
<td>15.8s</td>
</tr>
<tr>
<td>2</td>
<td>14.7s</td>
<td>397.1s</td>
<td>5.0s</td>
</tr>
<tr>
<td>3</td>
<td>18.0s</td>
<td>890.7s</td>
<td>1,903.3s</td>
</tr>
<tr>
<td>4</td>
<td>20.9s</td>
<td>169,461.0s</td>
<td>239,178.4s</td>
</tr>
</tbody>
</table>

It can be seen that results obtained running the two strategies are the same, their difference being due to the diverse rounding procedures. This shows solutions found with our approach to be optimal. These results gives us evidence about the fact that metaheuristics are used when proof of optimality is not required, still they can reach the global optimum. The same does not hold for running times: execution time required by CPLEX happens to be, by all accounts, greater than time requested by our strategy, as it can be shown by figure 3.1, where we report the running times of CPLEX, MOSEK and the SD+QP algorithm against the return value for the first 4 instances of the ORLib benchmark. The scale of the graphs is semi-logarithmic (on the y-axis) in order to enhance also small differences of performances.

Apart from the first instance, being small enough to be treated efficiently by exact solvers, running times of the three solvers are comparable only for the points for which cardinality constraints are not binding (i.e., in the right part of the frontier), whereas, in the left part of the UEF (where the minimum required return value is low and cardinality constraints are binding) the situation changes dramatically and both CPLEX and MOSEK running times scale very poorly with instance size. Our hybrid algorithms perform instead stably, and scale well with respect to both instance size and constraints strength.

The total running times for the computation of the whole CEF are shown in table 3.4; for SD + QP we report the average of the total running times for the 10 runs. The table confirms that SD + QP outperforms CPLEX and MOSEK in the computation of the whole frontier on all instances except of Instance 1, which is the smallest one.

We can conclude that exact solvers such CPLEX and MOSEK are to be preferred when the instance at hand is small-sized. Only in these cases they operate in a reasonable amount of time, other than certifying the optimality of the solution. Conversely, their use for bigger-sized instances appears to be impractical due to huge running time.

A better and ad hoc tuning of the MINLP solvers might help improve their running time, however this is out of the scope of this comparison.

### 3.6 Additional analyses

In this section we propose further experimental analyses with the aim to highlight the behaviour of the hybrid solver for various constraint settings and some features of the search landscape.

#### 3.6.1 Results for fixed cardinality portfolios

Fixed cardinality is used to obtain a given level of diversification, and to this goal it has been investigated by [106]. It is however a common practice amongst fund managers to decide in advance the number of assets to be included in the portfolio. Fixed cardinality has also been investigated by [3], but we cannot compare directly with their results, because of what we
Figure 3.1: Comparison of the SD + QP solver against a MINLP-encoding of the problem solved by CPLEX and MOSEK solvers.
Figure 3.2: \( \text{apl} \) found by our SD+QP solver varying \( k \), for instances 2 (left) and 5 (right).

think of as an implementation error in their approach. Nevertheless, we can use their very same parameter setting to investigate this phenomenon. We show some results of the Steepest Descent solver on the ORlib instances by setting \( k_{\text{min}} = k_{\text{max}} = K \), while quantity constraints (floor and ceiling) are set as in [137], i.e., \( \epsilon_i = 0.01 \), and \( \delta_i = 1 \) (no pre-assignment).

In Figure 3.2 we plot the behaviour of the \( \text{apl} \) found by our SD+QP solver at different values of \( k \) on two instances (2 and 5).

The curve is compared with the \( \text{apl} \) computed by the same solver but relaxing the minimum cardinality constraint to \( k_{\text{min}} = 1 \) (i.e., just allowing to include an increasing number of assets in the portfolios, but not obliging the solver to compel to a fixed cardinality).

From the figures we can notice an interesting phenomenon: the two curves are almost indistinguishable up to a value of \( k \) for which the fixed cardinality solutions tend to have a higher \( \text{apl} \). In a sense, this sort of minimum represents the best compromise in the cardinality, i.e., the optimal fixed number of assets \( k \) that minimises the deviation from the best achievable overall returns (i.e., the UEF values).

This result have a clear financial meaning: indeed, as discussed by [106], the marginal effect of diversification is decreasing up to a certain point, but is increasing afterwards because of the correlation between assets. This result shows that a good diversification can be obtained by including a proper subset of the assets universe, while the necessity of a sound cardinality constraint finds its justification, as imposing a too high strict cardinality constraint could vanish the diversification effect.

3.6.2 Results with pre-assignment constraints

In order to evaluate the impact of pre-assignment constraints on solution quality, for each benchmark instance we proceed as follows: we fix in turn one of the assets as pre-assigned and we compute the resulting CEF (imposing the other constraints as in the previous experiments, i.e., \( \epsilon_i = 0.01 \) and \( \delta_i = 1 \) for \( i = 1 \ldots n \), no cardinality constraints are set). This way, for each instance consisting of \( n \) assets, we obtain a family of \( n \) CEFs, one for each pre-assigned asset.

Intuitively, imposing pre-assignment constraints generally worsen the solution quality measured with \( \text{apl} \), unless the pre-assigned assets belong to an optimal solution for all the values of the required return \( r_c \). Moreover, the magnitude of this worsening effect might depend on the asset’s features. Specifically, when a low-return asset is pre-assigned the performances should
3.7 Search space analysis

Search space analysis is aimed to provide an explanation of the algorithm performance and to explain the hardness of the instance at hand when tackled by a metaheuristic. In our case, we want to investigate the search space obtained when imposing five equally distributed minimum...
3. A hybrid local search approach for portfolio selection

(a) Return vs \( \mathbf{apl} \): ORlib instances

(b) Return vs \( \mathbf{apl} \): S instances

(c) Risk vs \( \mathbf{apl} \): ORlib instances

(d) Risk vs \( \mathbf{apl} \): S instances

Figure 3.4: Return/risk vs \( \mathbf{apl} \) for one pre-assigned asset

Table 3.5: Pearson correlation between pre-assigned asset return/risk and the resulting \( \mathbf{apl} \)

<table>
<thead>
<tr>
<th>Inst.</th>
<th>( \rho(\mathbf{apl}, r_i) )</th>
<th>( \rho(\mathbf{apl}, \sigma_{ii}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.7907505 0.06618641</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-0.9107568 0.4910591</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-0.8249417 0.09394818</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-0.8138779 0.2904020</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-0.9280059 0.8029789</td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>-0.896169 0.825982</td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>-0.8816018 0.6919496</td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>-0.73096 0.1938812</td>
<td></td>
</tr>
</tbody>
</table>
required return values (called $R_1$ to $R_5$) over the five instances at hand (each required level of return defines an instance). This way, we aim to estimate the features of the search space tackled by our solver along the whole frontier. Furthermore, different parameter settings have been considered, w.r.t. cardinality constraints, since we investigated both maximum and fixed cardinality constraints, i.e., $(k_{\text{min}}, k_{\text{max}}) \in \{(1, 3), (1, 6), (1, 10), (3, 3), (6, 6), (10, 10)\}$.

A goal of the search space analysis is to provide an estimate of the local minima distribution. This is thought of as an estimator of the ruggedness of the search space, and is roughly negatively correlated with local search performance [71].

In order to estimate the number of minima in an instance, we run a deterministic version of Steepest Descent\(^5\) (called $SD_{\text{det}}$) starting from initial states either produced by complete enumeration (for very small instances) or by uniformly sampling the search space.\(^6\)

### Analysis for Real Instances

Our analysis shows that the instances of the benchmarks have a small number of local minima and only one global minimum, that is either a certified global minimum, when exhaustive enumeration is performed, or the best known solution, otherwise.

This analysis may provide an explanation for the very similar performance exhibited by SD and TS in terms of solution quality. To strengthen this argument, we also studied global and local minima basins of attraction, in order to estimate the probability of reaching a global minimum [128]. Given a deterministic algorithm such as $SD_{\text{det}}$, the basin of attraction $B(\overline{s})$ of a minimum $\overline{s}$, is defined as the set of states that, taken as initial states, gives origin to trajectories that ends at point $\overline{s}$. The cardinality of $B(\overline{s})$ represents its size (in this context, we always deal with finite spaces). The quantity $rBOA(\overline{s})$, defined as the ratio between the size of $B(\overline{s})$ and the search space size, is an estimation of the reachability of state $\overline{s}$. If the initial solution is chosen at random, the probability of finding a global optimum $s^*$ is exactly equal to $rBOA(s^*)$. Therefore, the higher is this ratio, the higher is the probability of success of the algorithm.

In Table 3.6 we report the outcome of the BOA analysis performed on instance 4, showing the number of local minima that are different from the global one (dashed entries indicates settings for which no feasible solution exists).

The stochastic nature of SD and TS, together with the escape mechanism provided by TS, lead us to treat the estimation of basins of attraction size related to $SD_{\text{det}}$ as a lower bound on the probability of reaching the global optimum when using SD and TS.

Our analysis shows that global minima have usually a quite large basin of attraction. Representative examples of these results are depicted in Figures 3.5(a–d); segments represent the basins of attraction: their length corresponds to $rBOA$ and their y-value is the objective value

### Table 3.6: Number of local (non-global) minima for instance 4.

<table>
<thead>
<tr>
<th>$k_{\text{min}}, k_{\text{max}}$</th>
<th>$R_1 = 0.00912$</th>
<th>$R_2 = 0.00738$</th>
<th>$R_3 = 0.00556$</th>
<th>$R_4 = 0.00375$</th>
<th>$R_5 = 0.00193$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1,6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>1,10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>3,3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>6,6</td>
<td>–</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10,10</td>
<td>–</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

\(^5\)Ties are broken by enforcing a lexicographic order of states.

\(^6\)Since the search space size is $\sum_{k=k_{\text{min}}}^{k_{\text{max}}} \binom{n}{k}$, uniform sampling is achieved by opportunely weighting, by binomial coefficients, the probability of choosing among the sets of solutions with exactly $k$ assets.
3. A hybrid local search approach for portfolio selection

Figure 3.5: Basins of attraction of minima on two benchmark instances with different cardinality and return constraints.

(a) Instance 4: $k_{\text{min}} = k_{\text{max}} = 3$, $R = 0.00375$
(b) Instance 4: $k_{\text{min}} = 1$, $k_{\text{max}} = 6$, $R = 0.00193$
(c) Instance 4: $k_{\text{min}} = 1$, $k_{\text{max}} = 10$, $R = 0.00375$
(d) Instance S3: $k_{\text{min}} = 1$, $k_{\text{max}} = 6$, $R = 0.260$

Figure 3.5: Basins of attraction of minima on two benchmark instances with different cardinality and return constraints.

Monolithic Search Basin of Attraction We have seen in the previous paragraph how the BOAs of local optima happen to be huge w.r.t. the whole search space. This sort of smoothness of the search landscape makes the problem easy to be tackled, and for this reason we also observe that a simple strategy such as SD exhibits as good performances as a more elaborated one (TS) does. Anyhow, we urge again the reader to understand that the BOA is not only problem-specific, but it also depends on the search strategy used. We can easily show this behaviour analysing the very same problem, over the very same instances, yet using a monolithic search strategy. To this extent, we use a variant of Threshold Accepting (see section 2.2.2) in which the threshold is kept fixed to 0 over all the search process: in this way, our Threshold Accepting behaves like an Iterative improvement algorithm. In this variant, the search space is not split in $z$ and $w$ variables, but only $w$ variables are considered: this means that the outcome of the two problems is the same (in the sense that we are interested in the same decision variables),
but the way of obtaining it differs from one strategy to the other. In the following we highlight the main features of this local search approach:

- **Search Space** The search space is not split in \( w \) and \( z \) variables as only one strategy is used, instead of a master-slave decomposition. For representing a state, we make use of a sequence \( W = w_1 \ldots w_n \) such that \( w_i \) is the fraction of asset \( b \) held in the portfolio. The number of non-zero elements in the sequence is in the range \([k_{\text{min}}, k_{\text{max}}]\), their sum equals 1, and their value is in the range \([\epsilon_i, \delta_i]\).

- **Neighborhood relations** A fraction (\( \text{step} \)) of the share is transferred from one asset \( a \) to another \( b \). If asset \( b \) is not in the portfolio, is then inserted. If one asset falls below \( \epsilon_i \) minimum is set to \( \epsilon_i \). If there is an attempt to decrease the share of an asset being set to \( \epsilon_i \) the asset is deleted and \( \text{step} \) is fixed to \( \epsilon_i \). The asset whose share is to be increased must be chosen so that \( x_i + \text{step} \leq \delta_i \).

- **Initial solution** The starting solution is created randomly. Cardinality, budget, floor and ceiling constraints are to be satisfied, while return constraint has not such limitation.

- **Cost Function** We use a penalty approach, as in the master-slave approach: the cost function is given by the sum of the portfolio variance (risk) and the degree of violation of return constraint.

- **Local Search Strategies** A greedy variant of Threshold Accepting.

In figure 3.6, results about BOAs analysis of this approach are outlined (rBOA versus risk are plotted, as in the previous figures).

It is clear how, using this strategy, there are several local minima, each with a narrow basin of attraction, and out of these local optima, none is the global one. The number of local optima is order of magnitude higher w.r.t. the number formerly found through our Hybrid Solver, and the solution quality is worse, even comparing the best result with the worst obtained through the Hybrid Method. This situation clearly indicates a situation in which the use of a more complex strategy should be desirable, and indeed, how clearly demonstrated by the literature on this topic, Iterative Improvement (since our greedy variant of Threshold Accepting can be thought of as an Iterative improvement) always led to unsatisfactory results when applied in this way (see section 2.2.2).

**Analysis for artificial instances** Our BOA analysis shows that instances at hand are relatively easy to solve, as local optima are few and far between, and their BOAs are huge: this is the reason for Best and Steepest Descent to work at least as good as more sophisticated strategies such TS. Using those instances we cannot even prove that our TS has good exploration capabilities. Indeed, we have enough elements to think that each real-world instance shows features comparable with the ones at hand, so the only way we have to test our TS is to use an artificially generated instance with a large number of local minima whose BOAs are narrow. So we create an artificial instance as follows: the number of assets \( n \) is generic (but even), \( r_i = 1 \) for \( i = 1 \ldots n \), the matrix \( \sigma \) is as follows: \( \sigma_{i,i} = 1 \) (for \( i = 1 \ldots n \)), \( \sigma_{1,2} = -1 \), \( \sigma_{i,i+1} = -0.9 \) (for \( i \) odd and \( > 1 \)), and \( \sigma_{i,j} = 0 \) for all other values. The covariance matrix is shown here below:
3. A hybrid local search approach for portfolio selection

![Graph](image)

(a) Instance 4: $k_{\text{min}} = 1$, $k_{\text{max}} = 10$, $R = 0.00375$

(b) Instance 4: $k_{\text{min}} = 1$, $k_{\text{max}} = 6$, $R = 0.00193$

Figure 3.6: Monolithic Basins of attraction of minima on two benchmark instances with different cardinality and return constraints.

\[ \sigma_{ij} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & \ldots & 0 \\ -1 & 1 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & -0.9 & 0 & 0 & \ldots \\ 0 & 0 & -0.9 & 1 & 0 & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & -0.9 & 0 & \ldots & 0 \\ 0 & 0 & -0.9 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 & 0 \\ 0 & \ldots & 0 & 1 & -0.9 & 0 \\ 0 & \ldots & 0 & -0.9 & 1 & \end{pmatrix} \quad (3.8) \]

In the unconstrained case, without loss of generalisation, the best solution is given, for each $r_e$, by the portfolio consisting of the first two assets, with others asset quantities set to 0, variance of such a portfolio being 0. Nevertheless, each portfolio consisting of assets $i$ and $i + 1$ ($i$ being
odd) is a local optimum, as any feasible neighbor chosen leads to a state with higher risk.

It is quite easy to show that in order to reach the global minimum $s^*$ from a state $s$ it is necessary to have $z_1 = 1$ or $z_2 = 1$ in $s$. In addition, it is necessary that there is no $i$, with $i > 1$ and odd, such that both $z_i = 1$ and $z_{i+1} = 1$ in $s$. In fact, such a pair would not be removed by any move. It follows that $B(s^*)$ is composed of the states in which $z_1 = 1$ or $z_2 = 1$ and there is no odd and $i > 1$ such that $z_i = 1$ and $z_{i+1} = 1$. It can be shown that $rBOA(s^*)$ is inversely proportional to $n$.

On the contrary, we have verified experimentally that for this instance both TS and CPLEX find easily the global optimum, with floor and ceiling constraints set as in previous cases ($\epsilon_i = 0.01$ and $\delta_i = 1$ for $i = 1 \ldots n$), no cardinality constraints nor pre-assignments. It has to be noticed that such an instance (or at least real markets that exhibit a similar structure) is unlikely to be found: though in principle such a matrix can be easily conceived, asset variances tend to be positively correlated, their prices being influenced by (at least) one factor (namely the market). Of course it can be possible to find negatively correlated assets, and this likelihood increases when considering other financial instruments to be part of the portfolio (i.e., options), but having assets just correlated in pairs seems to be more an exercise than a real world application.

It is worth mentioning that for the artificial instance presented here, the problem can be solved by using a larger neighborhood (i.e., a move that relocates two assets at once). Nevertheless, for any neighborhood made up of composite moves of length $k$, it is possible to create instances whose minima are composed by “clusters” of assets of size $k + 1$ that have to be moved jointly.

### 3.8 Conclusion

In this chapter we devised a hybrid solver for the Portfolio Selection Problem: this solver is based on a master - slave decomposition and resorts to local search and Quadratic Programming. Our results show that this solver compares favourably with the state-of-the-art local search approaches, attaining optimal performances in low running times. The same holds when comparing our solver with exact methods, showing once more that metaheuristics can reach the global optima, even though the proof of optimality is not required. A basin of attraction analysis has been performed, showing that, in this hybrid way, the search space appears to be smooth, differently from the monolithic local search approach.
Chapter 4

Metaheuristics for the index tracking problem

As seen in section 1.2 the literature on portfolio optimisation is predominantly concerned with active management, and this paradigm has driven our work so far. Nevertheless, the paradigm of passive management is becoming more and more compelling to both investors and academics: the rationale beneath this approach is that a portfolio should achieve roughly the same return as a specified benchmark (say, market index, such as FTSE, MIBTEL, NIKKEI) through an investment in a subset of the benchmark. Contrarily to active management, where investors want to choose high performance stocks w.r.t. their expectations through esteem of returns, in passive management no esteem has to be made, incurring in lower costs, but bearing the disadvantage that if the market (index) falls, the portfolio return will be falling as well, as it is linked to benchmark’s performances. In the following we are introducing the Index Tracking Problem, the most common problem faced by such passive managed fund managers. Some details were already discussed in section 1.1.2. In this section we will enter much into the problem, whilst in section 5 an approach combining active and passive strategies will be defined.

