



Adaptive evolutionary algorithms for portfolio selection problems

Gianni Filigrasso¹ · Giacomo di Tollo²

Received: 13 June 2022 / Accepted: 2 November 2022 / Published online: 15 February 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

In this contribution we propose to solve complex portfolio selection problems via Evolutionary Algorithms (EAs) that resort to adaptive parameter control to manage the Exploration versus Exploitation balance and to find (near)-optimal solutions. This strategy modifies the algorithm's parameters during execution, and relies on continuous feedbacks provided to the EA with respect to some user-defined criteria. In particular, our study aims to understand whether a standard EA can benefit from a robust method that iteratively selects the crossover operator out of a predefined set, in the context of optimised portfolio choices. We apply this approach to large-scale optimization problems, by tackling a number of *NP-hard* mixed-integer programming problems. Our results show that generic EAs equipped with single crossover operator do not perform homogeneously across problem instances, whereas the adaptive policy leads to robust (and improved) solutions, by alternating exploration and exploitation on the basis of the features of the current search space.

Keywords Portfolio optimization · Evolutionary algorithms · Adaptive parameter control · Parameter setting · Mixed-integer programming

1 Introduction

Evolutionary Algorithms (EAs) are high-level strategies that aim to solve optimization problems by mimicking the Darwin theory of natural evolution and selection. From a conceptual point of view, EAs manage (i.e., maintain and manipulate)

✉ Gianni Filigrasso
filigrasso.gianni@gmail.com

Giacomo di Tollo
giacomo.ditollo@unisannio.it

¹ Department of Economics, Ca' Foscari University of Venice, Sestiere Cannaregio 873, 30121 Venezia, Italy

² Dipartimento di Diritto, Economia, Management e Metodi Quantitativi, University of Sannio, Via delle Puglie 82, 82100 Benevento, Italy

a population of individuals, and modify them via the application of variation operators: this is done to identify a (near)-optimal solution that optimizes a given *fitness* function, used to assess the individuals' quality.

Since their introduction (Holland 1992; Goldberg 1989), Evolutionary Algorithms have been applied to many fields including logistics (Prins 2004; Archetti et al. 2012), meta-optimization (Birattari et al. 2002; Smit and Eiben 2009), economics (Tapia et al. 2007; Arifovic 1994) and finance (Dempster and Jones 2001; Hoogs et al. 2007): with respect to the last field, we mention in particular credit risk management (Wang et al. 2022; Brabazon and Keenan 2004), algorithmic trading (Brabazon and O'Neill 2004; Jeong et al. 2021) and the Portfolio Selection Problem (PSP) (Markowitz 1952, 1959), which represents one of the most studied subjects in finance.

The basic idea of portfolio selection developed by Markowitz (1952) involves the selection of promising assets in terms of their return and risk, and the allocation of capital to each of them: PSP was formulated as a bi-criteria optimization problem in a mean-variance framework, under the hypothesis that the distributions of asset returns are fully described by their means, variances and covariances: a growing body of literature outlines that the approximation of the exact maximum expected utility achieved by optimal mean-variance outperforms the one obtained by asymmetric risk measures for a variety of utility functions (see e.g. Carleo et al. 2017 for a comprehensive empirical study). However, normal returns or quadratic utility are sufficient but not necessary conditions for the optimality of mean-variance portfolios (Markowitz 2014), and since neither of the two assumptions hold in practice and given that mean-variance solutions are severely unstable (Corazza et al. 2013; Proccacci and Aste 2022), new approaches have been investigated, and will be discussed in what follows.

First, evidence shows that asset returns are asymmetric and display excess kurtosis, so that the mean-variance approach cannot fully describe the investor preferences: new risk measures (Chen and Wang 2008; Maillard et al. 2010) have been proposed to satisfy some formal properties (Artzner et al. 1999), to take into account non-normal empirical distributions (Gilli et al. 2006) and/or to deal with unstable and poorly diversified solutions arising from noisy estimates of expected returns (Kizys et al. 2022). Second, it has to be taken into account that realistic formulations of the optimization problem have to consider extra-features as integer constraints, leading to mixed-integer constrained problems that have been proven to be *NP-hard* (Moral-Escudero et al. 2006), and that cannot be solved by *traditional* methods: in this case, EA-based approaches are particularly attractive, on the basis of different features that will be detailed in what follows.

The first application of EAs to PSP dates back to the nineties' (Arnone et al. 1993; Loraschi et al. 1995; Loraschi and Tettamanzi 1995; Liu and Stefek 1995), and take into account different problem formulations and constraints (di Tollo and Roli 2008). More recent contributions improve the search component by embedding a hybrid mechanism that either add a local search component to refine found solutions, or introduce advanced constructive procedures to identify good individuals: the resulting strategies are referred to as *memetic algorithms* (Neri et al. 2012), and have been applied to Portfolio Selection by Maringer and Winker

(2003) and Maringer (2005), that enhance Evolutionary Algorithms by adding Simulated Annealing (Crama and Schyns 2003) and Tabu Search (Schaerf 2002) modules in the framework of different risk measures. Moreover, several studies propose to hybridize evolutionary algorithms with machine learning techniques (Chen et al. 2020; Yaman and Dalkılıç 2021), whereas Salehpoor and Molla-Alizadeh-Zavardehi (2019) discuss a strategy which combines genetic algorithms with electromagnetism-like algorithms, Particle Swarm optimization, genetic network programming and Simulated Annealing. Finally, Kalayci et al. (2020) employ a hybrid approach based on continuous ant colony optimization, genetic algorithms and artificial bee colony for solving a cardinality constrained PSP. One of the main advantages of Evolutionary Computation lies in its strength to tackle multi-objective formulations, as shown by the applications of *Multi Objective Evolutionary Algorithms* (MOEAs), see, e.g. Streichert et al. (2004a), Streichert et al. (2004b), Streichert et al. (2004c) and Lin et al. (2001), Diosan (2005) in the context of multi-objective portfolio selection problems; also single-objective formulations have been devised (Chang et al. 2000; Xia et al. 2000; di Tollo and Roli 2008; Wang et al. 2006), even though it has been reported that single-objective formulations are less apt to embed users' preferences (Subbu et al. 2005).

