A survey of computational approaches to portfolio optimization by genetic algorithms

Rifki, Omar

Ono, Hirotaka

出版情報：18th International Conference Computing in Economics and Finance, 2012-06. Society for Computational Economics
バージョン：
権利関係：

A survey of computational approaches to portfolio optimization by genetic algorithms*

Omar Rifki and Hirotaka Ono †
Department of Economic Engineering
Kyushu University
Fukuoka 812-8581, Japan

ABSTRACT
The portfolio optimization problem has become a standard financial engineering problem since the pioneering work of Markowitz on Modern Portfolio Theory. It aims to find an optimal allocation of capital among a set of assets by simultaneously minimizing the risk and maximizing the return of the investment. In the theoretical case of linear constraints, this problem is basically solved by quadratic programming. However, real-life financial market imposes some nonlinear constraints such as cardinality constraints, which limit the number of assets held in the portfolio, minimum transaction lots constraints, which require holding discrete units in assets, multiples of minimum lots, e.g., 100 or 200 shares, or transaction costs, which tend to eliminate small holdings. If we take into account these constraints, our problem becomes computationally intractable in theoretical sense, e.g., NP-hard. GA, genetic algorithm, is a collective term describing family of stochastic algorithms based on the natural selection principle – survival of the fittest, and is widely adopted in many fields. In fact, many empirical studies have reported that GA can find good approximate solutions for NP-hard problems. Already various GA-based approaches have been proposed to solve portfolio optimization problems. We survey more than 10 state-of-the-art approaches on the topic, categorize them, compare their computational results and provide brief descriptions of the techniques involved. The aim of this paper is to provide a good guide to the application of GA to portfolio optimization.

Keywords
Genetic algorithms, Portfolio optimization

1. INTRODUCTION
The portfolio optimization problem, POP for short, aims to find an optimal allocation of financial capital among a set of available assets. Markowitz [35], the creator of *Modern Portfolio Theory*, was the first one who proposed a model based on the trade-off between the risk and the expected return for choosing the optimal portfolio via the *Mean-Variance, M-V*, framework. This model wherein investors tend simultaneously to maximize the mean of asset returns and to minimize the variance, as a measure of risk, is now quite widespread in modern finance. Without taking into account some constraints that can occur in real-life financial market, the M-V model is solved by quadratic programming. The frameworks proposed after attempt essentially to find an adequate measure of risk. Actually, the financial risk is a subjective notion, several investors assess the risk in several manners. One way to quantify the risk is to measure the uncertainty of the underlying portfolio's assets. The uncertainty of an asset can be related on how much the returns deviate from the mean in a symmetric way, i.e., without distinction between above and below deviations, as the variance or the standard deviation in the M-V model. Konno [24] proposed another dispersion-based risk, *the absolute deviation*. Using *Mean-Absolute Deviation* model, MAD for short, the portfolio allocation problem becomes a linear programming problem, the computation is thus simplified and can handle large-scale portfolios. Konno and Yamazaki [26] showed that MAD generates an optimal portfolio altogether similar to the one based on Markowitz model. They based their empirical experiments on historical data of 224 stocks of the Nikkei 255 index. The risk can be also quantified, according to the conventional perception of investors, as the likelihood of losing parts of the investment's worth or its entirety. This risk is referred as a *downside risk*. A downside risk is an asymmetric measure that calculates the probability that the portfolio's return is above a certain level by estimating the maximum lost amount of the investment. Nowadays, the downside risks are very popular among portfolio investment institutions, specially the *Value-at-Risk, VaR*. Nevertheless, the first downside risk incorporated into the POP is the *semivariance*, proposed by Markowitz himself [37] as a correction of the variance in downside risk. Yet, the Mean-semivariance is still similar to the M-V model, apart from the fact that the return fluctuations above the mean are dismissed.

The original Markowitz portfolio optimization problem is solved using a convex quadratic programming procedure. However, real-life financial market imposes some nonlinear constraints such as *cardinality constraints*, which limit the number of assets held in the portfolio, *minimum transaction lots* constraints, which require holding discrete units in assets, multiples of minimum lots, e.g., 100 or 200 shares, or *transaction costs*, which tend to eliminate small holding. If we take into account these constraints, the POP becomes computationally intractable in theoretical sense, e.g., NP-hard.

Since the portfolio optimization problem may be computationally time consuming, heuristic searches are more suitable

*This work is partially supported by KAKENHI(No. 21680001 and 23310104). The authors would like to thank Peter Tzenov for his valuable comments on an earlier version of the paper.

†EMail adresses are omar.rifki.508@s.kyushu-u.ac.jp and hirotaka@en.kyushu-u.ac.jp .

solving tools. The current paper focuses exclusively on *Genetic Algorithms, GAs,* as an optimization tool. Initially conceived by Holland [23], GAs have been remarkably and widely adopted, in the recent years, to solve optimization problems in various domains [38], in particular those computationally intractable in theoretical sense. In fact, many empirical studies have reported that GAs can find good approximate solutions for NP-hard optimization problems. A new field has emerged known as Evolutionary optimization. GA is, in fact, a collective term describing family of stochastic heuristic algorithms based on the natural selection principle – survival of the fittest. The main concept behind these algorithms is to keep evolving a population of candidate solutions one generation after another to hopefully find a global optimum or a suboptimal solution in the worst case. In comparison with other heuristics, GAs have several advantages. Firstly, they are less problem-dependent in the sense that no more assumptions are made on the points of the problem. Secondly, they are performed in a representation rather than in the search space of the problem directly. Nevertheless, the first version of GAs, Holland's version, was restrictively associated to binary encodings. Afterwards, researchers adopted different encodings adequate to the problem, because, the binary strings do not always give a good representation for optimization problems. The reader can refer to Michalewicz [38], who applied different sorts of encodings to GAs to solve numerical optimization problems. In brief, GAs are currently seen as a powerful optimization tool balancing between exploration and examination; exploration of the search space and examination of fitter solutions. More information on this field can be found in Golberg [20], who describes the current shape of GAs.

Already various GA-based approaches have been proposed to solve POPs. Most of these approaches have emerged in the current decade, which indicates an increased interest in the subject. The problem can be formulated as a multiobjective optimization problem, in this case a *Multiobjective Evolutionary Algorithm, MOEA*, is applied. MOEAs are defined as variants of GAs handling multiobjective optimization problems. Schlottmann and Seese [43] and Castillo and Coello [8] presented two comprehensive surveys of MOEA in Economics and Finance, and the larger part of their applications, concerns the matter of portfolio optimization. This paper is (hopefully) intended to be a continuation of their work, but adding into account *Single Objective Genetic Algorithm, SOGA*, applications to POPs. SOGA refers to a standard GA where only one objective is considered in the underlying optimization problem. Our goal, beyond describing the state-of-art applications, is to focus on the computational results obtained by the different approaches.

Outline. In the remainder of the paper we describe, in Section 2, two widely used portfolio models, namely M-V and M-VaR models and specify a number of portfolio constraints often used in practice. A brief presentation of GAs is provided in Section 3. Applications of MOEA are reviewed in Section 4, while Section 5 deals with SOGA. Section 6 discusses the computational results of the regarded approaches. Finally, some prospected research directions are explored in Section 7 and we conclude in Section 8.

