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

Metaheuristics for the Portfolio Selection Problem

Giacomo di Tollo, Andrea Roli

November 17, 2006

Technical Report no. R-2006-005 *Research Series*

Metaheuristics for the Portfolio Selection Problem

Giacomo di Tollo, Andrea Roli

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

November 17, 2006

Abstract.
The Portfolio selection problem is a relevant problem arising in finance and economics. While its basic formulation can be efficiently solved through linear programming, its more practical and realistic variants, that include various kinds of constraints and objectives, have to be tackled by approximate algorithms. Among the most effective approximate algorithms, are metaheuristic methods that have been proven to be very successful in many applications. This paper presents an overview of the main formulations of the Portfolio selection problem and surveys the literature on the application of metaheuristics to it.

Keywords: *Portfolio Selection Problem, Metaheuristics*

Contents

1 Introduction

Portfolio selection is one of the most relevant and studied topics in finance. The problem, in its basic formulation, is concerned with selecting the portfolio of assets that minimizes the risk subject to the constraint of guaranteeing a given level of returns. Individuals and institutions prefer to invest in portfolios rather than single assets (or securities) because it enables them to dampen the risk, by diversification of the investments, without negatively affecting expected returns. The basic model of the portfolio selection problem (hereinafter referred to as PSP) is formulated in the seminal work by Markowitz[42]. In that work, the author rejects the hypothesis that investors wish to maximize expected returns, because this criterion does not imply that a diversified portfolio is preferable to a non-diversified one. Thus, he states that the goal is to select a portfolio with minimum risk at given minimal returns. Alternatively, the problem can be formulated as a multi-criteria optimization problem in which risk has to be minimized while return has to be maximized. Notwithstanding its potential in capturing the basic properties of the problem, the Markowitz model, referred to as *Mean-Variance model*, suffers from several drawbacks. First, it might be difficult to gather enough data and information for estimating risk and returns. Second, the estimation of return and covariance (used for defining the risk) from historical data is very sensitive to measurement errors. Finally, it is nowadays considered too simplistic for practical purposes, because it does not incorporate aspects of real-world trading that are non-negligible, such as maximum size of portfolio, minimum lots, transaction costs, preferences over the assets, management costs, etc. Adding those constraints 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[4]. Among such approaches can now be found the state-of-the-art solvers for the PSP.

In this work, we give an overview of the use of metaheuristic techniques to solve the PSP. We first present and discuss the different models from the literature and we also introduce a classification of them, that can provide a general scheme for analyzing and comparing such models. Then, we survey the most relevant metaheuristic approaches for the PSP. The distinction between model and solving technique is becoming particularly effective in the recent years, due to the development of constraint programming-oriented approaches, as demonstrated by recent successes of software tools such as Comet [23], ILOG Solver [26] and EasyLocal++ [17].

In Sec. 2 we introduce the Markowitz model along with the most relevant variants and improvements. The problem model is considered as an object with three attributes: decision variables and their domains, objectives and constraints. On the basis of such attributes, we also provide classification such that each actual problem formulation can be seen as an instance of a general abstract model, the basic (or *default*) instance of which is the Markowitz model. Sec. 3 presents the various metaheuristic approaches to the PSP by analyzing them through a general framework for metaheuristics called MAGMA [43]. First, the basic building block of algorithms based on metaheuristics are presented, such as the search space, the neighborhood structures and the cost function. Then, we overview the most important techniques from the literature, starting from solution construction procedures till advanced search strategies. Sec. 4 summarizes the most important works form the literature that explicitly address the issue of comparing different metaheuristic approaches for the PSP. Finally, in Sec. 5 we briefly summarize related works and we conclude with Sec. 6 outlining future research and application directions in the field.

2 PSP Modeling

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

In this section, we provide an overview of the PSP models that can be found in the literature trying to capture the diverse formulations by means of a unique classification, with the aim of giving a general view of PSP modeling along with the possibility of making comparison among the models. We first present the Markowitz model, that constitutes the basis upon which the other models are obtained as variations and extensions.

2.1 The basic model: Markowitz model

In the PSP 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 the portfolio. This measure is the objective function most commonly used in related works.

The Markowitz model [42] is as follows:

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

[1] In agreement with the main literature on the subject, here we consider the problem objective as the minimization of the risk measure. The problem can also be modelled as a maximization of returns or in other ways. See [16] for a brief discussion on this topic.

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 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 assets i and j. The objective function is the variance (herein called risk-measure) σ_p^2, given by $\sum_{i=1}^{n}\sum_{j=1}^{n}\sigma_{ij}x_i x_j$. The expected return of portfolio is given by r_p and $\sum_{i=1}^{n} r_i x_i$ represents the actual forecasted return of the chosen portfolio. Constraint (3) ensures that asset weights sum up to one, as they are considered as fractions of the whole amount of money to be invested.

Constraint (2) can also be written in the inequality form, that is the most commonly adopted constraint when metaheuristics are applied. In this case, constraint (2) becomes

$$\sum_{i=1}^{n} r_i x_i \geq r_p \tag{5}$$

Furthermore, it is also possible to optimize function (1) for a set of values of r_p[7]:

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

The Markowitz model can be considered as the most simple formulation of the PSP. Its conceptual representation is depicted in Fig. 1. Note that the three attributes, variables, objectives and constraints, can be directly instantiated, as in the case of constraints, or further detailed through qualifications. This basic model can be varied and extended in many ways. Every modification can be viewed as the result of the combination of simple variations, each of which affecting only one attribute. For instance, different risk measures can be chosen, or constraints that make the model more realistic can be added. The problem we consider in this paper is a 'single-period' (i.e., single-stage) problem; in particular, we do not take into account possible adjustments between estimated and actual returns, nor transaction costs. Moreover, the PSP formulations we discuss are deterministic.

In the following, we will detail the most important extensions of the basic model, by keeping in the background the conceptual model scheme.

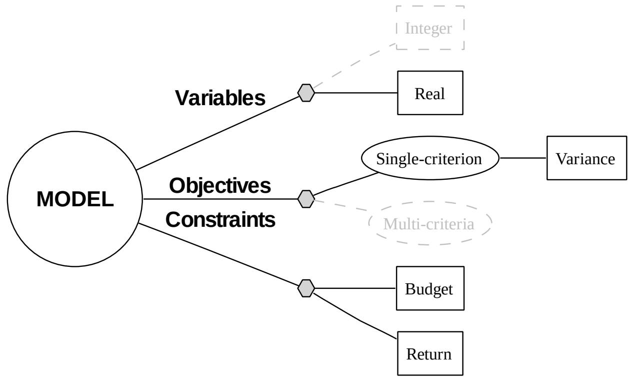

Figure 1. *Conceptual representation of the Markowitz model. Rectangles represent the instantiation of a qualification (ellipse). In gray, qualifications and instantiations not present in the model.*

2.2 Variables and domains

We first briefly discuss the possible choices for variable domains in a PSP model. In the Mean-Variance model, variables are real and they range between zero and one, as they represent the fraction of available money to invest in an asset. This choice is quite 'natural' and has the advantage of being independent of the actual budget. Conversely, another possibility is to choose integer values for variables and make them range between zero and the maximum available budget. When variables are integer, it is possible to add to the model constraints that involve actual budget values, such as minimum trading lots and also introduce more realistic objective functions. Advantages and disadvantages of the two approaches will be discussed in the following sections, in which variations in the basic model are presented.