All the above mentioned contributions take into account a single variation operator to perform the recombination of individuals. Although it has been the current practice for years, this could lead to underperformance in terms of the optimization results: this is due to the *Exploration vs Exploitation* (EvE) balance. In a nutshell, one may describe the behaviour of an Evolutionary Algorithm by using the concepts of *exploration* and *exploitation*: the former, also referred to as *diversification*, ensures that the algorithm is able to explore regions of the search space that are far from each other; the latter, which is more broadly referred to as *intensification*, identifies the ability of the algorithm to find good solutions for the problem at hand. The balance between exploration and exploitation represents an open research topic (Maturana et al. 2009, 2010; Črepinšek et al. 2013; Huang et al. 2020) and a key issue of the overall EAs' performances: this balance may be monitored and efficiently managed during the algorithm execution via the control of specified parameters, and in our case these are represented by the application rates of the different crossover operators available for the problem at hand.

Based on these considerations, in our contribution we want:

1. To give an overview of all crossover operators reported into the literature about Evolutionary approaches for Portfolio Selection;
2. To introduce a control procedure that uses all the mentioned operators during the algorithm run;
3. To perform an experimental analysis over different Portfolio Selection formulations.

As for the control procedure, we are exploiting the adaptive controller proposed by (di Tollo et al. 2015), that identifies the operator to be applied at each iteration on the basis of the past performances w.r.t. the criteria used to assess the

population's quality (average fitness) and diversity (entropy). In this way it is possible to devise adaptive strategies that select the operator needed in order to maintain the desired EvE balance, on the basis of an external parameter. Our goal is to show that the application of this control procedure leads to better (and/or more robust) solutions than the ones found by single-operator Evolutionary Algorithms.

The remainder of the paper is organized as follows. In Sect. 2 we present a literature review of evolutionary methods for portfolio management. In Sect. 3 we review the literature on crossover operators, with special focus on portfolio selection applications. In Sect. 4 we introduce a parameter control procedure and in Sect. 5 we perform a computational analysis over different formulations of the portfolio selection problem. Section 6 concludes.

2 Evolutionary algorithms for portfolio selection problems

Evolutionary algorithms are *bio-inspired* and *population-based* metaheuristics, which manage an evolving population of candidate solutions¹: a population is generated according to a user-defined random distribution, and modified by performing iteratively the following three steps:

1. Evaluate each individual according to a fitness function;
2. Identify the fittest individuals in order to converge to a (near)-optimal solution;
3. Modify the population through replacement of some individuals (selected w.r.t. a user-defined criterion) with new ones generated by applying crossover operators.

In the context of Portfolio Selection Problems, EAs have been applied to several extensions of the basic Markowitz model (Kolm et al. 2014; Gunjan and Bhattacharyya 2022) to obtain realistic results and/or to take into account real-world constraints. In what follows, we review the main literature about the applications of evolutionary algorithms to portfolio selection, before outlining some of the main components of the algorithm in the next sections.

The papers of Arnone et al. (1993) and Loraschi et al. (1995) are pioneering contributions in the field of evolutionary algorithms for Portfolio Selection Problems: they introduce basic approaches in order to determine the (approximated) constrained frontier of efficient portfolios, from a mean-semivariance point of view: they propose to exploit a genetic approach with a problem-specific crossover operator to tackle down-side risk measures, showing promising results.

Chang et al. (2000) propose to extend the standard mean-variance portfolio optimization model to take into account cardinality constraints and to impose

¹ Metaheuristics may be partitioned in trajectory-based and population-based ones: in the former case, the search can be represented as a trajectory in the search space; in the latter, the search can be seen as an evolution of a discrete set of points (solutions). Also hybrid approaches that combine the two paradigms exist: please refer to Blum and Roli (2003) for more details.

a constraint on the proportion of each asset held in the portfolio. They address a NP-hard multi-objective mixed-integer quadratic programming (MIQP) problem (Moral-Escudero et al. 2006), by using a single-objective formulation in which the minimum return constraint and the risk aversion coefficient λ are incorporated into the objective function. A similar approach is devised by Xia et al. (2000), in which the authors propose to model the PSP with an order of the expected returns of securities and with transaction costs. Also Chang et al. (2009) examine a genetic algorithm for single-objective problems with standard evolutionary features: they compare all the Pareto frontiers generated from portfolios with different number of assets, showing that low cardinality portfolios tend to outperform more complex portfolios with a larger number of assets.

The contribution of Lin and Liu (2008) is arguably the first one that proposes to solve the classic mean-variance quadratic program with minimum transaction lots constraint through evolutionary algorithms. The key point of this contribution is to consider a mixed-integer linear program in which the amount invested in each asset must be a multiple of the minimum amount that can be purchased of that security (roundlot).

Huang (2007, 2008) propose to overcome the limitations of mean-variance models by taking both the asymmetry and the uncertainty of stock returns into account, by using fuzzy logic (Zadeh 1965) to model the portfolio's returns, allowing the user to refine the Markowitz model, by relaxing the usual hypothesis of normally distributed returns through the assumption that returns are only partially known, i.e. the mean μ and the standard deviation σ of the distribution are fuzzy. These contributions introduce hybrid algorithms combining genetic algorithms and random fuzzy simulation to provide a general solution method. More recently, several studies (Kar et al. 2019; Yu et al. 2020) contribute to the literature by introducing new fuzzy hybrid methodologies, which combine genetic algorithms with machine learning techniques, in order to deal with multi-objective and multi-period portfolio problems, which take also into account higher-order moments.

Soleimani et al. (2009) extend the mean-variance model with the integer constraints proposed by Chang et al. (2000), and include a market capitalization constraint: the possible market sectors² are sorted in decreasing order with respect to their market capitalization, and the total weight of assets in the portfolio belonging to a given sector must be consistent with its market cap ranking. This constraint is formulated as follows:

$$\sum_{i \in |j|} x_i + (1 - \delta_j) \geq \sum_{i \in |j+1|} x_i \quad j = 1, 2, \dots, s - 1 \quad (1)$$

where x_i is the weight of security i in the portfolio and δ_j is a binary variable, which is equal to 1 if the asset is included in sector j and 0 otherwise.

