Università degli Studi "G. D'Annunzio"
Dipartimento di Scienze

Portfolio Selection by Metaheuristics: An Annotated Bibliography

Giacomo di Tollo

November 13, 2006

Technical Report no. R-2006-002 *Research Series*

Portfolio Selection by Metaheuristics: An Annotated Bibliography

Giacomo di Tollo [1]

Dipartimento di Scienze
Università degli Studi "G. D'Annunzio"
Viale Pindaro, 42 – 65127 Pescara, Italia

November 13, 2006

Abstract.

Portfolio Selection Problem (PSP) is one of the most studied issues in finance: It is concerned with selecting the portfolio of assets which minimize the risk given a certain level of returns. The PSP belongs to the class of combinatorial optimization problems and adding constraints to the basic formulation lead the problem to be NP-Hard, so metaheuristic approaches can be succesfully applied to solve the problem. This work is aimed in developing a conceptual analysis about Metaheuristic approaches developed in literature to this extent, introducing the problem, classifying the general concepts and outlining the most used strategies.

Keywords: *Portfolio Selection, Metaheuristics*

Contents

Introduction

Portfolio selection is one of the most studied issues in finance: The problem, in its basic formulation, is concerned with selecting the portfolio of assets which minimize the risk given a certain level of returns. Individuals and institutions prefer to invest in portfolios rather than single assets (or securities) because the first allow them to diversify risk without reducing expected returns. The basic model is formulated in the seminal work by Markowitz[38], in which the author rejects the hypothesis that investor maximize expected returns, because this idea doesn't imply that a diversified portfolio is preferable to a non-diversified portfolio, and states that they want to select a portfolio with minimum risk for given returns or more (and maximum returns for given risk or less), assuming that returns follow a multivariate normal distribution. This rule, as stated by the author, serves better as an explanation of investment as distinguished from speculative behavior: Its use is suitable for both theoretical analysis and practical purposes, but it presents shortcomings in the assessment of risk and returns.

Notwithstanding its success, the Markowitz model, referred to as Mean-Variance model, suffers from several drawbacks: At first it is difficult to gather necessary data and to estimate return and covariance from historical data and results are too sensitive to estimation errors of mean and covariances; Furthermore it is now judged too simplistic for practical purposes because it lacks incorporating lot of aspects of real-world trading: Maximum size of portfolio, minimum lots, transaction costs, preferences of which assets to include in the portfolio and by how much, management costs etc. Adding those issues to the original formulation makes the problem very hard to be solved by exact methods. Hence the need for designing efficient approximate algorithms, such as "metaheuristics"[3].

In this work we are aimed to give an overview of the use of "metaheuristic" techniques to solve the portfolio selection problem (hereinafter referred to as PSP). As PSP can be modeled as an optimization problem, in section 1 we will introduce the objective functions used in the literature. The formulation of the problem can be enriched by adding constraint, and they are introduced in section 2. The following sections (3–5) tackle "metaheuristic" approaches for the PSP as developed in literature: Our work is aimed in discussing about the choice of optimal portfolio when the initial wealth of investor is given by liquid in capital, so portfolio revision and portfolio re-balancing are not taken into account.

1 Objective functions

As stated before, the PSP can be modeled as an optimization problem in which an objective function has to be minimized (or maximized). In the process of portfolio selection in canonical form we want to find a portfolio that minimizes the risk at given levels of return rate[1]. In the Markowitz formulation the risk measure is given by the variance of portfolio, and this measure represents the most common objective function used in the majority of related works.

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

subject to

$$\sum_{i=1}^{n} r_i x_i = r_p \tag{2}$$

$$\sum_{i=1}^{n} x_i = 1 \tag{3}$$

$$x_i \in [0,1] \qquad i = 1, \ldots, n \tag{4}$$

where n is the number of risky assets, x_i is the proportion of money invested in asset i, r_i is the expected return (per period) of asset i, σ_{ij} is the real-valued covariance of expected returns on securities i and j[2]. The expected return of portfolio is given by r_p, whilst $\sum_{i=1}^{n} r_i x_i$ represents the actual forecasted return. The variance of portfolio (herein referred to as risk-measure) σ_p^2 is given by $\sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} x_i x_j$, constraint (2) refers to the expected return of the portfolio. Constraint (3) ensures that asset weights sum up to one, as they are considered as fractions of the whole portfolio instead of their actual value (constraint (4)) .

In order to handle the problem it is possible to optimize function (1) for several values of r_p[8].

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} x_i x_j \tag{5}$$

subject to

$$\sum_{i=1}^{n} r_i x_i \geq r_p \qquad r_p = 0 \ldots r_{max} \tag{6}$$

$$\sum_{i=1}^{n} x_i = 1 \tag{7}$$

$$x_i \in [0,1] \qquad i = 1, \ldots, n \tag{8}$$

[1] Here we handle the problem as consisting in a minimization of the risk measure. Of course the formulation can be expressed as a maximization of returns or in other several ways. See [16] for a brief discussion.

[2] For understandability, and to satisfy specific algorithm requirements, covariances are stored in the "covariance matrix" Cov, a $n \times n$ triangular matrix where element $Cov[i,j]$ (if $i \neq j$) represents covariance between assets i and j, whilst $Cov[i,i]$ represents variance of asset i.

Another approach consists on including constraint (2) on the objective function parameterizing risk and return to yield a parametric objective function [5] [58] [28]

$$\max(1-\lambda)\sum_{i=1}^{n} r_i x_i - \lambda \sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} x_i x_j \tag{9}$$

subject to constraints (3) and (4), where λ is a trade-off coefficient belonging to the range [0, 1]: If $\lambda = 0$ the investor completely disregards risk and wants to maximize returns; otherwise when $\lambda = 1$ the investor is risk-adverse and only wants to minimize risk. Resolving the problem for several λ values we can estimate the *efficient frontier* for the Markowitz unconstrained problem (referred to as UEF), being afterwards able the investor to choose the portfolio he desires depending on his requirements. UEF is composed of *Pareto* optimal solution: We say a solution is *Pareto* optimal, in a multi-objective framework, if no criterion can be improved without deterioration of other criterion. In our example a solution s is said to be efficient (Pareto–optimal) if there is no other solution s_1 such that $(return(s_1) > return(s)$ and $risk(s_1) \leq risk(s))$ or $(return(s_1) \geq return(s)$ and $risk(s_1) < risk(s))$. As "metaheuristics" provides us an approximation of the actual Pareto-frontier, in the following we will distinguish between the true efficient frontier (UEF) and the approximated one (AUEF): Since we are going to introduce 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 since here that the unconstrained frontier generally dominates the unconstrained one, so they are different.

In a straightforward way the problem can be formulated with a two-objective function, in order to maximize returns and minimize risk[51][40]:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} x_i x_j \tag{10}$$

$$\max \sum_{i=1}^{n} r_i x_i \tag{11}$$

subject to constraint (3). Approaches (5)(9)(10) can be applied also if other risk and return measures are used: Indeed several variations were successively applied to the model, so involving different objective functions. Markowitz himself suggested the use of semi-variance instead of variance in order to assess portfolio risk, but he introduced the latter because it is more tractable computationally. Semi-variance can be defined as

$$semivar = \sum_{r_{el} \leq E[r]} p_{el}(r_{el} - E[r])^2 \tag{12}$$

where r_{el} refers to one of the elements of portfolio return distribution, p_{el} to its probability and $E[r]$ to the expected mean return.

This measure is equivalent to variance if return distribution is symmetric around the mean and captures the essence of risk as perceived by investors: The likelihood of incurring into 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 threshold τ. 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 *down-side risk*

$$DSR(\tau) = \sum_{r_{el} \leq \tau} p_{el}(\tau - r_{el})^q \tag{13}$$

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 τ 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 loosing their wealth because of a low-probability-high-impact-event[53]. It has been used as the threshold below which the investor perceives a loss[18][17]. The probability that portfolio returns falls below the VaR level is called *Shortfall Probability*

$$SP = p(r < VaR) \tag{14}$$

where, for sake of readability, r stands for $\sum_{i=1}^{n} r_i x_i$.
Expected Shortfall Probability is furthermore defined as the expected return of portfolio given that its value has fallen below VaR.

$$ES = E(r|r < VaR) \tag{15}$$

Amongst other approaches it worths to introduce the Mean-Absolute-Deviation model (MAD)[31], in which the risk is defined as the mean absolute deviation of the portfolio rate of return. This model doesn't 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.

$$\min E\left[\left|\sum_{i=1}^{n} r_i x_i - E\left[\sum_{i=1}^{n} r_i x_i\right]\right|\right] \tag{16}$$

Assuming $r_i = E[R_i] = \frac{\sum t{=}1^T r_{it}}{T}$, this equation can be re-formulated as follows:

$$\min \frac{\sum_{t=1}^{T} |\sum_{i=1}^{n}(r_{it} - r_i)x_i|}{T} \tag{17}$$

and in this formulation it presents an advantage on the easiness in constraining the number of assets, since restricting the time-horizon the cardinality is supposed to lower (an optimal solution will contain at most 2T + 2 assets regardless of the instance size, being T the number of observations).

Following the same ideas, in [48] risk is measured as the mean semi-absolute deviation of the rate of return below the average:

$$\frac{\sum_{t=1}^{T} |\min(0, \sum_{i=1}^{n}(r_{it} - r_i)x_i)|}{T} \tag{18}$$

This function is shown to be equivalent to MAD, as semi-deviation is equal to half of absolute deviation, but it is useful to clarify the formulation because it halves the number of constraints. The objective is modeled as a linear function through the introduction of ad-hoc and dummies variables and constraints.

Furthermore, due to the high computational complexity of the Mean-Variance formulation, several efforts have been made to reconduct the problem to a linear programming model in order to make it solvable with conventional methods (as the simplex method): Amongst them, besides the above cited ones, the approach proposed in Young[57] defines the risk as the minimum return achieved by portfolio over the considered time horizon.

Another drawback of the traditional Mean-Variance formulation is its two–dimensionality, as portfolios are not directly comparable in their qualities: Also comparing two solutions lying on the efficient frontier the investor must decide according to his risk-aversion. So one-dimensional measure have been used to evaluate and compare directly portfolios, as the reward-to-variability ratio[29][39][36] and the geometric mean[13].

If the problem is described as a multi-objective optimization, it is possible to use several functions to optimize: Subbu et al.[53] uses the following:

$$\begin{cases} max & \text{Portfolio expected return} \\ min & \text{Variance} \\ min & \text{Portfolio value at risk} \end{cases} \tag{19}$$

using a hybrid evolutionary approach embedded in a model able to handle preferences, the objective is enriched with other three metrics: *Market-yield*, *Dollar duration weighted Market-yield* and *transaction costs* (even if they are not exploited in local-search, but only to describe and structure ordinal preferences). The idea of using a multi-criterion objective function is common when the model is aimed to support decision processes: In Ehrgott et al.[14], the objective is to maximize 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.[42]. According to existing models, they assume portfolio risk being composed of the uncertainty risk and the relation risk: The uncertainty risk measures the possibilistic degree of future return rate, whilst relation risk measures the trending degree of the sequence. In this framework the objective is given by

$$\begin{cases} max & \text{Portfolio expected return} \\ min & \text{Uncertainty Risk} \\ min & \text{Relation risk} \end{cases} \tag{20}$$

proposing an interesting alternative to traditional Mean-Variance portfolios.