2.3 Objective functions

In general, the PSP objectives can be either to minimize the risk, while satisfying a given return, or maximize the return not exceeding a given maximum risk, or both. In the former cases, the problem objectives are single-criterion, while in the latter case they are multi-objective. In our classification, we consider these two qualifications for the attribute *objectives*, as shown in Fig. 2.

2.3.1 Single-criterion objectives

Although metaheuristics have been successfully applied to tackle both single and multi-criteria optimization problems, the PSP has been mostly modeled as a single-criterion optimization problem. A first simple variant of the single-criterion objective consists in including constraint (2) in the objective function in a Lagrangian relaxation fashion [6] [62] [33]:

$$\max\ (1-\lambda)\sum_{i=1}^{n} r_i x_i - \lambda \sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} x_i x_j \tag{7}$$

subject to constraints (3) and (4), where λ is a trade-off coefficient ranging in [0, 1]. If $\lambda = 0$ the investor completely disregards risk and aims to maximize returns; conversely, when $\lambda = 1$, the investor is risk-adverse and only wants to minimize risk. By resolving the problem for several values of λ it is possible to estimate the *efficient frontier* for the Markowitz unconstrained problem (referred to as UEF). The investor can then choose the portfolio depending on specific risk/return requirements. The UEF is composed of *Pareto* optimal solution, i.e., solutions such that no criterion can be improved without deteriorating any 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 an approximation of the actual Pareto frontier, in the following we will distinguish between the actual efficient frontier (UEF) and the approximated one (AUEF). Moreover, since we are going to introduce other classes of constraints in our discussion, we will refer to the constrained efficient frontier as CEF, whilst its approximation will be referred to as ACEF. We notice here that the unconstrained frontier dominates the unconstrained one.

So far we have only considered variance as the risk measure, but other different measures can be taken, thus defining different objective functions. Markowitz himself suggested the use of semi-variance instead of variance in order to assess portfolio risk. Semi-variance can be defined as

$$semivar = \sum_{j:r_j \leq E[R]} p_j (r_j - E[R])^2 \tag{8}$$

R is a distribution of returns, often statistically computed by enumerating the most probable scenarios, r_j is the return of the j-th element of the distribution, p_j its probability and $E[R]$ the mean of the distribution. This measure is equivalent to variance if return distribution is symmetric around the mean and captures the essence of risk as perceived by investors, characterized by 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* measures:

$$DSR(\tau) = \sum_{r_i \leq \tau} p_i(\tau - r_i)^q \tag{9}$$

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[58]. τ has been used as the threshold below which the investor perceives a loss[19][18]. The probability that portfolio returns fall below the VaR level is called *Shortfall Probability*:

$$SP = p(r < VaR) \tag{10}$$

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

$$ES = E(r | r < VaR) \tag{11}$$

Amongst other approaches it worths mentioning the Mean-Absolute-Deviation model (MAD)[35], in which the risk is defined as the mean absolute deviation of the portfolio rate of return. This model does not rely on probabilistic assumptions on returns (it is equivalent to the Markowitz model if returns are considered as normally distributed) and it is easier to handle because it does not require the covariance matrix:

$$\min E\left[\left|\sum_{i=1}^{n} r_i x_i - E\left[\sum_{i=1}^{n} r_i x_i\right]\right|\right] \tag{12}$$

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

$$\min \frac{\sum_{t=1}^{T} |\sum_{i=1}^{n} (r_{it} - r_i) x_i|}{T} \tag{13}$$

.

Following the same ideas, in [53] risk is measured as the mean semi-absolute deviation of the rate of return below the average:

$$\frac{\sum_{t=1}^{T} |\min(0, \sum_{i=1}^{n} (r_{it} - r_i) x_i)|}{T} \tag{14}$$

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

Furthermore, since the Mean-Variance formulation is non linear, efforts have been made to model the problem as a linear programming model. Amongst them, besides the above cited ones, the approach proposed by Young[61] defines the risk as the minimum return achieved by portfolio over the considered time horizon.

2.3.2 Multi-criteria objectives

In the multi-criteria variant of the PSP model, the objectives are usually the following [56][1]:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j \tag{15}$$

$$\max \sum_{i=1}^{n} r_i x_i \tag{16}$$

subject to constraint (3).

Moreover, it is possible to have several functions to optimize: Subbu et al.[58], for instance, propose the following:

$$\begin{cases} max & \text{Portfolio expected return} \\ min & \text{Variance} \\ min & \text{Portfolio value at risk} \end{cases} \tag{17}$$

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

The approach consisting in weighting the criteria of a multi-criteria objective function is common when the model is aimed to support decision processes. For example, 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.[46]. According to existing models, they assume portfolio risk being composed of the uncertainty risk and the relation risk. The uncertainty risk measures the uncertainty on future return rates, whilst relation risk measures the trending degree of the sequence. In this framework the objective is given by

$$\begin{cases} max & \text{Portfolio expected return} \\ min & \text{Uncertainty Risk} \\ min & \text{Relation risk} \end{cases} \tag{18}$$

.

Many other objective functions and utility measures have been proposed, an overview of which can be found in [30]. Among them, we mention the objectives

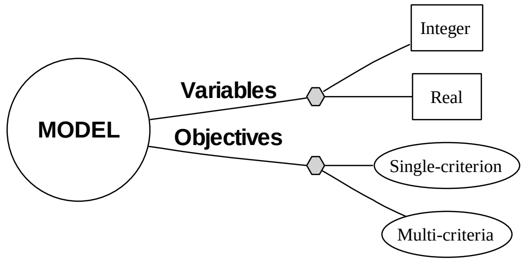

Figure 2. *Conceptual representation of the PSP model attributes variables and objectives.*

introduced in [5], in which 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:

$$\max \sum_{i=1}^{n} \left[\sin(50 \cdot x_i) \prod_{k=1, k \neq i}^{n} x_k \right] \tag{19}$$

$$\max \sum_{x=1}^{n} \left[x_i \prod_{k=1, k \neq i}^{n} (\lceil 50 \cdot k \cdot x_k \rceil \mathrm{mod} 8) \right] \tag{20}$$

A visual conceptual overview of the different kinds of objectives is depicted in Fig. 2, along with the possible choices for variable domains. Before discussing the third attribute of the model, i.e., *constraints*, we have to note that the estimation of returns from real-world data raises statistical and practical issues that have to be taken into account when the PSP is tackled. A discussion on this topic is out of the scope of this paper and we forward the interested reader to the specific literature on the subject [45][3][27][18][19][13][59].