² A stock market sector is a group of public companies that share similar business activities, outputs or characteristics. The U.S. stock market is partitioned into eleven sectors, according to the Global Industry Classification Standard (GICS).

Krink and Paterlini (2011) introduce a methodology based on Differential Evolution (Storn and Price 1997), which they call Differential Evolution for Multiobjective Portfolio Optimization (DEMPO): they propose some restrictions on turnover, by imposing a set of constraints involving the changes to the asset allocation (say, at time t) deriving from portfolio re-balance. For instance, they impose that the variation of the proportion of each asset must be greater than a threshold Δ_i , and they constrain the total turnover ratio by imposing a maximum turnover ratio TR , as follows:

$$|x_i^{t+1} - x_i^t| \geq \Delta_i \quad \text{or} \quad |x_i^{t+1} - x_i^t| = 0 \quad (2a)$$

$$\sum_{i=1}^N |x_i^{t+1} - x_i^t| \leq TR \quad (2b)$$

Anagnostopoulos and Mamanis (2011) tackle quantile-based risk measures, namely Value-at-Risk (VaR) and Expected Shortfall (ES): they include floor and ceiling constraints, cardinality and class constraints, which are used to limit the the proportion of invested capital in assets with similar features. The constraints are handled with a problem-specific solution representation based on real-encoding, and infeasible solutions are managed with a novel repair mechanism, which presents some similarities with the approach outlined in Chang et al. (2000).

Hochreiter (2015) suggests to exploit Evolutionary Algorithms in order to tackle portfolio selection problems based on risk parity (Roncalli 2013): the main idea behind this approach is to determine a portfolio whose assets are included in such a way that the risk contributions of all the securities are made equal; denoting the portfolio standard deviation with $\sigma(\mathbf{x})$, the marginal risk contribution of each asset x_i is given by:

$$\frac{\partial \sigma(\mathbf{x})}{\partial x_i} = \frac{x_i \sigma_i^2 + \sum_{i \neq j} x_j \sigma_{ij}}{\sigma(\mathbf{x})} \quad (3)$$

The general formulation of the risk parity problem (Maillard et al. 2010) can be managed via sequential quadratic programming (SQP), in which the authors propose to remove the long-only constraint and to include the floor and ceiling constraint; another approach, presented by Kaucic (2019), imposes a risk parity constraint into cardinality constrained portfolios.

Lwin et al. (2017) propose a hybrid multiobjective evolutionary algorithm for solving *mean-VaR* problems with real-world constraints, including cardinality, floor, ceiling, pre-assignment, roundlots and class constraints: a learning-guided solution generation strategy is incorporated into the optimization process in order to guide the search process towards promising regions. A slightly different perspective is discussed by Ranković et al. (2016): the authors propose to model a novel *mean-VaR* portfolio optimization problem, where VaR is estimated with a univariate *Generalized Auto-Regressive Conditional Heteroscedasticity (GARCH)* volatility model. They opt for a parametric VaR, under the hypothesis that returns follow a

standardized Student's t distribution, with d degrees of freedom, whereas the conditional portfolio volatility is estimated via univariate $GARCH(1, 1)$. The resulting multiobjective mean-VaR portfolio optimization problem with fixed asset holdings is solved by employing a *Nondominated Sorting Genetic Algorithm (NSGA II)*.

In a nutshell: a strand of the literature on portfolio optimization proposes to use evolutionary algorithms to solve problems with complex constraints and/or objective functions. All the considered contributions rely on a broad range of crossover operators, whose role in finding a good solution is discussed in Sect. 3, together with a literature review.

3 Crossover operators for portfolio selection problems

As we have discussed in the previous sections, Evolutionary Algorithms operate via the evolution of a population of individuals, which is implemented by the application of variation (crossover and mutation) operators. As pointed out by Herrera et al. (2005), the crossover operator is the main search operator which leads the search process towards different areas of the solution space, and a key point of success for EAs lies in generating offspring solutions in a neighborhood of the parents, given that the degree of proximity is typically relevant to guide the search process properly.³

Since the introduction of Genetic Algorithms, a broad range of such operators has been introduced, both general purpose operators such as one-point crossover (Holland 1992), uniform crossover (Syswerda 1993), arithmetic crossover (Michalewicz 1992), heuristic crossover (Wright 1991) and tailored to specific domains, such as fuzzy crossover (Voigt et al. 1995), unimodal crossover (Ono and Kobayashi 1999), parent centric crossover (Ballester and Carter 2004) and simulated binary crossover (Deb et al. 1995). In this section we want to review these operators, in order to detail those that will be used in our operators pool, that will be outlined in Sect. 4. Though the elements of the search space have been often represented by means of binary encoding (Goldberg 1989), we stick to the tradition of GA-based portfolio selection literature, by focusing on a real-valued representation. In what follows, we will describe the different crossover operators proposed in the literature, highlighting their behaviour and their relations with the portfolio selection literature.