Many other objective functions and utility measures have been proposed (see Kallberg and Ziemba[25] for an overview) but a citation must be given to Catanas[4], whose eight objective functions are defined to include more than the two canonical moments (indeed they are suitable to include mean, variance, kurtosis and skewness). Here below we report only two of them:

$$\max \sum_{i=1}^{n} \left[\sin(50 \cdot x_i) \prod_{k=1,k\neq i}^{n} x_k \right] \tag{21}$$

$$\max \sum_{x=1}^{n} \left[x_i \prod_{k=1,k\neq i}^{n} (\lceil 50 \cdot k \cdot x_k \rceil \text{mod} 8) \right] \tag{22}$$

1.1 Assessing returns

As stated before, the PSP takes into account two aspect of financial portfolios: Risk and expected returns. Several measures have been discussed for the risk-assessment of the portfolios, and they are all computable handling historical data. Assessing returns is a different topic, as it is clear that asset future returns are not deductable from past observation without the likelihood of incurring in errors: Mean-Variance analysis is claimed to paid not enough attention in estimating returns, while fund-managers and professionals, developing their decision-supporting models, require good estimate in order to *forecast* performances. That's why Mean-Variance portfolios are generally said to suffer from not satisfactory out-of-samples performances respect to simpler heuristic methods designed to specific purposes[41].

The assessment of returns can be made in two ways, depending on the aim of the work: If it consists of developing and testing the fitness of an algorithm for the PSP, returns are easily computed, i.e. considering returns as normally distributed and computing the mean of past returns (or using regression techniques). Benchmarks provides generally this computation of expected returns (The reference benchmark for the PSP was introduced by Beasley in its OR Library[2], now available at `http://people.brunel.ac.uk/ mastjjb/jeb/info.html`)

Conversely, if the work is aimed in depicting a good model of real world, it has to take into account uncertainty about future returns. This generally happens by modeling future returns generating a set of possible realizations, called scenarios[17][18][13].

In few words, since we don't know the future returns, we must choose under uncertainty, imagining a series of imaginable future state (scenarios) of future returns, and giving each of a probability of occurrence (their sum being 1) relying on a statistical model, past returns or experts opinions. Given p_s the probability of occurrence of scenario s the optimization technique must optimize a function taking into account the probability of occurrence of each scenario, i.e. with the following function:

$$\min \Big(\sum_s p_s \sum_{j=1}^{m} r_{sj} - r_{exp} \Big)^2 \tag{23}$$

where r_{sj} represent the return of asset j under scenario s. Anyway also with this technique it is difficult to reflect the situation of returns continuous distribution, so other techniques based on sampling procedures can be used [55].

2 Constraints

As stated before, a shortcoming of the original Markowitz formulation is that it lacks incorporating many aspects of real-world trading: Maximum size of portfolio, minimum lots, transaction costs, preferences of which assets to include in the portfolio and by how much, management costs, etc. These aspects can be formulated by introducing constraints and in the following we will introduce some of the most relevant ones.

2.1 Budget and Return Constraint

These are the most important constraints, stating that weights of assets must sum up to one and that the weighted sum of assets returns must be equal (or $\geq$) to the expected return. These constraints are included in the unconstrained Markowitz model and are used to norm the solution.

$$\sum_{i=1}^{n} x_i = 1 \tag{24}$$

$$\sum_{i=1}^{n} r_i x_i \geq r_p \tag{25}$$

Constraint (24) 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 (see section 5) can be relaxed allowing that it can assume value between two bounds C_0 and C_1[9][48][35]

$$C_0 \leq \sum_{i=1}^{n} x_i \leq C_1 \tag{26}$$

Furthermore, in the continuous formulation, imposing the budget constraint generally means that short sales are not allowed.

Return constraint (25) is very important as returns represent one of the two main aspects of the problem. For that reason it can be relaxed and included in the objective in a lagrangian style (see section 1). When belonging to constraint set it is used with both equality operator, as in equation (2)[8][5][44], or inequality operator ($\sum_{i=1}^{n} r_i x_i \geq r_p$) [45].

2.2 Cardinality Constraints

The number of assets in the portfolio has to be limited. Introducing a binary variable z equal to 1 if asset is in the portfolio and 0 otherwise, the constraint can be expressed as follows:

$$\sum_{i=1}^{n} z_i = k \tag{27}$$

This constraint can be defined also in inequality form, imposing that the portfolio must contain no more than k assets[45][8][28]; it is imposed to facilitate the portfolio management and to reduce its management cost. When the problem is facing this constraint it can be named "asset paring" problem[32]. There is evidence that when imposing this constraint ACEF tends to approach UEF when k increases [23] [5], whilst from a certain value of k onward ACEF tends to roll up toward special points: These points fix an average profit and risk from which the algorithms are not able to quit[40].

2.3 Floor and Ceiling Constraints

With these constraints we impose a minimum and maximum proportion (ε_i and δ_i) allowed to be held for each asset in portfolio, so that $\varepsilon_i \leq x_i \leq \delta_i$ $(i = 1 \ldots k)$; in other words the portion of the portfolio for a specific asset (each asset or some of them) must be included in a fixed interval. Floor constraint (we can also refer to it as a lower bound) is used to avoid the cost of administrating tiny portions of assets; ceiling constraints (we can also refer to it as an upper bound) to avoid excessive exposure to a specific asset and in some case is imposed by law.

$$\varepsilon_i z_i \leq x_i \leq \delta_i z_i \tag{28}$$

Note that this constraint can implicitly define a range for the cardinality constraint (For instance a lower bound of 0.1 for each asset implicitly defines a maximum cardinality of 10 assets in portfolio), so the two constraints, when both formulated, must be consistent.

Bounds must be carefully chosen in order to ensure that feasible solutions exist. Generally the upper bound is considered more relevant than the lower one and it is also possible to impose different upper and lower bounds for each asset, but this opportunity has been not exploited in literature so far.

2.4 Minimum lots

The unconstrained Markowitz model considers investments as perfectly divisible, so as to be represented by a real variable, whilst in real world securities are negotiated as multiples of minimum lots. For each asset there exists a minimum tradable lot, generally referred to as *round*: it is expressed in money so this constraint is generally encountered when dealing with the integer formulation [9][27][35], in which assets are labeled by their actual value rather than their ratio to the whole portfolio (see section 5). In integer values, if p_j is the price of asset j and tr_j its minimum tradable quantity, the minimum lot expressed in money is given by $c_j = tr_j p_j$. When using the continuous formulation its application consists in imposing that each weight must be multiple of a given fraction[51], but its meaning is different from imposing *rounds* in integer formulation.

Minimum lots seem to be relevant for small investors but negligible for big ones and their introduction has the effect to reduce the number of different assets in the optimal portfolio. Imposing them in the Markowitz formulation the PSP switches from a formerly continuous problem into a discrete one[51].

2.5 Turnover and trading constraints

These constraints define upper (in case of buying) and lower (in case of selling) bounds for the variation of asset values from one period to the next one; they can be implicitly inserted in the model imposing transaction costs and taxes either in the objective function or in the constraints. A formulation, conversely, can define lower bounds in case of buying and upper bound in case of selling: It means that investors are not interested in selling and buying small volumes of assets, due to contract clauses or fixed transaction costs. These constraints can be used in each combination in the formulation (they are not mutually exclusive), and were introduced in Crama and Schyns[8], but they are explicitly formulated so as to belong to a multi-period formulation, whilst the problem faced by these authors is indeed a single-period. In their work these constraints are to be satisfied considering solutions found in adjacent computation time w.r.t. the execution time of the algorithm, as they implicitly show converting turnover constraints into floor and ceiling constraints. This is not conceptually sound as those constraints have to be satisfied, in a multi-period formulation, over adjacent pairs of portfolios on the planing horizon. In other words, a search algorithm is run, returning a portfolio for each period; portfolios representing solutions in adjacent periods must satisfy these constraints, whilst no restrictions are imposed over neighbor solutions in the search space defined in order to find the best portfolio in one single period.

Equations (29) and (30) represent purchase and sale turnover constraints, whilst equation (31) represents trading constraint,

$$\max(x_i - x_i^0, 0) \leq \overline{B_i} \qquad 1 \leq i \leq n \tag{29}$$

$$\max(x_i^0 - x_i, 0) \leq \overline{S_i} \qquad 1 \leq i \leq n \tag{30}$$

$$x_i = x_i^0 \text{ or } x_i \geq (x_i^0 + \underline{B_i}) \text{ or } x_i \leq (x_i^0 - \underline{S_i}) \qquad 1 \leq i \leq n \tag{31}$$

where x_i^0 denotes weight of asset i in the previous period, $\overline{B_i}$ and $\overline{S_i}$ denote the maximum purchase and sale bounds (respectively) of asset i, $\underline{B_i}$ and $\underline{S_i}$ denote the minimum purchase and sale bounds (respectively) of asset i. Trading constraints are *aut–aut* constraints, as either the asset weight remains at the same level or it is modified by a minimum admissible value, so they are difficult to handle: They are similar to rounds (the new weight must be higher than the threshold, while introducing rounds must be a multiple of it), but they are included in the continuous model.

2.6 Class Constraints

In the real world of finance it may happen that investors ideally split all 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 the portfolio held in each class. Let M be the set of classes, L_m and U_m the lower and upper proportion limit (respectively) for class m, the class constraint can be defined as

$$L_m \leq \sum_{i \epsilon \Gamma_m} x_i \leq U_m \qquad m = 1 \dots m \tag{32}$$

being Γ_m the set of assets belonging to class m. This constraint is useful to diversify the portfolio amongst several economics areas, and reveals all its importance when we consider the multi-period *asset allocation* problem under the *fixed–mix* strategy, in which we determine a predetermined mix of classes (e.g. 30% insurance assets, 30% treasury bills, 40% naval assets) to be satisfied for each period, so imposing a re–balancing of the portfolio (to handle dividends and transaction costs) for reaching the desired mix-level. This problem was one of the fist asset allocation formulation to be attacked by a "metaheuristics"[19], so we will discuss about it in the following, even if it doesn't represent a typical single-period portfolio-selection formulation.

2.7 Compulsory assets

Ideally one investor may wish that some specific assets appears in the portfolio, in proportion fixed or to be determined. This constraint can be easily determined setting $z_i = 1$ for corresponding assets and imposing more or less restrictive upper and lower bound.

2.8 Non-negativity constraints

This constraint is defined in the formulation of the problem, when imposing $x_j \geq 0 \quad \forall j$, where j is the subscript indicating a specific asset. This constraint means that no short sales are allowed and it is imposed over almost all the works we are analyzing (a notable exception is given by [44], whilst in [8] short sales are allowed in the beginning formulation). Note that this constraint becomes redundant when imposing floor constraints.

It has been shown that the constrained PSP is NP-Complete[35]; when imposing constraints the feasible set of solutions, w.r.t. the LP model, becomes not connected and the problem becomes a non-convex optimization problem on the boundary of the feasible set[14].

The most used model in literature imposes cardinality, floor and ceiling constraints: These constraints are generally imposed to prevent the portfolio containing too much assets with tiny weights, a situation that generally occurs when solving the unconstrained case.