We will start from defining the main concept on which the model is based upon in section 4.1, then we will describe the problem through the same conceptual model used in chapter 1, defining the three attributes (variables, objectives and constraints) in sections 4.2.1, 4.2.2, and 4.2.4. We will be adding a further attribute, the temporal horizon (section 4.2.3) as different formulations of the problem can lead us to consider diverse moments in order to tackle decisions. Section 4.3 will introduce metaheuristics approaches for dealing with the Index Tracking (IT) Problem using the same framework introduced in chapter 2 for the PSP.

4.1 The basic model

Financial indices are often used to capture information about a certain region, market (or market segment) or industry. The actual computation of the index depends on its purpose. In some cases (in particular for stocks) the index, to be regarded as basket of assets, is based on their prices, and the weight for the different assets is based on the market capitalization of the respective companies. Unlike a price index, a performance index is computed as the weighted average of the returns. The latter reflects the market performance better as it includes the return additional rights and payments such as dividends yield; at the same time, however, it is more difficult to track in practice than the former which can be regarded as a basket with (or without) a fixed number of assets included[1].

Furthermore, the index structure can be re-adjusted, e.g., when one of the the stocks has
been suspended from the Stock Exchange regulation board or because its price has fallen and, in market capitalization terms, another stock merits inclusion in the index, or even because a company has grown enough to be included in the index or eventually because of mergers among companies, so a fully replication could make us bearing undesirable transaction costs, apart from being cumbersome to manage and monitor. In the Index-Tracking-Problem in canonical form we want to find a portfolio that reproduces the performance of a stock market index, yet without full replication of the index and having no knowledge on the index composition. Indeed, such kind of knowledge could lead us, in case we are neglecting transaction costs, to fully replicate the index.

Let $U$ be the set of assets the index is composed of (universe set) containing $n$ assets. Let $r_{i,t}$ be the asset return of asset $i \in U$ at time $t$, $r_{I,t}$ the return of index at time $t$, $x_{i,t}$ the integer quantity of asset $i$ held at time $t$. The portfolio at hand will be composed of the set\(^1\)

$$ P = \{i \mid x_i > 0 \}. $$

Let $S_{i,t}$ denote asset $i$’s price at time $t$, then the portfolio value at time $t$ is $v_t = \sum_{i \in P} x_{i,t} S_{i,t}$. In the Index Tracking problem we want our portfolio $P$ to mimic as close as possible the behaviour of index $I$ over a given time horizon, investing all the budget $B$ at hand. Given $r_p = \sum_{i=1}^{n} r_{i}x_{i}$ the return of portfolio $p$, this means that $r_p$ should be as close as possible to $r_I$ over our time horizon. The as close as possible clause can be translated into optimising different objective functions, as reviewed in section 4.2.2.

It is worthwhile to mention that the objects we observe are asset prices over a given time horizon, and these prices must be converted into assets return, following the procedure(s) we outlined in the introductory part to chapter 1.

Even if the main idea is to track an index with a subset of the index itself [109, 108], the formulation can easily be extended in order to include either stocks not belonging to the index using a superset instead of a subset (see [11, 60]) or even other financial instruments (say, bonds or derivatives; see [36, 35]).

### 4.2 A classification of index tracking models

As already pointed out in section 1.2, constrained optimisation problems can be defined by specifying variables, domains, objectives, and constraints among variables. We add to these entities the temporal horizon, being a feature of the Index Tracking Problem, and we remind that these entities play the role of model attributes and serve as the basis for a classification of the different models. In the following, we will classify the Index Tracking models by filling this conceptual model scheme. As in section 1.2, our description has not the goal of providing an overview of all the formulations for the IT, but rather of illustrating the diverse models of the problem as described in the specific literature on metaheuristics.

#### 4.2.1 Classification of variables and domains

Akin to the PSP discussed in section 1.2.1, variables of the Index Tracking Problem represent the actual amount of stocks held in the portfolio, and can be represented by both continuous and discrete values. It is useful to notice that, concerning the way of handling variables, the Index Tracking literature differs from the PSP one: where continuous variables are mostly used\(^1\)Theoretically, this could be formulated as $P = i | x_i \neq 0$. This would mean allowing short sales (see section 1.2.3). Although possible, this implementation has not been yet considered in the literature.
for the PSP (see section 1.2.1), discrete values are instead mostly used to tackle the Index Tracking Problem\cite{11, 60, 108, 109}. The motivations behind this choice can be manyfold:

1. Discrete values convey a more realistic representation of real-world trading, while continuous models represent a truly simplification of markets.

2. In several countries, stocks can be purchased as multiples of a minimum lot of units (rounds, see section 1.2.3). This behaviour cannot be handled by a continuous model because using it the actual amount invested in a stock will be represented by a fraction, with no means of checking if the equivalent amount of money is a multiple of the product between asset price and minimum lot.

3. When considering transaction costs, only proportional ones can be included on a continuous model, whilst fixed ones can only be embedded in a discrete formulation.

The formulation of the problem, in which prices change over time, makes necessary a compromise between these two formulations: by solving the problem in a discrete formulation at time $t$, we obtain the number of stocks that would best have tracked the index from the beginning of the considered period $t-1$, but as price have changed between $t-1$ and $t$, investing in the found number of stocks would result in a violation of the budget constraint. In this case, in order to determine the stock quantities, first the continuous-related solution is determined, so that

$$ w_{i,t-1} = \frac{x_i \cdot S_{i,t-1}}{\sum_{j \in U} x_j \cdot S_{j,t-1}} \quad i \in U $$

then these weights are used to create the portfolio at time $t$, determining $x_{i,t}$ so that $w_{i,t-1} = w_{i,t} \; \forall i \in U$, generally resulting in different number of assets due to price changes between $t-1$ and $t$ (see [60]).

$$ x_{i,t} = \frac{B \cdot w_{i,t-1}}{S_{i,t}} \quad i \in U $$

### 4.2.2 Classification of tracking error measures

The main aspects of the problem are the return of the portfolio at hand and its deviation from the actual index: it is straightforward to classify the objective functions of the problem depending on which of these two aggregates they care of.

#### Error based measures

Tracking error represents obviously the main measure of the problem, so different approaches have been proposed in the literature to deal with it. Most work related to Index Tracking literature define the tracking error as the variance of the difference between tracking portfolio return and index return \cite{31, 57, 98, 132, 158}. This approach has been also exploited, tackled by metaheuristics\cite{35, 36}:

$$ TEV = Var_T(r_P - r_I) $$

Although effective in capturing the main feature of the problem, the variance reveals itself lacking of practical implications, as a portfolio that constantly underperforms the index by a constant value has zero variance and so represents the optimal solution to the problem. In order to overcome this drawback, measures that penalise deviations (rather than measures penalising
different magnitudes of deviations) from the index have been introduced. These measures can be summarised by:

\[ TE = \frac{\left( \sum_T |r_P - r_I|^{\alpha} \right)^{1/\alpha}}{T} \] (4.5)

or its variant [109]

\[ TE = \left( \frac{\sum_T |r_P - r_I|^{\alpha}}{T} \right)^{1/\alpha} \] (4.6)

The parameter \( \alpha \) represents the magnitude of the penalization for differences between \( r_P \) and \( r_I \): Gilli and Këllezi [60] uses \( \alpha = 1 \) being the Mean Absolute Deviation; when \( \alpha = 2 \) the Tracking Error is defined as the Root Mean Squared Error or a variant of it [11]. The choice of the \( \alpha \) parameter can lead to differences in the optimisation results, as portfolios optimal w.r.t. an alpha value are not necessarily optimal for another.

### Performance based measures

In an Index tracking framework, we want to replicate an index as close as possible. Nevertheless, positive deviations from the index can be considered desirable. This phenomenon has been widely explored by [108], but we can basically formalise it introducing *Excess return*, defined as the portfolio return above the index return

\[ r_{\text{excess}} = \sum_{t=1}^{T} \frac{r_{p,t} - r_{i,t}}{T} \] (4.7)

and attempting to maximise this measure. Indeed, even though non-metaheuristic Index Tracking literature generally maximises return \(^3\) this measure has just been mentioned by [11, 60], but not used in their formulation.

The error based measures have a drawback in that they symmetrically penalise downside and upside deviations from the index: they do not capture actual preferences as investors might want to avoid negative deviations, but they should feel not harmed when the portfolio return is bigger than the index one (as of course they perceive downside deviations as more hurtful than upside ones). Two solutions have been proposed to overcome this issue: either considering only temporal points \( t \) over which the portfolio underperforms the index, or penalising negative deviations in the objective function.

The first approach has been proposed by Rudolf et al. [136]: after defining two risk measures, he introduces a variant considering, in order to compute the aggregate, only the temporal points over which the portfolio underperforms the index, so restricting the minimisation to the negative deviations between portfolio and index. This is called *downside risk* [69], but although widely used, it has hardly been employed in metaheuristic literature (it has been mentioned by [11, 60] but not used in the experimental phase). Namely, the two measures introduced in [136] are the Mean Absolute Deviation (see above) and a Min-Max criterion, where this latter function means that the maximum deviation between portfolio and index over the observation period is minimised, and represents a *worst case* approach (the same principle is introduced by [167] for portfolio selection problems). The approach used is a linear programming formulation to tackle the optimisation.

The second approach of penalising negative deviations in the objective function has been exploited by [108], where, advocating that TE does not maximise the investor’s utility, it is said

---

\(^3\)Or risk-related measures: [163] maximises expected returns, while [169] aims at maximising expected utility of terminal wealth.
4.2 A classification of index tracking models

that decision makers exhibit loss aversion by reacting more sensitively to losses than suggested by the traditional assumptions on risk aversion. For example, there is evidence that individuals are reluctant to realise losses and they stick to assets that generate losses [79]. In this context, a parameter $\gamma$ has to be introduced in order to penalise downside deviations

$$TE = \left( \frac{\sum_t (\Delta r_t)^2}{T} \right)^{\frac{1}{2}}$$  \hspace{1cm} (4.8)

$$\Delta r_t = \begin{cases} r_p - r_I & \text{if } r_p \geq r_I \\ (r_p - r_I) \cdot \gamma & \text{if } r_p < r_I \end{cases}$$  \hspace{1cm} (4.9)

$\gamma$ represents the loss aversion, as perceived by investors, so the greater $\gamma$, the higher their aversion to losses. When $\gamma = 1$ loss aversion has no additional effects on the optimisation, while the more $\gamma$ exceeds 1, the more losses are perceived to be harmful and to be avoided by decision makers. $\gamma < 1$ indicates a loss seeking individual, so it can be excluded from the analysis.

In order to combine the two main features of the problem, a weighted sum of risk and return is possible:

$$\min \lambda TE - (1 - \lambda)r_{excess} \quad 0 \leq \lambda \leq 1$$  \hspace{1cm} (4.10)

In principle, each measure can be used, but they must be defined so that risk and return are expressed in comparable units. This approach has been used by [11, 60], but they pose $\lambda = 1$, so that the problem turns into a minimisation of the risk measure.

Whatever the measure used, it should be noticed that, unless very unusual conditions, it should be impossible to obtain a null tracking error, even investing in the whole universe of assets, as indices compositions are frequently rebalanced over time.

4.2.3 Classification of temporal horizon

Index Tracking consists of mimicking the behaviour of a given index by optimising one of the function previously introduced (for a more comprehensive list of objective functions, see [20]). That said, the temporal dynamic of the tracking portfolio computation can be inspired to different principles.

We can define the Tracking Procedure as the creation of a tracking portfolio $p_t$ at a given time $t$ given we were endowed with a starting portfolio $p_0$ at time 0. This operation is performed over time.

The basic procedure would be creating a portfolio from scratch (Single Period Index Tracking, cash endowed\(^4\)), so imposing $p_0 = \emptyset$ (or, if we consider cash as a portfolio constituent, imposing $x_{\text{cash},0} = B$, where $B$ is the available budget). This is probably the most common approach, and it has been implemented by [11, 108, 109].

An extension of this basic approach consists in endowing the investor with a starting portfolio to rebalance, so that $p_0 = \{i : w_i \neq 0\}$ [35, 36]) (Single Period Index Tracking, portfolio endowed\(^5\)).

The natural extension is to consider a multi-period problem, as after portfolio selection, the issue of portfolio control arises, since portfolios need to be re-optimised due to new investments opportunities and infeasibility triggered by changes in the market. In order to do it, two approaches can be identified: the first consists in pre-determining temporal points in which portfolio is to be rebalanced (Static Multi-Period Index-Tracking Problem) [11, 60].

The second consists in defining a strategy in order to determine which time to rebalance the portfolio over (Dynamic Multi-Period Index Tracking Problem). For instance, in [35, 36] the

---


\(^5\)Referred by [11] as revision, or rebalancing problem.
portfolio is rebalanced each time it becomes unfeasible (due to price movements) w.r.t. floor and ceiling constraints.

Combining objective function and temporal horizon we can have several typologies of problems. The design of specific-oriented strategies would require intense efforts in implementation and parameter-tuning. Advantages of metaheuristics are, instead, to be quite robust w.r.t. the formulation and to be easily extended to other typologies of problems. Their use in this domain appears, so far, fully justified.

4.2.4 Classification of constraints

Constraints can be first distinguished into two classes: *Hard* and *Soft*. The first class includes constraints imposed by law regulations that must be fulfilled by any portfolio at any time, whose violation must be reported to supervising authorities; the second refers to constraints reflecting internal regulations and policies, whose fulfillment is not compulsory as they can be violated at some points. In the following we describe the major constraints as they are introduced in the literature. Note that constraints for the IT are mostly the same as previously introduced for the PSP (see sec 1.2.3) and in principle each sort of those constraints could be exploited in a IT framework.

**Budget constraint**

Also within the IT framework, this is a theoretical constraint, used in order to norm the solution, imposing that all the available budget must be invested, and the very same considerations made for the PSP still hold for the IT. Note that cash can be included in the portfolio, so the sum will encompass the cash amount held, either in the continuous formulation [35, 36] or in the discrete one [59].

**Floor and ceiling constraints**

As for the PSP, with these constraints we impose a minimum and maximum proportion (\( \varepsilon_i \) and \( \delta_i \) respectively) allowed to be held for each asset in portfolio, so that \( w_i = 0 \lor \varepsilon_i \leq w_i \leq \delta_i \) (\( i = 1 \ldots n \)). This holds in the continuous formulation with asset weights between 0 and 1. If the discrete formulation is used and prices change over time, a further explanation is needed: as stock \( i \) has an initial price \( S_{i,0} \), the fraction \( w \) invested in the stock is \( w_0 = \frac{x_i S_{i,0}}{B_0} \), where \( B \) is the initial budget and \( x_i \) the number of asset \( i \) to be bought. Anyway, as prices change over time, the \( w_i \) value will change accordingly, and it could become unfeasible w.r.t. these constraints. In this case, either we make the assumption that these constraints must hold only at period 0 so that the portfolio doesn’t need to be rebalanced when becoming unfeasible [108], or the portfolio need to be rebalanced each time it becomes unfeasible w.r.t. floor and ceiling constraints [36].

Unlike for the PSP, the opportunity of imposing different upper and lower bounds for different assets has been explored by [35, 36] where the general constraint is \( w_i \leq 0.1 \), but the share of DAX30-futures should not exceed 0.2 and cash must range between 0.03 and 0.1\(^6\).

**Cardinality constraints**

Akin to the PSP, the number of assets in the portfolio can be either set to a given number or bounded by a maximum (and-or minimum) value:

\(^6\)In the following we will explain constraints as to be imposed in an integer formulation, if not differently stated.
4.2 A classification of index tracking models

\[ \sum_{i=1}^{n} z_i \leq k \]  \hspace{1cm} (4.11)

As for the PSP, the inequality form is quite common \([36, 60]\), yet the equality form, i.e., \(\sum_{i=1}^{n} z_i = k\), can be used \([11]\). The latter formulation has been introduced by \([109]\), resolving the problem for several values of \(k\) in order to investigate the effect of the marginal cardinality increase, and as outcome of this setting it turned out that the higher the cardinality, the smaller the tracking error, but the marginal contribution of cardinality (in shrinking the error) decreases slowly (and, say, much slowly than Mean-Variance portfolios). Furthermore, tracking portfolios don’t consist of the whole universe of assets, even without cardinality restriction, due to short selling prohibitions and lower bounds. Also for this constraint, the same considerations made for the PSP (see section 1.2.3) still hold.

**Short sales**

The same considerations made in sections 1.1.2 and 1.2.3 remain valid in the IT framework. Alas, the short selling mechanism has not been taken into account over the literature on this formulation.

**Rounds**

As pointed out in section 1.2.3, it is typical for certain securities and markets that trades are based on multiples of minimum lots, referred to as *rounds*: the same holds in a IT framework. Let the integer \(\rho_i\) be the minimum lot for asset \(i\): the number \(x_i\) of asset \(i\) to be held in the portfolio must be a multiple \(y_i\) of the minimum lot

\[ x_i = y_i \cdot \rho_i \]  \hspace{1cm} (4.12)

As a result, when rounds are introduced in the formulation, they becomes the integer decision variables \(y\) \([60]\), so conclusions drawn by considering the problem consisting of real variables does not hold anymore.

**Class constraints**

Class constraints are introduced in an index-tracking framework by \([35, 36]\) in which assets of automobile industry are bounded to vary in the range \([0.15, 0.2]\) in order to reflect preferences, but specific constraints can also be imposed by law or regulations in several countries. For instance, the *KAGG* (Germany) requires that the share sum of assets with an individual share of at least \(5\%\) should not exceed \(40\%\), so resulting in the following constraint \([35, 36]\) in a continuous formulation:

\[ \sum_{i: w_i \geq 0.05} w_i \leq 0.4 \]  \hspace{1cm} (4.13)

Similar ideas are to be found in the *Asset Class Management*, in which the universe of assets is split into subsets of assets (classes) with similar features. The best representative(s) of each class is (are) selected and optimisation is performed on this pre-selection \([52, 66]\). This idea has been exploited in an Index Tracking framework by \([158]\) in which, tackling the Euro-Pac index, firstly portfolio for singular countries are determined; Then they are combined together to create a compromise solution.
4. Metaheuristics for the index tracking problem

Pre-assignment

Pre-assignment has been introduced in [35], where cash is pre-assigned in range [0.03, 0.1] in order to have enough liquidity if fund shares are returned; A more complex pre-assigning procedure is introduced in [11], where a Reduction Test is performed to determine which assets can be present in the tracking portfolio.

Transaction costs

Transaction costs consist of the amount of money to be paid in order to buy assets, and works exploiting metaheuristics to solve IT problems are used to embed transaction costs in their settings (a notable exception is given by [109, 108]), and this represents a clear element of difference w.r.t. metaheuristics for Portfolio Selection literature, since this latter corpus lacks in including them (see section 1.2.3).