- *Single point crossover [OPX]*

The single point crossover is the simplest crossover operator, which has been first proposed in the seminal works of Holland (1992) and Goldberg (1989). It generates an offspring by mating two parents by means of a *cutting point*, i.e. a randomly chosen position in the portfolio, by which the genetic material of each parent is split into two parts. It has been widely used in portfolio selec-

³ For practical purposes, from now on we focus our attention on genetic algorithms, hence in this section we review crossover methods specifically for GAs: nonetheless, note that our adaptive approach is flexible enough to manage a broad range of evolutionary algorithms.

tion literature, in particular by Chang et al. (2000) and Suksonghong and Boonlong (2021), who propose to solve mean-variance optimization with integer constraints through GAs, and by Lin and Liu (2008), that discuss a similar approach with minimum lots constraints. This operator works as follows. Given a population of N individuals $\{x_1, x_2, \dots, x_N\}$, each individual is characterized by n genes; let x_i denote a parent and let y_i be an offspring, then we have:

$$x_i = [x_i(1), x_i(2), \dots, x_i(N)] \tag{4}$$

$$y_i = [y_i(1), y_i(2), \dots, y_i(N)] \tag{5}$$

Consider now two parents x_a and x_b and a cutting point $k \in \mathbb{N}$ randomly chosen in the interval $[1, n - 1]$. Then, the two parents are mated to generate two offsprings, which in turn inherit the genes from their parents in a random fashion, as follows:

$$y_1(n) = [x_a(1), x_a(2), \dots, x_a(k), x_b(k + 1), \dots, x_b(N)] \tag{6}$$

$$y_2(n) = [x_b(1), x_b(2), \dots, x_b(k), x_a(k + 1), \dots, x_a(N)] \tag{7}$$

- *Multiple point crossover [MPX]*

The multiple point crossover (Goldberg 1989) is an extension of the single point crossover, in which the procedure includes more than one cutting point. In the context of portfolio selection problems, Skolpadungket et al. (2007) apply a three-point crossover to multiobjective portfolio optimization problems, while Hochreiter (2008) introduces an evolutionary approach based on stochastic programming, with the aim of solving multi-period portfolio selection problems by using a wide range of GAs, including a version with N-point crossover.

This method works as follows. Consider, for instance, a three point crossover; three numbers are randomly drawn, therefore three different cutting point are chosen (say k_1, k_2, k_3). The cutting points are drawn one after another, so that that the inequality $k_1 < k_2 < k_3$ is verified and the cutting points are defined over the $[1, n - 1]$ interval. Hence, an offspring inherits the first portion of genes (assets for PSP) $[1, k_1]$ from parent x_a , the second portion $[k_1, k_2]$ from parent x_b , the third one $[k_2, k_3]$ from parent x_a and finally the last portion $[k_3, N - 1]$ from parent x_b , as follows:

$$y_i = [x_a(1), \dots, x_a(k_1), x_b(k_1 + 1), \dots, x_b(k_2), x_a(k_2 + 1), \dots, x_a(k_3), x_b(k_3 + 1), \dots, x_b(N)] \tag{8}$$

The remaining values in the solution vectors, i.e. the first and the third portion of genes of parent x_a , the second and the last portion of genes of parent x_b may be allocated to a second offspring.

- *Linear crossover [LNX]*

The linear crossover has been devised by Wright (1991) to overcome some drawbacks of the n-point crossover: to our knowledge, there are no studies applying the linear crossover to GAs for portfolio optimization problems. The

authors propose to generate three offsprings: each k th gene in the offspring is computed as a linear combination of the k th gene in parent 1 and in parent 2.

$$\begin{cases} y_i(k) &= 0.5 \cdot x_1(k) + 0.5 \cdot x_2(k) \\ y_{i+1}(k) &= 1.5 \cdot x_2(k) - 0.5 \cdot x_2(k) \\ y_{i+2}(k) &= -0.5 \cdot x_1(k) + 1.5 \cdot x_2(k) \end{cases}$$

- *Uniform crossover [UX]*

The uniform crossover, devised by Syswerda (1993), generates an offspring by mating two parents: each k th gene is passed either from parent x_a with probability $p = 0.5$ or from parent x_b with probability $1 - p = 0.5$, for each $k \in [1, n]$ position, that is:

$$y_i(k) = \begin{cases} x_a(k) & \text{with probability } p = 0.5 \\ x_b(k) & \text{with probability } (1 - p) = 0.5 \end{cases}$$

In order to solve the optimal portfolio choice problem, the uniform crossover has been often applied in the literature: for instance, Chang et al. (2009) and Drenovak et al. (2022) apply this crossover operator to a set of non-convex PSPs. Moreover, Arnone et al. (1993), Loraschi et al. (1995) and Loraschi and Tettamanzi (1995) examine a GA with a modified version of uniform crossover. Denoting with TOT_{x_a} and TOT_{x_b} the total sums of the genes making up, respectively, parents x_a and x_b , and supposing that a gene k th from parent x_a has been chosen, the value of the k th gene in the i th offspring is:

$$y_i(k) = \min \left\{ \max \{x_a(1), \dots, x_a(k), \dots, x_a(n)\}, x_a(k) \frac{TOT_{x_a} + TOT_{x_b}}{2TOT_{x_a}} \right\} \quad (9)$$

- *Global uniform crossover [GUX]*

The global uniform, proposed by Simon (2013), is a generalization of the two-parent uniform crossover, and, up to the authors' knowledge, it has not yet found application for the portfolio selection problem. Hence, in what follows we propose to introduce this operator in the context of portfolio selection problems. The rationale of this operator is to extend the parent pool to the whole population, so that the k th gene is selected with probability $p = 1/N$ from the i th parent. This algorithm, though, chooses genes in a completely random fashion; as a consequence, one may redefine the probability of selecting the k th gene in accordance with a fitness-based criterion.

- *Queen-bee crossover [QBX]*

This method, discussed in Sung (2007), mimics the reproduction process of bees in nature; the author tests this recombination tool with a combinatorial problem and two continuous function optimization problems. We propose to introduce this crossover strategy for portfolio selection problems, as we have not found any application of this approach to GA-based portfolio optimization.

We point out that a major limitation of this operator is its fast convergence to (potentially) local solutions. The idea is the following. The *queen-bee* represents the fittest parent, which is mated, at each step, with a randomly chosen parent, which in turn is dropped at the following step and she is replaced with another parent and so on. In order to have the parents mated, a wide array of choices is available (e.g. simple crossover techniques, like uniform or multiple point crossover strategies).