3 Metaheuristic approaches for portfolio selection

3.1 General concepts

In order to apply "metaheuristics" techniques to the PSP a series of topics must be taken into account: Objective functions and constraints (we already discussed about it), search space, neighborhood relations, choice of the initial solution, cost function, performance measures. These topics will be tackled in the following subsections, before discussing about different strategies developed in literature for the PSP.

3.1.1 Portfolio representation and search space

When applying meta-heuristics to the PSP, we must first define the search space and how to represent it, considering that the historical data and universe n are kept fixed and stored in a non-modifiable data structure. We consider assets universe as represented by a list $U = [1 \dots n]$ whose elements represents asset indexes. For representing states reached by the search process during its execution, generally two ideas can be pursued: Either only a dynamical data-structure of assets is mantained, in which weights for each asset are stored, or two dynamical data-structures are mantained. In this latter case, generally a list $L = [a_{l1} \dots a_{lz}]$, $z \leq n$ is used to store indexes corrisponding to asset actually in the portfolio and another list is used to store weights associated to each previously introduced assets ($S = [x_{l1} \dots x_{lz}]$) so that x_{li} is the fraction of a_{li} in the portfolio. For instance, if we are facing a problem in which $n = 10$, we can represent a portfolio with the following lists: $L = [1, 4, 9, 10]$ and $S = [\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}]$: This representation means that the composition of the portfolio is the following: $x_1 = \frac{1}{2}$, $x_4 = \frac{1}{4}$, $x_9 = \frac{1}{8}$, $x_{10} = \frac{1}{8}$ (we refer to this representation as *two-fold-representation*).

If we decide to use only a data-structure, we will store in it weights of all assets in the universe: In this case if an asset does not belong to the actual portfolio found its weight will be 0. The representation of the same portfolio discussed above will be composed only by the following list $S = [\frac{1}{2}, 0, 0, \frac{1}{4}, 0, 0, 0, 0, \frac{1}{8}, \frac{1}{8}]$. This implementation can turn out to be inefficient as for each search step we must keep in memory weights for each asset and some ad-hoc mechanisms must be devised to perform specific operations (i.e. to determine the actual composition of the portfolio the array must be thoroughly browsed). This representation (whom we refer to as *one-fold-representation*) allows us to preserve all information about the current state of the search, but it can slow down too much the search process. Conversely, the *two-fold-representation* is more flexible and speedy, but it must be implemented with care.

A compromise between these two approaches in given by Streichert et al.[50][51][52] in which a *knapsack-based* representation of portfolios is used: Basically an *one-fold-representation* is used, but for each asset a bit (label) is added to specify if it belongs to portfolio or not. The asset representation will be $x_i = z_i \cdot s_i$ and, for the same portfolio discussed above, the representation will be the following: $Z = [1, 0, 0, 1, 0, 0, 0, 0, 1, 1]$ and $S = [\frac{1}{2}, 0, 0, \frac{1}{4}, 0, 0, 0, 0, \frac{1}{8}, \frac{1}{8}]$, in which each 0 in the list S can be replaced by any real number between 0 and 1 since to determine the real value of asset i, s_i will be multiplied for the corresponding z_i value. This value will be always 0 $\forall i | b_i = 0$. This representation simplifies the search process as adding or removing asset i in portfolio can be performed just setting $b_i = 1$ or 0. Similar is the representation by Kellerer and Maringer[28] in which a portfolio is represented by *two-fold-representation*, but it CAN contain also asset with weight 0 due to the implementation of the search process, as during the search if an asset i is forced to zero it is either kept in the portfolio with $x_i = 0$ (with probability p) or replaced by another asset j with random weight $x_j \in [0, (2 \cdot U_t)]$ (with probability 1-p): This representation is useful when cardinality constraint in inequality form is posed.

Whatever the representation of portfolios, they constitute the search space explored by the algorithm we develop for the problem: This search space presents the important feature to be more dense in the region corresponding to low returns, whilst less reachable portfolios belong to the upside part of the frontier, where returns are higher and the search space is more sparse. Please notice that explaining the previously introduced representation we didn't mention constraints: This is because there is no unanimous agreement about the feasibility of portfolios (and implicitly, constraint satisfaction) in the search space: The issue about feasibility of portfolios found during the search process doesn't appear to be addressed in an unanimous way. In other words, some works defines the neighborhood and the search strategies so that states must consist only of feasible portfolios, other works does not, allowing the search process reaching unfeasible solutions. A compromise can be stated by partitioning constraints in two classes:

1. *Hard constraints*, constraints that must be always satisfied by any candidate solution;

2. *Soft constraints*, constraints allowed to be not satisfied during the search process. Generally their violation is given a weight in the cost function.

so that a state can consist of a portfolio that strictly satisfies some constraints whilst not satisfying others.

Following these ideas, we can classify the search processes depending on how they handle infeasibility over the search space:

- *all feasible* approach: Each candidate solution s must satisfy constraints for any step in the search process (e.g Chang et al.[5]);

- *repair* approach, in which if a non-feasible solution is found, this is suddenly forced to satisfy constraints by an embedded repair mechanism (e.g Streichert et al.[51]);

- *penalties* approach: We allow moving toward infeasible solutions, but those will be assigned a penalty in the cost function, depending on the amount of violation (e.g Schaerf [45], see section 3.1.4). When using this mechanism the most reliable approach consists in satisfying all but the return constraint over the search process.

Sometimes 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. Repair mechanisms have the effect of considering a huge number of candidate solutions and to reduce run-time, but they can cause loss of information and wasting good partial solutions features as apart from increasing the exploration of the search space, they increase randomness reducing the exploitation of good solutions.

A typical repair mechanism is explained in Streichert et al.[52], referring to a formulation with cardinality and minimum lots constraints: This take as input a non normalized weight vector and repairs the solution in the following way (Please notice that this repair-mechanism is deterministic):

1. all weights are normalized so as to sum to one. This is done by setting weights $x'_i = x_i / \sum_i x_i$;

2. the obtained vector is normalized so as to meet cardinality constraint: Only the k assets with largest value of x'_i are held and then are normalized to the value x''_i;

3. a further modification is required to meet minimum lots constraints: Asset weights are forced to their previous roundlot level $x'''_i = x''_i - (x''_i \bmod c_i)$. The free portfolio amount is then redistributed so as to meet minimum lots constraint buying quantities of c_i on assets with biggest $(x''_i \bmod c_i)$ until all the remainder is spent.

If only feasible states are allowed we are sure to obtain valid portfolios during the whole search-process, but constraint can create troubles in finding them; conversely allowing infeasibility can help exploring a wider portion of the search-space, but can greatly slow down the convergence of the algorithm, so this issue must be addressed with care.

3.1.2 Neighborhood relations

Defining and understanding the neighborhood relation is a key point in order to develop and fathom powerful local search strategies. This is a crucial point in "meta-heuristics" as sometime in the literature the neighborhood is not explicitly defined, with the effect of making the algorithm unclear.

The first relations were introduced by Rolland[44] as, tackling the unconstrained problem, he introduced two neighborhoods that can be considered the basic structures for further developments. In the first (referred to as *RollandI*) the neighbor of a solution is defined as a new solution in which the weight of only one asset is increased or decreased of a given *step* size. The second (referred to as *RollandII*) is defined so that the weight of an asset is increased or decreased of a given value (*step* size) and the value of one other asset (randomly chosen) is increased or decreased of the same value. Note that with this approach assets contained in the final solution will constitute a subset of the starting portfolio since assets to be modified are chosen amongst assets present in the portfolio, but the author does not consider it as a problem as the problem is unconstrained and the portfolio is initialized with all assets in the universe N, with $x = \frac{1}{n}$ for each asset; furthermore *RollandI* is likely to produce infeasible solutions. These neighborhoods are well suited to the unconstrained case, as assets can assume every possible value (given their sum being one), but when imposing constraints the situation changes as assets cannot be present in the portfolio in any quantity. Hence those neighborhoods are to be enhanced enbedding insertion and deletion of assets. RollandII can be improved transfering a part of the portfolio from one assets i to another one j even if the latter does not belong to the portfolio: In this case j it will be inserted ([45]*TID*; [17]). Anyway this approach must include some mechanism to handle upper and lower bound if they are present in the formulation.

Also RollandI can be improved either ensuring feasibility for each step or allowing inserting and deleting assets. We can exploit feasibility defining a portfolio as neighbor of a solution if weight of one asset is increased while others are decreased[4]: If a solution is given by a a weight vector $(x_1 \dots x_n)$, the neighbor one is $(\frac{x_1}{1+step}, \dots \frac{x_i+step}{1+step} \dots \frac{x_n}{1+step})$, for only one i, $1 \leq i \leq n$. This neighborhood is proven to guarantee the finiteness of its size and that for a long enough sequence of moves, each solution of the problem is close enough to a move in the sequence (completeness). Completeness does not depend on the initial solution and holds iff $step \leq \frac{1}{n-1}$ (n is the number of assets and the optimal value of ***step*** is said to be $\frac{1}{n-1}$ for sake of speed-convergence). The latter improvement is given by inserting and deleting assets ([45]*idR*,[5]): This happens when a decrease is performed on asset i so that its value falls below its lower bound ϵ, so that asset i is deleted and another asset j is inserted on the portfolio. Conversely if an increase is performed on asset i so that its value exceeds its upper bound δ, the assumed value is set to δ. All other assets are normalized. Even if with this neighborhood asset contained in neighbor position are allowed to be different, the cardinality of neighbor portfolios is identical. A further improvement is reached allowing neighbor solutions to have different number of assets([45] idID): Three kinds of operations are allowed on a selected asset i:

- the value is increased of a given proportion (in this case asset i is already in the portfolio). If the obtained value exceeds its upper bound δ, the assumed value is set to δ;
- the value is decreased of a given proportion (in this case asset i is already in the portfolio). If the value falls below its lower bound ϵ, asset i is deleted and **not** replaced by any asset;
- asset is inserted in the portfolio with weight equal to its lower bound (in this case asset i is not yet in the portfolio);

Weights are afterward normalized.

After this brief overview we can propose a way of classifying the neighborhood relations in two classes:

1. relations in which neighbors are generated modifying weights of a pre-determined number of asset of the current solution;

2. relations in which neighbors are generated modifying all assets in portfolio in order to ensure feasibility.

In order to fathom this classification, we can ideally define a neighbor of a portfolio starting from the portfolio itself, selecting one asset to be modified, specifying its amount of variation and performing the change: This asset, referred to as *pivot*[40], constitutes the asset neighbors are generated starting from. After doing it, we counter-balance this modification by changing weights of other assets: If only a pre-determined number of assets is selected to be modified[3] the neighborhood is said to belong to class 1, otherwise, if all assets belonging are to be modified the neighborhood is said to belong to class 2.

The neighborhood structured as class 1. can either consist only of feasible solutions (i.e. [45], structure *TID*) or allowing infeasible moves too (i.e [44], pag. 8). The simplest neighborhoods in this group are generated modifying the pivot weight and counterbalancing this change by modifying weight of only another asset ([44] *TID*; [17]). The process can be generalized introducing an integer c representing the number of assets to be modified in order to counterbalance the magnitude of pivot weight variation. Crama and Schyns[8] uses $c = 2$, but it is possible to set c at an higher number, even if there are no significative efforts in this direction. It is also possible to set $c = 0$, so allowing infeasible and worst moves, but in this case an implicitly formulated repair mechanism should be included (Rolland[44] repairs solutions after five consecutive infeasible moves). Using this framework we can define the distance *step* as the difference between pivot weights over the starting portfolio and its neighbors. In the previously introduced neighborhoods *step* is kept fixed between the starting portfolio and all its neighbors, but neighbors can be generated also varying its value: In Armañanzas and Lozano [40] neighbors are generated varying *step* from a minimum of $\frac{w_{pivot}}{n}$ to a maximum of w_{pivot} , being forced to assume as value all multiples of $\frac{w_{pivot}}{n}$, so letting *pivot* disappear in one neighbor and defining the cardinality of the neighborhood as n.