Two main approaches are pursued to include transaction costs in the Index Tracking Problem:

1. Including costs in the objective functions.
2. Including costs in constraints.

For the first case, several routes can be chosen. For example, one might modify the TE definition by using the portfolio returns that reflect the transaction costs incurred, i.e., by computing the returns based on prices net of all cost. Alternatively, transaction costs can be treated as another part of the objective function by means of a weighting approach, defining a new objective function such as

$$\min \lambda T E + (1 - \lambda)TC_t$$

where $TC_t$ captures the transaction costs. Let $|x_{i,t_1} - x_{i,t_0}|$ be the turnover (referred to as $TO(x_{i,t_1}, x_{i,t_0})$), i.e. the asset weight change between two temporal instants. Total transaction costs are clearly sensitive to the amount of assets to be sold and bought and they can be modeled as

$$TC_t = f \left( \sum_{i=1}^{n} TO(x_{i,t_1}, x_{i,t_0}) \right)$$

where $f$ can be chosen such that it reflects the sort of costs arising. The most common approach is to consider transaction costs as proportional to the total turnover

$$TC_t = c \cdot \left( \sum_{i=1}^{n} p_{i,t_1} \cdot TO(x_{t_1}, x_{t_0}) \right)$$

where $c$ is a positive coefficient. Taking into account fixed costs [21] makes the problem even more difficult, as the unconstrained case cannot be handled by convex methods.

In the second case transaction costs are modeled as a constraint so that the amount of costs must not exceed a given limit: \footnote{Indeed, this constraint can be implemented constraining either the turnover [35] or transaction costs [60] being not greater than $UTC_t$.}

$$TC_t \leq UTC_t$$

Although the value of $UTC_t$ can be chosen arbitrarily [35], it is better to consider it as a fraction of the portfolio value, in order to better assess the cost effects when portfolios are different [11, 60]. When imposing transaction costs as a constraint, it is possible to solve the problem for several values of $UTC_t$ [35]: the fund manager can therefore control the turnover volume and consider transaction costs in the optimisation.
4.3 Metaheuristic techniques for the index tracking problem

In this section, we provide an overview of the models appearing in the literature about metaheuristics applied to the Index Tracking Problem, trying to capture the diverse formulations by means of a unique classification, with the aim of giving a general view of Index Tracking modelling along with the possibility of making comparison among the models. As already pointed out for the PSP, this description (also in this case built upon MAGMA [118]) has not the goal of providing an overview of all the Tracking Error formulations, but rather of illustrating, in a unifying view, the diverse models of the problem as described in the specific literature on metaheuristics. Indeed, due to the lack of a broad metaheuristic literature on the topic, several concepts could be directly imported from metaheuristic approaches for the PSP, and further research over IT should consider these concepts.

4.3.1 Metaheuristic attributes for the constrained index tracking problem

In defining the metaheuristic attributes for the Index Tracking problem we are using the same concepts introduced in section 2.2.1: metaheuristics are thought to be abstract classes whose attributes are the search space, the cost function and the neighborhood structure(s) that represent the basic components of the search strategy. In the following, we are going to introduce these classes as they are dealt with in the Index Tracking literature.
Search space

Usually, a solution to the IT is represented by an array of \( n \) variables \( x_1 \ldots x_n \), where \( x_i \) represents the amount invested in asset \( i \).

Besides those variables, auxiliary variables and data structures can be added for improving algorithm efficiency. An important distinction has to be made in the way the different approaches deal with constraint violations. Indeed, as for the PSP, some works define the search space explored by the algorithm as consisting of only feasible portfolios (i.e., satisfying all the constraints in the model), while in other works the search process is allowed to explore also unfeasible solutions: this leads us to the well-known classification of search processes depending on how they handle infeasibility:

- **All feasible** approach.
- **Repair**.
- **Penalty**.

All feasible approaches can be obtained either defining the search move s.t. constraints are implicitly satisfied at each step, or implementing a Check Mechanism surveilling feasibility by returning a boolean value: in principle, the solution can be accepted to move through or not depending on various criteria, yet in current implementations it is accepted only if recognised as feasible [35, 36].

It turns out that there is no reason for preferring one of these strategies when dealing with constraints, as these handling strategies are mostly used together, deciding, for each constraint, which is the most suitable way for handling them. A clear outline of this principle is given by [35, 36], where constraints are partitioned in three classes \( C_1, C_2 \) and \( C_3 \): class \( C_1 \) is composed of constraints the Check approach deal with (floor and ceiling); class \( C_2 \) is composed of constraints whose violation is penalised in the objective function (no short sales, class, pre-assignment and regulatory ones); class \( C_3 \) is composed of constraints impossible not to be fulfilled, as moves are defined so as to make them always satisfied (cardinality and budget constraint). In this approach, a repair mechanism is not implemented, but it can be found in other works: for instance, a typical repair mechanism (w.r.t. cardinality constraint) is given by [11], in which if after a move the solution is unfeasible, an heuristic procedure is used to add or remove assets until it contains exactly \( k \) assets.

A different approach is followed by [108, 109] in order to convert a non normalised vector \( w_s \) to a normalised vector \( w_a \) of decision variables, in a continuous formulation with upper and lower bounds and cardinality constraints, embedding both integer and continuous variables. In this case we cannot define it as a repair mechanism as since the search process operates over \( w_s \) space and the mapping procedure is called at each step in order to determine the actual portfolio, yet without affecting \( w_s \): this means that the next move will be performed over \( w_s \), neglecting what happens to \( w_a \) (the only operation to be performed over \( w_s \) will be storing it if represents the best solution yet found). Once \( w_s \) has been reached by the search move, the set \( W_+ = \{ i \mid w_i > 0 \} \) is determined. Three different cases are to be distinguished:

1. if \( \#w_+ < k_{\text{min}} \) \( w_+ \) will contain the \( k_{\text{min}} \) elements with largest values;
2. if \( \#w_+ > k_{\text{max}} \) \( w_+ \) will contain only \( k_{\text{max}} \) elements with largest values;

---

8Or the fraction of the portfolio in the continuous model, but obviously in this case a vector \( w_1 \ldots w_n \) of continuous variables must be created instead.

9In the following we refer to \( k_{\text{min}} \) and \( k_{\text{max}} \) as the implicit lower and upper cardinality bounds determined by imposing floor and ceiling constraints, see section 1.2.3.
4.3 Metaheuristic techniques for the index tracking problem

3. If \( \#w_+ > k_{\text{max}} \), each \( i \in w_+ \) will be selected, and this allows the actual cardinality to equal or fall below the allowed maximum cardinality.

After that each \( w_i \) \( (i \in w_+) \) is assigned the value \( \varepsilon_i \) and their weights are increased in proportion of the corresponding \( w_s \) until they sum up to 1 and no weight exceeds upper bound 1 (when some \( w_i \) exceed \( \delta_i \) the surplus part is redistributed) \(^{10}\). The actual stock quantity \( x_i \) is afterward computed by rounding \( w_i \cdot B_0 \cdot p_{i,0} \) to the closest integer. Note that with this mapping procedure the budget constraint might be not satisfied exactly, yet the differences are insignificant; furthermore there can be slightly differences in the implementation depending on the cardinality constraint being used (inequality or equality form), but principles remain valid.

Cost function

As already stated for the PSP in section 2.2.1, in many metaheuristic algorithms the objective of the problem is used as evaluation function \([109, 108]\), but sometimes different cost functions can better guide the search toward promising solutions.

An example of cost function for the Index Tracking is provided by \([60]\) in a formulation with transaction costs (plus other constraints, see section 4.3.2) where a cost function in which the cost associated to the violation of constraint over transaction costs is combined with the original objective function (tracking error): solutions for which transaction costs constraint is not satisfied are allowed to be accepted by the algorithm, but obviously they become less desirable due to a worsening of the associated cost value, as the overall cost function to be minimised is a sum of the two components \(^{11}\).

\[
\min f(P) = F_{t_1, t_0} + \max(TC_{t_1} - UTC_{t}, 0) \tag{4.18}
\]

where \( F_{t_1, t_0} \) represents the tracking error, \( TC_{t_1} \) is the rebalancing cost at time \( t_1 \) and \( UTC_t \) is the limit over transaction cost, computed as a fraction of the portfolio value before rebalancing.

This approach can be extended allowing other constraints violations being penalised on the cost function. For instance, \([35, 36]\) partition the constraint set into three classes: constraints belonging to one of these classes \((C_2)\) are penalised on the objective function through a weighting mechanism

\[
\min f(P) = F_{t_1, t_0} + \sum_{c \in C_2} \lambda_c \cdot \text{violation}_c \tag{4.19}
\]

where \( \lambda_c \) is the magnitude of penalization of constraint \( c \). Notice that in this case \( F_{t_1, t_0} \) can be composed of either the tracking error or a weighted sum of tracking error and transaction volume.

A different approach is followed by \([11]\), yet still in a context whose issue is about how to penalise infeasibility. Instead of embedding both objective function and constraint violation in a single aggregate, \([11]\) handles them separately for each solution, so drawing a value for assessing the fitness and another value for assessing the unfitness. This two values are used to determine if to accept the newly produced solution (child) or not.

\(^{10}\)This mapping procedure can be explained stating that genotype (i.e., the non-normalised vector of assets) and phenotype (i.e., the normalised vector of assets) are taken into account as separate entities through the search process: only the phenotype of an individual is altered by the mapping procedure while the genotype remains unaltered. We have already discussed of this approach, introduced in the PSP framework by \([151, 150, 152]\).

\(^{11}\)Note this being not a weighted sum, whilst weighted sum of the two components, whose weights change over time due to shifting mechanism are implemented in a PSP framework by \([137]\).
4. Metaheuristics for the index tracking problem

Neighborhood relations

Quite differently from the PSP literature, there are not many neighborhood relations defined in the Index Tracking literature, due to the scarcity of works on the topic. Due to the nature of the problem, almost all relations used in PSP (see section 2.2.1) can be tailored to Index Tracking, but for the time being the only relation is defined by transferring a given amount (step) between two assets: the same relation introduced in a PSP framework by [133], being compared to a sell a given amount of asset i, buy the same amount of asset j operation. Several ways of assessing this amount can be used, as it can be defined as a random amount [35], a fixed amount [60] or the amount leading to the greatest decrease in tracking error [36].

Of course, this transfer mechanism should embed some mechanisms in order to ensure feasibility w.r.t. all feasibility-handled-constraints at hand. Gilli and Kellezi [60] take it into account by selecting asset to be sold in the portfolio and asset to be bought either in the portfolio itself, if maximum cardinality is already obtained, or in the entire asset universe, if there is room for inserting another asset in the portfolio. Furthermore, if the actual step leads to an unfeasible state, it is re-determined in order to ensure feasibility.

Feasibility is handled by [35] by randomly choosing assets to be modified: if both are not in the current portfolio the choice is performed again; if both are contained in the portfolio a random step is selected and transferred between assets. Furthermore, if just one asset has positive share, two subcases are distinguished: if the current portfolio has a cardinality less than the maximum allowed, a random step is selected and transferred between assets; otherwise, the shares of the two assets are swapped.

Last, [36] define the amount to be transferred as the amount leading to the greatest decrease of the tracking error, but if this transfer leads to an unfeasible state, the step is performed in the other direction, so leading to an up-hill move. This approach can be considered as a good compromise between intensification and diversification.

4.3.2 Metaheuristic search components

Following the same schema pursued in section 2.1, in this section, we describe the search methods composing the metaheuristic approaches for the Index-Tracking problem. We first present trajectory based strategies, such as simulated annealing and threshold accepting, and then we introduce population-based metaheuristics, such as evolutionary algorithms and differential evolution.

Initial solution

As already stated in section 2.2.2, it has been proven that metaheuristics are quite robust w.r.t. the starting solution. In practical applications, it is sometimes found that a random initial solutions is favorable to one coming from another search technique when the latter is close to a strong local optimum; nonetheless, the choice of the starting solution does usually not represent a critical aspect. The only issue to be taken care of is to generate a solution that corresponds to the search process; i.e., that is feasible, or at least feasible w.r.t constraints we choose an all feasible approach to deal with [35, 36].

Iterative improvement

Iterative improvement is usually embedded into a more complex strategy, rather than being used as a stand alone local search. It has been experimented by [36] as a modification of SA (see here below), conveying a faster minimisation of the tracking error, but providing overall
unsatisfactory performances, as it does not manage to reduce the penalty term in the Cost Function.

Simulated annealing

SA is used by [35, 36] in order to implement a metaheuristic based decision-support-system generator (DSS) introducing cardinality, floor and ceiling constraint (different limits for each asset), short selling prohibition, class constraints, pre-assignment over cash and constraints stemming from regulatory issues (this latest constraint, along with class one, is to be penalised in the objective function). Firstly, they tackle a Single Period Index Tracking (portfolio endowed) problem, then they tackle a Dynamic Multi-Period Index Tracking Problem, rebalancing the portfolio at any time it becomes unfeasible w.r.t. given constraints. Interestingly, SA is able to track the index in both cases, resulting in portfolios whose assets have weights close to the limits given by constraints. The main shortcoming of this approach is that transaction costs are not embedded in the optimisation, i.e. they are considered after determining the portfolio, and they might affect return values. To overcome it, a new experimental setting is defined introducing transaction costs in the objective to minimise equation (4.19), resulting in a smaller transaction volume in both approaches. It is shown that results obtained by SA, though in a simplified formulation, favorably compare with standard QP solvers such CPLEX [29].

Threshold accepting

TA has been applied to the Index Tracking by [60], in a formulation with floor and ceiling, cardinality, and rounds constraints, also introducing transaction costs, operating over a set of five instances (ORLIB) and then pooling them together in order to obtain a bigger set composed of 528 assets. The index is artificially built, randomly selecting \( k \) assets: although correct, this choice jeopardise assessing the real effectiveness of the strategy, designed to mimic the behaviour of an index whose components (and weights) change continuously over time. That said, obtained portfolios track the index almost exactly. An extension is given tackling a Static multi-period IT problem: after computing the tracking portfolio from scratch, it is rebalanced over time at fixed intervals, yet without constraining transaction costs, rather embedding them in the cost function.

Evolutionary algorithms

Often EA-based metaheuristics are enhanced by hybridising EAs with other advanced strategies in order to improve the quality of the solutions constructed by the EA; this has been done, dealing with IT, introducing memetic algorithms. Beasley et al.[11] exploit a formulation with transaction costs, cardinality, floor and ceiling constraints using an EA whose individual evaluation is based on two measures so that their approach can be seen as a multi-criteria framework. The first function is the objective function of the problem (weighted sum of tracking error and portfolio return), that guides the search, since it is used in order to choose parents (through binary tournament selection); the second function is referred to as unfitness, given by the degree of violation of transaction cost constraint, and it is used, in conjunction with the first, in order to decide which portfolio (individual) to replace.

A population of \( p \) portfolios is created, and the two individuals with the best objective values (chosen by tournament selection) are selected to be parents. The child will contain all assets belonging to both parents, whilst assets belonging to one parent only will have 0.5 likelihood to be children components. Also mutation is implemented, adding or subtracting 5 percent of the value of a randomly selected stock. The child must always replace a member of
the former population, and in order to choose which one, the population is partitioned into 4 classes, composed of:

1. individuals whose both objective function and unfitness are worse than child ones.
2. individuals whose objective function is worse and unfitness is better than child ones.
3. individuals whose objective function is better and unfitness is worse than child ones.
4. individuals whose both objective function and unfitness are better than child ones.

As said before, in this implementation a child always has to replace a member of the population, chosen accordingly to the aforementioned functions, even if this lead to replace an individual with better objective function. Indeed, firstly, set 1) is selected: if it is not empty the worst individuals are replaced, otherwise the procedure is repeated, in row, for sets 2, 3, 4. A reduction test is furthermore performed in order to reduce the search space size (i.e., deleting the less promising assets and determining which assets must always be included in the tracking portfolio), and an artificial index is created by selecting $k$ assets (similar to [60]). Firstly a single-period tracking portfolio (portfolio endowed) is computed for several transaction cost limits, showing that increasing them reduces the tracking error; then its performance is assessed out-of-sample, showing that an in-sample reduction also leads to out-of-sample error reduction. A further analysis is performed by tackling a static multi-period IT problem choosing some (seven) decision points out of the 290 weeks available to rebalance the portfolio on. Even in this case it is shown that the tracking error decreases when the transaction costs limit increases.

An interesting hybrid genetic algorithm has been proposed by [138, 46], where Quadratic Programming and Genetic Algorithm are combined in a problem decomposition in which a genetic algorithm searches the space of assets only (i.e., on the binary variables $z_i$) and at each step the QP solver determines the optimal allocation over them. The same decomposition has been proposed by [122], combining a heuristic to decide which assets to be inserted into the portfolio and GA to optimise weights. This decomposition, already applied to the PSP by [119], is central to this thesis, as it has been applied to the PSP in the previous chapter, while it will be applied to a multi-objective approach, combining IT and PSP, in the next chapter.

**Differential evolution**

Differential Evolution (DE) [148] is a population based strategy using a population $S$ of vectors, where vector elements represent decision variables. The idea is to generate new solutions as a combination of three current vectors: a base vector $s_b$ is added the weighted difference of other two vectors $s_1$ and $s_2$. Of course, this difference will be significant as long as $s_1$ and $s_2$ will have variables with different values, whilst if this difference is null (or negligible) the new solution will inherit features by the base vector only. This obtained solution is afterward combined (crossed over) with an existing solution $s_p$ generating a new solution $s_n$: if the obtained solution is better than $s_p$, it will replace it, otherwise it will disappear. Extensions can be made by adding noise (in order to avoid premature convergence) or reinforcing the best solution found so far (elitist principle) [129].

In [109] DE is used in order to investigate the reduction of TE by increasing the portfolio cardinality (see section 1.2.3) in a setting including upper and lower bounds, cardinality and integer constraints over variables (yet without transaction costs). Both in-sample and out-of-sample analysis are performed, finding out interesting financial conclusions: firstly, the magnitude of tracking error (and its decay) is hardly affected by sample length, but changes over time, especially in years dramatic events were verified on; then, tracking error is due to upside deviations (so the tracking portfolio outperforms the index, yet at cost of higher volatility)
and the portfolio Sharpe Ratio (see section 1.1.2) is greater than the index one; last, in-sample findings also hold for out-of-sample analysis.

In [108] the formulation is enriched by introducing loss aversion (see section 4.2.2). From a financial point of view, this means investors reaction to losses are bigger than a traditional risk aversion approach, and either they want to reduce losses or they want bigger upside deviations to outweight losses perceptions. This results in a stronger preference for positive skewness of return differences between index and portfolio (obviously, the higher $\gamma_{12}$ the higher the skewness), although it can result in volatility increase. Differences come over when comparing loss averse and loss neutral individuals: the frequency of losses is the same in both classes, but this results in higher upside deviations when the portfolio outperforms the index. Also in this case, out-of-sample findings are similar to in-sample ones, with a notable exception: out-of-sample kurtosis (of return differences) is smaller than in-sample one, as a consequence of both returns overfitting affection over in-sample data and the stability of the underlying assets’ return.

In both works, DE was shown to require a small population of vectors (100, a rather small set considering the universe being composed of 65 assets) and to be quite robust w.r.t. parameter tuning.