2. PORTFOLIO OPTIMIZATION MODELS

2.1 Mean-Variance model

The M-V portfolio model, created by Markowitz, is the first framework for optimal asset allocation. It can be formalized as an optimization problem where the inputs are the returns of N risky assets, assumed to be multivariate normally distributed, and the output is an N-dimensional real vector $w = (w_1, w_2, ..., w_N)^T$ where each weight $(0 \leq w_i \leq 1)$ is the fraction held in the ith asset:

M-V model - General Formulation

$$Minimize_w \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij} = \sigma_p^2(w) \quad (1)$$

$$Maximize_w \sum_{i=1}^{N} w_i \mu_i(w) = \mu_p(w) \quad (2)$$

$$subject\ to \quad \sum_{i=1}^{N} w_i = 1 \quad (C1)$$

$$\forall i \in [|1, N|],\ 0 \leq w_i \leq 1 \quad (C2)$$

where μ_i and σ_{ij} are respectively the expected return of the ith asset, and the covariance of returns between the ith and jth assets, such that $\sigma_{ij} = \sigma_i^2$ is the variance of the ith asset. $(C1)$ expresses the budget constraint, i.e., the entire budget is invested. The M-V model can be considered a Multiobjective optimization problem, since the objective is simultaneously maximizing the portfolio mean μ_p, and minimizing the portfolio variance σ_p^2. All feasible solutions of this problem describe a curve in the plane (σ_p, μ_p) called *efficient frontier*. There also exist alternative formulations that eventually lead to the same efficient frontier. An investor can target, under a certain level of expected return μ_0, to minimize σ_p^2 or to maximize μ_p given a certain level of risk σ_0^2. The variation of μ_0 or σ_0^2 leads to the same efficient frontier. A scalarization in the same objective function of μ_p and σ_p^2 provides another formulation of the M-V model. The scaling parameter λ is used as an indication the risk-aversion, the smaller λ is, the more the investor is risk averse. Stein et al. [47] solved efficiently this problem by parametric quadratic programming for all the values of λ.

M-V model - Risk-aversion Formulation

$$Maximize_w\ \lambda\, \mu_p(w) - (1-\lambda)\, \sigma_p^2(w) \quad (3)$$

$$s.t.(C1) + (C2)$$

Adding a risk-free asset in the Markowitz model, Tobin [52] proved to have a more efficient portfolios set, in form of line so-called *Capital Market Line (CML).*

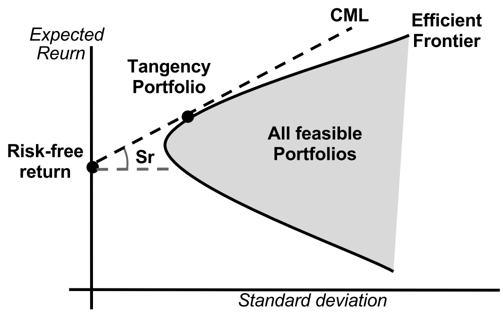

Figure 1: The efficient frontier and the CML.

2.2 Mean-Value-at-Risk model

Certainly the most popular downside risk measure, Value-at-Risk, introduced by J.P. Morgan research center [21], allows to calculate the maximum anticipated loss of the portfolio value under a certain level of confidence $0 \leq \alpha \leq 1$ over a specified time horizon. If r_i is a random variable of the ith asset return and P the probability function, the formal mathematical expression of VaR is given by:

$$VaR_\alpha(w) = min\{R|P(-\sum_{i=0}^{N} w_i r_i \geq R) \leq \alpha\},$$

Three approaches are commonly used to compute VaR, the parametric approach, the historical simulation and the Monte Carlo simulation. Nowadays, even non-financial institutions use VaR to manage their internal risks. However, VaR suffers from some inconvenient mathematical properties [1] such as non-convexity, which makes the optimization of M-VaR model computationally expensive, and also subadditivity, i.e., the risk of a portfolio can be greater than the sum of the assets' risks calculated separately, which is opposite to the principle of diversification in the portfolio theory.

Mean-VaR model

$$Minimize_w \; VaR_\alpha \qquad (4)$$

$$s.t. \sum_{i=1}^{N} w_i \mu_i(w) = \mu_0 \;\; and \qquad (C1) + (C2)$$

2.3 Real-life portfolio constraints

Real-life financial market imposes some additional constraints to POP. Portfolio managers usually use these constraints to efficiently handle their investment. A distinction can be made between soft constraints which are mainly linear as *holding weights constraints*, *turnover constraints* and *risk factor constraints* and hard constraints which are generally integer and combinatorial in nature as *cardinality constraints*, *buy-in thresholds constraints*, *minimum transaction lots constraints* and *transaction costs*. Hard constraints lead to a non-convex search space and the POP is transformed, depending on the cases, to a mixed-integer programming problem or integer programming problem.

Holding weights constraints (HWC)

In the perspective to control diversification in the portfolio, the maximum and minimum holding per asset is limited, such that l_i and u_i are respectively the lower and the upper bound on the ith weight:

$$l_i \leq w_i \leq u_i, \quad (C3)$$

By taking $l_i = 0$ short-selling is not permitted in the portfolio. And similarly to $(C3)$, it is possible to argue about the whole holding ($L \leq \sum_{i=1}^{N} w_i \leq U$) and limit the exposure of the portfolio, or on a classes of assets , e.g., oil stocks, energy stocks etc, such that the invested capital is concentrated in specified groups.

Minimum Transaction Lots constraints (MTL)

These constraints, called also *roundlots constraints*, require holding discrete units multiples of a minimum lot for each asset. For example, if 100-share is the minimum lot for asset A, the amounts purchased must be in 100, 200, 300, etc, trading units. Mansini and Speranza [34] demonstrated that POP constrained to MTL is NP-hard, independently of the risk function. If we denote by C the total capital invested and by T_i the MTL of ith asset, the corresponding weight is then:

$$w_i = \frac{x_i \, T_i}{C} \quad s.t. \quad x_i \in \mathbb{Z} \;\; and \;\; C = \sum_{j=1}^{N} x_j \, T_j. \quad (C4)$$

Cardinality constraints (CC)

Cardinality constraints limit the number of assets held in a portfolio, such that:

$$\sum_{i=1}^{N} \rho_i = K, \quad (C5)$$

where K is a pre-specified positive integer number, $(0 \leq K \leq N)$ and ρ_i is a binary function that indicates whether the ith asset is included in the portfolio, $\rho_i \in \{0, 1\}$. These constraints can be generalized in the form $\sum_{i=1}^{N} \rho_i \leq K$.

Buy-in thresholds constraints (BT)

With regard to reducing transaction costs, if an asset is held in the portfolio at least some proportion l of it is purchased:

$$|w_i| \geq l \, \rho_i, \quad (C6)$$

These constraints reduce visibly small holdings.

Transaction costs (TC)

Trading in financial markets is accompanied with costs. Two kinds of costs can be found in the literature; the fixed costs, which are independent of the volume of transactions, and the proportional costs. Usually, these costs are incorporated as a penalty function in the objective function of POP. However, they involve complicated nonlinear functions, a proper modeling is then necessary. For example, the fixed transaction costs function have been modeled as a V-shaped function by Yoshimoto [55] and as picewise constant function by Konno and Wijayanayake [25].

3. OPTIMIZATION TECHNIQUES: GA

3.1 Basic Description

GAs differ from other metaheuristics by starting with a random set of candidate solutions, denoted $P(0)$ and called *population*. A population is a constant-size set of individuals called *chromosomes* evolving through successive iterations, called *generations*. Each chromosome is a representation of a possible solution, and it is created by choosing an appropriate *encoding*. During each generation, new chromosomes are generated, the *offspring*, by selecting potential parents which are merged by pairs via the *crossover operator* and slightly changed via the *mutation operator*. Thereafter, the offspring replace their parents to keep the population size constant. A fitness function is formed to assess the quality of each chromosome. The termination criterion for iterations, can be either reaching a pre-specified maximum number of generations, or detecting that the best fitness of the population, does not change significantly during successive generations.

Representation scheme
The first step of applying GAs on a problem is to find a suitable encoding of chromosomes which can store the problem specific information. This may be a conventional representation as binary encodings, real-valued encodings, or more complex data structures, e.g., graphs. A string chromosome is commonly composed by a series of units called *genes*, which are the smallest elements manipulated. Each gene belongs to a set of symbols, termed the *alphabet*. In binary encodings, the alphabet is {0,1}. All the elements, the chromosome length, the alphabet and the encoding, are called the *representation scheme*. The following genetic operators depend strongly on this representation.