2.4 Constraints

Constraints can be first distinguished into two classes: *theoretical* and *practical*. The first class includes budget and return constraints, while practical constraints are motivated by actual problem requirements, such as minimum lots imposed by law.

2.5 Budget and Return Constraints

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

$$\sum_{i=1}^{n} x_i = 1 \tag{21}$$

$$\sum_{i=1}^{n} r_i x_i \geq r_p \tag{22}$$

Constraint (21) means that all the capital must be invested. If an integer formulation is used, in which assets are represented by their actual value rather than their ratio to the whole portfolio, it can be expressed in the following way [8][53][39]:

$$C_0 \leq \sum_{i=1}^{n} x_i \leq C_1 \tag{23}$$

where C_0 and C_1 are respectively the lower and upper bound on the budget.

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

Return constraint (22) is very important as returns represent one of the two main aspects of the problem.

As stated above, a shortcoming of the original Markowitz formulation is that it does not incorporate many aspects of real-world trading, such as maximum size of portfolio, minimum lots, transaction costs, preferences of which assets to include in the portfolio, management costs, etc. These aspects can be modeled by introducing constraints of the type that we have called 'practical', that are introduced in the following.

2.6 Cardinality Constraints

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

$$\sum_{i=1}^{n} z_i \leq k \tag{24}$$

This constraint is imposed to facilitate the portfolio management and to reduce its management costs. When the model contains this constraint, it can be named "The asset paring problem"[36]. It has been experimentally shown that, when the cardinality constraint is imposed, the ACEF tends to tightly approximate the UEF for high values of k [28][6]. The inequality form is quite common (see, for instance, [50][7][33]), however the constraint can also be expressed in the equality form, i.e., $\sum_{i=1}^{n} z_i = k$. When k is greater than a threshold value, the ACEF returned by the algorithms tends to collapse around one attractor point[1].

2.7 Floor and Ceiling Constraints

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

$$\varepsilon_i z_i \leq x_i \leq \delta_i z_i \tag{25}$$

Floor constraint (i.e., lower bound) is used to avoid the cost of administrating very small portions of assets; ceiling constraints (i.e., upper bound constraints) are introduce to avoid excessive exposure to a specific asset and in some case are imposed by law. Generally the upper bound is considered more relevant than the lower bound and it is also possible to impose different upper and lower bounds for each asset, but this opportunity has not yet been explored in the literature.

2.8 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, referred to as *round*. Rounds are usually measured in unities of money, so this constraint is generally encountered in the PSP integer formulation [8][32][39]. If p_j is the price of asset j and ρ_j its minimum tradable quantity, the minimum lot c_j of asset j, measured in unities of money, is given by $c_j = \rho_j p_j$. When using the continuous formulation its application consists in imposing that each weight must be multiple of a given fraction[56], and, obviously, 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 of reducing the number of different assets in the optimal portfolio.

2.9 Class Constraints

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

$$L_m \leq \sum_{i \epsilon \Gamma_m} x_i \leq U_m \qquad m = 1 \dots M \tag{26}$$

2.10 Compulsory assets

An investor may wish that some specific assets be included in the portfolio, in proportion fixed or to be determined. This constraint can be imposed by setting $z_i = 1$ for the corresponding assets and imposing more or less restrictive upper and lower bounds.

2.11 Turnover and trading constraints

For the sake of completeness, we also mentioning a class of constraints that arise in the multi-period formulation of the problem. These constraints define upper and lower bounds, respectively in case of buying and selling, for the variation of asset values from one period to the next one. Moreover, they are usually combined with transaction costs and taxes. These constraints have been introduced by Crama and Schyns in [7] in a variant of the single-period formulation.

It has been shown that the constrained PSP is NP-Complete[39]. Moreover, exact methods fail to solve large instances of the constrained PSP, therefore researchers and practitioners have to resort to approximate algorithms and, in particular, to metaheuristics and hybrid techniques.

The complete classification of the PSP model variants that can be found in the main literature on the subject is depicted in Fig. 3. In the next section we present an overview of the main metaheuristic approaches for tackling the PSP.

3 Metaheuristic techniques for portfolio selection

Metaheuristics are solving strategies based on which approximate algorithms for combinatorial optimization problems can be designed and implemented. In general, metaheuristic-based algorithms can not prove the optimality of the returned solution, but they are usually very efficient in finding (near-)optimal solutions. Some techniques, such as tabu search, iterated local search, variable neighborhood search, ant colony optimization and evolutionary algorithms have proven to be very successful in tackling real-world problems. For further details on metaheuristics we forward the reader to [4] and [24]. In this section, we provide a review of the most relevant metaheuristic approaches to the PSP. To this purpose, we will adopt the standpoint provided by MAGMA, a general framework for metaheuristics [43]. MAGMA (MultiAGent Metaheuristics Architecture) provides a framework for classifying and designing metaheuristic as a multi-agent system. Metaheuristics can be seen as the result of the interaction among different kinds of agents: the basic architecture contains three levels, each hosting one or more agents. At each level there are one or more specialized agents, each implementing an algorithm. LEVEL–0 provides a feasible solution (or a set of feasible solutions) for the upper level, therefore it can be