Conversely, neighborhoods of class 2 are generally encountered using population-based algorithms with particular mention to genetic algorithms, in which crossover and mutation operators are most likely to return an infeasible state. In this case the above specified two-phase mechanism (Choose an asset and modify its weight; Decide what to do with other assets) generally does not hold because the search is structured as evolution of portfolios generated by assemblying existing portfolios, so it turns out being impossible to determine the *pivot*. We can say that whenever a repair mechanism is implemented, the neighborhood belongs to class 2: Obviously this doesn't represent the only case, and other approaches can be reconducted to this class[5][4][45], even if they rather appear reconductable to examples of class 1 with $c = k - 1$.

[3] This number can even be 0, see [44], pag. 8.

Neighborhoods described above can be used in conjunction when using strategies such VNS (see sec. 3.6). Indeed, this aspect has not been exploited enough for the PSP, and the few efforts in this direction [19] [14] [48] cannot be said to belong exactly to the PSP in its canonic formulation.

3.1.3 Initial solution

Usually "metaheuristics" are quite robust w.r.t. the choice of the initial solution: This assertion has proven formally to be true, for the portfolio selection problem, by Catanas [4] referring to its neighborhood relation (see section 3.1.2), and empirically holds for the most works on this issue. For that reason most of the works assume as starting solution(e.g. Ehrgott et al.[14]), a randomly generated one, embedding a mechanism[8] to ensure feasibility of the starting portfolio.

An extension of this idea is generally employed when using genetic algorithms, when a population of solutions (chromosomes) must be initialized. The portfolios belonging to the initial population are randomly generated (generally ensuring feasibility) but some mechanism can be employed to insert in some of them particular features. For instance in Wang et al.[9] 200 portfolios are randomly generated, but two amongst them (randomly selected) are suddenly substituted by solutions of two single objective programming problems (max expected return and min variance) s.t. constraints in the formulation, where these solutions represent the global maximum return and the global minimum variance solution respectively.

Also *Rules of thumb* are used i.e. selecting the best amongst randomly generated portfolios [45] or selecting n assets with $x = 1\backslash n$ for each i [44] (unconstrained case). If satisfying cardinality constraint is considered essential greedy initialization can be used: Ehrgott et al.[14] uses, besides the initialization before described, a procedure in which new assets are successively added following a greedy strategy until the desired number of assets is reached. Indeed is it possible first to randomly select k assets and then determining their weights in order to satisfy constraints [40].

3.1.4 Cost function

In order to apply "metaheuristics" to the PSP it is important to distinguish between objective function and cost function: The first represents the function to be optimized to solve the problem, so the function being minimized (or maximized) in the final state reached by the search algorithm; The latter represents the function leading the search process over the search space toward solutions, so allowing to evaluate and accept (or not) new solutions encountered during the search. Generally when dealing with combinatorial optimization problems the objective of the problem is often used as evaluation function, but sometimes different cost function can better guide the search toward promising solutions; in other cases, when dealing with constraints, the cost function can be defined starting from the objective and adding some mechanism to embed and evaluate their violation (if any). In fact there is a debate in "meta-heuristic" literature about how to head the search trajectory over the search space: The major issue is if to force the search to visit only feasible solutions or to allow visiting infeasible ones. When the search process is allowed to explore infeasible regions the solution can consist in a infeasible solution because some constraint has been violated. In this case we must assign a cost to each violation and incorporate it in the cost function. Schaerf[45] defines a cost function in which the cost associated to the violation of budget constraint ($f_1(X)$) is combined with the original objective function $\big(f_2(X)\big)$: The overall cost function to be minimized is an overall weighted sum of the two components $w_1 f_1(X) + w_2 f_2(X)$, in which initially w_1 is set to a much larger value then w_2, while it varies during the search according to a *shifting penalties* mechanism.

$$\min w_1 f_1(X) + w_2 f_2(X) \tag{33}$$

$$f_1(X) = \max\Big(0, \quad \sum_{i=1}^{n} r_i x_i\Big)$$

$$f_2(X) = \sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} x_i x_j$$

A similar approach is followed in Gilli and Këllezi[17], when they want to test their TA approach (see below) on Mean-Variance formulation. They choose to minimize the following:

$$\min\left(\sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} x_i x_j + \underline{p}\Big(r_p - \sum_{i=1}^{n} r_i x_i\Big)\right) \tag{34}$$

$$\underline{p} = \begin{cases} c & \text{if returns are higher than } r_p \\ 0 & \text{otherwise} \end{cases}$$

where $\underline{p}$ is the penalty term. Similarly to Schaerf, in this approach only the return constraint is allowed not to be satisfied, as the algorithm is implemented so that other constraints are always satisfied.

If the search is allowed to visit only feasible moves generally cost and objective function are equivalent. Anyway, it must be understood if the search trajectory moves only in feasible areas because of its formulation or because an implicit repair mechanism is embedded (see section 3.1.1).

3.1.5 Performance measures

Generally performance measures are obtained by comparing constrained results (ACEF) with the ones obtained in the unconstrained case for each level of return (each point on the UEF) and drawing statistical measures (mean and median percentage error, standard deviation etc.) for the overall frontier. There are however many ways to assess error measure: Chang et al.[5] considers the distance of the point from the UEF, defined as the minimum between the percentage distance on the x-axis direction and the percentage distance on the y-axis direction[4]. Here the distance is computed with respect to both x and y axis, and both measures (considered as averaged) are used to compare the performances, showing, for instance, that results by GA are the best respect to a measure (x-axe: Variance) but poor respect to the other (y-axe: Mean returns). This approach does not provide us with an unique performance measure, but in our opinion is the most suitable for grasping useful information when comparing different algorithms on the same instance.

An unique measure can be found in Streichert et al.[50] [51] [52] where the performance is given by the percentage difference between the area below the UEF (considered as reference solution) and the obtained ACEF. This approach appears to be the most reliable as areas below frontiers are computed using the S-Metric[60], explaining that the cardinality of their ACEF is limited due to the limited size of the archive used (see section 3.7) and the point on the frontier are equally distanced.

We must recall that several issues arise when trying to evaluate a frontier and compare it with other ones specially when dealing with multi-objective formulations [43] so that the comparison could turn out to be unfair and lead to misleading conclusions. To this aim also statistical tests are used [23] [10], expecially to determine if UEF and CEF are equal, and some works introduces measures to determine the best portfolio in a frontier[13], also if this approach tends to waste all the financial and preference-oriented considerations about the problem at hand.

When the goal is to compare the proposed implementation against standard tools (e.g. Branch and Bound), performance is evaluated comparing the obtained ACEF with the one obtained by standard quadratic programming packages such as LINGO[26] or CPLEX[7]. The problem is that exact methods, when constraints are imposed, generally fail to generate the whole CEF within the given run-time limit. So the comparison is made between single points on the CEF, computing an error measure for each point and afterward computing, e.g., the mean or median error over all points. If exact methods do not provide us the solution we are looking for, solutions are compared against an approximation, e.g. the solution of the continuous relaxation, if it is involved in the search process[35], or lower-bound. This approximation can be coarse, so the comparison might not be meaningful, but the procedure is widely used and represents the only way to operate.

3.2 Iterative Improvement

This strategy represents the most basic local search approach: It is easy to understand and simple to implement, but it offers the poorest performances[45][40].

[4] A similar approach is given by Fernandez and Gomez [15]

Nevertheless it can be embedded in a more complex hybrid approach, as in Glover et al.[19] where if no further improvement are made the algorithm switches to another neighborhood (this is the basic idea of Variable Neighborhood Search, see section 3.6) or in Armañanzas and Lozano[40] where a greedy search "refines" solutions found by ants (see section 3.9).

3.3 Simulated Annealing

SA was applied to the PSP generally showing good performances: Crama and Schyns[8] use SA investigating empirically each class of constraints as alone (floor, ceiling and turnover first, then trading and cardinality), then they impose simultaneously all constraints in a single model. They use three kinds of strategies:

- performing independent runs from the same initial solution;

- performing several runs, using as initial solution the best one found in the previous cycle;

- first performing a run to create a list P of promising solutions, then performing $|P|$ independent runs, using as initial solutions the ones stored in P.

The fact that there is no clear dominance amongst the these strategies can be considered as a proof of validness regarding the insensitiveness of such search-processes to the initial solution (see section 3.1.3). Their SA is able to draw out exactly the UEF but, apart from good performances when imposing floor, ceiling and turnover constraints, ACEF drawn out when imposing trading constraints appears to be quite rugged. Anyway in all cases SA was able to approximate the CEF in reasonable run-times for medium-sized instances.

Good performances of SA are found in works comparing performances of different algorithms on the same instance (see section 4). Conversely it is easy to understand, browsing the same works, that SA suffer the comparison with other strategies as Genetic Algorithms, appearing to be always the best solvers. Keeping it in mind it becomes spontaneous to try improving performances of SA by embedding this strategy in a population-based framework, i.e. mantaining a set of portfolios instead of only one portfolio and letting them evolve: This approach has been used either modeling the problem as to include cardinality constraint[28] or including a wider set of constraints (Gomez et al.[20] uses the same set of constraints introduced by [8]). The basic approach [28] consists of starting from an initial population of random portfolios and generating, for each portfolio p_o, a new portfolio p_n by selecting some assets i and modifying them according to the following rule[5]:

$$w_{x_{in}} = \max(w_{x_{io}} + s, 0) \tag{35}$$

where s is randomly chosen in the range $[-U_t, U_t]$ and this bandwidth decrease over time. Weights are then normalized to sum up to 1 and p_n is evaluated and accepted or not depending on the Metropolis function. Performing this step for each portfolio produces a new population, that is furthermore refined replacing worst portfolios either by a clone of a probabilistically selected portfolio with high fitness with probability r or by a portfolio composed of assets with average weights over best portfolios with probability (1-r). The algorithm correctly draw out the ACEF, stopping after a fixed number of steps and returning the best portfolios found.

An interesting improvement of SA can be thought for a multi-objective formulation, in which better solution are always accepted whilst worst are evaluated annealing separately each objective: In other words, a worst solution is accepted also if just only one objective satisfies the acceptance criterion, so letting other objectives worsen while offering the possibility of improve later[40]. Noticeably, this approach seems to find good solutions in the downside part of the frontier, where risk and profits are kept small.

3.4 Threshold Accepting

Threshold Accepting shares some analogies to simulated annealing: It tries to avoid getting stuck in local optima by accepting solutions which are not worse than a given threshold which is decreased progressively, assuming the value zero over the last epoch. Threshold sequence is defined by estimating the distribution of distances between objective functions of neighbor positions (a similar approach is followed in SA to determine the initial temperature).