Selection scheme
The selection is the operation of selecting the best-quality chromosomes to "evolve" to the next generation. The result of the selection scheme is called the *mating pool*, $M(t)$, a set of chromosomes with a constant size equal to the population size, N. The best-known selection schemes are *tournament* scheme and *roulette-wheel* scheme. The roulette-wheel scheme consists firstly of constructing a roulette-wheel of chromosomes where the number of slots assigned to a chromosome is proportional to it fitness. After, the choice of $M(t)$ is performed by N stochastic separate selections on the roulette wheel. While, the concept of tournament scheme involves selecting successively N times 2 random chromosomes from $P(t)$ and putting the fitter of the two into $M(t)$.

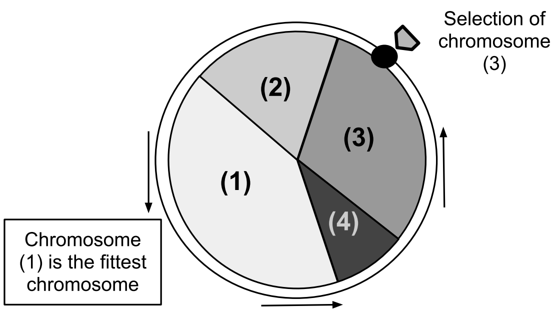

Figure 2: The roulette-wheel scheme

Crossover operator
The crossover takes a pair of parent chromosomes from the mating pool with a probability p_c (for each chromosome), and generates a pair of offspring chromosomes by combining the features of parents, e.g., exchanging substrings. Many types of crossover operators have been introduced [18], such as *one-point crossover*, *multi-point crossover* and *order crossover*. One-point crossover, the simplest crossover operator, involves merely exchanging substrings from a random site of the parents, called *one-cut point*.

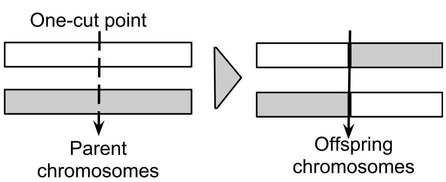

Figure 3: The one-point crossover

Mutation operation
Mutation is seen as a background operator relative to the crossover. The purpose of the operator is to maintain some degree of diversity in the population, and thereby avoid stagnation at a local optimum. The mutation operator takes each chromosome from the mating pool, and alters one or more genes (symbols) with a probability p_m, usually fairly low, e.g., 0.01. Many types of mutation operators have been introduced [18], such as *flip-bit mutation*, *insertion mutation* and *inversion mutation*. The flip-bit mutation reverses the value of a randomly selected gene as follows.

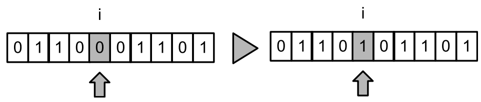

Figure 4: The flip-bit mutation

Genetic drift
The main issue of GAs is known as *Genetic drift*, where individuals lose their genetic diversity and the population ends up quickly with several similar chromosomes which may represent a local optimum of the problem. Mutation and random initialization are not enough to avoid the lack of diversity. In fact, if p_m is set to high value, GAs will turn on a primitive random search.
The basic steps of GAs can be summarized as follows:

Algorithm 1 Algorithmic description of GA

maxGen: maximum number of generations,
popSize: population size, p_c: crossover probability,
p_m: mutation probability

1: Set t:=0, initialize ($popSize$) the population $P(0)$;
2: Evaluate $P(t)$;
3: If termination criterion fulfilled ($maxGen$) then DONE;
4: Select the mating pool $M(t)$ from $P(t)$, $M(t) := s(P(t))$;
5: Apply the crossover operator $M(t) := c(M(t), p_c)$;
6: Apply the mutation operator $M(t) := m(M(t), p_m)$;
7: Form $P(t+1) := M(t)$
8: Set $t := t + 1$; go to step 2

The operations s, c and m denote respectively the selection, crossover and mutation operators, where t represents the generation number and $P(t)$ the population at generation t. In order to reinforce the performance of GAs, some authors add a genetic operator called *elitism strategy*. This strategy consists simply of copying integrally the fittest chromosome of the current population into the following one. Some empirical results show that elitism has a considerable impact on the performance of GAs [41]. It specially ensures that the best fitness of the population can never be reduced throughout generations.

3.2 Multiobjective Evolutionary Algorithms

Since GAs are population-based algorithms, several solutions can be kept simultaneously throughout all the process. Consequently, GAs are capable to handle multiobjective optimization problems, where different solutions can be found based on the notion of *non-dominance*. A solution is called non-dominated if it is not dominated by any other solution (see below the definition). In this context, GAs are called

Multiobjective Evolutionary Algorithms, MOEAs. The main advantage of MOEAs is the ability to construct in only one single run the set of *Pareto-optimal solutions*, which is the set of all feasible non-dominated solutions. This set corresponds indeed, in the case of the POP, to the efficient frontier. The difference that can be highlighted between GAs and MOEAs concerns the ranking of chromosomes. While in MOEAs a ranking based on non-dominance is used, single-objective GAs just return the fitter chromosome without any ranking (the objective function is as well as the the fitness function). In addition, MOEAs must maintain a degree of diversity among chromosomes. The *NSGA-II*, one of the-state-of-the-art MOEAs, have enjoyed considerable attention. A presentation of it is provided in the following part.

DEFINITION 1. ***Definition of domination***
Given a problem of the type $[mini/max_x f_1(x), ..., f_N(x)]$, *such that* N *is the number of objectives and* F *is the set of feasible solutions. Without loss of generality, we assume that the problem is a minimizing problem (max* f_k *is equivalent to min* $(-1 \times f_k)$*) and N=2. We say that* $x \in F$ *dominates* $y\ (\neq x) \in F$ *(denoted* $x \prec y$*) if* $\big(f_1(x) < f_1(y) \wedge f_2(x) \leq f_2(y)\big) \vee \big(f_1(x) \leq f_1(y) \wedge f_2(x) < f_2(y)\big)$

NSGA and NSGA-II

The *Non-dominated Sorting Genetic Algorithm, NSGA*, has been proposed by Deb et al.[46]. The only difference between a GA and a NSGA is the redefinition of the selection operator. In fact, two steps have been established on this operator. The first one starts by pulling the non-dominated chromosomes in a front and crediting them the same fitness value (fictitious) which is inherently high. Thereafter an operation, called *sharing* by the authors, redefines the fitness of each chromosome according to his neighborhood. Indeed, a neighborhood is defined by specifying the *sharing parameter* which is the maximum distance between two individuals in the same neighborhood. This step is repeated until all the population is classified in several fronts. Noting that the fictitious fitness value given to the first front is larger than the second and so on. The computational cost of all previous step is $O(MN^3)$ where M is the number of objective functions and N is the population size. The second step is a stochastic selection operator that will construct the mating pool such that the chromosomes with larger fitness value have more chance to be picked.

Seven years after Deb et al.[12] introduced a new version: NSGA-II, where the convergence and the spread of the solutions are improved. The adjustments are: designing a new algorithm of creating fronts with a better computational complexity $O(MN^2)$, adding an elitism approach and changing the process of diversification. Henceforth, the maintaining of diversity relies on a new operator without any input parameter called the *crowded-comparison*. Hence, the chromosomes are classified by favoring those of the first fronts and in the same front, the more isolated ones, i.e., with a less *crowded distance*. In the case of two objectives, the crowding distance of a chromosome is the perimeter of the rectangle bordered by the nearest right and left neighbors. At the end of the evolution operators, the offspring are added to their parents, thereafter the whole population of $2N$ individuals are ranked according to fronts repartition and crowded-comparison. The new population is created from the N first ranked chromosomes (the elitism approach). The authors validate their algorithm using nine test problems. The results shows that NSGA-II presents better results and diversity support compared with two other elitist MOEAs, PAES and SPEA. Lastly, NSGA-II was simulated with two different encodings; binary and real-valued.

4. MULTIOBJECTIVE EA

To categorize approaches using MOEA for solving POP, we propose to rely on the nature of portfolio models, substantially on hard constraints which make the optimization computationally hard. Notice that M-VaR is a category apart.