It is worthy to notice here the lack of an exhaustive literature about Index Tracking by metaheuristics: the strategies applied to the problem sum up to 5, while well-established stochastic local search methods haven’t found application yet. Furthermore, no work comparing different strategies over the same set of instances has been produced, making difficult a fair comparison between methods. Little effort in designing hybrid metaheuristics have been made, and exact methods have also been applied as stand-alone methods: these methods provide us with the optimal solution of the given instance, but are not suitable for solving large-sized instances. Indeed, either they are used in conjunction with mathematical approximations that make their use possible or they are employed as components of hybrid metaheuristics. In order to design robust hybrid metaheuristics, an analysis of simpler heuristics should be made: for instance, in [154] a heuristic is introduced in order to minimise the expected value of the squared differences $(r_p - r_i)^2$ with fixed cardinality constraint, even if computational results cover only a small instance (15 stocks); attention should be paid to linear methods, even if the nature of the strategy imposes some limits over the objective to optimise.

A brief outline about algorithms that have been applied to the Index Tracking will be described in Appendix A.3.

### 4.3.3 Benchmark instances

The selection of data to be used represents a sensitive choice: it depends of course on subjective factors, but results obtained over an instance can be contradicted over other instances. Furthermore, in the same instance itself the problem of overfitting can arise, as results over in-sample data might not be valid over out-of-sample data belonging to the same index. The best choice would be to test the algorithm over a set of different instances, comparing and generalising obtained results, and analysing both in-sample and out of sample results. Out-of-sample analysis has been performed by [109, 108, 11], showing that tracking portfolio performances are similar over both in-sample and out-of-sample data (for a detailed explanation see section 4.3.2). Several instances analysis have been introduced by [11], defining a set of five instances and making

---

12$\gamma_{12}$ is the loss aversion parameter, see section 4.2.2.

13This means, respectively, $\gamma \geq 1$ and $\gamma = 1$, see section 4.2.2.

14We are referring to such strategies as Tabu Search, ACO, Particle Swarm, Variable Neighborhood Search, GRASP.
4. Metaheuristics for the index tracking problem

<table>
<thead>
<tr>
<th>Paper</th>
<th>Index Origin</th>
<th>Index size</th>
<th>Universe origin</th>
<th>Universe size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beasley et al; Gilli and Kellezi</td>
<td>Hang Sen</td>
<td>$k$</td>
<td>Hang Sen</td>
<td>31</td>
</tr>
<tr>
<td>Beasley et al; Gilli and Kellezi</td>
<td>DAX100</td>
<td>$k$</td>
<td>DAX100</td>
<td>85</td>
</tr>
<tr>
<td>Beasley et al; Gilli and Kellezi</td>
<td>FTSE100</td>
<td>$k$</td>
<td>FTSE100</td>
<td>89</td>
</tr>
<tr>
<td>Beasley et al; Gilli and Kellezi</td>
<td>S&amp;P100</td>
<td>$k$</td>
<td>S&amp;P100</td>
<td>98</td>
</tr>
<tr>
<td>Beasley et al; Gilli and Kellezi</td>
<td>Nikkei 225</td>
<td>$k$</td>
<td>Nikkei 225</td>
<td>225</td>
</tr>
<tr>
<td>Derigs and Nickel (2003a, 2003b)</td>
<td>DAX30</td>
<td>30</td>
<td>DAX30 + STOXX200</td>
<td>202</td>
</tr>
<tr>
<td>Maringer (2007)</td>
<td>DJIA64</td>
<td>65</td>
<td>DJIA64</td>
<td>65</td>
</tr>
<tr>
<td>Maringer and Oyewumi (2007)</td>
<td>DJIA64</td>
<td>65</td>
<td>DJIA64</td>
<td>65</td>
</tr>
<tr>
<td>Shapcott</td>
<td>FTSE100</td>
<td>100</td>
<td>FTSE100</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 4.1: The benchmark instances

them publicly available in order to lead further research and comparisons over them. These instances have also been used by [60], where they are also pooled in order to create a super-set to test the algorithm against.

Most works deal with real indices to track and, by assumption, index composition is not known when attempting to track it. Nevertheless, artificial indices can be created as well: [60, 11] define an artificial index composed by selecting $k$ assets in the available set. Operating in this way, they are sure it is possible to track the index with 0 tracking error whatever the measure used. Indeed, although useful to test the algorithm performance, this approach is of little practical interest, as in real-markets we are concerned with tracking an index whose components and weights change over time.

When deciding which data to be used, asset return heteroscedasticity should be taken into account: this mean that assets have constant mean, yet their variance is time-dependent. This feature is influenced by observation frequency, as it increases as frequency decreases, meaning that daily data are more heteroscedastic than monthly (or annual) ones. As a result of this consideration, weekly data are used by [11, 60], but it is worthwhile noticing that other works use daily data [109, 108, 35, 36]. A summary of the main instances features is to be found in table 4.3.3.

Looking at the table it is worthy to notice that the widespread definition of Index Tracking as tracking an index with a subset of its components it is not mostly used when dealing with metaheuristics, as universes whose cardinality is bigger than the relative index one are used: in both [60] and [11] indeed this statement should be conceived as the other way round, as the index is artificially generated as a subset of the universe picking $k$ assets out of it, whilst [35, 36] really use a universe set whose cardinality (202) is greater than the index at hand one (30).

4.4 Conclusion

In this chapter we have classified, following the same pattern already defined and used in chapters 1 and 2, the main literature about Index Tracking by metaheuristics. It turns out that although the literature on portfolio choice has so far focused on portfolio selection problem (PSP), there does exist a body of literature that is concerned with passive strategies such as Index Tracking. All aspects detailed in this chapter will turn to be useful in the next chapter, where we will define a multi-objective framework combining PSP and IT to tackle optimised portfolio choices.
Chapter 5

A multi-objective approach for portfolio choice

A situation in Index Tracking (referred to as Index Beating) arises when the portfolio manager wants to beat the benchmark index: in this situation, common practice is to maximise the excess return (say, the difference over time between portfolio’s returns and index’ returns, as seen in section 4.2.2). It has been shown that just focusing on excess return leads to inefficient results [132], as the portfolio risk is always bigger than the index one. A remedy to this situation would be to optimise the excess-return while adding constraints over the Tracking Error, but this approach has the greatest shortcoming in that the solution is independent from the index at hand, so we would obtain the very same portfolio maximising excess return w.r.t. MIBTEL, FTSE or NIKKEI; Furthermore, the obtained portfolios are never efficient in terms of Mean Variance analysis, as even increasing the allowed tracking error, an increase on the volatility will be observed as the result of focusing on excess return instead of total return. As a further remedy, a constraint over portfolio risk (variance) can be introduced, so facing three variables: Risk, Index Tracking and Excess Return. In this framework it can be seen that the frontier obtained optimising for several allowed variance values collapses to the mean-variance one when increasing the allowed tracking error value [78]. It has been shown that using these three variables in conjunction (rather than just tracking the index) can be useful in situations where the benchmark is relatively inefficient: adding a constraint over portfolio variance can lead to a consistent decrease of portfolio risk at the cost of a negligible return loss [78]. In the following, we will introduce a multi-objective analysis of the portfolio choice combining Index Tracking and PSP (absolute return will take the place of excess return discussed previously): though in the previous chapter we have classified portfolio strategies as belonging either to active strategies or to passive strategies, the two approaches happen to be not mutually exclusive and can be used together in a hybrid strategy. The goal of such a hybrid approach is to reflect several degrees of both investors attitude toward risk and their trust in the market. Furthermore, in order to overcome limitations of mean-variance portfolios, multi-objective approaches including Index Tracking have been proposed by [147], but the experimental phase is missing.
5.1 Our approaches

The main idea of our approach is to combine the Mean Variance framework with Index Tracking, so leading to a multi-objective formulation as follows:

\[
\begin{align*}
\text{min} & \quad \text{Risk} \\
\text{max} & \quad \text{Return} \\
\text{min} & \quad \text{Tracking Error}
\end{align*}
\]

Indeed, the idea of combining PSP with the Index Tracking problem has already appeared in literature. For instance, multi criteria formulations with Index Tracking as one of the objectives have been suggested by [147] and tackled jointly with Sharpe Ratio maximization\(^1\) [22] and Downside Risk Measures [168]. Other multi-objective approaches have been proposed by [134].

As already seen in chapters 1 and 4, several measures can be used to define what the above mentioned objectives are. In the following we will stick to the Mean-Variance representation for returns and risk (i.e., using mean return as return measure and variance as a risk measure), whilst three different measures will be dealt with to quantify the Tracking Error.

The implementation we followed for the multi-objective approach is the very same we used to tackle Mean-Variance: we choose a measure to optimise, while constraining the others in the formulation. In our case, we aim at minimising variance while imposing a minimum required return \(r_e\) and a maximum allowed tracking error \(TE\). By resolving the problem for different pairs of \((r_e, TE)\) we obtain a three dimensional surface. This surface cannot be referred to as the Pareto front, as it consists of an approximation without proof of optimality; furthermore it contains only a subset of the front’s points, and, above all, it contains points that are dominated, and that will be deleted to obtain our real AUEF (Approximated Unconstrained Efficient Frontier, as defined in section 1.2.2).

As already pointed out in section 4.3, several performance measures can be introduced to assess how much the portfolio stays close to the index in a given period. In our work, we experimented three different error measures, each defining a different formulation of the problem and being implemented by adding further constraints to equations (1.1) to (1.4). Let \(T\) be the number of observations, \(n\) the number of assets in the universe, \(w_a\) the fraction of the portfolio to invest in asset \(a\), \(r_{a,t}\) the return of asset \(a\) over the observation \(t\), \(r_{I,t}\) the index return at time \(t\). The three formulations we are dealing with are the following:

1. The Index-Beating Problem, requiring the portfolio to perform better than the index for each day of the observation horizon. In this formulation, the constraint set to be added is

\[
\sum_{a=1}^{n} w_a r_{a,t} \geq r_{I,t} \quad \forall t \in T
\]  \hspace{1cm} (5.1)

In this fashion, the formulation turns to be adding \(T\) constraints to the original Mean-Variance formulation (1.1)-(1.4).

2. The canonical index tracking problem, in which we want the sum of differences between the index’s and portfolio’s return not to be worse than a given threshold. In this approach, the extra-constraint to be added to the formulation is

\[
\sum_{t=1}^{T} \left| \sum_{a=1}^{n} w_a r_{a,t} - r_{I,t} \right| \leq TE
\]  \hspace{1cm} (5.2)

\(^1\)See [40].
5.2 Experimental settings

where $TE$ is the maximum desired tracking error over the observations (i.e., the sum of daily tracking errors must be smaller than $TE$). Note that in this formulation, the number of constraints to be added to the formulation is $2^T$, since the aforementioned formula must be converted into $2^T$ linear constraints in order to be dealt with by our Quadratic Programming.

3. The daily Index Tracking Problem, where we want the difference between portfolio's and index' return to be smaller (or equal) than a given error for each of the observation points. This means adding to the formulation the following constraints:

$$\left| \left( \sum_{a=1}^{n} w_a r_{a,t} \right) - r_{I,t} \right| \leq TE \quad \forall t$$

(5.3)

The cardinality of this set of constraints is $2 \cdot T$. Note that in this last formulation $TE$ represents the daily tracking error, and not the global one such as the one introduced in case 2).

It is easy to see that the cardinality of the constraint set to be added to the formulation is reasonable for cases 1) and 3) while it explodes easily for case 2). Indeed, if we had an horizon of 128 days to feed our problem (so, an even modest number of daily observations), the number of constraints to be added should be $2^{128} (= 340282366920938463463374607431768211456)$. It is clear that it is too big a number for traditional enumerative methods to process. Note that we did not introduce any further constraint (i.e., cardinality, floor, ceiling etc.) in the original model, as our goal is only to understand the behaviour of the Pareto Frontier when combining these two models. For the experimental phase, we used the same approach outlined in chapter 3, so operating a master-slave problem decomposition and introducing a two-components hybrid metaheuristic approach in which a metaheuristic is defined to tackle the space of assets only, while Quadratic Programming determines, for each asset combination, the optimal asset weights w.r.t. variance minimisation fulfilling the constraint added to the model. Every local search component (search space, cost function, neighborhood relations, initial solution) is to be defined in the very same way we did in chapter 3. The only difference w.r.t. our previous approach, is that the constraint matrix to be processed by the quadratic solver, is to contain each additional constraint deriving from our new formulations to be tackled.

5.2 Experimental settings

As instance set we used a group of five instances taken from the repository ORlib available at the URL http://mscmga.ms.ai.ac.uk/~jeb/orlib/portfolio.html. These instances have been used by [11] and are referring to five well-known stock exchange indices. For each instance we have 290 daily observations. Tables 5.1 and 5.2 show our benchmark features (origin, cardinality and average variance) when using both continuous and discrete computations.

These instances are conceived as daily observed stock prices (for assets) and index value. In order to apply our formulations to these instances, prices have been converted to returns using both discrete and continuously compounded formula (see the introductory part to chapter 1), in order to understand if the diverse formulas lead to different optimisation results:

$$r_{a,t} = \log \left( \frac{S_{a,t+1}}{S_{a,t}} \right)$$

(5.4)

$$r_{a,t} = \frac{S_{a,t} - S_{a,t-1}}{S_{a,t-1}}$$

(5.5)
5. A multi-objective approach for portfolio choice

<table>
<thead>
<tr>
<th>Inst.</th>
<th>Origin</th>
<th>assets</th>
<th>Avg. Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hong Kong</td>
<td>31</td>
<td>0.0068247366</td>
</tr>
<tr>
<td>2</td>
<td>Germany</td>
<td>85</td>
<td>0.0059415526</td>
</tr>
<tr>
<td>3</td>
<td>UK</td>
<td>89</td>
<td>0.0052871267</td>
</tr>
<tr>
<td>4</td>
<td>USA</td>
<td>98</td>
<td>0.0055660108</td>
</tr>
<tr>
<td>5</td>
<td>Japan</td>
<td>225</td>
<td>0.0020211282</td>
</tr>
</tbody>
</table>

Table 5.1: The benchmark instances, continuously compounded returns.

<table>
<thead>
<tr>
<th>Inst.</th>
<th>Origin</th>
<th>assets</th>
<th>Avg. Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hong Kong</td>
<td>31</td>
<td>0.0084706972</td>
</tr>
<tr>
<td>2</td>
<td>Germany</td>
<td>85</td>
<td>0.006815315</td>
</tr>
<tr>
<td>3</td>
<td>UK</td>
<td>89</td>
<td>0.0056192979</td>
</tr>
<tr>
<td>4</td>
<td>USA</td>
<td>98</td>
<td>0.0050672006</td>
</tr>
<tr>
<td>5</td>
<td>Japan</td>
<td>225</td>
<td>0.002376324</td>
</tr>
</tbody>
</table>

Table 5.2: The benchmark instances, discretely compounded returns.

where $r_{a,t}$ is the return of asset $a$ at time $t$ and $S_{a,o}$ is the price of asset $a$ at time $o$. We remember that the lower bound for portfolio returns (when forbidding short selling, as in our case) is the return of the minimum variance point, whilst its upper bound is the maximum amongst asset returns. This computation was made with a QP solver by Luca Di Gaspero publicly available at http://www.diegm.uniud.it/digaspero/index.php?page=software.

As formulation 2 turned to be unfeasible (see section 5.3) we analysed formulations 1 and 3. Again, the two optimisation problems were tackled using the Master-Slave decomposition used in [37] and explained in detail in chapter 3, so a combination of Local Search and Quadratic Programming solver.

Experiments were run on a cluster\(^2\) composed of 96 CPUs dedicated to computations and 2 CPUs for administrative purposes.

### 5.3 Experiments with index beating

Although the number $T$ of constraints added is not large, they make the problem very difficult to be solved, as they request the portfolio always being better performing than the index. As the portfolio is forbidden to be rebalanced, this task happens to be very difficult for fund managers, and likewise for our Quadratic Programming. We performed experiments on the 5 instances, but it turns out that for every instance, there are no feasible solutions found. It is not clear if this result can be generalised or if it depends strictly on instance’s features. Further experiments over different instances are to be made in order to investigate this aspect.\(^3\)

\(^2\)Located at IRIDIA, Université Libre de Bruxelles, Belgium.

\(^3\)For sake of completeness, we note that, for checking purposes, another formulation has been tried to find a solution always underperforming the index but also this problem turned to be unfeasible.
5.4 Experiments with index tracking

The only formulation we are able to tackle is the daily index tracking problem, obtained adding the following constraint set:

\[
\left| \sum_{a=1}^{n} w_a r_{a,t} - r_{I,t} \right| \leq TE \quad \forall t \quad (5.6)
\]

to the Markowitz formulation. The whole formulation becomes

\[
\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} w_i w_j \quad (5.7)
\]

subject to

\[
\sum_{i=1}^{n} r_i w_i \geq r_e \quad (5.8)
\]

\[
\sum_{i=1}^{n} w_i = 1 \quad (5.9)
\]

\[
w_i \geq 0 \quad i = 1 \ldots n \quad (5.10)
\]

\[
\left| \sum_{a=1}^{n} w_a r_{a,t} - r_{I,t} \right| \leq TE \quad \forall t \quad (5.11)
\]

still using the continuous variables representation. Note that we are neglecting all additional constraints but the index tracking ones. Solving this problem for an arbitrarily huge value of \( TE \) we obtain the Mean-Variance efficient frontier, while imposing other values has the effect of getting results that are different from the unconstrained case. We computed, for each \( TE \) value, a discretised AUEF composed of 100 equally distributed values of the expected return \( r_e \). We draw this frontier for 100 equally distributed \( TE \) values, obtaining a three dimensional surface to be further refined by deleting dominated points, as points obtained optimising \( \text{w.r.t.} \) a given \( r_e \) can be dominated by points obtained optimising \( \text{w.r.t.} \) another \( r_e \) value.

5.4.1 Local search settings

We used, for the experimental phase, the same approach as defined in chapter 3. The only remarkable difference is that extra Index Tracking constraints (5.11) are handled directly by the QP solver. No additional constraints are added (i.e., cardinality, floor, ceiling, pre-assignment etc.), as our goal is to outline the global behaviour of our formulation rather than to discuss the influence of constraints parameter settings. In the following list, we summarise the main local search features:

- **Search Space** Similarly to what we already discussed in section 3.1.1, the search space \( S \) is composed of the all \( 2^n \) possible configurations of \( Z \), while the QP solver receives as input assets included in the state under consideration, producing values to the corresponding \( w_i \) variables. Note that, if the QP solver is unable to produce a feasible solution, the configuration that gives the highest return, using only the assets included in \( S \), is determined.

- **Neighborhood relations** The very same relations defined in section 3.1.3 are defined: insertion, deletion and replacement.
• **Initial solution** We have noticed in section 3.1.4 that three strategies can be used to this extent. Indeed, results have shown themselves to be insensitive to the initial solution choice, so we determine our initial solution as randomly generated (previously referred to as RandomCard).