TA was applied to the PSP by Dueck and Winker[13] and Gilli et al.[17] [18] but interestingly works using TA are rather interested in comparing risk measures, so the algorithm represents the technical mean to investigate financial aspects. In example in [13] different risk measures are compared: Variance, generalized semi-variance and geometric mean. At *first*, ACEF is drawn w.r.t. a risk measure. Afterwards other risk measures are computed for each portfolio belonging to the formerly drawn ACEF, drawing out another frontier and showing that this is generally not efficient w.r.t. the *new* risk measure chosen: This happens for each function used as *first* and *new*. In this way it is possible, also visually, to understand how risk measures are different and by how much, drawing out interesting financial conclusions. For instance it is stated that ACEF is not as smooth as it seems, since it results being

[5]Please notice that t stands for the runtime of routine, not for the temporal line over which the portfolio must be optimized: We recall once more this work does not take into account multi-period portfolio optimization.

composed of linear fragments, and the curve switches from a segment to another one when the fraction held in a particular asset changes sharply.

After using TA to solve Mean-Variance portfolio selection, Gilli and Këllezi[17] tackle a more realistic problem in a downside-risk framework in which decision variables are considered as integers (see section 5). The problem is formulated as maximizing future returns, while value-at-risk and expected-shortfall are compared as risk measures. In a further development[18] TA is used to compare three different risk measures: *Value at Risk (VaR)*,*Expected Shortfall (ES)* and Omega measure (defined as ratio of the weighted conditional expectation of losses over the weighted conditional expectation of gains) in a formulation with cardinality, upper and lower bounds constraints. Results show that mean-VaR portfolios are more diversified than those obtained with other measures, while ES frontier dominates other two (in this context it means that extreme losses are less likely using ES measure, but small losses are more likely using ES measure than others).

Interestingly all these works give evidence to the fact that much attention is to be paid to the choice of the appropriate risk measure, as efficient portfolios for a risk measure are usually not efficient w.r.t. other measures and under different utility functions efficient portfolios are very different from each other: Investors can choose the risk measure accordingly to their preferences, as this choice plays an important role in determining the final portfolio features. Furthermore, in all TA implementations by these different authors, portfolios are allowed to contain liquid cash over the search process, while this choice is not taken into account with other techniques.

3.5 Tabu search

The application of TS to the PSP has its milestone in the works of Rolland [44] and Glover et al.[19]. Indeed these works belong to different formulations of the problem and indeed Glover tackles a multi-period formulation (while our work is concerned only with single-period portfolio selection), but both deserve to be analyzed due to the richness of concepts stated, being the basis for further works and explanations. Rolland uses a tabu search for the unconstrained problem: Moving in the previously mentioned neighborhood relation (see section 3.1.2), the author tackles two problems of minimizing variance and minimizing variance given an expected level of returns, even if its paper is more oriented in finding a single point (describing the trajectory drawn by the algorithm over time to reach it) rather than drawing out the whole UEF. The two approaches differ in the repair mechanism: The *minimum variance* one, after generating five non-feasible solutions, repairs the solution as follows:

- If investment exceeds budget ($\sum x_i > 1$), find the asset with maximum sum of covariance referring to other assets (i so that $\sigma_{ij} x_i x_j$ is maximized) and decrease x_i in order to ensure feasibility;

- If investment is less than budget ($\sum x_i < 1$), find the asset with minimum sum of covariance referring to other assets (i so that $\min \sigma_{ij} x_i x_j$) and increase x_i in order to ensure feasibility.

The *minimum-variance-given-return* approach instead initially tries to reach the desired level of returns, repairing the solution as follows (after moving over five non-feasible solutions):

- find i so that
$$\left|\left(\left|1-\sum x_j\right| \cdot r_i\right) - \left|\left(\sum x_j r_j - r_p\right)\right|\right| \tag{36}$$
is minimized;

- If investment exceeds budget ($\sum x_i > 1$), decrease x_i in order to ensure feasibility, otherwise;

- If investment is less than budget ($\sum x_i < 1$), increase x_i in order to ensure feasibility.

Afterwards, 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 "home in" on the requested minimum-variance point after the requested return level has been reached. *Step* length can assume two distinct values: $\frac{1}{10}$ and $\frac{1}{100}$ and over the search process it toggles after 100 non-improving steps. Even if the proposed TS is said to offer good performances, it is useful only to find single point instead of the whole UEF, so this implementation does not represent the most powerful solution for real-world problems; nevertheless it can be useful when only one desired level of return is given.

Glover et al. tackle a problem close to the portfolio selection: The asset-allocation with fixed-mix (we defined it in section 2.6). This is a multi–period problem in which we want, for each period, to respect the proportions of asset classes (generally assets, bonds and treasury bills) to 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 (they are considered as proportional in this case) 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 assets classes with returns below average. Both cases with and without transaction costs are investigated and Tabu search is used in conjunction with *variable scaling*: With this term we indicate a strategy in which 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* length are defined and ranked in decreasing order, and an Iterative Improvement 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 used *step* size is the last of the rank. At this point if improvements were reached over the list, the process restarts from the first value, otherwise the procedure stops. This procedure has the effect of moving away from local optima by a change of the *step* size, enforcing indeed intensification until it is not possible to escape anymore from the local optimum by changing the neighborhood: At this point a TS search is performed; if improvements are gained the search switches back to variable scaling and the process continues until no improvements are made. *Step* size is crucial, and in TS it is set higher than in Variable Scaling to escape from local optimum. ACEF is compared with the frontier obtained with exact global optimization, showing that they are almost identical, in both cases with or without transaction costs. Interesting conclusions are drawn: 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.

Tabu Search has been widely applied to solve the PSP: It is easy to find it in works comparing different algorithms on the same instance (see section 4). Indeed the application of TS produced very different performances when compared with other strategies: Apart from Schaerf[45], TS is generally not competitive enough, even compared with SA[5] [4] [14]. In our opinion this defect comes from non optimized enough implementations and TS represents a very attractive and powerful strategy for the PSP.

3.6 Variable Neighborhood Search

Variable Neighborhood Search[21] (VNS) is a "metaheuristic" that dynamically changes neighborhood structures during search, so that a neighborhood is substituted by another one when the actual solution cannot be improved using the former structure. There is no explicit application of VNS to the portfolio selection problem, but, as the algorithm is very general, its principles can be found in works exploiting other kinds of algorithms. This is the case of Glover et al.[19], where the implementation of *Variable Scaling* can be easily re-conducted to a VNS, as a new neighborhood is exploited changing the *step* when no further improvement is possible, also if the problem is in the multi-period formulation and neighborhoods with the same cardinality are defined. The ideas underlying this approach can be found in Ehrgott et al.[14], where search switches between two neighborhoods.

A special mention must be given to Speranza[48], in which an heuristic is defined and applied to Milan Stock Market using an integer formulation. Here, in order to satisfy 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 increase is unsuccessful, decreased) by one unit: If the new solution is feasible, the algorithm stops, otherwise the procedure is repeated over $x_2 \dots x_n$. If no feasible solution is found, the cycle is repeated increasing assets by two units, then three and so on. This mechanism represents a naive VNS strategy, even if the neighborhood cardinality appears to be kept constant over the whole process and the neighbor selection process is deterministic.

3.7 Evolutionary and Genetic Algorithms

The terms Evolutionary Strategies and Genetic Algorithms assume different meanings in literature: The first is used when decision variables are represented by their actual real value, whilst the latter is used when real decision variables are encoded with a binary representation. We consider these differences as details of the implementation, encompassing in this section works belonging to both strategies.

The first applications of Evolutionary algorithms to the unconstrained PSP are given by Tettamanzi et al.[1][33] [34]. Initially[1] they applied genetic algorithms to portfolio selection (using a down-side measure of risk) in a sequential machine, in which only one population is handled and individuals are generated according to investor preferences: A specie is defined for each λ (where λ is the trade-off coefficient between return and risk); individuals are generated belonging to a specie with a probability proportional to the investor's interest in that species and for each new generation a new individual replace the worst in the previous one. As both portfolio selection problem and genetic algorithm are plenty of scope for parallelization, in a further work[34] a distributed genetic algorithms is applied in which, using parameterized objective function, each λ value is associated to a subpopulation. Obviously, as AUEF is composed drawing out a point for each λ, the greater the cardinality of populations, the finer the resolution of the frontier. The algorithm is implemented so as to permit migrations of individuals between populations referring to neighbor values of λ in order to avoid premature convergence, and individuals are allowed to mate only with individual of the same population or of adjacent ones (the same holds for the previous sequential implementation, but in that case we handle different species rather than populations). This implementation outperforms the previous sequential version, and in [33] an enriched outline of implementation and risk measures is given, even if results does not provide further novelties or improvements.

Performances of GA are generally not affected by imposing constraints: If only cardinality constraint is imposed is it possible to search in parallel several ACEF corresponding to each number of requested assets k we are interested in, using information from each of these to improve the search process of others. With this approach ACEF is shown to approach UEF as k increases and a statistical test is performed showing that constrained optimal portfolios are not significantly different from unconstrained ones, except for tiny number of assets and lowest level of risk. Cardinality constrained PSP is also tackled by Streichert et al.[50][51][52] using a two-objective optimization structure, 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 discussed in section 3.1.1, comparing it with the *standard* one and they further investigate the use of *lamarckism*: To better explain the meaning of this term, we must precise that these works embed a repair mechanism that prevents to reject infeasible solutions (The mechanism is explained in section 3.1.2). We must recall that the genetic process provides us, for each portfolio, a vector consisting of non-normalized values. For the sake of simplicity, we say that this vector represents the genotype, while the normalized vector(so that their weights sum to one) represents the phenotype. In each case mutation and crossover operators affect each genotype components Z and X separately from each other, simplifying asset insertion and removal in the portfolio, but in experiments *without lamarckism* only the phenotype of an individual is altered by the repair mechanism, while the genotype remains unaltered; in experiments *with lamarckism* the repair mechanism modifies the genotype too, according to the phenotype. In each case, this representation performs better than the *standard* one, but noticeably lamarckism helps improving standard algorithms performances too. Furthermore different variable representations (binary and real-valued) are compared and different coding [51] and crossover operators [52] are examined. Indeed these works investigates first the case in which cardinality constraint is imposed alone, then the case in which this constraint is imposed in conjunction with floor and minimum lots. Strategies used in these works can be better encompasses in *memetic algorithms*, as a local search for feasible solutions is added to improve performances of EA: This approach has been exploited in [37], comparing cases in which the local search steps are performed by both SA and TA. It turns out that TA is more suitable when the search space is rough, so when *VaR* is used as risk measure, whilst when using such a measure as *ES* the search space appears smoother, and SA performs better. In this work the introduction of *elitist* is investigated: With this term we refer to a strategy in which at a certain stage of the local search procedure, the next state is determined by comparing the the current solution (or, in cases of populations-based-algorithms, the solution of one agent) with the best solution found so far, instead of one of its neighbors. This strategy is used as the SA and TA acceptance criteria allows moving away from the current optimum, in order to escape from local optima. The principle is good, but it can be disadvantageous if the search space is smooth. The introduction of elitists, besides making the algorithms less dependent to parameter setting, reduces this drawback, as witnessed by the fact that performances using *ES* as risk measure are greatly improved.