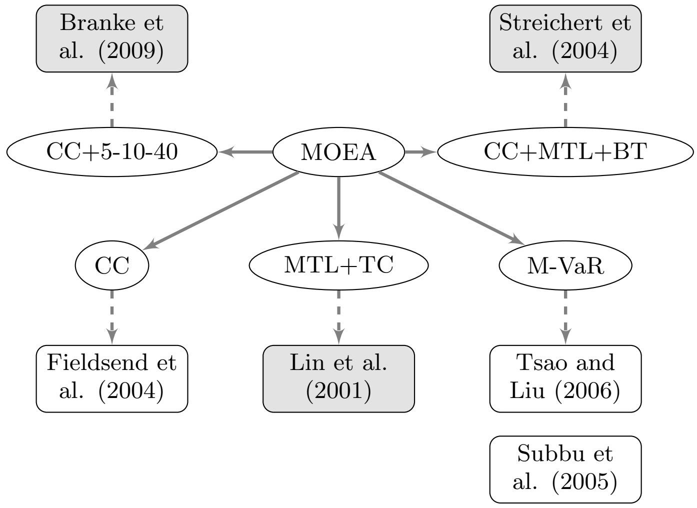

Figure 5: Classification of approaches using MOEA

Lin, D., Wang, S., and Yan, H. (2001)
Lin et al. [33] considered a M-V portfolio model with MTL, fixed transaction costs (TC) and linear constraints on capital invested similar to the holding weights constraints. A NSGA-II based algorithm is proposed to solve this constrained problem. Instead of a real-valued encoding, the authors adopted an integer encoding. As a consequence, the genetic operators used in NSGA-II are, in order to respect this encoding, altered largely by using the truncation function. The altered operators are the tournament selection, *Simulated Binary Crossover, SBX* [11] and *Parameter based Mutation* [12]. However, the persistent issue is the feasibility of chromosomes with respect to the constraints, i.e., the evolution operators can not guarantee the feasibility of offspring even the parents are completely feasible. The authors chose a single objective GA called *Genetic for Numerical Optimization of Constrained Problems, GENOCOP* [38] to handle the constraints. However, GENOCOP necessitates the feasibility of all initial individuals. Therefore, in the initialization step of the overall algorithm, an initial feasible population is constructed by using NSGA-II based algorithm on an optimization problem minimizing the violation of the constraints. In addition, two typical chromosomes are added to the initial population (by replacing two randomly chosen individuals). The idea is to inject the better variance and the better expected return into the population to reduce the *range-dependency* [6] between the non-dominated solutions. This phase is called by authors the *Fitness Scaling*. For clarity, we remind the structure of the main hybrid algorithm:

Algorithm 2 Hybrid Algorithm of Lin et al.

1: Construct an initial feasible population by using the NSGA-II algorithm with integer encoding and compute the fitness scaling;
2: Evolve the population with SBX crossover and PM mutation with the use of GENOCOP ideas;
3: Evaluate the population;
4: Select chromosomes with the tournament selection scheme and apply the elitism strategy established in NSGA-II;
5: If the number of generations is less than the maximum pre-specified number of iterations, go to step 2, else DONE.

Fieldsend, J. E., Matatko, J., and Peng, M. (2004)
Fieldsend et al. [17] chose a cardinality constrained portfolio optimization with the M-V model. Normally in order to achieve some diversification, the number of assets tends to vary inversely with the correlation between asset returns, i.e., the more asset returns are independent, the greater the number of assets is required to lead to a better portfolio. With CC, the cardinality is usually specified in advance. However, because an accurate assets number leading the investors to a better portfolio is not originally known, the authors cover in their model all possible cardinalities. Thus, the cardinality of portfolios is considered as a third objective to be minimized. The reason behind this choice is back to the possibility to find for a greater cardinality an equivalent portfolio with a lower cardinality. Actually, portfolios with higher cardinality might include a considerable group of assets zero-weighted according to the finance theory. Though this technique works only if no other constraints than CC are taken into account. Further, solving each cardinality range and gathering the results (efficient frontiers) after is certainly more computationally expensive. A heuristic based on the MOEA *(1+1)-evolution strategy* [16] is used to solve this computationally hard problem.

The used algorithm depends on m parallel heuristic searches related to each other, where m is the higher portfolio cardinality (an input of the algorithm). Concretely, the algorithm makes evolve throughout generations a container set of m size, where each element is a set H_k of non-dominated solutions given a certain cardinality $1 \leq k \leq m$. During each iteration, a cardinality k is chosen uniformly at random, thereafter, a portfolio is selected from H_k. The following operations perform on a copy of this portfolio, starting with the significant *adjustment operator*. Adjusting a portfolio takes two forms: rebalancing of the nonzero weights, or changing the cardinality plus the rebalancing. Both situations relate to the *Dirichlet distribution*, since this distribution guarantees the budget constraints (the weights add up to one). Before starting a new iteration, the portfolio result, whether it keeps the same cardinality $l = k$ or not $l = k'$, is evaluated (calculating r_p and σ_p) and compared to the elements of H_l, to check if it is non-dominated to keep it in H_l, or otherwise to remove it.

Streichert, F., Ulmer, H., and Zell, A. (2004)
Streichert et al. [48] dealt with a M-V portfolio model, constrained to cardinality, buy-in thresholds and MTL constraints. For the resolution algorithm, they used the NSGA [46] heuristic with real-valued encoding and binary encodings. For binary strings, both natural binary and gray codings are considered. The genetic operators used are tournament selection, discrete 3-point crossover, one-point mutation and bit-flip mutation for binary encodings. By noticing in their preliminary experiments that the efficient frontier of the POP is generally composed of a restrictive number of the initial available assets (particularly in the case of CC), the authors outline the analogy with the one-dimensional binary knapsack problem. Thereafter, a new hybrid representation based on the binary knapsack problem is proposed, since this problem is already approached by GAs. In the new representation a chromosome is expressed with two vectors of the same size, namely the real-valued vector of weights W and a binary vector B where each bit indicates if the corresponding asset is used or not in the portfolio. Hence, the advantage is to make adding/removing assets in the portfolio much simpler for GA, by using bit-flips. During the evolution process, the elements of the knapsack representation (W and B) are affected separately by mutation and crossover.

Since hard constraints are adopted, handling offspring chromosomes (generated by evolution operators) which violate these constraints is crucial. The authors propose a repair algorithm to deal with these infeasible individuals. The first step is to satisfy CC and buy-in threshold constraints. The $(N - K)$ smallest weights of the infeasible chromosome and weights that do not meet the buy-in threshold are set to zero, with N and K are respectively the total number of assets and the specified cardinality. After, the weights are normalized. The following step rounds the remaining weights to satisfy MTL constraints. This repair mechanism is deterministic. To examine its efficiency, the authors run the GA with and without Lamarckism. The *Lamarckian strategy* assumes that the improvements of an individual during its lifetime can alter the way in which it is encoded (*genotype*). Hence, in the experiment where the Lamarckism is adopted, the authors keep the repaired solution in the population.