- *Arithmetic and average crossover [AX]*

The arithmetic crossover, devised by Michalewicz (1992), produces two offspring by mating two parents at each iteration. In the context of portfolio selection problems, this crossover strategy has been applied by Xia et al. (2000) and Mazraeh et al. (2022) to solve the classic mean-variance problem. The offsprings are generated, gene by gene, as a weighted mean of their parents genes, displaced in the same position of the resulting offspring gene, as follows:

$$y_i(k) = \beta x_a(k) + (1 - \beta)x_b(k) \quad (10)$$

The average crossover, mentioned in Michalewicz (1992) and also discussed in Simon (2013), simply derives the k th gene in the offspring as an average of his parents genes, drawn exactly from the same position in the vector representing an individual, as follows:

$$y_i(k) = (x_a(k) + x_b(k))/2 \quad (11)$$

Note that the average crossover is a special case of the arithmetic operator, i.e. we simply impose $\beta = 0.5$.

- *Simplex crossover [SPX]*

The design of the simplex crossover has been proposed by Renders and Bersini (1994) and it can be viewed as a generalization of the arithmetic crossover (Michalewicz and Schoenauer 1996). Up to the authors' knowledge, no application of simplex crossover to portfolio optimization models exists in literature, hence we propose to test its behaviour for PSPs. The operator identifies a groups G of possible parents, and selects $k > 2$ parents; then it determines the best and the worst individual (in terms of fitness) within G , and computes its centroid c , by removing the worst individual x_i^w from it, as follows:

$$c = \sum_{x_i \in G - x_i^w} x_i / (k - 1) \quad (12)$$

Then the 'reflected point' is computed:

$$x_r = c + (c - x_i^w) \quad (13)$$

If x_r is better (in terms of fitness) than the best selected individual x_{best} , an expanded point x_e is determined:

$$x_e = x_r + (x_r - c) \quad (14)$$

If x_e is better than x_r , the offspring is x_e , else the offspring is x_r ; otherwise if x_r is not better than x_{best} but it is better than x_w , then the offspring is x_r .

- *Geometrical crossover [GX]*

With this operator, an offspring can be generated in this way:

$$y_i(k) = \sqrt{x_a(k) \cdot x_b(k)} \quad \text{for each } k\text{-th gene} \quad (15)$$

The geometrical crossover has been first proposed by Michalewicz et al. (1996), in which an offspring can be generated in this way:

$$y_i(k) = \sqrt{x_a(k) \cdot x_b(k)} \quad \text{for each } k\text{-th gene} \quad (16)$$

where y_i denotes an offspring, x_a and x_b denote a given pair of parents. The mating process is repeated until a new population of N individuals is generated. Some simple tests are carried out to compare its performance to standard crossover operators, and though the authors have shown its effectiveness in the context of constrained problems, we have not found any application of this crossover strategy in the context of portfolio optimization, hence we propose to test its performance on a variety of portfolio selection problems.

- *Direction-based crossover [DBX]*

The direction-based crossover, devised by Arumugam et al. (2005), is a variant of the linear crossover in the sense that the population is biased towards fitter individuals, i.e. problem-specific information is introduced in the search process.

The direction-based crossover uses the fitness function information to determine the direction of the search in the following way. First, we denote with $f(\cdot)$ a generic fitness function and with r a random number such that $r \in U[0, 1]$. For two given parents x_a and x_b , assuming that $f(x_b) < f(x_a)$, the operator generates an offspring y_i according to the following rule:

$$\begin{cases} y_1 = r(x_b - x_a) + x_b & \text{if } f(x_b) < f(x_a) \\ y_1 = r(x_a - x_b) + x_a & \text{otherwise} \end{cases}$$

- *Heuristic crossover [HX]*

The heuristic crossover, devised by Wright (1991), is a more sophisticated variant of the direction-based crossover: to our knowledge, Yusuf et al. (2019) is the only application of this version of the heuristic crossover in the context of Portfolio Selection Problems. It works as follows: the parents are chosen in the mating pool and their fitness, before running the usual procedure of gene transmission, is compared. Afterwards, a random number r is generated from a uniform distribution $U[0, 1]$ and the genes are chosen according to the rule given in Eq. 17. Note that the resulting population is expected to display less diversity and a better mean vector after the crossover stage. For maximization problems we have:

$$y_i(k) = x_b(k) + r(x_a(k) - x_b(k)) \quad \text{if } f(x_b) > f(x_a) \quad (17)$$

- *Flat crossover [FX]*

The flat crossover, proposed by Radcliffe (1991), generates the offsprings from a uniform distribution, tweaking the classical crossover techniques in order to improve the search process in terms of exploration, since it extends the search process to every possible value contained in a uniform distribution bounded as follows, for a given k th gene of a generic offspring y_i :

$$y_i(k) \sim U[\min(x_a(k), x_b(k)), \max(x_a(k), x_b(k))] \quad (18)$$

Still, note that the offspring is generated from a uniform distribution bounded by the parents genes, which actually is not that beneficial for exploration purposes, according to the framework sketched in Herrera et al. (2005). Some applications of a variant of flat crossover to portfolio selection problems are discussed in the following item.

- *Blend crossover [BLX]*

The blend crossover, proposed by Eshelman and Schaffer (1993), deals with the limited search capabilities of the flat crossover by adding a parameter α whose aim is to widen or to shrink the search domain. Hochreiter (2008) uses a GA with blend crossover to solve multi-period portfolio selection problems, with a wide range of structurally different risk measures applied for empirical simulations. Furthermore, Streichert et al. (2004b) investigate the impact of different crossover operators for a real-valued GA on a mean-variance cardinality constrained portfolio optimization problem, showing that blend crossover tends to display better performance compared to standard 1-point and N-point crossover approaches. More recently, Mazraeh et al. (2022) devise a combined model based on Grey Wolf optimizer and machine learning preselection methods, in which the blend crossover is used.

Given two parents x_a and x_b , then the k th gene of an offspring is drawn from a random uniform distribution, as follows:

$$x_{min}(k) = \min(x_a(k), x_b(k)), \quad x_{max}(k) = \max(x_a(k), x_b(k)) \quad (19)$$

Then we have:

$$y_i(k) \sim U[x_{min}(k) - \alpha\Delta_x, x_{max}(k) + \alpha\Delta_x] \quad (20)$$

with $\Delta_x = x_{max} - x_{min}$. As highlighted by Herrera et al. (2005), a negative alpha encourages exploitation, whereas a positive alpha steers the crossover operator towards exploration; note that the blend crossover is a simple generalization of the flat one: indeed, for $\alpha = 0$, they are equivalent.