The case in which floor, ceiling and cardinality constraints are imposed is the most commonly used in literature when applying GA [32][5] [14] [15], accordingly to how stated in section 2. Minimum lots are noticeably encountered not only when dealing with integer formulation ([9], where also transaction costs are taken into account, see section 3.9 and 5) but also when dealing with the continuous model[50] [51] [52].

The extremely high potential of GA, together with the enormous interest provoked in the worldwide scientific community, caused a high degree of experimental works also for the PSP, in which GA where also used in conjunction with formulation differing from the canonical Mean-Variance one in order to define more realistic customer oriented frameworks. An interesting example is represented by Xia and coworkers[58], where the objective function to maximize is given by equation 9 but they solve this objective for different values of λ rather than drawing out the whole frontier, showing that in the obtained portfolios return is higher than the optimal one provided by optimization software for Mean-Variance (LINGO[26]) even if they are more risky. Furthermore the expected return is considered as a variable, belonging to 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. This is explained by the following constraints:

$$r_i \geq r_{i+1} \qquad i = 1 \ldots n-1 \tag{37}$$

$$a_i \leq r_i \leq b_i \qquad i = 1 \ldots n \tag{38}$$

$$x_i \geq 0 \qquad i = 1 \ldots n \tag{39}$$

Three ways for assessing returns and variance are described: Mean of historical data, single-index and multi-index models. Single index relies on the correlation between stock returns and returns of a market-index, while multi-index denotes return-correlation with several indexes (or factors) to capture non-market influences. The genetic algorithm creates a pre-defined number of chromosomes (portfolios) in which assets are ordered according to their expected returns (equation 37) using a weighted sum of arithmetical mean, historical return tendency and balance-sheet based forecast of future returns. Afterwards, portfolios are ranked according to their objective value, and this rank is used to probabilistically define the crossover arguments as the higher the position, the higher the likelihood to be selected as parents (with no regard to the actual evaluation function value). Indeed crossover is performed between an individual and its closest neighbor, and after crossover mutation is performed. At this stage the new population is ready for the next evaluation. V-Shaped transaction costs are also investigated for portfolio revision, but they are only considered as proportional[6]. Transaction costs (embedded in a MAD objective function) and single λ values analysis are also exploited in Wang et al.[55] in which a sample procedure for

[6] In a further work[56] risk-free asset are introduced and the formulation is based on a linear program in a compromise formulation.

stochastic returns is introduced instead of the classical scenario analysis (see section 1.1).

Other approaches are pursued, generally developing a more complex and tailored architecture for practitioners in order to help decision making introducing other measures either to define an ordinal-preference framework in which other measures are added to the formulation [53] [14], or to predict the future return rate and to obtain the uncertainty risk of the future return rate when the sample is small [42].

We must finally notice that comparative studies stated that GA represents the best performing solver respect to other "metaheuristics" such as TS, SA and Iterative Improvement. This is due to the fact that GA generates, in each step of the search process, solutions in different areas of the search landscape, allowing to exploit a better diversification: This feature is compelling when using a multi-objective formulation and indeed the most used genetic algorithm for the PSP are encompassed in the family of MOEAs (MultiObjective Evolutionary Algorithms) [10] [50] [51] [52] [9] [42].

3.8 Particle Swarm

The new nature-inspired paradigm referred to as *Particle* swarm represents a teasing alternative to other search paradigms. Indeed, its application to the PSP is nowadays still limited, and the few works on this topic does not tackle the standard formulation, being rather oriented in finding an unique portfolio optimal w.r.t a measure (the reward-to-variability ratio, referred to as *Sharpe ratio*) out of a given set of assets than drawing out the whole efficient frontier[29][39].

3.9 Ant Colony Optimization

ACO has found not yet enough application to the PSP, if we think about excellent results achieved in combinatory optimization. Initially conceived for discrete spaces, ACO has been tailored for continuous spaces too[47] so it can be applied to handle the Markowitz PSP formulation. Anyway little efforts was made in that direction.

ACO can be successfully applied when cardinality constraint is imposed: The common approach is to define populations of n ants in order to explore a completely connected graph composed of n nodes, so as they can easily move from an asset to another. The number of populations can be related to the number of objective functions to optimize[40] or, if there is one only function, can be just defined to optimize parameter-tuning[36]. In [40] when satisfying the convergence criterion, a greedy search refines the solution. Noticeably, compared with other techniques (SA and II) this approach seems to find best solutions in the upside part of the frontier, where risk and profits are kept big.

ANT colony optimization has found application in problems similar to portfolio selection one like the so-called multi-objective project portfolio selection, a generalization of the bin-packing problem in which we want 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[11][12].

We must recall that Ant Colony Optimization was initially conceived for discrete search space, while in the problem we are considering variables can assume any continuous value belonging to [0, 1] range. If we would like to use this technique for our problem, the basic idea would be to discretize the space: This operation might not be conceptually sound since we only consider that assets weights must sum up to one, regardless of the total amount to be invested. We might decide to apply a discretization at 0.000005 intervals, without any trouble for the formulation, but it is clear that if we have to invest 1,000,000,000 euros the discretized minimum admissible lot will be 5,000 euros, while if we consider 10 euros to be invested it will amount to 0.00005 euros. This will not turn into errors or warnings, but it is clear that the meaning of the efficient frontier could be strongly misleading depending on the invested amount. We already stated that a version of ACO was tailored for continuous space, so the issue is not so heavy, but it is important in order to understand the importance of the formulation; the trouble is more marked if we consider transaction costs (they are often not taken into account in works about "metaheuristics"): If they are considered only as proportional they can be easily inserted in the formulation, while a cost function as defined in [30] requires formulation efforts that, to date, did not bring satisfactory results. In our opinion further research must be aimed at trying to define a standard-formulation in order to take into account transaction costs and to define a versatile "metaheuristic" approach, so that different risk measures can be used (a comparison amongst several risk measures should also be extensively pursued in further research). The dilemma is about the integer formulation (as defined in section 5) versus continuous fractional formulation (in which weights must sum up to one), universally used as standard approach in "metaheuristics" formulation.

The local search approach is, in our opinion, robust w.r.t the formulation, so it is able to handle the integer version too, ensuring important advantages: Lack of necessity of discretisation, correctness of meanings of formulation, easiness in including transaction costs and rounds and we hope researchers will switch their attention in this direction.

4 Comparative studies

Comparing works and implementation by different authors turns out to be a difficult task: The most common trouble is that each author claims the superiority and the advantages of his model, but too often it is not possible to fully justify this assertion as too many factors enter into account: Data-sets are not the same, algorithm implementations can be different and lead to wrong comparison, shortcomings of actual implementation are not clearly stated, utility and performances measures are different; furthermore ad-hoc implementations are created to fit a data at hand and the model performs well over those data, without guarantee of robustness to data-set-features (say, without warranty that this model can be successfully applied to other data-sets). We must indeed note that different works can pursue different goals: One can just test the efficiency of his algorithms; others can aim to test the suitableness regard to the problem formulation; others can aim to discard the standard problem-formulation (because it is not able to capture real-world-features) and to claim their formulation and algorithm are more appropriate; others wants to create a model with good performances to help decision making in professional activities.

Due to those factors, comparison amongst several works is not possible: Our sole valuable resources in that goal are works comparing different algorithms on the (purportedly) same instance.

One of the first works of this kind was by Catanas[4], but his paper is more oriented in investigating properties of the proposed neighborhoods (see section 3.1.2) than to develop a solution for the portfolio problem. He uses TS and SA, implemented in both *robust* and *dynamic* way: In the *robust* implementation *step* is kept fixed during all iterations, while in *dynamic* one it is let decreasing as to tend to zero during the execution. Furthermore he defines a schema for the variation of *step*, letting its value to increase if solution quality worsens, to decrease if solution quality improves, and defines a threshold on the minimum value of *step*, as tiny values can make the search stagnate. The variation scheme is the following:

$$step = \begin{cases} \min\left(\frac{1}{n-1}, F_{step} \cdot \frac{1}{1+|\overline{\Delta}|} \cdot step\right) & \text{if } \overline{\Delta} < 0 \\ \min\left(\frac{1}{n-1}, F_{step} \cdot (1+|\overline{\Delta}|) \cdot step\right) & \text{otherwise} \end{cases} \tag{40}$$

where F_{step} is a given number belonging to the range (0, 1] and $\overline{\Delta}$ is the difference between values of the objective function at runtime t_x and t_{x+1}. Results show that SA performs better than TS and dynamic implementation is more performing than robust in both SA and TS; furthermore *robust TS* performances are extremely sensitive to *step* values and *dynamic* version performs better when $F_{step} = 1$ for both TS and SA.

Chang et al.[5] introduce cardinality and minimum and maximum holding constraints. They state that when introducing those constraints, 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 may be invisible (due to minimum proportion constraints) for a classical λ-weighting drawing approach. They use an evaluation procedure to ensure feasibility of all solutions and use GA, TS and SA to solve the problem. Results show that GA is able to better approximate the UEF with 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 three heuristics 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. Sets obtained by the three heuristics are then pooled to draw the ACEF. This approach shows that for the constrained problem ACEF approximate UEF one when increasing the asset cardinality k (as already stated in [23], see section 3.7).

Results obtained by [5] are compared, in Jobst et al.[24], with two heuristic methods: The first is an *integer-restart* procedure drawing out CEF starting from the highest return and its corresponding risk to lower return and reduced risk, in which the result obtained is supplied as starting point to the next (lower return) iteration, considering it as first feasible and upper bound value[7]; the second, following an idea similar to [48] (see section 5), first solves a continuous relaxation without any constraints, then uses the k highest weights as input of a problem in which constraints are imposed (this heuristic is referred to as *re-optimization heuristic*). Both heuristics are embedded in a branch-and-bound and are said to outperform "metaheuristics" used in [5]. Anyway, if this assertion can be shown to be true for integer-restart heuristic, we must note that re-optimization heuristic is not able to draw the whole frontier: Points are missing when the first continuous relaxation produces a portfolio with less than k assets, and this should be noted as drawback, even if the obtained performance measure overall drawn points is better than all procedure introduced in [5].

Schaerf[45] takes into account floor, ceiling and cardinality constraints. He defines three neighborhood relations, in which all but the budget constraints are to be satisfied, defining the objective function as mentioned above (see section 3.1.4). The initial state is selected as the best amongst 100 randomly generated portfolios with k assets. Initially he uses *best* and *first* Iterative improvement, SA and TS as single solvers, then he uses TS (said to be the most promising solver in his implementation) combining neighborhood relationships in various token-ring strategies. In this case the *step* length has a great impact in the first solver to attain diversification, while is set at a smaller value in the last solvers to attain intensification; furthermore the *step* length assessment can either have a fixed value for each iteration or include a random component. Experimental results show that the best performances are obtained by token ring solvers with random steps, even if fixed steps seem to behave well too. Single solvers do not attain comparable good results.

Armañanzas and Lozano[40]compare II, SA and ACO in a multiobjectives formu-

[7]The process of providing the actual solution as input of the next search iteration is defined as *warm-start*.

lation with cardinality, floor and ceiling constraints. Algorithms used are tailored to the multiobjective problem, and ACO seems to outperform other techniques, whilst the simple greedy search (II) offer poor performances if used alone, while its use is valuable when refining solutions provided by ACO. Interestingly, ACO and SA best performances are found in different areas of the frontier: The first in the upside part of the frontier, the latter downside.