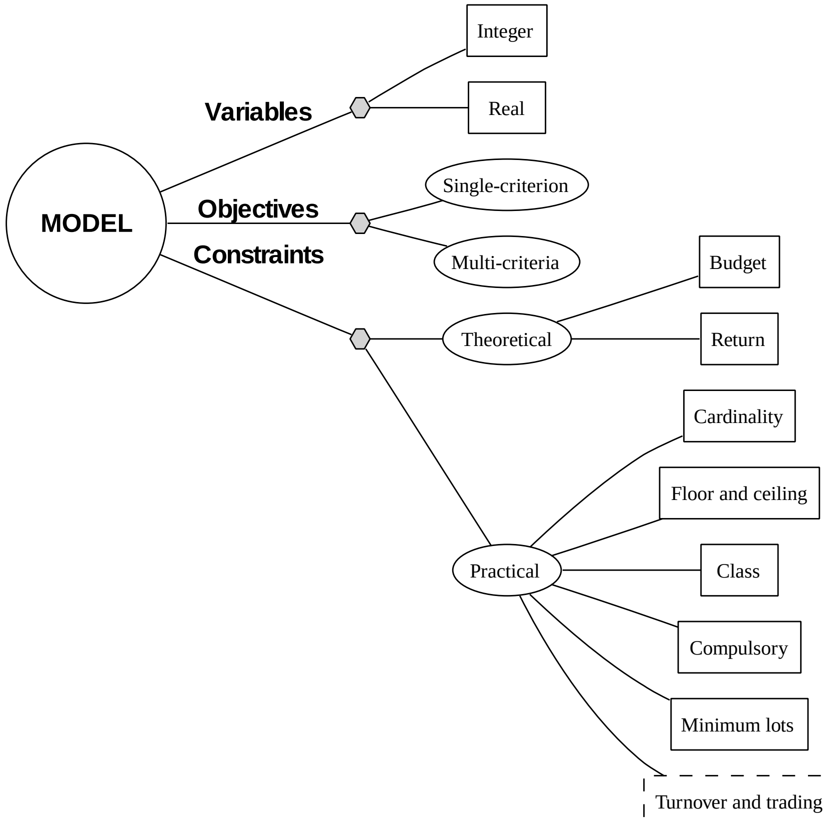

Figure 3. *Conceptual representation of all the PSP model attributes (variables, objectives and constraints).*

considered as the (initial) solution construction level. LEVEL–1 deals with solution improvement and agents perform a trajectory in the search space until a termination condition is verified (basic local search level). LEVEL–2 agents have a global view of the space, or, at least, their task is to guide the search toward promising regions and provide mechanisms for escaping from local optima (long term strategy level). Classical metaheuristic techniques, such as tabu search, can be easily described via these three levels. This basic three level architecture can be enhanced with the introduction of a fourth level of agents, LEVEL–3 agents, coordinating lower level agents. With this fourth level, the framework can also describe hybrid techniques such as large neighborhood search, in which complete solvers are integrated into metaheuristics [9][48]. We first survey the basic concepts metaheuristics for PSP are based upon, i.e., the various choices for defining the set of feasible solutions, the neighborhood structure(s) and the cost function. Then, we give an overview of the techniques level by level, starting from the solution construction till the most general search strategies.

3.1 Metaheuristic attributes

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

3.1.1 The search space

Usually, a solution to the PSP is represented by an array of n variable $x_1, \ldots, x_n$, where x_i represents the fraction of the amount invested in asset i (or the actual amount of money in the integer variable model). Besides those variables, auxiliary variables and data structures can be added for improving algorithm efficiency. An important distinction has to be made in the way the different approaches deal with constraint violations. Indeed, some works define the search space explored by the algorithm as consisting of only feasible portfolios (i.e., satisfying all the constraints in the model), while in other works the search process is allowed to explore also infeasible solutions. For the sake of simplicity, we partition the constraints in two classes:

1. *Hard constraints*, that must be always satisfied by any candidate solution;
2. *Soft constraints*, allowed to be not satisfied during the search process. Generally, their violation is evaluated in the cost function by means of penalties.

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

- *All feasible* approach: Each candidate solution s must satisfy the constraints at any step of the search process (e.g Chang et al.[6]);
- *Repair* approach, in which if an infeasible solution is found, this is immediately forced to satisfy the constraints by means of an embedded repair mechanism (e.g Streichert et al.[56]);
- *Penalty* approach: It is allowed to visit infeasible solutions, but those will be assigned a penalty in the cost function, depending on the amount of violation (e.g Schaerf [50]).

Repair mechanisms are very often used for they provide a tradeoff between exploration and intensification. A typical repair mechanism is explained in Streichert et al.[57], referring to a formulation with cardinality and minimum lots constraints. This mechanism takes as input a non-normalized solution vector and repairs the solution through the following deterministic procedure:

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

2. the obtained vector is normalized so as to meet the 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 the largest roundlot level less or equal than the current asset weight, i.e., $x'''_i = x''_i - (x''_i \bmod c_i)$. The residual amount of budget is redistributed so as to meet minimum lots constraints by buying quantities of c_i of assets with the largest $(x''_i \bmod c_i)$ until all the budget is spent.

3.1.2 Cost function

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

A prominent example of a cost function for the PSP is provided by Schaerf[50] who 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 $(f_2(X))$. The overall cost function to be minimized is a weighted sum of the two components $w_1 f_1(X) + w_2 f_2(X)$, where w_1 and w_2 vary during search according to a shifting penalties mechanism.

$$\min w_1 f_1(X) + w_2 f_2(X) \tag{27}$$

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

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

A similar approach is followed by Gilli and Këllezi in [18]. They choose the following objective function:

$$\min\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sigma_{ij}x_ix_j + \overline{p}\Big(r_p - \sum_{i=1}^{n} r_ix_i\Big)\right) \tag{28}$$

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

where $\overline{p}$ is the penalty term.

3.1.3 Neighborhood relations

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

The first examples of neighborhood relations in local search for the PSP were introduced by Rolland[49]. These neighborhoods are defined for the unconstrained model, i.e., the one with only theoretical constraints as explained in Sec. 2, and can be considered the basic structures upon which further developments have been designed. In the first structure (referred to as *RollandI*) the neighbor of a solution is defined as a solution in which the weight of only one asset is increased or decreased of a given quantity, called *step*. The second neighborhood (referred to as *RollandII*) is defined so that the weight of an asset is increased or decreased of a given *step* and the value of one other asset is respectively decreased or increased of the same value. With these two neighborhood structures, the assets contained in the final solution are a subset of the starting portfolio, since the assets to be modified are chosen amongst the ones present in the portfolio. Anyway, this does not prevent the search from being able to explore all the possible asset combinations, because the model is unconstrained and the portfolio is initialized with $x_i = \frac{1}{n}$, for each asset $i = 1, 2, \ldots, n$. We also observe that *RollandI* might move the search to infeasible solutions. These neighborhoods are well suited for the unconstrained model, but have to be modified for the constrained models because assets can not be present in the portfolio in any quantity. Hence, these neighborhood structures are modified by embedding asset insertion and deletion operations. *RollandII* can be modified by transferring a quantity from one assets i to another asset j even if the latter does not belong to the portfolio. In this case, asset j will be inserted in the portfolio (see the neighborhood called *TID* in [50] and [18]). This approach should also include some mechanism to handle upper and lower bounds, in case they are present in the formulation. *RollandI* can be modified by enforcing the satisfaction of the budget constraint and by allowing insertions and deletions of assets. Feasibility w.r.t. the budget constraint can be enforced by increasing the weight of one asset and decreasing the other asset weights [5]. More precisely, if a solution is given by a weight vector $(x_1 \ldots x_n)$, the neighboring one is $(\frac{x_1}{1+step}, \ldots \frac{x_i+step}{1+step} \ldots \frac{x_n}{1+step})$, for only one i, $1 \leq i \leq n$. This neighborhood is proven to be complete, i.e., for a long enough sequence of moves, each solution can in principle be reached. Completeness does not depend on the initial solution and holds iff $step \leq \frac{1}{n-1}$. The possibility of having asset insertions and deletions lead to neighborhoods defined in [50], called *idR*, and [6]. This neighborhood takes into account the case that an asset i is decreased so that its value falls below its lower bound ϵ; hence, asset i is deleted and another asset j is inserted in the portfolio. Conversely, if asset i is increased so that its value exceeds its upper bound δ, then its weight is set to δ and all other asset weights are normalized. Observe that all these variants do not change the number of assets in the portfolio. A further improvement is thus possible by allowing neighbor solutions to have different number of assets (see [50],*idID*), defined by allowing three kinds of operations on the selected asset i:

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

After these operations, asset weights are normalized.