• **Local Search Strategies** We implemented the three mentioned local search techniques, namely Steepest Descent, First Descent and Tabu Search. Concerning the parameter settings, only TS needs to be optimised, and this has been done by means of F-Race [14]. For TS a dynamic-size tabu list is used and we search for the next state by exploring the full neighborhood at each iteration.

### 5.5 Experimental results

In the following, the experimental results of our approach will be discussed. We will detail the correlation analysis made before the optimisation, before entering into algorithm’s results details, such as the computational time, the equal points (i.e., the intersection between portfolios found by two strategies) and the coverage measure (i.e., how many points found by a strategy are dominated by points found by another strategy). Note that we are not using the aggregate defined in chapter 3 (apl) to assess performances. This is due because results (frontiers) from chapter 3 consisted of equidistanced points, and through them we could have an idea about the global algorithm’s behaviour, so summarising its performance via a unique number. In the multi-objective we are dealing with, instead, points composing the frontier are not equally distributed (because of deletion of non-dominated points, as previously stated, or because constraints are too binding, or eventually because the solver does not manage to find a feasible solution). For this reason, instead of using an aggregate to quantify the absolute performance of an algorithm, we used measures to compare algorithms with each other.

#### 5.5.1 Correlation analysis

Before deciding to optimise the three objectives as stated before, we performed a correlation analysis to understand if the objectives were correlated, to avoid optimising features that are very highly correlated. We computed the rank-based pairwise correlation between return, variance and the tracking error we are dealing with over a randomly generated sample composed of 1000 portfolios with random cardinality and with no constraint other than the non-negativity. Results are shown in table 5.3. It can be seen that correlation over the two cases (continuous and discrete) are of comparable values and, furthermore, negligible when analysing \((TE, r_p)\) and \((\sigma^2, r_p)\) (an exception is given by \((\sigma^2, r_p)\) on instance 5, but still the value does not appear to be too high). Conversely, higher values are found observing \((TE, \sigma^2)\), where the value is almost always higher than 0.5. Nevertheless, correlation values are not too close to one, so that the behaviour on different instances can be considered sufficiently different.

---

4We decided to show here the rank-based correlation rather than, i.e., Pearson correlation, in order to better evidence non-linear features between variables. It is nonetheless worthwhile to notice that rank based correlation and Pearson indicator lead to comparable results (a little exception is shown in \((TE, \sigma^2)\), where the value is almost always higher than 0.5 using rank-based, whilst always higher than 0.6, so showing a higher correlation, when using Pearson). For interested readers, Pearson correlations amongst these variables are shown in Appendix C.

5The only noticeable difference is between \((TE, r_p)\) values on instance 1, as it is positive in the discrete case and negative in the continuous, still the value of the difference is not significant and anyhow the two measures turn to be uncorrelated.
5.5 Experimental results

Correlation between Return, Variance and Tracking Error over random portfolios, continuously compounded

<table>
<thead>
<tr>
<th>Instance</th>
<th>((TE, \sigma^2))</th>
<th>((TE, r_p))</th>
<th>((\sigma^2, r_p))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4898572</td>
<td>-0.0420849</td>
<td>-0.0685356</td>
</tr>
<tr>
<td>2</td>
<td>0.5766356</td>
<td>-0.1343672</td>
<td>-0.1214940</td>
</tr>
<tr>
<td>3</td>
<td>0.7370264</td>
<td>0.0282333</td>
<td>0.1520472</td>
</tr>
<tr>
<td>4</td>
<td>0.8032893</td>
<td>0.0127949</td>
<td>0.2671934</td>
</tr>
<tr>
<td>5</td>
<td>0.5788888</td>
<td>-0.3371507</td>
<td>-0.4971540</td>
</tr>
</tbody>
</table>

Correlation between Return, Variance and Tracking Error over random portfolios, discretely compounded

<table>
<thead>
<tr>
<th>Instance</th>
<th>((TE, \sigma^2))</th>
<th>((TE, r_p))</th>
<th>((\sigma^2, r_p))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4864088</td>
<td>0.0705872</td>
<td>-0.1786201</td>
</tr>
<tr>
<td>2</td>
<td>0.5788265</td>
<td>-0.1458709</td>
<td>-0.0732799</td>
</tr>
<tr>
<td>3</td>
<td>0.7447293</td>
<td>0.0247673</td>
<td>0.1442864</td>
</tr>
<tr>
<td>4</td>
<td>0.8016539</td>
<td>0.0691124</td>
<td>0.2482710</td>
</tr>
<tr>
<td>5</td>
<td>0.7427990</td>
<td>-0.2622887</td>
<td>-0.2913642</td>
</tr>
</tbody>
</table>

Table 5.3: Random portfolios: rank based correlation analysis.

5.5.2 Algorithm results

For each instance, 100 efficient frontiers have been determined by each strategy, one for each level of maximum allowed tracking error (ranging from 1 to 100). Each frontier consists of minimum variance points for 100 uniformly distributed minimum required returns. In whole, for each multidimensional three-objective surface, 10,000 optimisation processes have been attempted, so each frontier’s cardinality, for each strategy and for each instance, has the upper bound of 10,000 points. In reality, this number is never reached, as in many cases either constraints are too stringent, or the solver returns an unfeasible state when it does not succeed in finding a feasible one. Furthermore, as previously stated, we are interested in non-dominated points only.\(^6\)

In the following tables we show, for each pair of (strategy-instance), the number of solutions found and the number of non-dominated points out of them (respectively, first and second numbers between parenthesis), for both continuously compounded and discrete formulas.

<table>
<thead>
<tr>
<th>Method</th>
<th>Instance 1</th>
<th>Instance 2</th>
<th>Instance 3</th>
<th>Instance 4</th>
<th>Instance 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD+QP</td>
<td>(16224 – 166)</td>
<td>(16288 – 223)</td>
<td>(17728 – 151)</td>
<td>(17428 – 151)</td>
<td>(10 – 1)</td>
</tr>
<tr>
<td>TS+QP</td>
<td>(16384 – 170)</td>
<td>(16552 – 218)</td>
<td>(18030 – 152)</td>
<td>(16876 – 156)</td>
<td>(6 – 1)</td>
</tr>
</tbody>
</table>

Table 5.4: Found versus Non-Dominated Portfolios, Continuously Compounded Returns. The first number represents the number of portfolios found, the latter the non-dominated portfolios out of them.

It can be seen that, in the continuous case, the FD+QP strategy produces the largest number of feasible portfolios; nevertheless, it is also the strategy that produces the smallest number of non-dominated points. In the discrete case instead, more points are generated by more sophisticated strategies such as SD+QP and TS+QP. In the continuous case, TS+QP reaches the highest

\(^6\)For instance, two points with the same risk and return can be independently found in AUEF corresponding to different \(r_e\) values, but only one of them (namely, the smallest index-tracking valued) will be non-dominated over the three-dimensional surface.
A multi-objective approach for portfolio choice

Table 5.5: Found portfolios versus Non-Dominated Portfolios, Discretely Compounded Returns. The first number represents the number of portfolios found, the latter the non-dominated portfolios out of them.

<table>
<thead>
<tr>
<th>Method</th>
<th>Instance 1</th>
<th>Instance 2</th>
<th>Instance 3</th>
<th>Instance 4</th>
<th>Instance 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD + QP</td>
<td>(7579 − 208)</td>
<td>(8791 − 141)</td>
<td>(9570 − 144)</td>
<td>(9700 − 118)</td>
<td>(10 − 4)</td>
</tr>
<tr>
<td>SD + QP</td>
<td>(7482 − 216)</td>
<td>(8892 − 148)</td>
<td>(9575 − 145)</td>
<td>(9713 − 119)</td>
<td>(6 − 5)</td>
</tr>
<tr>
<td>TS + QP</td>
<td>(7605 − 221)</td>
<td>(8992 − 142)</td>
<td>(9583 − 148)</td>
<td>(9711 − 111)</td>
<td>(2 − 1)</td>
</tr>
</tbody>
</table>

Table 5.6: Computational time.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>I1</th>
<th>I2</th>
<th>I3</th>
<th>I4</th>
<th>I5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FD + QP</td>
<td>51min 2.42s</td>
<td>58min 0.94s</td>
<td>52min 18.63s</td>
<td>57min 53.82s</td>
<td>47min 21.13s</td>
</tr>
<tr>
<td>SD + QP</td>
<td>55min 8.51s</td>
<td>60min 40.98s</td>
<td>61min 23.92s</td>
<td>56min 17.92s</td>
<td>42min 59.10s</td>
</tr>
<tr>
<td>TS + QP</td>
<td>58min 31.62s</td>
<td>61min 33.38s</td>
<td>62min 5.39s</td>
<td>56min 23.33s</td>
<td>43min 40.35s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strategy</th>
<th>I1</th>
<th>I2</th>
<th>I3</th>
<th>I4</th>
<th>I5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FD + QP</td>
<td>51min 33.99s</td>
<td>54min 41.66s</td>
<td>53min 42.65s</td>
<td>54min 58.00s</td>
<td>51min 52.54s</td>
</tr>
<tr>
<td>SD + QP</td>
<td>56min 15.65s</td>
<td>54min 52.23s</td>
<td>52min 53.88s</td>
<td>60min 42.60s</td>
<td>49min 15.59s</td>
</tr>
<tr>
<td>TS + QP</td>
<td>59min 59.40s</td>
<td>59min 47.22s</td>
<td>57min 30.60s</td>
<td>62min 53.88s</td>
<td>53min 12.36s</td>
</tr>
</tbody>
</table>

number of non-dominated portfolios for 3 instances, while in the remaining 2, SD+QP attains the best; in the discrete case exactly the opposite is observed: TS+QP reaches the highest number of non-dominated portfolios for 2 instances, while in the remaining 3 SD+QP attains the best.

A singular phenomenon is furthermore observed: using the continuous case, more feasible portfolios are generated, and the difference between the two cases is consistently high (always around the 10,000 unities, except for the fifth instance, that is the biggest one and cannot be dealt with efficiently by any method). This big difference does not translate itself in a likewise big difference between efficient portfolios, whose value is surprisingly close between the two cases. It is out of the scope of this work to investigate this phenomenon, but it could be taken into account for future extensions.

Computational Time

Table 5.6 shows the computational time required by each strategy on each instance, for both continuous and discrete representations. Computational times of one run are reported, as the other two are comparable.

It turns out that the least time-consuming strategy is FD+QP (except for instance 4-continuous and instance 3-discrete) that happens to be, as we will see afterwards, the least performing. Interestingly the less time-consuming instance is the fifth, being the biggest one. Its low computational times are justified by the fact that for most of the points, it happens to be impossible to find a feasible solution, and the computation ends when the idle iterations expire. As expected, on every instance TS+QP requires more time than SD+QP. Indeed, for each method, most time is dedicated to the Quadratic Programming solver, and the dimension of the constraint matrix to be processed (proportional to the observations, as seen in section 5.1), justifies the large computational amounts requested.
5.5 Experimental results

In the following we are describing the outcome of our experiments, quantifying equal points found through the diverse methods and the coverage measures amongst these methods.

5.5.3 Equal points

Firstly, we determined how many points are the same between the different strategies on the 5 instances. To this extent, we computed the pairwise Mean Equal Points measure, averaging the equal points found using two strategies over three runs for each instance. A clarification has to be made: also comparisons between the same strategies (FD versus FD, SD versus SD and TS versus TS) have been taken into account. Indeed, this is done to assess the robustness of the single strategy, and the mean is not computed averaging nine observations, but just three, so averaging across all the pairwise comparisons.\(^7\) Instance 5 results are not produced as it turns out that there are no equal points amongst strategies. Data presented in the following tables are expressed through their absolute values, rather than in percentages (that will be used for the coverage measure). This choice is to be justified by the fact that this measure counts the intersection between two sets: a point in the set \(A\) belongs to this intersection if there exists the same point in the set \(B\). If more points on the set \(B\) are equals to the \(A\)-set point, just one of them will be considered, the others being negligible also for comparisons with other points (except in the case there are other \(A\)-points with the same value). On the contrary in the next subsection we will compute the coverage measure determining if a point in the set \(A\) is dominated by some points in the set \(B\). If more points on the set \(B\) dominate the \(A\)-set point, just one of them is enough to be considered, but the others can be relevant also for comparisons with other (different valued) points. For this reason, we choose to use, for the coverage measure, the percentage representation.

Continuously compounded returns

We show here the equal points obtained computing returns as to be continuously compunded.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Instance 1</th>
<th>Strategy</th>
<th>Instance 2</th>
<th>Strategy</th>
<th>Instance 3</th>
<th>Strategy</th>
<th>Instance 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FD+QP</td>
<td>40</td>
<td>81</td>
<td>82</td>
<td>FD+QP</td>
<td>149</td>
<td>110</td>
<td>168</td>
</tr>
<tr>
<td>SD+QP</td>
<td>81</td>
<td>147</td>
<td>147</td>
<td>SD+QP</td>
<td>110</td>
<td>167</td>
<td>136</td>
</tr>
<tr>
<td>TS+QP</td>
<td>82</td>
<td>147</td>
<td>145</td>
<td>TS+QP</td>
<td>168</td>
<td>136</td>
<td>207</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FD+QP</td>
<td>118</td>
<td>123</td>
<td>126</td>
<td>FD+QP</td>
<td>114</td>
<td>115</td>
<td>118</td>
</tr>
<tr>
<td>SD+QP</td>
<td>123</td>
<td>138</td>
<td>137</td>
<td>SD+QP</td>
<td>115</td>
<td>144</td>
<td>144</td>
</tr>
<tr>
<td>TS+QP</td>
<td>126</td>
<td>137</td>
<td>138</td>
<td>TS+QP</td>
<td>118</td>
<td>144</td>
<td>149</td>
</tr>
</tbody>
</table>

Table 5.7: Mean Equal Points, continuous

This measure can be conceived as a measure of stability of results, and consequently as a coarse indicator of the method capabilities, since low values indicate that different runs lead to very different results and vice-versa. From the tables it can be argued that FD+QP is the method that has the lowest number of points shared, meaning that different runs may obtain rather

---

\(^7\)Meaning that we will give the intersection between points obtained by the given methods over the indicated runs. In the following, the aggregate \((m_1 - int_1 vs m_2 - int_2)\) means Mean equal points between the \(int_1\)th run of strategy \(m_1\) and the \(int_2\)th run of strategy \(m_2\).
different results. The other two strategies have more stable results, even if this difference is not reflected in standard deviations, whose value over the three strategies are similar.

Note that FD+QP performs worst in terms of equal points w.r.t. other strategies (and w.r.t. its own). Furthermore from that, SD+QP and TS+QP obtain similar performances to each other, except for instance 1, where SD has more consistent results amongst various runs, and on instance 2, where TS is more stable.

### Discretely compounded returns

In principle, the same considerations stated for the continuous case can be tailored to the discrete one. Also in this case we will present data as consisting of their absolute value rather than percentages. FD+QP is the method showing the smallest consistency over the different runs, while SD+QP and TS+QP show comparable results, a part from instance 4, where SD+QP shows more equal points than TS+QP, and with lower standard deviation.

### Table 5.8: Standard Deviation Equal Points, continuous

<table>
<thead>
<tr>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FD+QP</td>
<td>54.211</td>
<td>56.146</td>
<td>56.836</td>
</tr>
<tr>
<td>SD+QP</td>
<td>56.146</td>
<td>1.8856</td>
<td>2.5434</td>
</tr>
<tr>
<td>TS+QP</td>
<td>56.836</td>
<td>2.5434</td>
<td>3.2998</td>
</tr>
<tr>
<td>Instance 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FD+QP</td>
<td>15.577</td>
<td>79.178</td>
<td>17.510</td>
</tr>
<tr>
<td>SD+QP</td>
<td>79.178</td>
<td>94.752</td>
<td>96.677</td>
</tr>
<tr>
<td>TS+QP</td>
<td>17.510</td>
<td>96.677</td>
<td>0.4714</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FD+QP</td>
<td>1.8856</td>
<td>3.0225</td>
<td>2.3570</td>
</tr>
<tr>
<td>SD+QP</td>
<td>3.0225</td>
<td>2.4944</td>
<td>2.1082</td>
</tr>
<tr>
<td>TS+QP</td>
<td>2.3570</td>
<td>2.1082</td>
<td>0.9428</td>
</tr>
<tr>
<td>Instance 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FD+QP</td>
<td>1.6997</td>
<td>3.5277</td>
<td>4.2947</td>
</tr>
<tr>
<td>SD+QP</td>
<td>3.5277</td>
<td>2.1602</td>
<td>1.4907</td>
</tr>
<tr>
<td>TS+QP</td>
<td>4.2947</td>
<td>1.4907</td>
<td>0.8165</td>
</tr>
</tbody>
</table>

### Table 5.9: Mean Equal Points, discrete

Also in this case, there are no equal points on instance 5, so results are not needed.

### 5.5.4 Coverage measure

Coverage measure is needed in order to assess the performance of a multi-objective algorithm respect to another’s. It says how many points on a Pareto frontier obtained by an algorithm are
5.5 Experimental results

<table>
<thead>
<tr>
<th>Instance 1</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>1.4142</td>
<td>2.5142</td>
<td>4.1306</td>
<td></td>
<td>FD+QP</td>
<td>5.7155</td>
<td>5.2868</td>
<td>4.4222</td>
</tr>
<tr>
<td>SD+QP</td>
<td>2.5142</td>
<td>1.6330</td>
<td>3.5416</td>
<td></td>
<td>SD+QP</td>
<td>5.2868</td>
<td>1.6997</td>
<td>2.3094</td>
</tr>
<tr>
<td>TS+QP</td>
<td>4.1306</td>
<td>3.5416</td>
<td>3.6818</td>
<td></td>
<td>TS+QP</td>
<td>4.4222</td>
<td>2.3094</td>
<td>2.4944</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance 2</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>5.7155</td>
<td>5.2868</td>
<td>4.4222</td>
<td></td>
<td>FD+QP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD+QP</td>
<td>5.2868</td>
<td>1.6997</td>
<td>2.3094</td>
<td></td>
<td>SD+QP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TS+QP</td>
<td>4.4222</td>
<td>2.3094</td>
<td>2.4944</td>
<td></td>
<td>TS+QP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance 3</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>3.2666</td>
<td>3.0912</td>
<td>2.7666</td>
<td></td>
<td>FD+QP</td>
<td>2.1602</td>
<td>2.4545</td>
<td>8.9318</td>
</tr>
<tr>
<td>SD+QP</td>
<td>3.0912</td>
<td>1.2474</td>
<td>3.1662</td>
<td></td>
<td>SD+QP</td>
<td>2.4545</td>
<td>2.0548</td>
<td>8.1891</td>
</tr>
<tr>
<td>TS+QP</td>
<td>2.7666</td>
<td>3.1662</td>
<td>1.2472</td>
<td></td>
<td>TS+QP</td>
<td>8.9318</td>
<td>8.1891</td>
<td>7.7889</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance 4</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>10.08</td>
<td>10.48</td>
<td></td>
<td></td>
<td>FD+QP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD+QP</td>
<td>10.08</td>
<td>10.48</td>
<td></td>
<td></td>
<td>SD+QP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TS+QP</td>
<td>10.08</td>
<td>10.48</td>
<td></td>
<td></td>
<td>TS+QP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.11: Standard Deviation Equal Points, discrete

Table 5.10: Standard Deviation Equal Points, discrete

dominated by points belonging to a frontier obtained by another algorithm. Coverage measure can be expressed in absolute values or in percentages. In our experiments, as already stated in the previous section, we chose to use for the coverage measure the percentage representation.