Also Ehrgott et al.[14] 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 heuristic local search similar to a VNS embedding a random escape mechanism to avoid stagnation at a local minimum, testing them over both random and real instances. Results on both instance classes show that GA appears to be the most performing solver. Even the heuristic-local search and SA obtain good results, while TS performances appear to be the worst ones.

Different judgement can be found comparing those studies: In Schaerf Tabu Search appears to be the most promising solver, while in Ehrgott et al. is said to be the less reliable. It worths to notice here that in both Ehrgott et al. and Chang et al. GA represents the most performing solver, whilst unluckily there is no possibility for a thoroughly evaluation of ACO due to the lacking of comparative studies applying it.

A further interesting comparison is made by Fernandez and Gomez[15], in which "metaheuristics" by Chang et al. are compared with a neural net approach: An Hopfield network[8] is used to draw out the ACEF when cardinality constraint and bounds (lower and upper bounds) are imposed in the formulation. Their results show that there is no significative difference between their neural network and "metaheuristics" such as GA, TS and SA. In order to improve performances, portfolios from the four approaches are pooled and dominated ones are deleted, so obtaining a better ACEF (the same approach pursued by Chang). Noticeably, the portion of this frontier composed of portfolios formerly obtained by neural net are strongly higher than the portion composed of portfolios by other heuristics: This give us evidence about the quality of neural net results[9]. Nevertheless, the total amount of portfolios computed by neural net is strongly lower than the total portfolios computed by other heuristics so, even if the quality is high, stand-alone results by neural nets are not suitable for a complete analysis.

[8]Hopfield networks[22] are neural network composed of a single layer of neurons fully connected and are widely applied in combinatorial optimization[46].

[9]The neural net skill of grasping non linear and underlying-model-lacking relations amongst variables is well suited for forecasting future returns without relying on the assumption of normally distributed stock returns. This idea has been exploited in [49] and [59] in order to optimize portfolio-management, even if the two implementation are different.

5 Other heuristic approaches

For sake of completeness, in this section we briefly review works that cannot be said belonging to local search, being oriented to linear programming. Indeed their algorithms seems to be too problem-specific instead of defining a general strategy, but these approaches are useful to understand the problem and can serve as components of more robust strategies. These works are important because they introduce the integer formulation, in which assets assume integer (or anyway discrete) values corrisponding to the actual amount of money to be invested in each asset. This formulation can easily obtained from the continuous Mean Variance formulation replacing (2) and (3) by other equations in order to impose that the return of the portfolio must be equal to an amount of money obtained by multiplying the initial wealth times the expected return (42) and that asset weights must sum up to the initial weight C (43), even if in the works we consider in this section this requirement is relaxed allowing that assets weight can be included between a lower and upper bound (see section 2.1).

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j \tag{41}$$

subject to

$$\sum_{i=1}^{n} r_i x_i = r_p \cdot C \tag{42}$$

$$\sum_{i=1}^{n} x_i = C \tag{43}$$

$$x_i \quad integer \qquad i = 1, \ldots, n \tag{44}$$

Besides works we are going to introduce, integer formulation was also applied in conjunction with *proper* meta-heuristics[9][13][17][18], even if the continuous Markowitz model is the most commonly and widely used. Speranza[48] models the problem including transaction costs, minimum lots, cardinality, floor and ceiling constraints, introducing two dummy (binary) variables to determine if a security has fixed transaction costs and if it belongs to the portfolio. As the introduction of integer variables make this problem difficult to solve when the number of assets or desired return rate increases, the idea is to relax the integer constraint on quantities, transforming the problem to a linear programming one (to be solved efficiently even when the number of securities is high) and finding a solution to it. Obtained asset weights are afterwards rounded to the closest integer and heuristics are applied to force the solution satisfying capital and rate of return requirements. If the algorithm ends without solutions, bounds on capital are iteratively re-defined in less restrictive way: In this way solutions are easier to find. The algorithm (referred to as *ROUND–LP*) shows good performances when the total number of assets is low and when the capital is large this heuristic method obtains a solution close to the optimal one.

In Mansini and Speranza[35], the formulation of the problem includes minimum lots and proportional taxes. After stating that in presence of minimum lots the problem of finding a feasible portfolio is NP-complete, they provide three heuristic algorithm based upon the idea of solving sub-problems of the original formulation, involving subsets of initial universe of assets.
In the first heuristic (referred to as *SINGLE-LP*) they solve the relaxation of the problem, then using the vector of asset quantities as input of *MILP* procedure.
As desirable assets can be excluded from the final solution, the second heuristic (referred to as *Reduced–cost–MILP–heuristic*) considers a number of assets greater than the vector x_R of assets i s.t. $x_i \neq 0$ as input of MILP-procedure, so including also assets whose quantity in the solution of the relaxed problem is 0.
The third method consists in an iterated routine: After solving the relaxed problem, the vector x_R (previously defined) is used as input of a MILP procedure. After each step, half of assets i s.t. $x_i = 0$ is deleted and half is replaced in the solution. Process ends when a limit number of securities has been considered. This third heuristic is the most effective but requires more computational time, even if the best solution found with these heuristics methods is obtained after less time than exact ones. These heuristics performs reasonably better than simple problem specific heuristics proposed in [48] and they have the advantage of being more general and are also used in Kellerer et al.[27] in a formulation enriched by introducting fixed transaction costs and minimum lots. Heuristics are applied over different models:

- the first refers to the case in which investors pay a fixed amount (≥ 0) for each security in the portfolio;
- in the second the previous model is enriched because of the introduction of rounds;
- the third refers to the case in which fixed cost is applied if the amount of money invested in a security exceeds a minimum threshold (this is a way to facilitate small investors);
- the fourth represents the generalization of the third by introducing rounds.

The conclusion of this work is that the introduction of fixed costs reduces the number of securities in the optimal portfolio, and this effect is more evident when introducing rounds. Increased fixed costs lead to a more risky portfolio as this will include more risky securities with higher rate of return, but diversification depends more on the introduction of fixed costs rather than their increase, as the investor bears their burden when they are introduced, independently from the quantity of assets.

The same heuristic approach can be used to solve the *mutual funds portfolio selection problem*: Mutual funds are portfolios of assets and securities, and they have become a strong form of saving. As several funds are available on the market, being generally offered by banks as a form of fidelization, the investor is asked to choose which funds to select and how much money to invest in each of them in order to achieve the lowest risk for each given return rate. Chiodi et al.[6] tackle this problem considering minimum lots, entering and management commissions, implementing *round-lp*, *single-lp* and another heuristic referred to as *mult-lp*. In this last heuristic they define a set of pairs of capital and return rate, solving the relaxed continuous formulation for each of them. Selected funds for each pair (funds appearing in the solution of the relaxed problem) are pooled in a set S: This set is used to solve the original integer formulation for each pair capital-return. Using monthly data, this procedure almost always finds the optimal solution, and error does not increase significantly when the size increases but decreases when required return (or capital) increases. This heuristic performs better than *round-lp* and *single-lp*, even if the computational cost is higher. Notably, the average number of selected funds decreases when expected return rate increases.

The shortcoming of these works is the heuristic error measurement: This is defined measuring the deviation from the optimal solution supplied by exact methods for the same instance. When exact method fail in finding optimal solutions, the deviation is measured referring to the solution of the relaxed formulation, entailing an overestimation of error. For low levels of capital this solution does not represent a good approximation of the optimum, but the higher the invested capital (or the expected rate of return) the higher the explanating power of this value. We must finally consider that the more complex the heuristic, the more accurate the result, but at the same time, the higher the computational time: This can represent a crucial point for the choice of the heuristic to use, as for lower values of capital or return rate simpler heuristics seem to provide good performance. In this case the use of a more complex strategy can reveal itself useless, just improving the performance measurement of few decimal digits and increasing the running time of more than one order of magnitude.

The best advantages of these works is their clear approach and formulation, as they give a good explanation of models, variables, constraints: Generally "meta-heuristic" oriented papers show lack of clarity in these points that are instead essential for a better understanding of the problem. However the greatest difference with the literature is the introduction of integer variables and in our opinion this formulation better captures the underlying problem and facilitates the introduction of additional features.

5.1 Effect of costs

We decide to present here a short overview about costs because they have been discussed in the above cited works, but there are just a few works in the literature giving a sufficient outlook and treating them correctly in proposed "metaheuristics". As stated in Konno and Wijayanayake[30] 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 decrease as size increase), then increases linearly to another certain point (unit-transaction costs are here constant) and then becomes convex due to the illiquidity premium (unit prices increases due to the shortage of supply). Total transaction costs are function of several variables: VAT rate, fixed costs, brokerage rate, illiquidity premium,transaction size,marketable securities, tax rate. In this scheme an important role is played by illiquidity premium: It can be introduced in different ways in the formulation but there is empirical evidence that it is discontinuous over the amount of transaction. Indeed it is general opinion illiquidity premium function being smooth, but this assumption simplifies real-world features.

Transaction costs can be plotted as a V-Shaped function and this representation provide us with a realistic way for taking global transaction costs into account. Nevertheless just a few authors embedded it in their formulation (e.g. [9], [54], [58]). More in general, the *statum-of-the-art* "metaheuristics" literature lacks in including transaction costs.

We must consider that, even if Modern Portfolio Theory states that diversified portfolio are preferable to undiversified ones, there is evidence that investors choose undiversified portfolios. This is due to the action of transaction costs, since they were not included in the original model. Considering all typologies, global transaction costs tend to reduce portfolio diversification: This is partially due to the introduction of fixed costs, while proportional ones do not have effects because they generate only a decrease in returns rate. Note that this assertion must be taken *cum grano salis*, because investor behavior depends on subjective factors too: In Glover et al.[19] 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.

Conclusions

In this work we defined a framework for classifying metaheuristic approaches for the PSP, introducing the main aspects of the problem (objective function and constraints), the general concepts about the application of local search techniques to this problem and the main strategies used to solve the problem.

The PSP is only a representative of a class of problem consisting on the management of portfolio of several nature. There is plenty of scope for applying "metaheuristic" techniques to this classes of problems as to date they appear to be not investigated enough: Indeed features of "metaheuristics" make them suitable to solve several problems concerning "portfolios", since the usual PSP can be viewed in different functions and at different abstraction levels (portfolio of assets, optimal funds consisting in several portfolios, combining mutual funds with different risk profiles etc.). These problems can be viewed either in a static formulation (as the Markowitz one) or over a temporal line, hence issues of *re-balancing*, *index-tracking* and *re-optimization* arise; furthermore we can consider also financial portfolios, in which the risk-assessment plays an important role as they consist of credits. Considering also slightly different classes of problems we find the *project-portfolio* problem, the *product-portfolio* problem (in which a firm must chose the mix of products to be manufactured or sold on the market). Due to several constraints imposed in order to satisfy management requirements and to make the model the most realistic as possible, conventional methods are not able to solve exactly the problem, so each of this classes can be successfully attacked with "metaheuristics".

Indeed a further effort must be tackled in order to ground metaheuristics approaches with theoretical and empirical achievements about the PSP. It may turns to be useful to move away from the canonical Mean-Variance approach in order to include real-world features and to help the analysis of markets: To this extent it is necessary to develop a framework in order to reflect customer behaviour, also including transaction costs, recalling that statum-of-the-art literature lacks in including them in the analysis.