Subbu, R., Bonissone, P. P., Eklund, N. H. W., Bollapragada, S., and Chalermkraivuth, K. C. (2005)
Subbu et al.[51] introduced an hybrid evolutionary multi-objective optimization approach allying linear programming and evolutionary computation (generic term referring to GAs). They also introduced a new model where different measures of risk are considered in order to capture different aspects of portfolio risks, in particular from the point of view of the *asset-liability management, ALM.* Thereby, surplus variance is added to VaR as a second measure of risk. Whereas in ALM approach the investor have to match portfolio assets with the liabilities which are influenced by several risk factors, further linear constraints are added. These constraints serve to limit *Duration mismatch* and *Convexity mismatch*; financial accounting variables which can be easily linearized. The used model is as follows:

$$\begin{aligned}
&Maximize\ \ \mu_p\ (Portfolio\ Expected\ Return)\\
&Minimize\ \ VaR_p\\
&Minimize\ \ Surplus\ Variance_p\\
&subject\ to\ \ Duration\ mismatch < l_1\\
&\quad and\ \ Convexity\ mismatch < l_2\\
&\quad and\ \ linear\ portfolio\ investment\ constraints
\end{aligned}$$

The hybrid used algorithm relies on *Pareto Sorting Evolutionary Algorithm, PSEA*, to handle the non-dominated chromosomes. This algorithm keeps the non-dominated solutions found throughout generations in an archive, though the population is supposed to have a small size. To further preserve the diversity during the search, the authors added three mechanisms to the algorithm: a new crossover operator, inclusion of a new randomly generated solutions in each iteration and addition of a non-crowding filter to have more dispatched solutions. PSEA is initialized using *Random Linear Programming, RLP* algorithm. The aim of using RLP is to provide initial solutions which are likely to meet the problem constraints (which are linear). By solving multiple linear programs and gathering them with randomly generated weights, RLP can give extreme limits sampled from the search space. The authors used also a *Fast Dominance Filter, FDF*, which is a method that splits the dominated and non-dominated chromosomes of a population. According to them, FDF improves the overall computational complexity. Actually, FDF decomposes the set of solutions into smaller subsets before comparing elements inside each subset. In brief, PSEA manipulates the non-dominated solutions. However, the factual search is done by *Target Objective Genetic Algorithm, TOGA* [15]. TOGA is a non-Pareto, non-aggregating algorithm which drives the global search toward the nearest chromosomes according to a pre-specified target. A graphical tool allowing the projections of efficient frontiers is used during the run of the hybrid algorithm. The decision maker can hence incorporate its preferences during the process of optimization by downselecting assets.

Tsao, C., and Liu, C. (2006)
The paper of Tsao and Liu [53] relates to the application of NSGA-II [12] to the Mean-VaR portfolio framework (described in section 2.2). Only the budget constraint is considered in the model. However, due to the non-convexity of the VaR function in two of its calculation methods, i.e., the historical simulation and the Monte Carlo simulation, the optimization becomes computationally expensive in terms of time. Thus, the standard mathematical techniques can not be applied in this case. The authors have brought some changes to the NSGA-II, though the binary encoding is kept. In fact, the random initialization is modified by adding a threshold $U \in [0,1]$ to the value randomly generated. If the value uniformly generated from $[0,1]$ is greater than U, 0 is chosen, otherwise it is maintained. The second alteration concerns the method to spot the non-dominated solution set given a population of chromosomes. Three approaches are known; *naive and slow approach, continuously updated approach* and *Kung et al.'s efficient method* [27]. The last method is computationally efficient with a small number of objective functions, therefore it was chosen by the authors. Kung et al's efficient method starts by dividing the population in half, the top part T and the bottom part B, after sorting individuals according to the first objective. Directly afterward, the method is recursively applied to both T and B. In each call of the function, iteratively all individuals of B are compared to those of T. If an individual of B is not dominated by any other of T, it is stored in a set M. At the end the function return $T \cup M$ which represents the non-dominated solutions.

Branke, Scheckenbach, Stein, Deb and Schmeck (2009)
Branke et al. [7] have proposed to solve a M-V portfolio optimization with CC and 5-10-40 constraints. The latter constraint is a rule from the German investment law which stipulates that the share of each asset is not superior than 10% of the net asset value of the fund and the total holdings exceeding 5% are less than 40% (of the net asset value of the fund). Plus, asset shares of the same issuer are no more than 5% (of the net asset value of the fund). The 5-10-40 rule is a hard constraint, i.e., leading to a non-convex search space. The authors' approach, called *envelope-based MOEA (E-MOEA)*, incorporates *critical line algorithm* into a MOEA based on NSGA-II. The major point here is that E-MOEA is not point-based where each point is a portfolio solution, but rather handles continuous fronts as individuals. The concept is to confer the managing of hard constraints to the MOEA in the sense that chromosomes represent problems with convex constraints. In instance for CC, portfolio solutions are repaired (some weighs are forced to zero) and the convex corresponding problems are simply problems where the total number of assets matches assets of non-zero weights. Each convex problem is solved by *critical line algorithm*, which is no more than the algorithm used by Markowitz for computing the whole efficient frontier [36].

The solved convex problems correspond henceforth to continuous fronts so-called *envelopes*. The authors adapt NSGA-II to handle these envelopes by adjusting the non-dominated sorting and the crowding distance calculation. As an envelope scarcely dominates completely other envelopes, *aggregated fronts* are introduced. These fronts are defined as union of the non-dominated parts of all envelopes. And the crowding distance is substituted by the calculation of the contribution of an envelope to the aggregated front (length of the segment). Hence, it is possible to maintain a ranking of individuals. Concerning the genetic operations, permutation encoding is adopted and the uniform order based crossover and swap mutation are used to evolve the population. To test E-MOEA, the authors chose the point-based approach of Streichert et al. [50] [48] [49] as a reference. They made however some adaptations of the approach of Streichert et al. by considering 5-10-40 constraint and abandoning MTL constraints (just by adjusting the repair algorithm).

5. SINGLE OBJECTIVE GAS

Similarly to MOEA, Single Objective GAs applications of POP are categorized according to the hard constraints.

Eddelbüttel, D. (1996)
Eddelbüttel [14] formulated an index-tracking problem which is a particular form of passive management. The purpose of this problem is to duplicate within a target portfolio the behavior of a stock market index chosen as a benchmark. A formulation as an optimization problem consists in minimizing the *tracking error variance*, i.e., the variance of the differences in terms of return between the market index and its tracking portfolio. However, the expected return difference between the market index and the tracking portfolio, defined as *expected tracking error*, is specified in advance. By constraining the portfolio solution to a fewer stocks, the author applied implicitly the cardinality constraints. Thus, the problem is henceforth computationally hard, which drives the author to use a hybrid GA. Actually, during each gen-

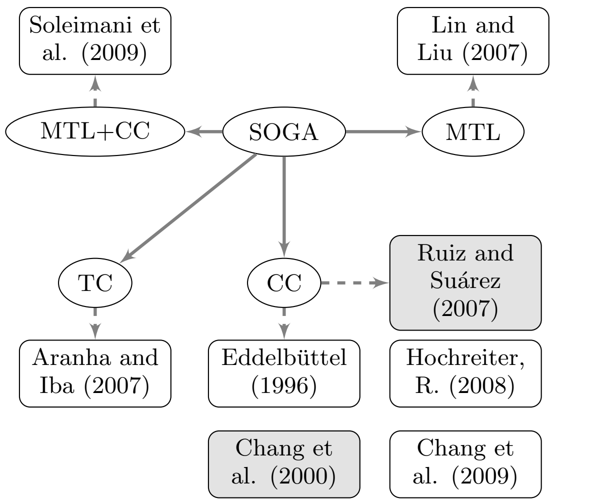

Figure 6: Classification of approaches using SOGA

eration two steps are performed. First, using the global search of GA to select assets to include into the portfolio. Thereafter, a quadratic programming solver is run to assign optimal weights for the tracking portfolio stocks. For the encoding, a binary vector of length $5k$ is used to represent a possible chromosome. The empirical experiments of the author simulated the DAX *(Deutscher Aktien IndeX)* tracking portfolio using one year of daily closing prices. These simulations show promising results on portfolios with significant size, by comparing with the canonical GA previously presented by the author [13].

Chang, Meade, Beasley and Sharaiha (2000)
Chang et al. [9] chose to solve a M-V portfolio optimization including cardinality constraints and holding weights constraints. The considered model corresponds to the risk-aversion formulation (see Section 2.1). The main purpose of the paper is to find and analyze the efficient frontier of this hard constrained optimization using three heuristic algorithms; GA, tabu search (TS) and simulated annealing (SA). At first, the authors showed that the efficient frontier in the presence of CC becomes discontinuous, by a small example problem; four assets of FTSE index with a cardinality fixed to two. In fact, all the combinations of this smooth example problem can be listed and the constrained efficient frontier is obtained by gathering unconstrained cases (solved via mathematical programming). Although this approach is correct, it may hide, according to the authors, invisible portions of the constained efficient frontier. The exact approach is hence to solve the problem once, using weighting. On the other side, the three heuristics construct all the parts of the constrained efficient frontier (hidden or not). A computational comparison is done between the heuristics using computer time and different percentage errors. For unconstrained POP, GA gives the best approximation with an almost zero mean percentage error. SA is in the second position followed by TS. For cardinality constrained POP, no heuristic is uniformly superior, i.e., in the sense that the percentage errors vary indifferently from a case to another and no algorithm emerges as winner. The authors simulate also a tradeoff experience by pooling the result of each heuristic. In this case, GA performs better by contributing the most to the final efficient frontier.