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

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

We can ideally define a neighbor of a solution by selecting one asset to be modified, specifying the amount of variation and performing the change. This asset is referred to as *pivot*[1]. Then, this modification is counter-balanced by changing the weights of some other assets. If only a pre-determined subset of assets is selected to be modified the neighborhood is said to belong to class 1, otherwise the neighborhood is said to belong to class 2.

The neighborhood structures of class 1 can either consist only of feasible solutions (e.g., [50], structure *TID*) or allowing infeasible moves too (e.g., [49]). The simplest neighborhoods in this group are generated by modifying the pivot weight and counterbalancing this change by modifying the weight of only one other asset ([49] *TID*; [18]). This structure can be generalized by introducing an integer c representing the number of assets to be modified in order to counterbalance the pivot weight variation. Crama and Schyns[7] use $c = 2$, but it is possible to set c at any number, even 0, thus allowing infeasible moves[2]. In the previously described neighborhoods, the *step* size is set before choosing the assets involved in the modification; however, neighbors can also be generated by varying this value. For example, in [1] neighbors are generated by varying *step* from a minimum of $\frac{w_{pivot}}{n}$ to a maximum of w_{pivot}, being forced to assume all multiples of $\frac{w_{pivot}}{n}$.

Neighborhoods of class 2 are generally used in population-based algorithms, especially in genetic algorithms, in which crossover and mutation operators could return infeasible solutions. In this case, it is often impossible to determine which asset plays the role of *pivot*. Anyway, there are some representative cases in which the pivot is used, such as in [6][5][50].

3.2 Metaheuristic search components

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

3.2.1 Initial solution

It has been empirically observed that metaheuristics for the PSP are usually quite robust with respect to the choice of the initial solution. This assertion has been formally proven by Catanas in [5], subject to the specific neighborhood structure defined therein. For this reason, most of the works assume as starting solution a randomly generated one or a solution constructed by means of a simple heuristic procedure [14], possibly embedding also a mechanism to ensure feasibility of the initial portfolio [7].

3.3 Iterative Improvement

Iterative improvement can be considered as the simplest local search, as it performs a path in the search space by moving from a solution to a neighboring one with a lower cost. This search can be named *best improvement*, if the neighbor chosen is the best among the feasible neighbors, or *first improvement*, if the chosen neighbor is the first state found during the neighborhood enumeration that is better than the current one. Iterative improvement is usually incorporated into a more complex strategy, rather than being used as a stand alone localsearch. For instance, in Glover et al. [20] iterative improvement is the local search component of a variable

[2] In this case a repair mechanism should be included.

neighborhood search technique. As another example, we mention Armañanzas and Lozano [1] who use a greedy search to refine solutions found by an ant algorithm.

3.4 Simulated Annealing

The possibility of moving to solutions with a higher cost (i.e., performing degrading moves) characterizes Simulated annealing (SA). The probability of moving toward solutions worst than the current one depends on the cost difference between the two solutions and it also decreases during the search. This probabilistic acceptance criterion enables the search to escape from local optima. Crama and Schyns [7] apply SA to various PSP models by first considering in the model only one constraint class at a time (floor, ceiling and turnover first, then trading and cardinality), then they include all constraints in the model. The authors experiment with three strategies:

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

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

The concepts of SA can also be effectively utilized inside population-based algorithms, as done in [33] [21]. In the approach proposed in [33], an initial population of random portfolios is generated. Then, for each portfolio p_o in the initial population, a new portfolio p_n is created by selecting some assets i and modifying them according to the following rule:

$$w_{x_{in}} = \max(w_{x_{io}} + s, 0) \tag{29}$$

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

An interesting application of SA for a multi-objective formulation is presented in [1], in which moves are always accepted if at least one criterion is improved, while deteriorating moves are subject to the SA usual probabilistic acceptance criterion. This approach seems to find good solutions in the lower part of the frontier, where risk and profits are small.

3.5 Threshold Accepting

Threshold accepting (TA) shares some analogies with SA, as a degrading move can be accepted if the cost difference between the current and the new solution is within a given threshold, that is progressively decreased to zero. The threshold decreasing schedule is defined by estimating the distribution of distances between objective values of neighboring positions (an analogous parameter tuning procedure is undertaken also for SA). TA has been applied to the PSP by Dueck and Winker [13] and Gilli et al.[18] [19]. These works are primarily aimed at comparing risk measures, so the algorithm represents the technical mean to investigate financial aspects. For example, in [13] different risk measures are compared, namely variance, generalized semi-variance and geometric mean. Experimental results show that the solutions corresponding to a risk measure are generally not efficient w.r.t. another risk measure. In this way it is possible to directly compare risk measures. In [13] it is stated that the ACEF is not as smooth as it seems, since it turns out to be composed of linear fragments, and the curve switches from a segment to another one when the fraction held in a particular asset changes sharply.

Besides solving the mean-variance portfolio selection, Gilli and Këllezi [18] tackle a more realistic problem in a downside-risk framework in which decision variables are integers. The problem is formulated as a maximization of future returns, while value-at-risk and expected-shortfall are compared as risk measures. In a further work[19] TA is used to compare three different risk measures: *Value at Risk (VaR)*,*Expected Shortfall (ES)* and Omega measure (defined as the ratio of the weighted conditional expectation of losses over the weighted conditional expectation of gains) in a formulation with cardinality and upper/lower bounds constraints. Results show that Mean-VaR portfolios are more diversified than those obtained with ES criteria, while ES frontier dominates the other two.

These works stress the fact that much attention has to be paid to the choice of an appropriate risk measure. Indeed, efficient portfolios with respect to a risk measure are usually not efficient with respect to other measures and efficient portfolios are very different from each other with respect to different utility functions.

3.6 Tabu search

The Tabu search metaheuristic (TS) moves away from local optima by forbidding the search to execute the inverse of the last l recently performed moves. This simple mechanism, enhanced with the exploitation of the search history for intensifying and diversifying the search, makes TS one of the best performing local search strategies. The application of TS to the PSP has its milestones in the works by Rolland [49] and Glover et al.[20]. These works refer to different formulations of the problem and moreover Glover tackles a multi-period formulation (while our work is concerned only with single-period portfolio selection). Nevertheless, both deserve to be analyzed for the richness of concepts presented.