In the following we will show the average percentage coverage measure over the three runs for each pairwise comparison, on the five instances. Note that this measure is far from being symmetrical.

We can notice that over the continuous case TS+QP seems to generally perform better than the SD+QP, yet the percentage differences are rather tiny in this case. On the contrary, when dealing with the discrete case, SD+QP performs better, and differences are higher than in the continuous case. Anyhow is our idea that in both cases, the differences between percentage coverages are too small, and the standard deviation is still too high for us to have a clear perspective about which method performs best. In tables 5.11 and 5.13 the percentage $x_p$ corresponding to methods $MT_1$ (rows) - $MT_2$ (columns) means that $x_p$ percent of the points on the frontier given by $MT_1$ are dominated by points on the frontier given by $MT_2$. Tables 5.12 and 5.14 show the standard deviation of such measures.
<table>
<thead>
<tr>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance 1</td>
<td>FD+QP</td>
<td>10.07</td>
<td>10.11</td>
<td>SD+QP</td>
<td>0.52</td>
<td>0.35</td>
<td>TS+QP</td>
</tr>
<tr>
<td>Instance 2</td>
<td>FD+QP</td>
<td>13.92</td>
<td>14.46</td>
<td>SD+QP</td>
<td>0.54</td>
<td>0.35</td>
<td>TS+QP</td>
</tr>
<tr>
<td>Instance 3</td>
<td>FD+QP</td>
<td>0.01</td>
<td>0.30</td>
<td>SD+QP</td>
<td>0.51</td>
<td>0.61</td>
<td>TS+QP</td>
</tr>
<tr>
<td>Instance 4</td>
<td>FD+QP</td>
<td>0.93</td>
<td>0.27</td>
<td>SD+QP</td>
<td>0.39</td>
<td>0</td>
<td>TS+QP</td>
</tr>
<tr>
<td>Instance 5</td>
<td>FD+QP</td>
<td>40.82</td>
<td>40.82</td>
<td>SD+QP</td>
<td>37.09</td>
<td>33.33</td>
<td>TS+QP</td>
</tr>
</tbody>
</table>

Table 5.12: STD Percentage Coverage Measure, continuous
5.5 Experimental results

<table>
<thead>
<tr>
<th>Instance 1</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>–</td>
<td>1.53</td>
<td>1.65</td>
<td></td>
</tr>
<tr>
<td>SD+QP</td>
<td>0.53</td>
<td>–</td>
<td>0.58</td>
<td>–</td>
</tr>
<tr>
<td>TS+QP</td>
<td>1.45</td>
<td>0.15</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance 2</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>–</td>
<td></td>
<td>11.80</td>
<td>1.75</td>
</tr>
<tr>
<td>SD+QP</td>
<td>0.36</td>
<td>–</td>
<td>0.36</td>
<td>–</td>
</tr>
<tr>
<td>TS+QP</td>
<td>0.47</td>
<td>0.47</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance 3</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>–</td>
<td>0.97</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>SD+QP</td>
<td>0.23</td>
<td>–</td>
<td>0.38</td>
<td>–</td>
</tr>
<tr>
<td>TS+QP</td>
<td>0.07</td>
<td>0.44</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance 4</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>–</td>
<td></td>
<td>0.40</td>
<td>0.79</td>
</tr>
<tr>
<td>SD+QP</td>
<td>2.40</td>
<td>–</td>
<td>2.0</td>
<td>–</td>
</tr>
<tr>
<td>TS+QP</td>
<td>0.47</td>
<td>0.47</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance 5</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>–</td>
<td>8.33</td>
<td>15.00</td>
<td></td>
</tr>
<tr>
<td>SD+QP</td>
<td>26.67</td>
<td>–</td>
<td>6.67</td>
<td>–</td>
</tr>
<tr>
<td>TS+QP</td>
<td>25.00</td>
<td>0</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 5.13: Mean Percentage Coverage Measure, discrete

<table>
<thead>
<tr>
<th>Instance 1</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>–</td>
<td>0.51</td>
<td>0.52</td>
<td></td>
</tr>
<tr>
<td>SD+QP</td>
<td>0.56</td>
<td>–</td>
<td>0.67</td>
<td>–</td>
</tr>
<tr>
<td>TS+QP</td>
<td>0.20</td>
<td>0.22</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance 2</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>–</td>
<td></td>
<td>11.80</td>
<td>1.75</td>
</tr>
<tr>
<td>SD+QP</td>
<td>0.36</td>
<td>–</td>
<td>0.53</td>
<td>–</td>
</tr>
<tr>
<td>TS+QP</td>
<td>0.47</td>
<td>0.47</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance 3</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>–</td>
<td>0.49</td>
<td>0.52</td>
<td></td>
</tr>
<tr>
<td>SD+QP</td>
<td>0.34</td>
<td>–</td>
<td>0.50</td>
<td>–</td>
</tr>
<tr>
<td>TS+QP</td>
<td>0.22</td>
<td>0.57</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance 4</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>–</td>
<td></td>
<td>0.79</td>
<td>0.93</td>
</tr>
<tr>
<td>SD+QP</td>
<td>1.35</td>
<td>–</td>
<td>1.33</td>
<td>–</td>
</tr>
<tr>
<td>TS+QP</td>
<td>10.24</td>
<td>10.53</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance 5</th>
<th>Strategy</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD+QP</td>
<td>–</td>
<td>14.43</td>
<td>14.91</td>
<td></td>
</tr>
<tr>
<td>SD+QP</td>
<td>46.19</td>
<td>–</td>
<td>11.55</td>
<td>–</td>
</tr>
<tr>
<td>TS+QP</td>
<td>50.00</td>
<td>0</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 5.14: STD Percentage Coverage Measure, discrete
At the light of the considerations made so far (computational time, coverage measure, equal points), SD + QP is the recommended strategy: it performs comparably well (when not better) w.r.t to a more sophisticated strategy such as TS+QP, at a cost of a lower computational time. This can be justified by the fact that, a part for instance 5, that is the biggest one, instances at hand are not hard. A sound basin of attraction analysis should be done to justify this assertion, but the approach proposed in chapter 3 cannot be tailored to our present formulation.

### 5.6 Analysis of results

In this section we are going to analyse the experiments with the help of graphical representations. As we recommended the use of SD + QP, we will use figures from experiments ran using that technique, also considering that the shape of figures obtained through the other techniques are not much different considering a visual representation. The same happens when comparing continuously and discretely compounded instances, so for the time being we will stick to the continuous version. Further figures are available upon request.

Tables 5.15, 5.16 and 5.17 show the correlation between our three measures over non-dominated points of the 5 instances, as found with the different strategies. For this analysis only one run has been taken into account. Also here the rank based correlation has been computed (Pearson indicators are reported in Appendix C).

Analysing the correlation values, the first thing that becomes evident is that, in the continuous case, for every strategy, the correlation over the pairs \((TE, \sigma^2)\) and \((TE, r_p)\) obtained on instance 3 are considerably lower than all other instances’ (we are not considering instance 5 throughout this analysis: since portfolios found are few and far between, results for this instance are inconclusive).

This phenomenon over instance 3 clearly indicates to us a situation in which computing the returns in the two different ways (continuous or discrete) leads to different optimisation results: the search space from the 2 cases is different and all strategies found significantly different results. Other than being remarked by correlation values, this can be visually seen by comparing the continuous AUEF (figure 5.6) with the discrete one (figure 5.3).

Indeed, the difference between these values can be understood by plotting two-dimensional figures representing the trade-off between tracking error and return over optimised portfolios for instance 3, in both continuous (figure 5.1) and discrete (figure 5.2) classes. It can be seen...

<table>
<thead>
<tr>
<th>Continuous</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8831751</td>
<td>0.9008139</td>
<td>0.8954103</td>
</tr>
<tr>
<td>2</td>
<td>0.5854104</td>
<td>0.5156715</td>
<td>0.5170243</td>
</tr>
<tr>
<td>3</td>
<td>0.0041203</td>
<td>-0.0087223</td>
<td>-0.0167920</td>
</tr>
<tr>
<td>4</td>
<td>0.4121395</td>
<td>0.4970624</td>
<td>0.4803282</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>0.0997780</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discrete</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9314232</td>
<td>0.9240132</td>
<td>0.9216653</td>
</tr>
<tr>
<td>2</td>
<td>0.6181232</td>
<td>0.6677519</td>
<td>0.6700172</td>
</tr>
<tr>
<td>3</td>
<td>0.9208142</td>
<td>0.9099732</td>
<td>0.9065281</td>
</tr>
<tr>
<td>4</td>
<td>0.9074432</td>
<td>0.8619837</td>
<td>0.9007163</td>
</tr>
<tr>
<td>5</td>
<td>0.9500452</td>
<td>0.5020538</td>
<td>0.8000000</td>
</tr>
</tbody>
</table>

Table 5.15: Optimised portfolios: rank based correlation between TE and \(\sigma^2\)
### 5.6 Analysis of results

#### Table 5.16: Optimised portfolios: rank based correlation between TE and $r_p$

<table>
<thead>
<tr>
<th>Instance</th>
<th>Continuous</th>
<th>Discrete</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FD+QP</td>
<td>SD+QP</td>
</tr>
<tr>
<td>1</td>
<td>0.8999999</td>
<td>0.9134293</td>
</tr>
<tr>
<td>2</td>
<td>0.7265932</td>
<td>0.6958132</td>
</tr>
<tr>
<td>3</td>
<td>0.0416792</td>
<td>0.0341923</td>
</tr>
<tr>
<td>4</td>
<td>0.4272943</td>
<td>0.5183023</td>
</tr>
<tr>
<td>5</td>
<td>-0.2500000</td>
<td>-1</td>
</tr>
</tbody>
</table>

#### Table 5.17: Optimised portfolios: rank based correlation between $r_p$ and $\sigma^2$

<table>
<thead>
<tr>
<th>Instance</th>
<th>Continuous</th>
<th>Discrete</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FD+QP</td>
<td>SD+QP</td>
</tr>
<tr>
<td>1</td>
<td>0.9966293</td>
<td>0.9978231</td>
</tr>
<tr>
<td>2</td>
<td>0.9648102</td>
<td>0.9516932</td>
</tr>
<tr>
<td>3</td>
<td>0.9971649</td>
<td>0.9964920</td>
</tr>
<tr>
<td>4</td>
<td>0.9999998</td>
<td>0.9985629</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.16: Optimised portfolios: rank based correlation between TE and $r_p$

Table 5.17: Optimised portfolios: rank based correlation between $r_p$ and $\sigma^2$
how, for each level of tracking error, there exist many points with different return values (this is the trend generally followed in every instance) and this phenomenon is common to both figures. Nevertheless, in the continuous version, there are more low-return points associated to high-tracking error values, leading to a lower correlation value for that instance. Interestingly, this phenomenon happens when having an instance size for which the obtained efficient portfolios are of similar cardinality with the two approaches, as seen on tables 5.4 and 5.5. Again, looking at that tables, we can see how using the continuous representation we obtain more optimised portfolios on instances 2 and 4, while the discrete representation obtains more portfolios over the first instance, that is the smallest (and easiest, at least considering computational times\(^8\)). Interestingly, the third instance is the one for which the found optimal portfolios are of a comparable cardinality for both instance classes.

Indeed, looking at the figures, we can realise how in both instance classes the efficient points are step-wise distributed in the first part of the frontier, while they collapse into a more linear structure after a given tracking error threshold (note that the scale is different for each figure). Points over that linear segment are, as argued by computational results, the most difficult to be found\(^9\), and for this reason we say that our approaches attain the best results over the first two instances, where this linear trend is perceivable on the upper part of the frontier. In this context, we receive another hint to suggest that instance 1 is the easiest: our approaches were

\(^8\)Indeed, in the discrete case, SD+QP requires more time for solving I4 than for solving I2 and I3. The same holds with TS+QP, that also requires less time to solve I4 over the discrete instances, but these cases are the exceptions rather than the average behaviour.

\(^9\)Such points have been found on instances 1 and 2, considered to be the easiest instances, while they have been hardly detected in others instances.
Figure 5.2: Projection of the nondominated points in the three objectives problem onto the two-dimensional trade-off between tracking error and return. Instance 3, discrete.

Figure 5.3: AUEF: SD+QP, Instance 3, Discretely computed returns
able to find non-dominated portfolio with higher Tracking Error on this instance (and, with smaller magnitude, on instance 2); in the discrete case, we were able to find tracking error as great as 60 on the first instance (30 on the second), while it does not reach the value of 10 on the third and fourth (the same holds on continuous instances). Again, it turns out that high values for the tracking error refer to points on the linear structure over the frontier.

The more efficient portfolios located upon this linear structure are found, the higher \((TE, \sigma^2)\) and \((TE, r_p)\) are. For this reason the efficient frontier for Instance 1 (fig. 5.4) looks more like a line through the three-dimensional surface, while the most sparse is Instance 3, where no point over this structure is found.

Other points to be noticed are the differences between randomly generated and optimised portfolio’s correlation values. Firstly it can be seen how correlations over \((\sigma^2, r_p)\) and \((TE, r_p)\) assume low values over random portfolios, while their values show a strong correlation over optimised portfolios (also with the exception of instance 3 \((TE, r_p)\) on the continuous case, while in the discrete this correlation has higher values).
5.6 Analysis of results

As also intuitively valid, it can be seen that, for every instance, the stricter the Index-Tracking constraint, the less points are reachable on the efficient frontier. This also hold for the discrete case. The figures relative to discretely-computed-returns-instances-Pareto-fronts are provided in the Appendix to this work. We end this discussion about results showing the bi-dimensional representation of Pareto frontiers, providing for each instance two bi-dimensional plots, the first showing the trade-off between tracking error and risk and the second showing the trade-off between tracking error and return (trade-off between risk and return is not provided, as for each instance results in portions of efficient frontiers overlapping to each others). We can see how a linear trend (at least from a certain point on) over both sets is perceivable on instance 1 only. The behaviours over other instances (other than instance 3, discussed before) are comparable.

We can end saying that, after the optimisation phase, portfolios that are good w.r.t. return are not performing w.r.t. the Tracking Error: this can be seen by the correlation values between these two measures over optimised portfolios, whilst the correlation over random portfolios is negligible. The same high correlations are shown comparing variance with Index Tracking Error, though it means that portfolios that are good w.r.t. Tracking Error are also good w.r.t. variance (note that in this case a correlation is perceivable also over random portfolios). This indicates a situation in which the choice of a multi-objective shows features that are in conflict with each other, and an external criterion must be used in order to express preferences.

Figure 5.5: AUEF: SD+QP, Instance 2, Continuously computed returns.
Figure 5.6: AUEF: SD+QP, Instance 3, Continuously computed returns

Figure 5.7: AUEF: SD+QP, Instance 4, Continuously computed returns
5.6 Analysis of results

Figure 5.8: Projection of the nondominated points in the three objectives problem onto the two-dimensional trade-off between tracking error and return. 4 instances, continuous.

Figure 5.9: Projection of the nondominated points in the three objectives problem onto the two-dimensional trade-off between tracking error and variance. 4 instances, continuous.
5.7 Conclusion

In this chapter we have introduced a hybrid metaheuristic approach to solve a multi-objective Portfolio Selection Problem combining Mean Variance with Index Tracking. After defining three formulations, only one of them has been taken into account due to computational limits and unfeasibility of the other models. The same three strategies introduced in chapter 3 (FD+QP, SD+QP, TS+QP) have been used in the experimental phase for both discretely and continuously compounded return’s instances. Results show that SD + QP is the best solver, and the behaviour of the proposed strategies does not change when comparing discretely and continuously compounded return’s instances.
Chapter 6

Epilogue

Conclusions

In this thesis we have addressed the application of metaheuristic approaches to the Portfolio Selection Problem. This problem is no doubt amongst the most studied topics in finance, and there exists a broad literature about metaheuristic approaches deal with it. Three formulations have been tackled, the first being the standard one (Mean Variance analysis), the second an alternative and somewhat opposite approach (Index Tracking) and the third, a multicriteria problem combining the two previously introduced versions. A hybrid metaheuristic has been proposed to tackle Portfolio Selection Problems, combining Local Search with Quadratic Programming.

For each problem, our work has been tackling the following aspects:

• Outline of the model(s);
• Outline of the state of the art metaheuristics;
• Description of the introduced algorithms;
• Experimental Analysis;

This general scheme has been tailored to specific features of problems at hand. For instance, regarding the Index Tracking Problem, the last two points could not be implemented, as the nature of the problem makes it impossible to be solved with the hybrid metaheuristic we devised; regarding the multi-objective choice, the description of the state-of-the-art metaheuristics is missing as there are no approaches in the literature. The original contributions of the thesis are summarised in the following section.

Contributions of the thesis

The original contributions of the thesis are the following:

• The introduction of a classification of Portfolio Selection models (as tackled by metaheuristic approaches) and of metaheuristic strategies for the Portfolio Selection Problem. This contribution is the outcome of a joint work with Andrea Roli. The ideas, as expressed in this thesis, have been based on the work *Metaheuristics for the Portfolio Selection Problem* appeared on the *International Journal of Operations Research* [40]. The main focus of this contribution has been to classify and analyse a literature that to some extent lacks of coherence: in this way we provided the readers with a sound classification and with a
piece of knowledge that can be re-used. One of the key parts of this approach has been the separation between the model and the strategy used, so as not to hinder pieces of knowledge about strategies originally thought for a specific model, to be used to tackle formulation different from the one they were proposed for (and vice-versa).

- The introduction of a classification of Index Tracking models (as tackled by metaheuristic approaches) and of metaheuristic strategies for the Index Tracking Problem. This contribution is the outcome of a joint work with Dietmar Maringer and it is based on the work *Metaheuristics for the Index Tracking Problem*, accepted for publication as a book chapter to appear in *Metaheuristics for Service Industry* [39]. The same considerations expressed at the previous point hold.

- The introduction of a hybrid algorithm combining metaheuristic approaches and Quadratic Programming to solve the Mean Variance Portfolio Selection, along with an experimental analysis over a set of known instances, a comparison with results from the literature and a basin of attraction analysis to investigate instance hardness. This contribution is based on a joint work with Luca Di Gaspero, Andrea Roli and Andrea Schaerf presented at The Fourth International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR 07) [37] and currently being examined for a Journal Publication. In this approach, the particular structure of the problem, to be split on a decision problem and an optimisation one, is used to develop a master-slave approach that resorts to local search for the decision problem and to Quadratic Programming for the optimisation one. This decomposition performs far better than monolithic approaches, as witnessed by a Basin Of Attraction analysis on both monolithic and hybrid strategies.