Ruiz-Torrubiano, R., and Suárez, A. (2007)
Ruiz-Torrubiano and Suárez [42] have targeted to solve the standard M-V model with cardinality constraints and holding weights constraints (on each asset and/or groups of assets). They used for this purpose a hybrid algorithm pooling Evolutionary Algorithms (EA), Quadratic Programming (QP) and a pruning heuristic. In their approach, finding the optimal subset of assets to be held in the portfolio is handled by EAs, separately than solving the optimal weights which is achieved by a standard quadratic solver. In fact, the quadratic solver used relies on the so-called *inertia-controlling* methods [19], while for EAs two distinct instances are used and compared. The first instance, named RAR-GA, is a GA employing set representation and suitable evolution operators. Actually, these operators are specially conceived to maintain the cardinality of individuals. They are *Random Assortment Recombination*, RAR, [54] for crossover and for mutation an operator exchanging an asset belonging to the portfolio with another one outside the selection. On the other hand, the second instance is an implementation of *Estimation of Distribution Algorithms*, EDAs [28]. EDAs are evolutionary algorithms where the evolution operators are replaced by sampling a probability distribution which is estimated for each generation from the previous population. Different techniques are used by the authors for the estimation process, e.g., *Univariate Marginal Distribution Algorithm, UMDA* [39], *Probability-Based Incremental Learning, PBIL* [4], and *Estimation of Gaussian Networks Algorithm, EMNA* [29]. Regarding Pruning heuristic, it is a technique intended to reduce the number of assets included in chromosomes (portfolios), and hence to reduce the computational cost of the overall hybrid algorithm. Its method is to solve the corresponding unconstrained POP and then remove from the constrained optimization problem, assets assigned zero or negligibly small weight. The experiments conducted by the authors show that without using pruning heuristic, EDAs are more time consuming than RAR-GA. And pruning heuristic generally improves the execution time for both EDAs and RAR-GA and the accuracy in the case of EDAs.

Lin, C., and Liu, Y. (2007)
Lin and Liu [32] examined the extension of Markowitz' model to integrate the MTL constraints. Actually, their approach appears to be one of the firsts allying M-V model and MTL, since these constraints have been in previous studies specially considered with MAD model. As the POP is fundamentally a *Multi-Objective Decision-Making (MODM)* problem, where two conflicting criteria are needed in making decisions, solving tools as goal programming [30] and fuzzy programming are used by the authors. In fact, three possible models are considered. Starting by the first one which is simply an adaptation of the M-V model to the MTL. This model is formulated as a single-objective problem minimizing the portfolio risk given a rate of return r, besides that, the decision variables are integers. However, the target portfolio defined as (r, σ) may be unattainable using integer solutions, with σ is the corresponding risk of r from the unconstrained efficient frontier. As a consequence, the authors proposed another model which belongs to the category of

goal programming problem. The aim of this second model is to minimize the distance between the target portfolio (r, σ) and the portfolio result of the optimization (r_p, σ_p). Nevertheless, this model may increase the computational time due to the prerequisite drawing of the unconstrained efficient frontier. In addition, a proper choice of an instance of r, which reflects accurately the degree of risk/return preference of the *Decision Maker (DM)* or the investor, is delicate. A third model relying on the fuzzy programming approach is hence proposed. The advantages of this model is to consider the DM preferences without any pre-specified inputs and discard the incommensurability between the objectives and different distance metrics (an issue of the second model). Thus, The authors adjust the weighted max-min model for fuzzy MODM proposed by Lin [31] to the POP.

A GA is used to solve the 3 optimization problems. In this algorithm, the offspring does not replace directly the parent. It replaces instead a randomly selected chromosome (except the best one), only if is fitter than the worst chromosome of the population. To maintain the feasibility of individuals, the authors used both the penalty strategies and modifying genetic operators. Hence, the budget constraints are handled with an appropriate encoding/decoding from real to integer values while the constraints of respecting the rate of return of the portfolio r is handled as a penalty function. The empirical simulations consolidate the efficiency of GA in terms of time and the solution quality for all the models considered, knowing that the results of models 2 and 3 seems more effective. However, the authors recommended the fuzzy approach because the simplicity and the adaptability of dealing with investor preferences.

Aranha, C., and Iba, H (2007)
Unlike the other applications of this survey, Aranha and Iba [3] discussed the POP in terms of multi-state continuous optimization over time, where the objective, plus increasing the return and decreasing the risk, is to minimize transaction costs. More specifically, costs are minimized between each two consecutive time periods. The authors used an indirect approach for modeling costs as *Euclidean distance* of the weight vectors of the current position (time $t-1$) and the desired position (time t). In other terms, transaction costs are associated to the amounts bought or sold of assets between two time positions. The GA-based technique used to solve this problem represents each chromosome by two arrays: a binary array which indicate if an asset is present in the portfolio and a real-valued array of weights. For the evolution operators, the authors used tournament selection, simple linear crossover, bit-flip mutation, mutation by perturbation and elitism strategy. On the other side, two genetic techniques are introduced; first the *seeding* technique which improves handling POP over time by coping some individuals from the current (time $t-1$) to the next population (time t), and second, the *objective sharing* which compute the Euclidian distances. The empirical experiments are based on monthly historical returns from the NIKKEI and the NASDAQ indexes. They show that the implemented GA ends up by dominating the simple GA (without seeding and objective sharing) in terms of sharpe ratio (the chosen fitness function), cumulative returns and average distances.

Hochreiter, R. (2008)
Hochreiter [22] used the stochastic programming in order to incorporate directly in the portfolio model the underlying uncertainty. Hence, the uncertainty that can occur in estimating correct expected returns and variance matrix is handled by generating scenarios of possible realizations for asset returns (a probability is affected to each scenario). By considering a scenario-based risk-return approach, the domain of portfolio risk is extended to contain different measures. Indeed, the adopted portfolio risk measure is based on a general discrete *loss distribution* regardless its exact underlying structure. In fact, this distribution is the cross product of the regarded portfolio and the matrix of the set of scenarios. For empirical simulations, four structures of risk measure are applied: standard deviation, mean absolute downside semi deviation, VaR, expected shortfall. The author considered also some real-life constraints, namely the cardinality constraints and the holding weights constraints (upper and lower limits on asset weights). In the end, the resolution method relies on a evolutionary approach based on a GA using real-valued genes with N-point crossover, intermediate/blend crossover and mutate by factor. The numerical results are based on weekly historical data of 14 selected assets of Dow Jones Index.

Chang, T., Yang, S., and Chang, K (2009)
Chang et al. [10] considered the POP constrained to cardinality and linear holding weights constraints (HWC), within the risk-aversion formulation. Three different risk measures in addition to the variance are adopted, namely mean absolute deviation, semivariance and variance with skewness. The two latter risk measures are improvement of the variance; by taken into account only the returns below the mean for semivariance and by including skewness for variance with skewness. A GA is proposed to solve this problem with uniform crossover, binary tournament selection and a replacement strategy where instead of replacing the parents, the offspring chromosomes replace the worst fitter individuals in the population. To ensure the feasibility of chromosomes to HWC, a specific iterative procedure is established based on splitting the chromosomes on two parts; the minimum proportion part plus a complementary share. The empirical simulations of the authors show that the computation time increases with the growth of the cardinality and that efficient frontiers of higher cardinality are dominated by those of lower cardinality. Hence, according to the authors, an investor, in order to expect an effective portfolio, should not consider a cardinality greater than one-third of the total assets. Finally, because the risk measures illustrate different incompatible risks, the risk models are not directly compared. However, this application asserts the flexibility and the efficiency of GA to handle different risk models within the same framework.