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

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

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

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

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

- find i such that

$$\left| \left(\left| 1 - \sum_j x_j \right| \cdot r_i \right) - \left| \left(\sum_j x_j r_j - r_p \right) \right| \right| \tag{30}$$

  is minimized;

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

- If the total investment is less than budget (i.e., if $\sum x_i < 1$), increase x_i so as to make the solution feasible.

When the return level of the best solution found is within the $0,005\%$ of the desired level, the repair mechanism invoked is the one described for the minimum variance problem, so that the solution is feasible w.r.t. the requested minimum-variance point after the requested return level has been reached.

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

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

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

Tabu Search has been widely applied to solve the PSP. It is easy to find it in works aimed at comparing the performance of different algorithms on the same instance (see Sec. 4). A very successfull application of TS can be found in Schaerf[50], in which TS is improved by dynamically changing the neighborhood structures.

3.7 Variable Neighborhood Search

Variable Neighborhood Search[22] (VNS) is a metaheuristic that dynamically changes neighborhood structures during search, so that a neighborhood is substituted by another one when the current solution cannot be further improved. There is no explicit application of VNS to the PSP, however, as this strategy is very general, its principles can be found in some important works in the literature. This is the case of the work by Glover et al.[20], in which the implementation of *variable scaling* can be considered a VNS, as a new neighborhood is introduced by changing the *step* when no further improvement is possible. A similar technique can be found in [14], in which the search switches between two neighborhoods. Moreover, similar ideas can be found in [50], in which TS is implemented in a token ring sequence, in which runs using a different neighborhood structures are interleaved.

It is worth mentioning also the work by Speranza[53], in which a heuristic algorithm is defined and applied to Milan Stock Market using an integer formulation. Here, in order to satisfy the constraints on capital, assets are ordered and re–numbered in nondecreasing order of x_i in the portfolio; then x_1 is increased (and, if this move is unsuccessful, decreased) by one unit. If the new solution is feasible, the algorithm stops, otherwise the procedure is repeated over $x_2 \dots x_n$. At the end of this phase, if no feasible solution is found, the cycle is repeated increasing assets by two units, then three and so on. This mechanism can be considered as a kind of VNS, even if the neighborhood cardinality is constant over the whole process and the neighbor selection process is deterministic.

3.8 Evolutionary Algorithms

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

The first applications of EAs to the unconstrained PSP are presented by Tettamanzi et al. in [2][37][38]. In [2] a genetic algorithm (GA) is implemented for the PSP with down-side measure of risk. In the algorithm, one population is handled and individuals are generated according to investor preferences: a specie is defined for each λ (where λ is the trade-off coefficient between return and risk, as discussed in Sec. 2.3). Individuals are generated such as their probability of belonging to a specie is proportional to the investor's interest in that specie. At each generation, a new individual replaces the worst one in the previous population.

In a further work [38], a distributed genetic algorithm is applied in which each λ value is associated to a subpopulation. As the AUEF is composed by plotting a point for each λ, the greater the number of populations, the finer the resolution of the frontier. Migrations of individuals between populations corresponding to neighboring values of λ are permitted, in order to avoid premature convergence of the algorithm. Individuals are allowed to mate only with individuals of the same population or of adjacent ones. This implementation outperforms the previous sequential version, and in [37] a detailed description of the implementation and risk measures is provided. In parallel implementation of GA, if the cardinality constraint is imposed it is possible to search in parallel several ACEF corresponding to each value of k, using information from each of these to improve the search process of others. With this approach, the ACEF approximates the UEF with increasing precision, as k increases and constrained optimal portfolios are shown to be not significantly different from unconstrained ones, except for very small number of assets and very low risk levels.

Cardinality constrained PSP is also tackled by Streichert et al. in [55][56][57] using a two-objectives optimization model, enriching their implementation by adding an archive in order to store the frontier obtained so far. In their work, they introduce the knapsack representation of portfolios, comparing it with the *standard* one. The authors also investigate the use of *Lamarckism*. In fact, these algorithms embed a repair mechanism that prevents the search from rejecting infeasible solutions. In the GA version without Lamarckism, only the phenotype of an individual (i.e., the normalized vector of assets) is altered by the repair mechanism, while the genotype (i.e., the non-normalized vector of assets) remains unaltered. Conversely, in the version with Lamarckism, the repair mechanism modifies the genotype too, according to the phenotype. In each case, this solution representation leads to a better performance than the standard one. Moreover, Lamarckism helps improve performances too. Furthermore, different variable representations (binary and real-valued) are also compared and different coding [56] and crossover operators [57] are examined.

Memetic algorithms for the PSP are presented in [41], in which the use of SA and TA inside the EA framework is compared. The results discussed indicates that TA is more suitable when *VaR* is used as risk measure, while SA makes the algorithm perform better when *ES* is chosen. An explanation of this result is given by observing that *VaR* induces a rugged search space, while *ES* induces a smoother landscape. In that work also the use of a kind *elitist* strategy is investigated, that implement a sort of intensification of the search around the best found solutions. This strategy improves the performance of the algorithm when the search space is rugged, while it seems not to payoff when the search space is smooth. Moreover, the introduction of this kind of intensification makes the algorithms more robust against parameter values.

Liu and Stefek [36] tackle the PSP with cardinality and ceilings constraints, comparing GA with a heuristic proprietary method and they investigate crossover rates, population size and elitist strategy showing that GA can achieve good performances, even if worse than the heuristic, especially concerning execution time.

We should observe that the model with floor, ceiling and cardinality constraints is the most commonly used in literature when GA are applied [6][14][15]. GA have also been used in conjunction with formulations differing from the canonical Mean-Variance one, in order to define more realistic customer-oriented frameworks. An interesting example is represented by [62], in which the objective function to maximize is given by the usual weighted objective function (eq. 7), but they solve this model for different isolated values of λ rather than trying to plot the whole frontier. They show that in the obtained portfolios return is higher than the best one provided by optimization software for Mean-Variance (LINGO[31]) even if they are more risky.

One of the main contribution of that work is that the expected return is considered as a variable, rather than an instance datum. The return ranges in an interval in which arithmetical mean represents lower bound a if its recent history trend has been increasing, the upper bound b if its trend has been decreasing.V-Shaped transaction costs are also investigated for portfolio revision, but they are only considered as proportional[3]. Transaction costs (embedded in a MAD objective function) and single λ values analysis are considered in Wang et al.[59] in which a sample procedure for stochastic returns is introduced instead of the classical scenario analysis.