- *Simulated binary crossover and fuzzy recombination [SBX] and [FR]*

The simulated binary crossover, proposed by Deb et al. (1995), generates two vectors of offsprings from a probability distribution, which is dependent on the location of their parents: in other terms, this crossover is self-adaptive in the sense that the spread of the possible offspring solutions depends on the distance between the parents, which decreases as the population converge. Fuzzy recombination is based on a similar hypothesis and it can be derived from the simulated binary crossover, so we discuss their properties jointly: the difference between the two operators stays in that [FR] is based on a triangular probability

density, whilst the [SBX] design is based on a specific shape of the distribution, as we discuss below. Mehlawat et al. (2021) propose to use the simulated binary crossover to solve a mean-absolute semideviation problem, where the resultant multiobjective credibility model is solved with a real-coded genetic algorithm.

The simulated binary crossover is defined as:

$$y_1 = 0.5[(1 - \beta)x_a + (1 + \beta)x_b] \quad (21)$$

$$y_2 = 0.5[(1 + \beta)x_a + (1 - \beta)x_b] \quad (22)$$

x_a and x_b are independent samples from the population of parents and β is a sample from a random number generator with density:

$$p(\beta) = \begin{cases} 0.5(n + 1)\beta^n & \beta \leq 1 \\ 0.5(n + 1)\frac{1}{\beta^{n+2}} & \beta > 1 \end{cases}$$

This distribution can be obtained by sampling from a random uniform $U[0, 1]$ and then by the following transformation:

$$\beta = \begin{cases} (2u)^{\frac{1}{n+1}} & \text{if } U(0, 1) \leq 0.5 \\ [2(1 - u)]^{\frac{-1}{n+1}} & \text{otherwise} \end{cases}$$

The fuzzy recombination, proposed by Voigt et al. (1995), is very similar to the simulated binary crossover. The density of β has its maximum at $\beta = 1$. while the only difference between them is the shape of the probability density $p(\beta)$, which is triangular for the fuzzy crossover:

$$p_{\Delta}(\zeta) = \begin{cases} \zeta + 1 & \text{if } -1 \leq \zeta < 1 \\ 1 - \zeta & \text{if } -0 \leq \zeta \leq 1 \end{cases}$$

The β value in Eqs. 21 and 22 is then obtained as follows:

$$\beta = 1 + 2\zeta d \quad (23)$$

As Beyer and Deb (2001) suggest, d is a ‘strategy parameter’, which determines how far the offspring should be located from parents. The random number ζ with triangular distribution can be obtained as the sum of two independent numbers $U[0, 1]$:

$$\zeta = u_1 + u_2 - 1 \quad (24)$$

- *Laplace crossover [LX]*

The Laplace crossover, proposed by Deep and Thakur (2007), belongs to the family of parent-centric crossover operators, sharing many characteristics with the simulated binary crossover. Mittal and Srivastava (2021) introduce a refined version of the Laplace operator, which they call bounded exponential crossover [BEX], since it is based on a double exponential distribution. The authors show its superior performance in a set of numerical experiments,

within the context of mean-variance-skewness portfolio optimization under uncertain environment (Liu 2010).

Let us now comment briefly this approach. The density function of the Laplace distribution is given by:

$$f(x) = \frac{1}{2b} \exp\left(-\frac{|x-a|}{b}\right) \quad -\infty < x < \infty \quad (25)$$

where $a \in \mathbb{R}$ is the location parameter and $b > 0$ is the scale parameter. For $b = 0.5$ the probability of creating offsprings near the parents is higher and for $b=1$ distant points are likely to be selected as offsprings. Moreover, the authors proceed by drawing a random number $u \in \mathbb{R}$ from $U[0, 1]$; then a random number β is generated from the Laplace distribution. This can be obtained by inverting it:

$$\beta = \begin{cases} a - b \log(u) & u \leq \frac{1}{2} \\ 2ma + b \log(u) & u > \frac{1}{2} \end{cases}$$

Then, the offsprings are generated as follows:

$$y_1 = x_a + \beta |x_a - x_b| \quad (26)$$

$$y_2 = x_b + \beta |x_a - x_b| \quad (27)$$

- *Parent-centric normal crossover [PNX]*

The parent-centric crossover, devised by Ballester and Carter (2004), seeks to improve the simulated binary crossover by generating offsprings in regions of the search space neglected by SBX; in particular, it is a parent-centric and self-adaptive operator, i.e. the spread of offspring solutions depends on the distance between parents, which decreases as the population tends to converge to a solution. To our knowledge, this approach has never been discussed in the context of portfolio selection problems, hence we propose to introduce it and to evaluate its performance.

$$y_1(j) \sim N(x_a(j), |x_b(j) - x_a(j)|/\eta) \quad (28)$$

$$y_2(j) \sim N(x_b(j), |x_b(j) - x_a(j)|/\eta) \quad (29)$$

where $N(\mu, \sigma^2)$ is a normal random number, η is a tunable parameter. With j we denote as usual the j th component of parent a or b .

- *Unimodal normal distribution crossover [UNDX]*

The unimodal normal distribution crossover, devised by Ono and Kobayashi (1999), belongs to the family of mean-centric operators and it uses an ellipsoidal probability distribution to generate offsprings. We propose to introduce

this crossover strategy for portfolio selection problems, as we have not found any application of this approach to GA-based portfolio optimization.