Soleimani, H., Golmakani, H., and Salimi, M.(2009)
Soleimani et al .[45] suggested to solve a M-V portfolio optimization, where CC, MTL and constraints on sector capitalization are taken into account. The sector capitalization constraints suppose that some assets belong to sectors (sets of assets) and state that the capital invested in sector 1 is greater than the one invested in sector 2 and so on. The ad-

vantage of these constraints is let investors to invest in some sectors with high-value in a manner to reduce the overall risk. Regarding the formalization, it involves inequalities on the asset weights belonging to the sectors. However, the use of binary functions (if an asset is held or not) makes these constraints hard (combinatorial). To solve this mixed-integer programming problem, the authors used a GA with the RAR crossover operator [40] and a selection operator wherein half of the population is conducted to the following generation by choosing the fitter chromosomes, and the other half is composed of offspring chromosomes. The empirical experiments start by comparing the results of GA on a small assets problem with those of *LINGO*, which is an optimization modeling software. The error differences of both approaches are minimal (no more than 3%), however GA is undoubtedly superior in terms of computational time. The other experiments involve 2000 assets and show the efficiency of GA: computational time around 7min with 3.5% of risk error.

6. COMPUTATIONAL ANALYSIS

Genetic Algorithm presents an undeniable computational advantages. Shoaf and Foster [44] approximate the complexity of GA for the original Markowitz' POP to $O(n\,log(n))$, with n is the number of assets, which is better than a quadratic complexity. In this section, we discuss in brief the computational aspects of the approaches mentioned so far. However, for almost the techniques and the approaches we reported, from the published papers, only summaries of computational results and upper-level descriptions of algorithms are provided. Generally, these papers do not provide enough information to reproduce algorithms implementation directly. In addition, the applications do not share a unique portfolio model. A shown in Figures 5 and 6, the hard constraints differ for almost all the applications, which makes any fair computational comparison seemingly impossible. The test instances are also different in nature, i.e., asset returns are compounded depending the application from daily, weekly or even monthly returns. Despite that, a collection of applications share the same simulation benchmark from the OR-library [5], which is a compilation of publicly available test data sets for range of operation research problems. Those applications are displayed with dark background on Figures 5 and 6. In our case, this benchmark is the test data used by Chang et al. [9], which is weekly prices data from March 92 to September 97, of component assets of the indexes: Hang Seng, Dax100, FTSE100, S&P100 and Nikkei225. For the above mentioned reasons, we have been limited in our analysis to a theoretical discussion only, rather than computational simulations. To structure the discussion after, MOEA applications are examined separately from the SOGA ones, since they do not return the same result.

First, MOEA are believed to be more effective in terms of computational time than SOGA, as Anagnostopoulos and Mamanis [2] pointed out in their study where they compared on a cardinality constrained POP five advanced MOEAs, among them NSGA-II, and a SOGA similar to the one proposed by Chang et al. [9]. In the computational results revealed by Lin et al. [33] based on the simulation of the Hang Seng data from the OR-library, the designed algorithm accurately approximates the unconstrained efficient frontier when MTL are equal to 1 or 100. However, the lack of references when fixed transaction costs are added to the model does not allow them to compare the result. Streichert et al. [48] examined CC and BT constraints in addition to the MTL given the same data set. They showed that their hybrid algorithm has better results when the knapsack representation is adopted. More precisely, the hybrid knapsack-GA with 7bit gray-coding with Lamarckism is the best in terms of convergence speed and fitness results. In fact, 7bit coding performs better than the other coding because the evolution operators appear to be more productive. The emipirical results of Fieldsend et al. [17] approach, where a cardinality constrained POP has been targeted, accord that the computational performance of the constrained and the unconstrained POP are not considerably different in the case of small asset number. Recently Branke et al. [7] reported better computational results based on their algorithm E-MOEA by comparison to Streichert et al. approach [48]. The authors explained this by the recourse to envelopes (continuous fronts) and the critical line algorithm inside their hybrid algorithm. Tsao and Liu [53], by using the daily prices of stocks of the TSEC Taiwan 50 index, compared four efficient frontiers (EF), i.e., the M-V EF and the M-VaR EFs according to the three calculation methods of VaR. The experiments of their algorithm based on NSGA-II, show that the three M-VaR EFs are similar for the levels of confidence of VaR around 90% and 95%. When the level of confidence is increased to 99%, the M-V EF does not match the M-VaR EFs, especially for M-VaR based on historical simulation method which seemed not quite interlaced.

On the other hand, SOGAs have various applications with different goals. For instance, Eddelbüttel [14] constructed an index-tracking portfolio with CC using an hybrid GA. His empirical simulations, which are based on Monte Carlo experiments, show favorable results. More precisely, the hybrid GA outperform the canonical GA as long as the CC increase. In the computational study of Chang et al.[9], GA approximates better the unconstrained EF compared to tabu search or simulated annealing, with an average mean percentage error amounting to 0.0114% on OR-library data. Per contra, for the cardinality constrained EF no heuristic seems better than the others. To give an idea of the computational time, the processing of the Nikkei225 data approaches 33 hours. When the results of heuristics are pooled into a unique EF, GA in this case contributes the most to the EF, e.g., 39.5% for the Hang Seng data. In the same register of cardinality constrained POP, Ruiz-Torrubiano and Suárez [42] indicate the superiority in performance of RAR-GA over EDA algorithm. However, the use of pruning heuristic makes EDA competitive with RAR-GA. Lin and Liu [32] analyzed the POP with MTL. They obtained near-optimal results especially for the fuzzy model and the goal programming model. Taiwanese mutual fund data from 1997 to 2000 are used for this study. Concerning the other SOGA approaches; Aranha and Iba [3], Hochreiter [22], Chang et al. [10] and Soleimani et al .[45], the reader may refer to Section 5 for further information about the empirical results. Finally, about the technical characteristics of computers on which the experiments had been conducted, only some papers have specified it [7][9][32][45].

7. DISCUSSION

Almost all applications reviewed in this survey are not time-dependent but rather single-period investment operations. Drawing attention to further applications to portfolio management (over time) with the consideration of hard constraints seems quite worthwhile and practical. Some of the aspects to be explored more in GAs concern the possible ways to handle the constraints within the algorithm. Besides adopting penalty strategies and rejecting or repairing the infeasible individuals, finding a more appropriate representation scheme incorporating the problem-specific-knowledge is more appealing as known. For POP the use of a suitable representation beyond the binary or the real-valued encodings is particularly scarce in MOEA when compared with SOGA applications. Hence, investigating possible representations including problem knowledge of POP will certainly improve the performance and the power of MOEA. Among the models examined we can notice that the cardinality constrained POP has raised much attention starting with the work of Chang et al.[9]. Finally, GAs, besides being a good optimization tools to obtain approximate solutions, improve our understanding of the nature of the solution space, especially in the area of finance where the optimization techniques are widely needed.

8. CONCLUSIONS

In this paper, we have surveyed 14 state-of-the-art approaches to portfolio optimization problems that use genetic algorithms. For simplicity, we handled only cases where the target optimization model is theoretically hard, which includes portfolio models with nonlinear constraints or models involving non-convex risk measures as Value-at-Risk. As known, two different forms of GAs can be clearly categorized: multiobjective EA and single-objective GA. The classification of the 14 approaches follows this distinction, since MOEA and SOGA do not return the same output, that is a single portfolio for SOGA and the whole efficient frontier for MOEA. At the end, computational results of the approaches were summarized after providing descriptions of the techniques, models and algorithms involved.

9. REFERENCES

[1] Alexander, G., and Baptista, A. Economic implications of using a mean-var model for portfolio selection: A comparison with mean-variance analysis. *Journal of Economic Dynamics and Control 26*, 7-8 (2002), 1159–1193.

[2] Anagnostopoulos, K., and Mamanis, G. The mean-variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Systems with Applications* (2011).

[3] Aranha, C., and Iba, H. Modelling cost into a genetic algorithm-based portfolio optimization system by seeding and objective sharing. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on* (2007), IEEE, pp. 196–203.