More complex approaches are proposed aimed at helping decision making by introducing other measures either to define an ordinal-preference framework in which other measures are added to the formulation (see [58][14]), or to predict the future return rate and to estimate the uncertainty risk of the future return rate when the sample is small [46].

We must finally notice that GA, by being inherently effective in diversifying the search, show good performances especially in multi-objective formulations, as shown by the family of MOEAs (MultiObjective Evolutionary Algorithms) [10][55][56][57][8][46].

3.9 Particle Swarm

The nature-inspired paradigm referred to as *Particle swarm* is a promising search paradigm, especially when continuous optimization problems are tackled. Nevertheless, its application to the PSP is still limited, and the works on this topic do not tackle the standard formulation, being aimed at finding one portfolio optimal with respect to a measure such as the reward-to-variability ratio out of a given set of assets, rather than drawing out the whole efficient frontier[34][44].

[3] In a further work[60] risk-free asset are introduced and the formulation is based on a linear programming model.

3.10 Ant Colony Optimization

Ant colony optimization (ACO) is a population-based metaheuristic that is inspired by the foraging behavior of ants. Solutions are built component by component, according to a probabilistic procedure that bias the choice of the next solution component on the basis of the quality of the previous constructed solutions. Usually, ACO also incorporates some local search algorithm to improve the quality of the solutions built. Initially conceived for discrete spaces, ACO has been adapted also for continuous spaces, too (see, for instance, [52]). Nevertheless, the potential of ACO for tackling the PSP appears still not completely exploited.

A successfully application of ACO can be found in a PSP modeled with the cardinality constraint [1][40]. The approach consists in defining a population of n ants that explore a completely connected graph composed of n nodes. Assets and nodes are in one-to-one mapping and the path traversed by an ant corresponds to the assets to be chosen for the portfolio. Path lengths are of exactly k steps, where k is the portfolio cardinality. In the case of multi-criteria optimization, ants are divided in populations such that each population solves a problem corresponding to one objective function [1]. When ants terminate the exploration phase, a greedy search refines the solutions. This method finds better solutions than SA and iterative improvement and results are particularly striking in the upper part of the frontier, where risk and profits are high.

ACO has found application in problems similar to the PSP such as the so-called multi-objective project portfolio selection [11][12], 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 not 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.

4 Comparative studies

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

Performance measures are usually obtained by comparing constrained results (ACEF) with the ones obtained in the unconstrained case for each level of return (each point of the UEF) and computing statistical measures (mean and median percentage error, standard deviation etc.) for the overall frontier. There are however many ways to define an error measure. For instance, Chang et al.[6] consider the distance of the point from the UEF, defined as the minimum between the distance on the x-axis direction and the distance on the y-axis direction[4]. Another measure can be found in Streichert et al.[55] [56] [57], where the algorithm performance is computed as the percent difference between the area below the UEF and the obtained ACEF. The issue of comparing two frontiers is just an instance of the more general problem of comparing algorithms for multi-objective optimization[47]. Often, also statistical tests are used [28] [10], especially to determine if the difference between UEF and CEF is significant, and some works introduce measures to determine the best portfolio in a frontier[13].

One of the first comparative works is due to Catanas[5]. That paper is focused on investigating properties of the proposed neighborhoods (see Sec. 3.1.3). The author uses TS and SA, implemented in both *robust* and *dynamic* way. In the *robust* implementation, the *step* is kept fixed during all iterations, while in the *dynamic* one it is decreased to zero during the execution. Furthermore, a schema for the variation of the *step* is defined such that its value is increased if solution quality worsens and decreased if solution quality improves. Moreover, a threshold on the minimum value of *step* is introduced, since too small values can make the search stagnate.

Chang et al.[6] introduce cardinality and minimum and maximum holding constraints and observe that the CEF becomes discontinuous. This is due to the fact that feasible proportions of assets are dominated (because of the existence of portfolios with lower variance and higher return); furthermore portions of frontier could not be reachable for a classical λ-weighting drawing approach (due to minimum proportion constraints). In [6], the authors implement GA, TS and SA to solve the problem. Results show that GA is able to approximate the UEF with the lowest average mean percentage error. Regarding the constrained problem, GA seems to perform better than SA and TS, but differences are not as clear as in the unconstrained case, so they use portfolios from the three metaheuristics to draw out the ACEF. Their approach is to store, for each heuristic, all the improving solutions found in the search process and, finally, deleting the dominated ones. The sets obtained by the three heuristics are then pooled to draw the ACEF. This approach shows that for the constrained problem the ACEF approximate the UEF when the asset cardinality is high (as already stated in [28], see Sec. 3.8).

Jobst et al. [29] compare the results presented in [6] against two heuristic methods. The first is an *integer-restart* procedure that plots the CEF starting from the

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

highest return and its corresponding risk to lower return and reduced risk. The result obtained at each stage is supplied as starting point to the next (lower return) stage, considering it as first feasible value (this heuristic is referred to as *warm restart* heuristic). The second, inspired to an idea similar to [53], first solves a continuous relaxation without any constraints, then uses the k assets with highest weights as input for 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 [6]. Anyway, we should note that re-optimization heuristic could not able to draw the whole frontier when the continuous relaxation produces a portfolio with less than k assets.

Another important work that compares different techniques is the one by Schaerf[50], in which the model includes floor, ceiling and cardinality constraints. The author defines three neighborhood relations, that specify moves that satisfy the budget constraint, and defines a cost function that account for the violation of the other constraints. The initial state is selected as the best amongst 100 randomly generated portfolios with k assets. A first phase of experiments with *Best* and *First* Iterative improvement, SA and TS run as single solvers is performed. Then TS, the most promising solver in the preliminary experimental analysis, is chosen for an extensive experimental analysis, combining neighborhood relations in various token-ring strategies. In this case, the *step* length is set to a higher value in the first used solvers to favor diversification, while it is set to a smaller value in the last used solvers for intensifying the search. Experimental results show that the best performances are achieved by token ring solvers with random steps, even if fixed steps seem to behave well too. Single solvers do not attain comparably good results.

Armañanzas and Lozano [1] compare iterative improvement, SA and ACO in a multi-objective formulation with cardinality, floor and ceiling constraints. The algorithms used are tailored to the multi-objective problem, and ACO outperforms the other techniques. The simple greedy search (iterative improvement) shows poor performances if used alone, but turns out to be effective when used to refine solutions provided by ACO. Interestingly, ACO and SA best performances are found in different areas of the frontier: the first in the upper part of the frontier, the latter in the lower part.

Also Ehrgott et al.[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 local search similar to a VNS embedding a random escaping mechanism to avoid stagnation at local minima. They test the algorithms over both random and real-world instances. Results on both instance classes show that GA appears to be the best performing solver. The local search and SA achieve good results, while TS performances appear to be the worst ones.