References

[1] S. Arnone, A. Loraschi, and A. Tettamanzi. A genetic approach to portfolio selection. *Neural Network World, International Journal on Neural and Mass-Parallel Computing and Information Systems*, 3(6):597–604, 1993.

[2] J.E. Beasley. Or-library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.

[3] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, September 2003.

[4] F. Catanas. On a neighbourhood structure for portfolio selection problems. Technical report, Departamento de Metodos Quantitativos do ISCTE, Lisboa, Portugal, 1998.

[5] T.J. Chang, N. Meade, J.E. Beasley, and Y.M. Sharaiha. Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27(13):1271–1302, 2000.

[6] L. Chiodi, R. Mansini, and M.G. Speranza. Semi-absolute deviation rule for mutual funds portfolio selection. *Annals of Operations Research*, 124:245–265, 2003.

[7] CPLEX Optimization Inc. *Using the CPLEX callable library and the CPLEX mixed integer library*, 1993.

[8] Y. Crama and M. Schyns. Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research*, 150:546–571, 2003.

[9] S. Wang D. Lin and H. Yan. A multiobjective genetic algorithm for portfolio selection. Technical report, Institute of Systems Science, Academy of Mathematics and Systems Science Chinese Academy of Sciences, Beijing, China, 2001.

[10] L. Diosan. A multi-objective evolutionary approach to the portfolio optimization problem. In *International Conference on Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*. IEEE Conference Proceedings, 2005.

[11] K. Doerner, W. Gutjahr, R. Hartl, C. Strauss, and C. Stummer. Ant colony optimization in multiobjective portfolio selection. In *Proceedings of the 4th Metaheuristics Intl. Conf., MIC'2001, July 16–20, Porto, Portugal*, pages 243–248, 2001.

[12] K. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, and C. Stummer. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of Operations Research*, 1:79–99, 2004.

[13] G. Dueck and P. Winker. New concepts and algorithms for portfolio choice. *Applied stochastic models and data analysis*, 8:159–178, 1992.

[14] M. Ehrgott, K. Klamroth, and C. Schwehm. An MCDM approach to portfolio optimization. *European Journal of Operational Research*, 155(3):752+, 2004.

[15] A. Fernandez and S. Gomez. Portfolio selection using neural networks. *Computers & Operations Research*, 2005.

[16] K.V. Fernando. Practical portfolio optimization. Technical Report TR2/00 (NP3484), NAG Ltd, Oxford, UK, 2000.

[17] M. Gilli and E. Këllezi. A global optimization heuristic for portfolio choice with VaR and expected shortfall. In *Computational Methods in Decision-making, Economics and Finance*, Applied Optimization Series. Kluwer, 2001.

[18] M. Gilli, E. Këllezi, and H. Hysi. A data-driven optimization heuristic for downside risk minimization. *The Journal of Risk*, 3:1–19, 2006.

[19] F. Glover, J.M. Mulvey, and K. Hoyland. Solving dynamic stochastic control problems in finance using tabu search with variable scaling. In *Proceedings of the Meta-Heuristics International Conference MIC-95*, pages 429–448. Kluwer Academic Publishers, 1995.

[20] M.A. Gomez, C.X. Flores, and M.A. Osorio. Hybrid search for cardinality constrained portfolio optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation, Seattle, USA*, pages 1865–1866. Elsevier Science, 2006.

[21] P. Hansen and N. Mladenović. An introduction to variable neighborhood search. In S. Voß, S. Martello, I. Osman, and C. Roucairol, editors, *Meta-heuristics: advances and trends in local search paradigms for optimization*, chapter 30, pages 433–458. Kluwer Academic Publishers, 1999.

[22] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. In *Feynman and computation: exploring the limits of computers*, pages 7–19. Perseus Books, Cambridge, MA, USA, 1999.

[23] J. Matatko J.E. Fieldsend and M. Peng. Cardinality constrained portfolio optimisation. In *Proceedings of the Fifth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'04), Lecture Notes in Computer Science (LNCS 3177)*, pages 788–793. Springer, August 2004.

[24] N.J. Jobst, M.D. Horniman, C.A. Lucas, and G. Mitra. Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. *Quantitative Finance*, 1:489–501, 2001.

[25] J.G. Kallberg and W.T. Ziemba. Comparison of alternative utility functions in portfolio selection problems. *Management Science*, 29:1257–76.

[26] J. Kallrath. *Modeling Languages in Mathematical Optimization.* Springer, 2004.

[27] H. Kellerer, R. Mansini, and M.G. Speranza. On selecting a portfolio with fixed costs and minimum transaction lots. *Annals of Operations Research*, 99:287–304, 2000.

[28] H. Kellerer and D. Maringer. Optimization of cardinality constrained portfolios with a hybrid local search algorithm. *OR Spectrum*, 25(4):481–495, 2003.

[29] G. Kendall and Y. Su. A particle swarm optimisation approach in the construction of optimal risky portfolios. In *Proceedings of the 23rd IASTED International Multi-Conference ARTIFICIAL INTELLIGENCE AND APPLICATIONS*, pages 140–145, 2005.

[30] H. Konno and A. Wijayanayake. Mean-absolute deviation portfolio optimization model under transaction costs. *Journal of the Operational Research Society of Japan*, 42(4), 1999.

[31] H. Konno and H. Yamazaki. Mean absolute deviation portfolio optimization model and its applications to Tokyo stock market. *Management Science*, 37:519–531, 1991.

[32] S. Liu and D. Stefek. A genetic algorithm for the asset paring problem in portfolio optimization. In *Operations Research and Its Application. Proc. First International Symposium, ISORA*, pages 441–449, 1995.

[33] A. Loraschi and A. Tettamanzi. An evolutionary algorithm for portfolio selection within a downside risk framework. In *Forecasting Financial Markets*, Series in Financial Economics and Quantitative Analysis, pages 275–285. Chichester, 1996.

[34] A. Loraschi, A. Tettamanzi, M. Tomassini, and P. Verda. Distributed genetic algorithms with an application to portfolio selection problems. In D. W. Pearson, N. C. Steele, and R. F. Albrecht, editors, *Artificial Neural Networks and Genetic Algorithms*, pages 384–387, Wien, 1995.

[35] R. Mansini and M.G. Speranza. Heuristic algorithms for the portfolio selection problem with minimum transaction lots. *European Journal of Operational Research*, 114(2):219–233, 1999.

[36] D.G. Maringer. Optimizing portfolios with ant systems. In *International ICSC congress on computational intelligence: methods and applications*, pages 288–294, 2001.

[37] D.G. Maringer and P. Winker. Portfolio optimization under different risk constraints with modified memetic algorithms. Technical Report 2003-005E, University of Erfurt, Faculty of Economics, Law and Social Sciences, 2003.

[38] H. Markowitz. Portfolio selection. *Journal of Finance*, 7(1):77–91, 1952.

[39] L. Mous, V.A.F. Dallagnol, W. Cheung, and J. van den Berg. A comparison of particle swarm optimization and genetic algorithms applied to portfolio selection. In *Proceedings of Workshop on Nature Inspired Cooperative Strategies for Optimization NICSO 2006*, pages 109–121, 2006.

[40] R. Arma nanzas and J.A. Lozano. A multiobjective approach to the portfolio optimization problem. In *The 2005 IEEE Congress on Evolutionary Computation, CEC 2005*. Springer Verlag, 2005.

[41] D.N. Nawrocki. Portfolio optimization, heuristics and the "Butterfly Effect". *Journal of Financial Planning*, pages 68–78, Feb 2000.

[42] C.S. Ong, J.J. Huang, and G.H. Tzeng. A novel hybrid model for portfolio selection. *Applied Mathematics and Computation*, 169:1195–1210, October 2005.

[43] L. Paquete and T. Stuetzle. A study of stochastic local search algorithms for the biobjective QAP with correlated flow matrices. *European Journal of Operational Research*, 169:943–959, 2006.

[44] E. Rolland. A tabu search method for constrained real number search:applications to portfolio selection. Technical report, Dept. of accounting and management information systems, Ohio State University, Columbus. U.S.A., 1997.

[45] A. Schaerf. Local search techniques for constrained portfolio selection problems. *Comput. Econ.*, 20(3):177–190, 2002.

[46] K.A. Smith. Neural networks for combinatorial optimization: a review of more than a decade of research. *INFORMS J. on Computing*, 11(1):15–34, 1999.

[47] K. Socha. ACO for Continuous and Mixed-Variable Optimization. In Marco Dorigo, Mauro Birattari, and Christian Blum, editors, *Proceedings of ANTS 2004 – Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence*, volume 3172 of *LNCS*, pages 25–36. Springer-Verlag, Berlin, Germany, 5-8 September 2004.

[48] M.G. Speranza. A heuristic algorithm for a portfolio optimization model applied to the Milan stock market. *Comput. Oper. Res.*, 23(5):433–441, 1996.

[49] M. Steiner and H. G. Wittkemper. Portfolio optimization with a neural network implementation of the coherent market hypothesis. *European Journal of Operational Research*, 1(3):27–40, july 1997.

[50] F. Streichert, H. Ulmer, and A. Zell. Evolutionary algorithms and the cardinality constrained portfolio optimization problem. In *Selected Papers of the International Conference on Operations Research (OR 2003)*, pages 253–260, Heidelberg, Germany, 3-5 September 2003. Springer Verlag.

[51] F. Streichert, H. Ulmer, and A. Zell. Comparing discrete and continuous genotypes on the constrained portfolio selection problem. In *Genetic and Evolutionary Computation Conference - GECCO 2004*, volume 3103 of *LNCS*, pages 1239–1250, Seattle, Washington, USA, June 26-30 2004. Springer Verlag.

[52] F. Streichert, H. Ulmer, and A. Zell. Evaluating a hybrid encoding and three crossover operators on the constrained portfolio selection problem. In *Congress on evolutionary computation (CEC 2004) Portland, OR, USA, Proceedings part I*, pages 932–939, 2004.

[53] R. Subbu, P. Bonissone, N. Eklund, S. Bollapragada, and K. Chalermkraivuth. Multiobjective financial portfolio design: A hybrid evolutionary approach. In *IEEE Congress on Evolutionary Computation (CEC 2005), Edinburgh UK*, 2005.

[54] G. Vedarajan, L.C. Chan, and D.E. Goldberg. Investment portfolio optimization using genetic algorithms. In *Late Breaking Papers at the Genetic Programming 1997 Conference*, pages 255–263. Stanford Bookstore, Stanford University, California, 1997.

[55] S.M. Wang, J.C. Chen, H.M. Wee, and K.J. Wang. Non-linear stochastic optimization using genetic algorithm for portfolio selection. *International Journal of Operations Research*, 3(1):16–22, 2006.

[56] Y. Xia, S. Wang, and X. Deng. Compromise solution to mutual funds portfolio selection with transaction costs. *European Journal of Operational Research*, 134(3):564–581, 2001.

[57] M.R. Young. A minimax portfolio selection rule with linear programming solution. *Management Science*, 44(5):673–683, 1998.

[58] S. Wang Y.S. Xia, B.D. Liu and K.K. Lai. A model for portfolio selection with order of expected returns. *Computers & Operations Research*, 27(5), 2000.

[59] H.G. Zimmermann and R. Neuneier. Active portfolio-management with neural networks. In *Proceedings of Computational Finance, CF1999*. Springer Verlag, 1999.

[60] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.