Fundamentally, $(\mu - 1)$ parents are randomly chosen (say, $\mu = 3$) and their midpoint x_p is computed; thereafter, the difference vector (primary search direction) as $d = x^2 - x^1$. Furthermore a third parent x^3 is picked up randomly, and let D denote the distance between x^3 and the line connecting x^1 and x^2 :

$$D = |x^3 - x^1| \times \left(1 - \left(\frac{(x^3 - x^1)^T (x^2 - x^1)}{|x^2 - x^1| |x^3 - x^1|} \right)^2 \right)^{1/2} \quad (30)$$

An offspring is generated by the equation:

$$y_1 = x_p + \xi d + \sum_{i=1}^{n-1} \eta_i e_i D \quad (31)$$

where ξ is a random number following a random distribution $N(0, \sigma_\xi^2)$, η_i are $n - 1$ random numbers sampled from a normal distribution $N(0, \sigma_\eta^2)$ and e_i an orthogonal basis vector. Ono and Kobayashi (1999) suggest to use $\sigma_\xi^2 = 0.25$ and $\sigma_\eta^2 = (0.35)^2/n$ and $\mu = 3$ to 7.

4 A controller for evolutionary algorithms

In this section we will outline the structure of the adaptive *controller* that is to be connected to a basic Evolutionary Algorithm in order to perform Adaptive Operator Selection (AOS), i.e., to choose at each iteration what operator to be used, out of the ones detailed in Sect. 3, depending on the criteria to be defined by the user (or by the adaptive strategy).

As we have seen into the Introduction, the EvE (*exploration* versus *exploitation*) balance is a key point in the Evolutionary Algorithm design, and in our contribution we aim to address this trade-off in a dynamic way: given the set of operators detailed in Sect. 3, the adaptive controller chooses what operator to be applied on the basis of the past operators' performances with regard to some user-defined criteria: it is well known that some operators are designed to foster exploitation, and other to foster exploitation (Eiben and Smith 2015), but for most operators, their effect on these two criteria depends on the current state of the search. Our controller assigns a reward to each operator after each iteration, and may adopt a deterministic (pre-defined) or an adaptive strategy in order to manage the EvE trade-off over the time.

Our controller is based on Reinforcement Learning, and its main components are iterated in a *state-action-reward* sequence; it defines an input/output interface with the EA (solver) in which the EA itself sends the last applied operator identifier and its performance metrics w.r.t. the desired criteria, and selects the operator to be applied next on the basis of the following modules: *Aggregated Criteria Computation*, *Reward Computation*, *Credit Assignment*, *Operator Selection* (Maturana et al.

2010; di Tollo et al. 2015). These modules are executed sequentially and are detailed in what follows:

- *Aggregated Criteria Computation* this module stores the impact of successive applications of a crossover operator on the desired criteria: in our contribution, these criteria are the variation of population quality and diversity, and are computed as the variation of the value of population's *entropy* (ΔD) and *mean fitness* (ΔQ). These values are stored in a sliding window W_{ij} of size T , where i is the operator and j the criterion taken into account. A function $F(\cdot)$ computes the final impact, which could be instantiated to *max* (to detect outliers) or *avg* if one wants to register a smooth behaviour (di Tollo et al. 2015). The inputs of this module are the identifier of the last applied operator and the observed variation of each criteria. The output, which is sent to the *Reward Computation* module, is a vector containing a scalar value for each criterion; for our experiments, $F(\cdot)$ will be instantiated with *avg*(\cdot); preliminary experiments showed that the behaviour of the AOS is rather robust with regards to the instantiation of $F(\cdot)$. This module is described in Algorithm 1.

Algorithm 1 Aggregated Criteria Computation

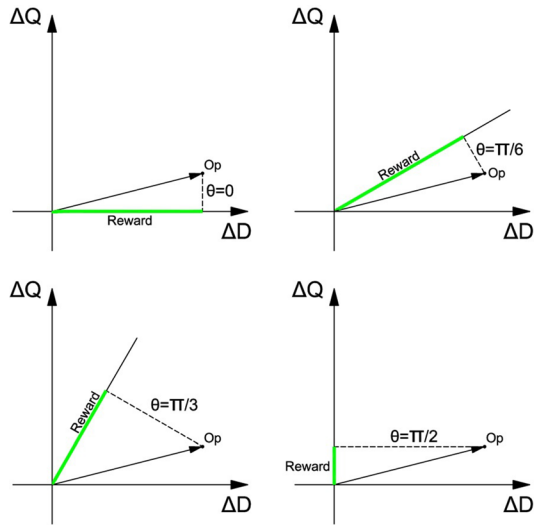
```

1: Input
2:    $op_i$  Operator  $i$ 
3:    $\Delta D$  Observed variation of diversity criterion
4:    $\Delta Q$  Observed variation of quality criterion
5: Output
6:    $op_i$ 
7:    $F(W_{i\Delta D}, T)$ 
8:    $F(W_{i\Delta Q}, T)$ 
9: if  $type = extreme$  then
10:   $F(W_{i\Delta D}, T) \leftarrow max(W_{i\Delta D}, T)$ 
11:   $F(W_{i\Delta Q}, T) \leftarrow max(W_{i\Delta Q}, T)$ 
12: else
13:   $F(W_{i\Delta D}, T) \leftarrow avg(W_{i\Delta D}, T)$ 
14:   $F(W_{i\Delta Q}, T) \leftarrow avg(W_{i\Delta Q}, T)$ 
15: end if
16: if  $normalize = True$  then
17:   $norm\_func \leftarrow \max_{i=1, \dots, K} F(\cdot)_{i=1, \dots, K}$ 
18:   $F(\cdot) \leftarrow F(\cdot) / norm\_func$ 
19: end if

```

- *Reward Computation* the reward strategy computes a reward for the operator identified by the previous module, and can be implemented by using several functions (Fialho 2010). In our case we have chosen the *Compass*, method, proposed by Maturana and Saubion (2008), that employs the input from the previous module (a vector) and computes the reward on the basis of an hyperparameter $\theta \in [0, \pi/2]$, that identifies a search angle in the $\Delta D/\Delta Q$ plane. The value of θ can be set a priori or computed through a dynamic strategy. Via this

Fig. 1 Reward computation based on a compass method, adapted from Maturana et al. (2010) and di Tollo et al. (2015)



module, each operator’s performances are represented in the $\Delta D/\Delta Q$ plane, according to the aggregated impact vector previously computed and associated to an additional vector, that will be used to store the rewards. The main idea is to attribute a reward to each operator by computing the distance between the performance point and the plane defined by θ , as shown in Fig. 1: in this way, $\theta = 0$ indicates a scenario that fosters diversity and neglects quality, whilst $\theta = \pi/2$ indicates a scenario in which quality is fostered. Intermediate values indicate a weighted compromise between these two criteria. Algorithm 2 outlines this behaviour.