A further interesting comparison is made by Fernandez and Gomez[15], in which metaheuristics by Chang et al. are compared with a neural net approach. An Hopfield network[5] is used to plot the ACEF when cardinality constraint and bounds (lower and upper bounds) are imposed. Their results show that there is no significative difference between their neural network and metaheuristics such as GA, TS and SA. In order to improve the performance, portfolios from the four approaches are pooled and dominated solutions are deleted, so as to obtain an improved ACEF (the same approach pursued by Chang). The quality of solutions returned is high, making this neural nets approach successful[6]. Nevertheless, the number of different portfolios returned by the neural net is lower than the number returned by other heuristics, therefore, even if the quality is high, stand-alone neural nets approaches are not suitable for solving the problem in the whole frontier.

5 Related work: LP-based heuristic approaches

For the sake of completeness, in this section we briefly review heuristic approaches based on linear programming, that can be very useful as components of more robust and complex metaheuristic strategies. These works are also important because they deal with integer formulations of the PSP, in which assets are assigned integer values corresponding to the actual amount of money to be invested in each asset. The model they use is as follows:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j \tag{31}$$

subject to

$$\sum_{i=1}^{n} r_i x_i = r_p \cdot C \tag{32}$$

$$\sum_{i=1}^{n} x_i = C \tag{33}$$

$$x_i \quad integer \qquad i = 1, \dots, n \tag{34}$$

where C is the total amount of available budged to be invested. Speranza[53] models the problem by including transaction costs, minimum lots, cardinality, floor and ceiling constraints and by introducing two auxiliary binary variables to indicate whether

[5]Hopfield networks[25] are neural network composed of a single layer of neurons fully connected and are widely applied in combinatorial optimization[51].

[6]Indeed, neural nets can capture non linear relations among variables and do not need model assumptions, therefore they are suited for forecasting future returns without relying on the stock returns normal distribution assumption. This idea has been also exploited in [54] and [63] in order to optimize portfolio management.

a security has fixed transaction costs and whether it belongs to the portfolio. The idea presented is to relax the integer constraint on quantities, transforming the problem into a linear programming one (to be solved efficiently even when the number of securities is high) and finding a solution to it. Fractional asset weights are then rounded to the closest integer and heuristics are applied to force the solution to satisfy capital and rate of return requirements. If the algorithm terminates without solutions, less restrictive bounds on capital are iteratively set. This algorithm (referred to as *ROUND–LP*) shows good performance when the total number of assets is low and reaches a solution close to the optimal one when the capital is large.

In Mansini and Speranza[39], the formulation of the problem includes minimum lots and proportional taxes. The authors provide three heuristic algorithms based upon the idea of solving sub-problems of the original formulation, involving subsets of initial universe of assets. In the first heuristic (referred to as *SINGLE-LP*) they solve the continuous relaxation of the problem. Then, they use this solution to feed the mixed integer-linear programming solver. The second heuristic (referred to as *Reduced–cost–MILP–heuristic*) considers a vector x_R with a number of assets greater than the vector of assets i s.t. $x_i \neq 0$ as input of MILP-procedure, thus including also assets whose quantity in the solution of the relaxed problem is zero. The third method consists in an iterated routine: after solving the relaxed problem, the vector x_R is used as input for a MILP procedure. After each step, half of the assets i s.t. $x_i = 0$ is deleted and half is replaced in the solution. The process ends when a given number of securities has been considered. This third heuristic is the most effective, but requires more computational time. These heuristics performs reasonably better than simple problem specific heuristics proposed in [53] and they have the advantage of being more general and are also used in Kellerer et al.[32] in a formulation enriched by introducing fixed transaction costs and minimum lots. These heuristics are applied to four different models that include rounds and fixed costs applied if the amount of money invested in a security exceeds a minimum threshold.

6 Conclusions

In this work, we have defined a framework for classifying metaheuristic approaches for the PSP, introducing the main aspects of the models of the problem and the general components of the metaheuristics developed to tackle it.

The PSP is only a representant of a class of problems consisting in the management of portfolios of different 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, metaheuristics provide flexible and powerful solving strategies that can effectively and efficiently tackle the various instantiations of the PSP, from the basic Markowitz formulation, to more elaborated models including also side constraints. Moreover, we believe that metaheuristic and hybrid approaches could be very successful also to tackle dynamic and multi-period formulations of portfolio selection, in which issues of *re-balancing*, *index-tracking* and *re-optimization* arise.

The works we discussed in this paper show, on the one hand, the potential of such a solving strategies and, on the other hand, the modelling and algorithm design issues that have to be addressed for implementing effective tools. Future research is now focusing on the development of methodologies for designing and implementing constraint-based metaheuristics and hybrid techniques. Furthermore, the practical importance of stochastic optimization has contributed to increase the efforts in providing effective solvers for such a kind of problems, both off-line and on-line. Finally, it is important to recognize that research on the PSP is inherently interdisciplinary and, for these problems to be effectively attacked, it requires a cross-fertilization between algorithmics and finance.

Acknowledgments

We thank Andrea Schaerf for comments and suggestions on a former draft of this paper.

References

[1] R. Armañanzas 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.

[2] 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.

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

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

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

[6] 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.

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

[8] 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.

[9] B. De Backer, V. Furnon, and P. Shaw. Solving Vehicle Routing Problems Using Constraint Programming and Metaheuristics. *Journal of Heuristics*, 6:501–523, 2000.

[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), 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] L.Di Gaspero and A.Schaerf. EasyLocal++: an object-oriented framework for the flexible design of local-search algorithms. *Software Practice & Experience*, 3(8):733–765, 2003.

[18] 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 Academic Publishers, 2001.

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

[20] 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.

[21] 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.

[22] 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.

[23] P.Van Hentenryck and L.Michel. *Constraint-Based Local Search*. The MIT Press, 2005.

[24] H.H. Hoos and T. Stützle. *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann Publishers, 2004.

[25] 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.

[26] ILOG. *ILOG Solver 5.1 User's manual*, 2001.

[27] J.Beasley. Benchmarks for the portfolio selection problem. http://people.brunel.ac.uk/ mastjjb/jeb/info.html, Visited on July 2006.

[28] 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.

[29] 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.

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

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

[32] 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.

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

[34] 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 on artificial intelligence and applications*, pages 140–145, 2005.

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

[36] 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.

[37] 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.

[38] 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.

[39] 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.

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

[41] 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.

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

[43] M. Milano and A. Roli. MAGMA: A multiagent architecture for metaheuristics. *IEEE Trans. on Systems, Man and Cybernetics – Part B*, 34(2):925–941, April 2004.

[44] 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.

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

[46] 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.

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

[48] G. Pesant and M. Gendreau. A Constraint Programming Framework for Local Search Methods. *Journal of Heuristics*, 5:255–279, 1999.

[49] 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.

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

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

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

[62] 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.

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