- The application of the concepts used to develop the previous point (hybrid algorithm for the Portfolio Selection Problem) to implement a hybrid algorithm to solve a three-objective extension of the Portfolio Selection Problem, combining the two major problems discussed in this thesis (Portfolio Selection Problem and Index Tracking). This formulation has been suggested in the literature but experimental results have not been provided and investigated. To the best of our knowledge, this thesis is the first work to provide an experimental analysis for the Multi Objective Portfolio Selection Problem.

**Perspectives**

The work of this thesis has been concerned with the application of metaheuristic algorithms to the Portfolio Selection Problem. In our application, we decided to tackle a risk-return formulation, hence introducing (at most) two measures of risk and one of returns, but several measures can be introduced for the same objectives, so determining several multi-objective frameworks to be tackled by meta-heuristics. Indeed, it is out of our scope to tackle other formulations, just switching amongst objective functions: we introduced and tested an approach to be used for optimised portfolio choices, in which the user has to make his own choices w.r.t. some practical details, objective function being the most important, but it is out of the scope of our work to investigate a systematic parameters choice. Our next focus is to investigate in detail structural properties of the problem at hand, together with extension of our approach (and not alternative approaches).

First of all, a detailed search space analysis has to be performed over the three-objective framework we deal with: we have shown that the hybrid basin of attraction (BOA) is much simpler than the monolithic one, and we want to make the same comparison over the three
objective surface: such a work will give us proper insight on the structure of multi-objective PSP, so far unknown as un-investigated. It is furthermore our idea that conclusions obtained implementing the TE measure we used will hold also using other measures.

All empirical work we have done so far relies on non-negativity constraint over variables: $x_s$ (or $w_s$) value must be positive. Indeed, we have already mentioned the case of assets to be sold short, and this extension of the problem has absolutely to be taken into account for future achievements: nowadays, short selling are referred to as a common investment strategy, so there is no point in forbidding them in the formulation. One of the hurdles to taking short selling into account is the diverse way they are allowed in different countries regulation systems; Nevertheless, the objective (or cost) function should be the same no matter the system in use, and different constraints should be used to model the actual law-regulation system at hand.

Short selling presents lot of aspects to be modelled: for instance, USA reg T can be introduced

$$\sum_{i=1}^{n} |x_i| \leq 2 \quad (6.1)$$

but it can easily be circumvented, i.e., taking out a loan at one bank, and taking the money to his broker claiming it is equity. Furthermore, an investor can also borrow from himself, but this trigger problems when considering the following aspects: the investor borrows stocks and sells them over the market: the proceeds of a short selling must be invested in cash equivalent [73]; the borrower does not get interest on the short selling proceeds if it is a small-investor, otherwise he gets a part of interests (short-rebate). Now, these technical aspects are already difficult to be taken into account as their application (or not) depends on investor’s market power, and even more difficult if the investor borrows from himself. Anyhow, an optimisation procedure should be used just as an advice, so it should have a general explanation power: it is clear that all practical aspects cannot be included in the formulation, but this does not hinder the validity of the approach for general purposes. First, when introducing short selling, an analysis must be made over the unconstrained case, just using an exact solver (i.e., QP) taking into account different formulations: in detail, differences between assets-only universe and assets-cash universe must be taken into account. The assets-cash models it is caused by the introduction of others parameters into the analysis, such as the short rebate and the cash-equivalent collateral. So, first just optimising a risk measure as we did before, erasing the non-negativity constraint and tackling the asset-only universe, has to be performed. Optimising short-selling PSP without extra-constraints leads to defining a frontier to be used as upper bound for further analysis, as the risk-return combinations we will obtain in this case are thought to dominate the ones we will obtain introducing extra-features, which we can define a loss measure with reference to. Then different practical aspects (i.e., collateral and short rebate) have to be taken into account. Only after this preliminary operation, aimed in understanding the different impact of these aspects over the unconstrained case, we can introduce constraints as the ones we introduced in our analysis (cardinality, floor-ceiling, pre-assignments etc.) to define a proper optimisation setting and to compare it with the long-only framework. To this extent, a BOA analysis has to be done to understand the structure of the instance, with particular attention to variations occurred when optimising w.r.t. different minimum required returns. It will be possible to compare all work done so far with the formulation obtained just introducing short selling, leading to a better understanding of the diverse formulations at hand.

A further direction can be investigated just considering the Pareto nature of the Efficient Frontier. In the common approach, a rational investor should use an external criterion in order
to choose a portfolio out of the set at hand. But, is there any guarantee that the portfolios over the frontier are realistically non-dominated? We want to check whether others dominate some of the point in the frontier in a probabilistic sense. This is motivated by the fact that the two objectives we are taking into account are not unrelated quantities, but rather the mean and the variance of the same random variable: asset returns. In this sense, it is easy to devise a procedure able to show that two points are not non-dominated, and an analytical study will be carried out in order to understand if portfolios over the Pareto frontier are truly non dominated or not. As already stated, in a bi-dimensional space of risk and return, a solution \( s \) is said to be efficient (Pareto-optimal) if there is no other solution \( s_1 \) such that \( \text{return}(s_1) > \text{return}(s) \) and \( \text{risk}(s_1) \leq \text{risk}(s) \) or \( \text{return}(s_1) \geq \text{return}(s) \) and \( \text{risk}(s_1) < \text{risk}(s) \); furthermore, as seen before, minimising the variance for several levels of return leads us to frontier composed of (purportedly) non dominated points.

Let us take two portfolios (points) over this frontier and let us denote the first point as \( h \) and the second as \( l \). We must be aware that, if we stick to the Markowitz model, these points are computed from historical data, but we can conceive them as states representing portfolio’s future outcome: this outcome could be described as a normal distribution with a given mean (expected return) and a given variance (risk), so we can describe the two points as follows

\[
\begin{align*}
h & \sim N(\mu_h, \sigma_h^2) \\
l & \sim N(\mu_l, \sigma_l^2)
\end{align*}
\] (6.2) (6.3)

Note that is not necessary to assume normality of return distribution in order to carry on with our discussion, as the Chebyshev inequality holds for any distribution. Anyhow, following the well-established assumption, we will think of returns as normally distributed. Note that we will try to devise a method just relying on statistical and analytical procedures, so without taking into account more complex portfolio theory such as utility analysis and statistical dominance [161, 6]. In terms of confidence intervals (see tab 6.1), we can draw that the true return of portfolio \( l \) is supposed to be in the interval

\[
[\mu_l - k \cdot \sigma_l, \mu_l + k \cdot \sigma_l]
\] (6.4)

with Confidence \( (k) \) likelihood. For example, the true return of the portfolio has 99.7% likelihood to stay in the interval

\[
[\mu_l - 3 \cdot \sigma_l, \mu_l + 3 \cdot \sigma_l]
\] (6.5)

Given that, we can introduce a new normally distributed variable \( d \), defined as the difference between the two portfolios and represented as follows (for sake of simplicity the two portfolios are thought to be independent):

\[
d \sim N(\mu_h - \mu_l, \sigma_h^2 + \sigma_l^2)
\] (6.6)

<table>
<thead>
<tr>
<th>( k )</th>
<th>Confidence ( (k) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>84.2 %</td>
</tr>
<tr>
<td>2</td>
<td>97.6 %</td>
</tr>
<tr>
<td>3</td>
<td>99.7 %</td>
</tr>
</tbody>
</table>

Table 6.1: Percentiles of a distribution

Now we can state that portfolio \( h \) statistically dominates portfolio \( l \) at confidence level \( \text{Confidence}(k) \) if
\[ \mu_d - k \cdot \sigma_d \geq 0 \implies \mu_d \geq k \cdot \sigma_d \]  

This becomes

\[ \mu_h - \mu_l \geq k \cdot \sqrt{\sigma_h^2 + \sigma_l^2} \]

\[ \mu_h \geq \mu_l + k \cdot \sqrt{\sigma_h^2 + \sigma_l^2} \]

and eventually

\[ \mu_h - \mu_l \geq k \cdot \sqrt{\sigma_h^2 + \sigma_l^2} \]  

This means that, picking two portfolios \( h \) and \( l \) over the Mean Variance Space, we can state that portfolio \( h \) probabilistically dominates \( l \) if equation 6.10 holds. At a first glance, dominated portfolios are not likely to be found out of Mean-Variance efficient frontiers, but this is not hindered by theoretical assumption of the model. Indeed, confidence intervals tend to be wider as the portfolio return increase, so it is likely that the confidence interval for a high return portfolio will encompass between its range a smaller-return-portfolio confidence interval, even for tiny \( k \)-values. The main objection is that this can hold for the Mean Variance Problem, when we have just risky assets at hand. But what if we deal with a framework including risk-free assets? In this case the efficient frontier would be a straight line, whose slope depends on the risk-free rate. Furthermore, within the Brennan Framework the frontier will be composed of two different straight lines with different slopes (other than a chunk of the Mean Variance frontier). In this framework, the possibility of having dominated portions of the frontier seems to be more realistic. Experiments will be designed to understand if this is the case or not.

The previously introduced extension tries to introduce a generalisation that still relies on mean and variance estimates, but introduces uncertainty treating them as intervals in their values space rather than points; nevertheless, its main shortcoming is that it can be only dealt with when taking into account frontiers, as the goal is to compare two portfolios already thought to be non-dominated. Obviously, when the optimisation is concerned only with finding one single portfolio, i.e., when we consider Sharpe-Ratio maximisation or Index Tracking, the specific feature of the problem makes this approach unfeasible, so we need to device another approach to model uncertainty by another account. To this extent, the use of Estimated Based Local Search will be pursued: this is a technique based on the idea that in a wide range of combinatorial optimization problems, such as vehicle routing, resource allocation and scheduling, only a part of the information needed for assessing the cost of a solution is available. In order to solve these problems, researchers and analysts often consider a framework in which the cost of each solution is a random variable and the goal is finding a solution that minimises some statistics of the latter. In this particular framework the optimization is performed with respect to the expectation. Two types of solution techniques have been discussed inside this framework: first, using a well established analytical development for computing the expectation; and latter, using Monte Carlo simulation to estimate the expectation. This latter technique is referred to as \textit{empirical estimation}.

The empirical estimation approach for stochastic combinatorial optimization falls into the so-called sample average approximation: the given stochastic optimization problem is transformed into a so-called sample average optimization problem, which is obtained by considering several samples of the random variable and by approximating the cost of a solution with a sample average function. In this framework, our aim will be to design a local search algorithm that adopts the empirical estimation approach to tackle the Index Tracking problem. In this framework, the estimation approach can be used under the assumption that both index’s and
portfolios’ return can be described as a random variable and the first two moments (mean and variance) suffice to explain the behaviour of an index. Provided this, the index tracking objective will be given by a metric of the difference between samples of portfolio and index returns.

Our conjecture is that this approach will lead to robust out-of-sample results. To this goal, several ways can be taken for the sampling-optimisation assessment: in the canonical Index Tracking Problem, optimisation is performed over historical data, for both portfolio’s and index’s return, as well as the testing (evaluation) phase. Introducing estimation, we have a choice (between historical and sampled returns) to be done, in turn, for portfolio’s and index’s returns in the optimisation phase, whilst the testing phase still has to be done over historical data. This way, we can conduct experiments over four class of instances: historical assets returns and estimated index return; estimated assets returns and historical index return; estimated returns for both assets and index; and historical returns for both assets and index (this last instance being the Index Tracking in canonical form). Comparisons have to be made amongst these instances, to understand which class performs better and why.

Lastly, other methods can be applied to Portfolio Selection Problem. A wide application and comparison between methods will be impossible and therefore out of the scope of our future work. Nevertheless, it is our idea to devise a Neural Network procedure to deal with the problem. Indeed, Neural Networks have already been used to deal with optimised portfolio choices, but our contribution will consist in applying networks able to grasp the dynamical features of series (returns) at hand to tackle Index Tracking Problem: this will be done by means of Recurrent Neural Networks, that have not been used in this context so far. Furthermore, little effort has been made to pay attention to pre-processing of data and Neural Network tuning when applied to Portfolio Selection: these aspects will also be investigated.
Appendices
Appendix A

Algorithms for optimised portfolio choice

In this Appendix we are outlining non-metaheuristic methods that have been proposed in the literature for both the Portfolio Selection Problem (in Appendix A.1) and Index Tracking (in Appendix A.3). Furthermore, in Appendix A.2 the critical line algorithm, due to its historical importance, will be outlined.

A.1 Algorithms for the portfolio selection problem

Since the introduction of the Mean Variance Model in 1952 a great amount of work has been done to develop algorithms for the Portfolio Selection Problem. In this section we want to provide the reader with the main algorithms proposed during this broad period. We are not aiming to give a systematic overview, but rather an insight about the most common approaches.

The Mean-Variance model can be formulated either as a variance minimisation or as a return maximisation. Nowadays, the minimum variance formulation seems to be the standard one, but there are several approaches dealing with the return maximisation, justified by the impossibility to deal with some classes of constraints (or integer variables) when minimising a quadratic function in a, say, Quadratic Programming framework. Algorithms for the PSP can be classified as Complete or Incomplete algorithms: algorithms belonging to the first class provide a solution certified to be optimum, while the ones belonging to the second class don’t.

The first algorithm developed for the PSP is an exact one and is referred to as Critical Line Algorithm[112, 115, 114]. For the historical importance it has, it will be explained in detail in the Appendix. Another exact method has been provided by [162], introducing the simplex method for Quadratic Programming. Elton, Gruber and Padberg [50], extending a previous work[49] about optimisation under the Tobin framework under specific assumptions about the covariance matrix and the constraint set, introduce an exact criterion to trace out the Mean Variance efficient frontier. All these algorithms minimise variance and have nice formal properties, but they cannot tackle even single constraints added to the formulation (i.e., floor constraint in [50]). Later exact algorithms succeeded in dealing with more complex constraint sets. For instance, Konno and Yamamoto[88] solve a portfolio optimisation problem under concave and piecewise constant transaction cost, formulated as a nonconcave maximization problem under linear constraints using absolute deviation as a measure of risk, and solving it by a global-optimisation branch and bound algorithm. Another exact algorithm is provided by Mansini e Speranza[103], also in order to encompass transaction costs, together with integer rounds. The branch and cut algorithm by Bienstock[13] shows good performances in minimising
A. Algorithms for optimised portfolio choice

variance over a class of real-world problems, still computational time seems to be too sensitive to the constraint set (i.e., cardinality constraint). An extension of this approach has been made by Bertsimas and Shioda[12], but their approach seems to be much problem-specific, based on a pivoting algorithm[93], that takes advantage from the special structure of the problem. A general branch-and-bound algorithm for mixed integer nonlinear programs has been presented by Borchers and Mitchell[18], which is well suited to solve the Portfolio Selection being apt to be solved by it.

Lagrangian relaxation are also introduced and dealt with exact methods: Li, Sun and Wang[94] propose a convergent Lagrangian and contour-domain cut method to solve a general class of discrete-feature constrained portfolio selection problems by exploiting some prominent features of the mean–variance PSP. A similar approach has been followed by [142]

A widely used approach is the Interior Point algorithm: it has been introduced by Takehara [155] to solve three variants of the PSP (mean-variance, index tracking and the multiple factor) and by Lee and Mitchell[92] within a parallel branch-and-bound framework to solve nonlinear mixed-integer programming problems.

Nevertheless, exact algorithms encounter troubles when dealing with large instances, hence the need for approximated and heuristic (incomplete) algorithms. One special class of such algorithms consists of methods that simplify the covariance matrix using either single[139] or multi[48, 126] factor models. The need for such kind of analysis was in that using historical covariance matrix could lead problems when observed stocks were more than observations (daily, weekly, monthly or whatever): in this case the matrix becomes singular, preventing algorithms requiring the covariance matrix to be singular (i.e., critical line) to operate. Factor models rely on asset returns being driven by other criteria. For example, the market model is usually addressed in single factor models, and it can be shown that the covariance matrix, when introducing this analysis, becomes diagonal \(^1\). When introducing multi factor analysis, influences can be either common or asset(firm) specific. Also Sharpe[141] developed an algorithm based on a covariance matrix simplified by means of a piecewise linear approximation, in order to maximise returns.

Other incomplete algorithms rely on linear programming and use other risk measures. They have been outlined in more detail in section 2.4.

A.2 The critical line algorithm

The critical line algorithm has been proposed by Markowitz to solve his model. In order to give a brief outline we firstly need to introduce some preliminary concepts. Note that it is out of the scope of this work to give an exhaustive review of the algorithm's foundations and applications: what we emphasise in this section is to have the reader aware of the existence of an analytical method to solve the unconstrained PSP.

Indeed, this method happens to have just a theoretical concern, as it can handle only a far too restricted class of constraints we defined before. Nevertheless, its historical importance makes it necessary to be dealt with in this thesis. Interested readers are forwarded to [114, 112, 135] for getting more insight about this subject.

Before introducing the algorithm we need to introduce some preliminary concepts about affine models: the reason will be clear in the following.

\(^1\)The same happens with scenario analysis.
A.2 The critical line algorithm

A.2.1 Affine models and portfolio selection problem

We define a set to be an affine set \( S \in \mathbb{R}^n \) if and only if
\[
X, Y \in S \implies tX + (1 - t)Y \in S \quad \forall t \in \mathbb{R} \tag{A.1}
\]

When the affine set also includes a conventional \( X = 0 \) point, it is called a linear subspace. In the context of a PSP, an affine model is conceived as a model having only equalities \((Ax = b)\) as constraints. The most common instance of this model class is the Black model, imposing \( \sum_{i=1}^{n} x_i = 1 \) as the sole constraint (thus allowing short selling). Feasible solutions to affine models (if any) consist of an affine set (or, as a special case, of a single point).

An interesting affine set’s property is itself to be a cone, defined as follows:

A set \( S \subset \mathbb{R}^n \) is a cone, \( X_0 \) being its vertex, if \( \forall y \in S \ (y \neq X_0) \), the ray connecting \( X_0 \) and \( y \) lies in \( S \).

Note that it is not necessary \( X_0 \) to stay in \( S \): when it does not belong to the set, \( S \) will be referred to as a cone with vertex deleted.

The property of being a cone, satisfied for instance by the Black Model, is not necessary for the remainder of the work, and will be not subject of further explanation.

Now, let’s try to solve a PSP with affine constraint set, so that our problem would be
\[
\min F(x) \quad s.t \tag{A.2}
\]
\[
g_i(x) = b_i \quad i = 1 \ldots m \tag{A.3}
\]

It is possible to define a Lagrangian formula from the previous, just minimising
\[
L(x) = F(x) - \sum_{i=1}^{m} \lambda_i g_i(X) \tag{A.4}
\]

It is possible to demonstrate that if a solution minimising \( L(x) \) also satisfies \( g_i(x) = b_i \), that very same solution minimises \( F(x) \).