[4] Baluja, S. Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning. Tech. rep., DTIC Document, 1994.

[5] Beasley, J. Or-library. online, 2006. `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`.

[6] Bentley, P., and Wakefield, J. Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms. *Soft Computing in Engineering Design and Manufacturing, Part 5* (1997), 231–240.

[7] Branke, J., Scheckenbach, B., Stein, M., Deb, K., and Schmeck, H. Portfolio optimization with an envelope-based multi-objective evolutionary algorithm. *European Journal of Operational Research 199*, 3 (2009), 684–693.

[8] Castillo Tapia, M., and Coello, C. Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on* (2007), IEEE, pp. 532–539.

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

[10] Chang, T., Yang, S., and Chang, K. Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications 36*, 7 (2009), 10529–10537.

[11] Deb, K., and Agrawal, R. Simulated binary crossover for continuous search space. *Complex Systems 1*, 9 (1994), 115–148.

[12] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on 6*, 2 (2002), 182–197.

[13] Eddelbüttel, D. A genetic algorithm for passive management: Creating index funds with fewer stocks. *Arbeitspapier, präsentiert bei: Forecasting< s Financial Markets: Advances for Exchange Rates, Interest Rates and Asset Management, London 27* (1996), 29.

[14] Eddelbüttel, D. A hybrid genetic algorithm for passive management. In *Working Paper presented at the Second conference on computing in economics and Finance, Society of computational economics, Geneves, Suisse* (1996).

[15] Eklund, N. *Multiobjective visible spectrum optimization: A genetic algorithm approach.* PhD thesis, Rensselaer Polytechnic Institute, 2002.

[16] Fieldsend, J., Everson, R., and Singh, S. Using unconstrained elite archives for multiobjective optimization. *Evolutionary Computation, IEEE Transactions on 7*, 3 (2003), 305–323.

[17] Fieldsend, J. E., Matatko, J., and Peng, M. Cardinality constrained portfolio optimisation. In *IDEAL* (2004), pp. 788–793.

[18] Gen, M., and Cheng, R. *Genetic algorithms and engineering optimization*, vol. 7. Wiley-interscience, 2000.

[19] Gill, P., Murray, W., Saunders, M., and Wright, M. Inertia-controlling methods for general quadratic programming. *Siam Review* (1991), 1–36.

[20] Goldberg, D. *Genetic algorithms in search, optimization, and machine learning.* Addison-wesley, 1989.

[21] Guaranty, M. Riskmetrics technical document. *New York: Morgan Guaranty* (1994).

[22] Hochreiter, R. Evolutionary stochastic portfolio optimization. *Natural Computing in Computational Finance* (2008), 67–87.

[23] Holland, J. *Adaptation in natural and artificial systems.* No. 53. University of Michigan press, 1975.

[24] Konno, H. Portfolio optimization using l1 risk function. *IHSS report 88*, 9 (1988), 139–156.

[25] Konno, H., and Yamamoto, R. Integer programming approaches in mean-risk models. *Computational Management Science 2*, 4 (2005), 339–351.

[26] Konno, H., and Yamazaki, H. Mean-absolute deviation portfolio optimization model and its applications to tokyo stock market. *Manage. Sci. 37*, 5 (May 1991), 519–531.

[27] Kung, H., Luccio, F., and Preparata, F. On finding the maxima of a set of vectors. *Journal of the ACM (JACM) 22*, 4 (1975), 469–476.

[28] Larranaga, P., and Lozano, J. *Estimation of distribution algorithms: A new tool for evolutionary computation*, vol. 2. Springer, 2002.

[29] Larranaga, P., Lozano, J., and Bengoetxea, E. Estimation of distribution algorithms based on multivariate normal and gaussian networks. *Dept. Comput. Sci. Artif. Intel., Univ. Basque Country, Vizcaya, Spain, Tech. Rep. KZZA-1K-1-01* (2001).

[30] Lee, S., and Lerro, A. Optimizing the portfolio selection for mutual funds. *The Journal of Finance 28*, 5 (1973), 1087–1101.

[31] Lin, C. A weighted max–min model for fuzzy goal programming. *Fuzzy sets and systems 142*, 3 (2004), 407–420.

[32] Lin, C., and Liu, Y. Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operational Research 185*, 1 (2008), 393–404.

[33] Lin, D., Wang, S., and Yan, H. A multiobjective genetic algorithm for portfolio selection problem. In *Proceedings of ICOTA 2001, Hong Kong* (December 2001).

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

[35] Markowitz, H. Portfolio selection. *The journal of finance 7*, 1 (1952), 77–91.

[36] Markowitz, H. *Mean-variance analysis in portfolio choice and capital markets*. Blackwell, Oxford [u.a.], 1987.

[37] Markowitz, H., and Markowitz, H. *Portfolio selection: efficient diversification of investments*, vol. 16. Wiley, 1991.

[38] Michalewicz, Z. *Genetic algorithms+ data structures*. Springer, 1996.

[39] Mühlenbein, H. The equation for response to selection and its use for prediction. *Evolutionary Computation 5*, 3 (1997), 303–346.

[40] Radcliffe, N. Forma analysis and random respectful recombination. In *Proceedings of the fourth international conference on genetic algorithms* (1991), San Marco CA: Morgan Kaufmann, pp. 222–229.

[41] Rudolph, G. Evolutionary search under partially ordered sets. *Dept. Comput. Sci./LS11, Univ. Dortmund, Dortmund, Germany, Tech. Rep. CI-67/99* (1999).

[42] Ruiz-Torrubiano, R., and Suárez, A. Use of heuristic rules in evolutionary methods for the selection of optimal investment portfolios. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on* (2007), IEEE, pp. 212–219.

[43] Schlottmann, F., and Seese, D. Financial applications of multi-objective evolutionary algorithms: Recent developments and future research directions,Ó. *Applications of Multi-Objective Evolutionary Algorithms* (2004), 627–652.

[44] Shoaf, J., and Foster, J. The efficient set ga for stock portfolios. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on* (1998), IEEE, pp. 354–359.

[45] Soleimani, H., Golmakani, H., and Salimi, M. Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm. *Expert Systems with Applications 36*, 3 (2009), 5058–5063.

[46] Srinivas, N., and Deb, K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation 2*, 3 (1994), 221–248.

[47] Stein, M., Branke, J., and Schmeck, H. Efficient implementation of an active set algorithm for large-scale portfolio selection. *Computers & Operations Research 35*, 12 (2008), 3945–3961.

[48] Streichert, F., Ulmer, H., and Zell, A. Comparing discrete and continuous genotypes on the constrained portfolio selection problem. In *Genetic and Evolutionary Computation–GECCO 2004* (2004), Springer, pp. 1239–1250.

[49] Streichert, F., Ulmer, H., and Zell, A. Evaluating a hybrid encoding and three crossover operators on the constrained portfolio selection problem. In *Evolutionary Computation, 2004. CEC2004. Congress on* (2004), vol. 1, IEEE, pp. 932–939.

[50] Streichert, F., Ulmer, H., and Zell, A. Evolutionary algorithms and the cardinality constrained portfolio optimization problem. In *Selected Papers of the International Conference on Operations Research (OR 2003)* (2004), pp. 253–260.

[51] Subbu, R., Bonissone, P. P., Eklund, N. H. W., Bollapragada, S., and Chalermkraivuth, K. C. Multiobjective financial portfolio design: a hybrid evolutionary approach. In *Congress on Evolutionary Computation* (2005), pp. 1722–1729.

[52] Tobin, J. Liquidity preference as behavior towards risk. *The Review of Economic Studies 25*, 2 (1958), 65–86.

[53] Tsao, C., and Liu, C. Incorporating value-at-risk in portfolio selection: An evolutionary approach. Tech. rep., Working Paper, 2006.

[54] Whitley, L. *Foundations of genetic algorithms 2*, vol. 2. Morgan Kaufmann, 1993.

[55] Yoshimoto, A. The mean-variance approach to portfolio optimization subject to transaction costs. *Journal of the Operations Research Society of Japan 39*, 1 (1996), 99–117.