We can write an affine portfolio selection problem in the form of equation (A.2) as follows\(^2\)
\[
\min V = X'CX \quad s.t \tag{A.5}
\]
\[
Ax = b \tag{A.6}
\]
\[
\mu'X = E \tag{A.7}
\]

where \( V = X'CX \) indicates an homogeneous quadratic form
\[
\sigma_{1,1}x_1^2 + \sigma_{1,2}x_1x_2 + \cdots + \sigma_{1,n}x_1x_n + \cdots \tag{A.8}
\]
\[
\sigma_{n,1}x_nx_1 + \sigma_{n,2}x_nx_2 + \cdots + \sigma_{n,n}x_n^2 \tag{A.9}
\]
\[
Ax = b \] indicates a system of \( m \) equations in \( n \) variables

\(^2\)In the following we are using matrix notation for ease of computation. This notation will be discarded for the remainder of the work.
\begin{align*}
a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n &= b_1 \\
\cdots \\
a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n &= b_n
\end{align*}

(A.11) \quad (A.12) \quad (A.13)

and \( \mu x = E \) represents the weighted sum of the variables \( E = \mu_1 x_1 + \mu_2 x_2 + \cdots + \mu_n x_n \).

It is possible to simplify the formula using \( V_2 \) instead of \( V \) and formulating \( L(x) \) as

\[
L(x) = \frac{1}{2} X'CX + \lambda'AX - \lambda_E \mu'X
\]

(A.14)

We recall that the point obtained minimising \( L(x) \) and satisfying \( g_i(x) = b_i \) also minimises equation (A.2). This holds for some fixed values of \( \lambda_i \), but this doesn’t help us in finding the proper \( \lambda_i \) values that make both \( L(x) \) and \( g(x) \) are minimised. The first step for understanding the Markowitz algorithm is to find such \( \lambda_i \) values.

A necessary condition for \( L(x) \) to be minimised is to pose \( \frac{\partial L}{\partial x} = 0 \), which in our case becomes

\[
CX + A'\lambda - \lambda_E \mu = 0
\]

(A.15)

or

\[
(CA'\mu)
\begin{pmatrix}
  X \\
  \lambda \\
  -\lambda_E
\end{pmatrix} = 0
\]

(A.16)

The left term of equation (A.16) will be referred to as \( \eta \). As we want our solution to satisfy either \( (Ax = b) \) or \( (\mu'X = E) \), we can add these constraints to equation (A.16), obtaining the following:

\[
\begin{pmatrix}
  C & A' & \mu \\
  A & 0 & 0 \\
  \mu' & 0 & 0
\end{pmatrix}
\begin{pmatrix}
  X \\
  \lambda \\
  -\lambda_E
\end{pmatrix} =
\begin{pmatrix}
  0 \\
  b \\
  E
\end{pmatrix}
\]

(A.17)

Solving equation (A.17) for \( x_s \) and \( \lambda_s \) we are sure to find \( X \) minimising \( L \) for the given (just found) \( \lambda_s \) subject to \( (Ax = b) \) and \( (\mu' = E) \). Assuming

\[
M = \begin{pmatrix}
  C & A' & \mu \\
  A & 0 & 0 \\
  \mu' & 0 & 0
\end{pmatrix}
\]

(A.18)

to be non-singular\(^3\) (meaning that an \( M^{-1} \) matrix exists so that \( M \cdot M^{-1} = I \)), the values of \( X \) and \( \lambda \) which minimise \( L \) are given by

\[
\begin{pmatrix}
  X \\
  \lambda \\
  -\lambda_E
\end{pmatrix} = M^{-1} \cdot 
\begin{pmatrix}
  0 \\
  b \\
  E
\end{pmatrix}
\]

(A.19)

Furthermore, assuming

\[
\overline{M} = \begin{pmatrix}
  C & A \\
  A & 0
\end{pmatrix}^T
\]

(A.20)

to be not singular and rewriting (A.17) as

\[
\begin{pmatrix}
  C & A \\
  A & 0
\end{pmatrix}
\begin{pmatrix}
  X \\
  \lambda
\end{pmatrix} = 
\begin{pmatrix}
  0 \\
  b
\end{pmatrix} + \lambda_E \begin{pmatrix}
  \mu \\
  0
\end{pmatrix}
\]

(A.21)

\(^3\)For an analytical development, see Markowitz-Todd.
we can find out that
\[
\begin{pmatrix}
X \\
\lambda
\end{pmatrix}
= M^{-1}\begin{pmatrix}
0 \\
b
\end{pmatrix} + \lambda E \begin{pmatrix}
\mu \\
0
\end{pmatrix}
\] (A.23)
provides us with a vector \((X, \lambda)\), solving (A.20) for given \(\lambda_E\). This is what we call critical line, that shows the relationship between \(X\) and \(\lambda_E\), to be written as
\[
\begin{pmatrix}
X \\
\lambda
\end{pmatrix}
= \alpha + \beta \lambda_E
\] (A.24)
where \(\alpha\) and \(\beta\) depend on the \(IN\) and \(OUT\) chosen sets (see later). This finding is of the utmost importance because it is directly related to the basic Markowitz algorithm, based upon the critical lines. The importance we advocate to affine models lies in that the Mean-Variance efficient set is composed of portfolios belonging to critical lines obtained solving affine models related to the original problem by forcing some of the \(x\) variables to be 0 and removing short-selling prohibitions (so allowing \(x\) to be \(\leq 0\)). Indeed, it has been shown that every (feasible) PSP model has a (piecewise linear) complete and non-redundant\(^4\) set of efficient portfolios composed of critical lines segments.

### A.2.2 The critical line algorithm

The analytical procedure described above could help us in solving Mean-Variance PSP, but it presents two strong cons:

1. It requires all the variables (stocks) to be present in the final solution, as the procedure applies to all asset whose information is reported in the \(M\) matrix, so we should compute an \(M\) matrix for each non-null subset of \(N\) and eventually finding the best one.

2. It allows short-sellings, which is generally not allowed by most works in the literature (see [40] for an overview).

Even taking into account just the first of the aforementioned shortcomings, it turns out that we should define an \(M\) matrix for every possible subset of the original asset universe, meaning that even for a small sample (for example when taking into account such a small benchmark as the Hang Sen, composed of 31 assets) the possible combinations are no less than \(2^{31} - 1\): it is clear that this makes the problem, when not impossible to deal with, at least cumbersome. For this reason, a more refined algorithm, the true critical lines algorithm has been introduced by [112], with further refinements. In our discussion, we will rely on a later description given in [73].

The algorithm relies on the previous analytical solution for affine sets, but instead of creating all the possible candidate solutions, it uses subsets in a very ante-litteram local-search fashion. Let’s partition the universe \(N\) in two complimentary sets \(IN\) and \(OUT\), so that \(IN\) will be composed of a subset of numbers between 1 and \(n\) and \(OUT\) of numbers between 1 and \(n\) not belonging to \(IN\). Let’s further define \(C_{IN}\) as the \(C\) matrix with \(i\)th row and column replaced by \(e_i'\) and \(e_i\) respectively\(^5\) for each \(i \in OUT\). Accordingly, let’s define \(A_{IN}\) as the \(A\) matrix.

\(^{4}\)The attribute complete means that there exists a portfolio for each EV efficient combination, whereas non redundant means that there exists only one portfolio providing the efficient EV combination.

\(^{5}\)\(e_i\) is defined as a vector all of whom’s members are 0 but its \(i\)th (to be set to 1).
whose $ith$ column is replaced by a 0 vector and $\mu_{IN}$ as the $\mu$ vector whose $ith$ component is replaced by 0, still for each $i \in OUT$. The development introduced in the previous subsection still holds when using $C_{IN}, A_{IN}$ and $\mu_{IN}$ where

$$M_{IN} = \begin{pmatrix} C_{IN} & A'_{IN} & \mu_{IN} \\ A_{IN} & 0 & 0 \\ \mu_{IN} & 0 & 0 \end{pmatrix}$$

(A.25)

is non-singular. Indeed, portions of critical lines can happen to have singular $M_{IN}$, but the algorithm just deal with non-singular ones. For a given critical line, we will have

$$x_i = 0 \quad \forall i \in OUT$$

(A.26)

furthermore equations (A.22) and (A.23) still hold using $M_{IN}$. Now, after stating that even $\eta$ has a linear relationship with $\lambda_E$

$$\eta = \gamma_{IN} + \delta_{IN} \lambda_E$$

(A.27)

, we find that equation A.22 implies

$$\eta = 0 \quad \forall i \in IN$$

(A.28)

and we can deduct (after equations (A.26) and (A.28)) that if a point on the critical line satisfies

$$x_i \geq 0 \quad \forall i \in IN$$

(A.29)

$$\eta_i \geq 0 \quad \forall i \in OUT$$

(A.30)

$$\lambda_E > 0$$

(A.31)

it is an efficient point, due to Kuhn-Tucker conditions. Generally in such critical lines there exists an interval whose points are efficient, referred to as efficient segment. In order to reduce the aforementioned complexity, the critical lines algorithm search recursively efficient portfolios lowering the given $\lambda_E$. We can describe the algorithm using the following template:

**Attributes** $(CL, IN, M_{IN}, \lambda_E)$ with $CL$ to be a critical line, $IN$ the given set with $M_{IN}$ as previously defined and $\lambda_E$ as the starting point.

**Description** The critical line algorithm traces the efficient frontier form high to low $\lambda_E$, reacting to special cases to be introduced afterwards. The value $\lambda_E$ is needed to compute $(\alpha_{IN}, \beta_{IN}, \gamma_{IN}, \delta_{IN})$ at each step and it is recursively reduced.

**Preconditions** $M_{IN}$ to be non-singular.

**Special Cases** if $x_i \downarrow 0$ asset $i$ moves from $IN$ to $OUT$ and all parameters are recomputed; if $\eta_i \downarrow 0$ $i$ moves from $OUT$ to $IN$ and all parameters are recomputed; if $\lambda_E \downarrow 0$ an efficient portfolio has been found

The behaviour can be summarised as follows: If $x_i \downarrow 0$, it moves from $IN$ to $OUT$ on the next segment, and its $\eta_i$ will increase; Conversely, if $\eta_i \downarrow 0$ it moves from $OUT$ to $IN$ on the next segment, and its $x_i$ will increase. The new $M$ matrix is nonsingular, obtained just by adding or deleting one column and the corresponding row, so $M^{-1}, \alpha_{IN}, \beta_{IN}, \gamma_{IN}, \delta_{IN}$ can be computed efficiently.
A.3 Algorithms for the Index Tracking

As done for the PSP, we are giving in this section a brief outline of algorithms for the Index Tracking Problem. The same considerations made for the PSP hold: we are not aimed in giving a complete annotated bibliography on the topic, but rather to give a high-level overview and to suggest guidelines for further analysis.

We divide algorithms for the IT as to be Complete and Incomplete, and we begin our discussion detailing complete methods:

Meade and Salkin [116, 117] discuss some assumptions related to the tracking error in order to solve the problem by using Quadratic Programming. Indeed, when formulated as a Mixed-Integer Quadratic Program, branch-and-bound has been applied to Index Tracking [90]. Standard solvers (Cplex) have been used to solve mixed-integer linear Index tracking formulations (including transactions cost and cardinality constraints) by [23]. Standard methods have also been used as components of more robust strategies. For instance, Jansen and van Dijk [76] presented a problem decomposition (similar to the one we proposed in this thesis), employing standard Quadratic Programming after the set of assets has been decided.

Furthermore, exact and efficient parametric simplex algorithms have been applied to a variant of the Index Tracking problem by [85] (partial optimisation).

Also incomplete methods have been applied to the Index Tracking:

Focardi and Fabozzi [55] employed clustering to solve the problem: proposing Euclidean distances between stock price series as a basis of the hierarchical clustering, they select one (or more) stocks from each cluster to be inserted in the tracking portfolio. Another clustering approach has been proposed by Dose and Cincotti [44], by firstly selecting a subset of stocks and then determining their weights as the result of an optimisation process (asset allocation).

Tabata and Takeda [154] suggested to decompose the problem in two phases: firstly the choice of assets to be included must be performed, then assets’ weights must be determined. A heuristic approach has been defined to tackle this formulation.

Linear methods have been proposed by [136, 163, 27, 130]. Furthermore, a mathematical program to be solved by linear programming has been generated by fuzzy theory in [51].

Other heuristic approaches have been proposed by Gaivoronski et al. [58] and Corielli and Marcellino [28] (in the context of a factor model).

Stochastic programming has been exploited by Yao et al. [166] (using semidefinite programming) and Stoyan and Kwon [149] (using a two-stage stochastic program). Amongst other methods, scenario optimisation [163, 34, 7] and constraint aggregation [123] have been proposed.
Appendix B

Discrete Approximated Pareto fronts

In this Appendix we are giving additional figures relative to discretely computed returns instances Pareto fronts.
Figure B.1: AUEF: SD+QP, Instance 1, Discretely computed returns

Figure B.2: AUEF: SD+QP, Instance 2, Discretely computed returns
Figure B.3: AUEF: SD+QP, Instance 3, Discretely computed returns

Figure B.4: AUEF: SD+QP, Instance 4, Discretely computed returns
Figure B.5: Projection of the nondominated points in the three objectives problem onto the two-dimensional trade-off between tracking error and return. 4 instances, discrete.

Figure B.6: Projection of the nondominated points in the three objectives problem onto the two-dimensional trade-off between tracking error and variance. 4 instances, discrete.
Appendix C

Pearson correlations amongst multi-objective measures
Table C.1: Random portfolios: Pearson Correlation analysis.

<table>
<thead>
<tr>
<th>Instance</th>
<th>( \rho(TE, \sigma^2) )</th>
<th>( \rho(TE, r_p) )</th>
<th>( \rho(\sigma^2, r_p) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6589436</td>
<td>-0.0013849</td>
<td>0.1437935</td>
</tr>
<tr>
<td>2</td>
<td>0.8773389</td>
<td>-0.1746821</td>
<td>-0.1214940</td>
</tr>
<tr>
<td>3</td>
<td>0.8927029</td>
<td>0.0268381</td>
<td>0.1185914</td>
</tr>
<tr>
<td>4</td>
<td>0.9289325</td>
<td>0.0693839</td>
<td>0.2309638</td>
</tr>
<tr>
<td>5</td>
<td>0.7309253</td>
<td>-0.1212082</td>
<td>-0.5048555</td>
</tr>
</tbody>
</table>

Correlation between Return, Variance and Tracking Error over random portfolios, discretely compounded

<table>
<thead>
<tr>
<th>Instance</th>
<th>( \rho(TE, \sigma^2) )</th>
<th>( \rho(TE, r_p) )</th>
<th>( \rho(\sigma^2, r_p) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6384629</td>
<td>0.1168086</td>
<td>0.2765043</td>
</tr>
<tr>
<td>2</td>
<td>0.8625881</td>
<td>-0.1810025</td>
<td>-0.1092650</td>
</tr>
<tr>
<td>3</td>
<td>0.8890381</td>
<td>0.1188086</td>
<td>0.2108899</td>
</tr>
<tr>
<td>4</td>
<td>0.9208972</td>
<td>0.1109356</td>
<td>0.2644789</td>
</tr>
<tr>
<td>5</td>
<td>0.7371075</td>
<td>-0.3196788</td>
<td>-0.5012652</td>
</tr>
</tbody>
</table>

Table C.2: Optimised portfolios: Pearson correlation between Index Tracking Error and Return

<table>
<thead>
<tr>
<th>Instance</th>
<th>FD+QP</th>
<th>SD+QP</th>
<th>TS+QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9838264</td>
<td>0.9452859</td>
<td>0.9689109</td>
</tr>
<tr>
<td>2</td>
<td>0.9198262</td>
<td>0.8694472</td>
<td>0.8908287</td>
</tr>
<tr>
<td>3</td>
<td>0.3791247</td>
<td>0.3484607</td>
<td>0.3391185</td>
</tr>
<tr>
<td>4</td>
<td>0.8465497</td>
<td>0.8234686</td>
<td>0.8209287</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-0.9998859</td>
<td>-</td>
</tr>
<tr>
<td>Discrete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9838264</td>
<td>0.9452859</td>
<td>0.9689109</td>
</tr>
<tr>
<td>2</td>
<td>0.9198635</td>
<td>0.9566343</td>
<td>0.9524813</td>
</tr>
<tr>
<td>3</td>
<td>0.9088353</td>
<td>0.9063244</td>
<td>0.9019436</td>
</tr>
<tr>
<td>4</td>
<td>0.9363362</td>
<td>0.9194625</td>
<td>0.9353125</td>
</tr>
<tr>
<td>5</td>
<td>0.7810302</td>
<td>-0.9093942</td>
<td>-</td>
</tr>
</tbody>
</table>
Table C.3: Optimised portfolios: Pearson correlation between Return and Variance

<table>
<thead>
<tr>
<th>Instance</th>
<th>Continuous</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FD+QP</td>
<td>SD+QP</td>
<td>TS+QP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.8778265</td>
<td>0.8774337</td>
<td>0.8831495</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.7246013</td>
<td>0.7930128</td>
<td>0.7923934</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.9313572</td>
<td>0.9295367</td>
<td>0.2973364</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.9479613</td>
<td>0.9468227</td>
<td>0.9477524</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.9989791</td>
<td>-0.3703063</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Discrete

<table>
<thead>
<tr>
<th>Instance</th>
<th>Continuous</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FD+QP</td>
<td>SD+QP</td>
<td>TS+QP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.8778955</td>
<td>0.8811079</td>
<td>0.8828453</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.7246521</td>
<td>0.7296863</td>
<td>0.7305646</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.8675143</td>
<td>0.8750358</td>
<td>0.8752318</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.8199518</td>
<td>0.8358669</td>
<td>0.8331235</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-0.0852347</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Table C.4: Optimised portfolios: Pearson correlation between Index Tracking Error and Variance

<table>
<thead>
<tr>
<th>Instance</th>
<th>Continuous</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FD+QP</td>
<td>SD+QP</td>
<td>TS+QP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.8696677</td>
<td>0.8711558</td>
<td>0.8736682</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.6857134</td>
<td>0.6482741</td>
<td>0.6456677</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.0594818</td>
<td>0.0561420</td>
<td>0.0320780</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.4846590</td>
<td>0.5028774</td>
<td>0.5011948</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>0.0997780</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Discrete

<table>
<thead>
<tr>
<th>Instance</th>
<th>Continuous</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FD+QP</td>
<td>SD+QP</td>
<td>TS+QP</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.8696365</td>
<td>0.8723581</td>
<td>0.8948482</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.6857187</td>
<td>0.7253341</td>
<td>0.7241152</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.9011923</td>
<td>0.8968315</td>
<td>0.8893325</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.835478</td>
<td>0.8189894</td>
<td>0.8402524</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.8065421</td>
<td>0.6959137</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
Bibliography