Algorithm 2 Reward Computation (Compass)

- 1: **Input**
 - 2: op_i
 - 3: $F(W_i\Delta D, T)$
 - 4: $F(W_i\Delta Q, T)$
 - 5: **Output**
 - 6: R_{op_i} Reward of operator op_i
 - 7: $D_{op_i} \leftarrow avg(diversity_{op_i})$
 - 8: $Q_{op_i} \leftarrow avg(quality_{op_i})$
 - 9: $V_{op_i} \leftarrow (D_{op_i}, Q_{op_i})$
 - 10: $\alpha_{op_i} \leftarrow |atan(\frac{Q_{op_i}}{D_{op_i}}) - \theta|$
 - 11: $R_{op_i} \leftarrow |V_{op_i} \cdot cos(\alpha_{op_i}) - min_{i=1, \dots, K}(|V_i| \cdot cos(\alpha_i))|$
-

- **Credit Assignment** the credit amount summarizes the reward obtained by an operator during recent applications: it takes as input the reward of each operator op_i at time t , which is then is stored into a sliding window of size T' . Then, the module computes an aggregate reward by specifying an aggregation function

$F(\cdot)$ (which could be instantiated to $\max(\cdot)$ or $\text{avg}(\cdot)$ over the period T'). These aggregated values represent the credit, i.e., the output of the module sent to *Operator Selection*.

Algorithm 3 Credit Assignment

```

1: Input
2:    $R_{op_i}$  Reward of operator  $op_i$ 
3: Output
4:    $\mathcal{U}_i$  Credit of operator  $op_i$ 
5: if  $type = extreme$  then
6:    $F'(W'_i, T', R_i) \leftarrow \max(W'_i, T', R_i)$ 
7: else
8:    $F'(W'_i, T', R_i) \leftarrow \text{avg}(W'_i, T', R_i)$ 
9: end if
10: if  $normalize = True$  then
11:    $norm\_func \leftarrow \max_{i=1, \dots, K} F'(\cdot)_{i=1, \dots, K}$ 
12:    $F(\cdot) \leftarrow F'(\cdot) / norm\_func$ 
13: end if
14:  $\mathcal{U}_i \leftarrow F'(W'_i, T', R_i)$ 

```

- *Operator Selection* this module receives as input the credits of all operators and determines the next crossover operator to be applied by the solver, according to a predefined selection scheme. The input of the operator selection module is the credit estimate \mathcal{U}_i , while the output is the identifier op_{next} , which is sent to the EA. In our experiments we have performed operator selection by using Probability Matching (PM), as outlined by Algorithm 4; for a summary of other selection methods, we forward the reader to Maturana et al. (2009), Lardeux et al. (2006), and Maturana et al. (2010).

Algorithm 4 Probability Matching (PM)

```

1: Input
2:    $\mathcal{T}$  Termination criterion
3:    $\mathcal{A}$  Set of  $K$  operators
4:    $P_{min}$  Minimum selection probability for each  $K$ 
5:    $\alpha$  Adaptation rate
6: Output
7:    $\mathcal{P}_s$  Probability vector
8: for  $i \leftarrow 1 : K$  do
9:   % initial probability vector
10:  % initial credit vector
11:   $\mathcal{P}[i] \leftarrow 1/K$ 
12:   $\mathcal{U}[i] \leftarrow 1.0$ 
13: end for
14: while termination criterion  $\mathcal{T}$  not met do
15:    $s \leftarrow \text{Select\_Operator}(\mathcal{P})$ 
16:    $R_s[t] \leftarrow \text{Get\_Reward}(s)$ 
17:    $\mathcal{U}_s[t+1] \leftarrow \mathcal{U}_s[t] + \alpha(R_s[t] - \mathcal{U}_s[t])$ 
18:   %update probability vector
19:   for  $s \leftarrow 1 : K$  do
20:      $\mathcal{P}_s[t+1] \leftarrow P_{min} + (1 - K \cdot P_{min}) \frac{\mathcal{U}_s[t+1]}{\sum_{i=1}^K \mathcal{U}_i[t+1]}$ 
21:   end for
22: end while

```

In order to determine the value of θ , we have implemented the framework devised by di Tollo et al. (2015), that uses four high-level dynamic strategies:

1. *Increasing* this strategy splits the number of iterations into n epochs and increases θ value in equally spaced steps in $[0, \pi/2]$;
2. *Decreasing* this strategy splits the number of iterations into n epochs and decreases θ value in equally spaced steps in $[0, \pi/2]$;
3. *Always moving* this strategy splits the number of iterations into n epochs and alternates $\theta \in \{0, \pi/2\}$ in equally spaced steps;
4. *Reactive moving* this strategy sets $\theta = 0$ if $\frac{\Delta D(P,t)}{D(P,t-1)} < 1e - 01$, else if $\frac{\Delta Q(P,t)}{Q(P,t-1)} < 1e - 01$, set $\theta = \pi/2$, where $Q(P, t)$ and $D(P, t)$ denote respectively the mean fitness and the entropy of population P at time t . Otherwise, if neither conditions are satisfied, $\theta = 0$.

Note that strategies 1), 2), and 3) define a deterministic schedule, whereas *reactive moving* is adaptive: the schedule induces an increase in population entropy when the search procedure is stagnating in a local minimum, by selecting exploration operators for the next operations. These strategies will be used in our computational experiments in Sect. 5.

5 Computational experiments

In this section we describe the computational experiments aimed to optimize the portfolios with regards to specific risk measures, and to evaluate the performance of the crossover operators detailed in Sect. 2. After detailing our experimental setting in Sect. 5.1, we will evaluate the behaviour of each crossover operator in Sect. 5.2, in order to understand the specific features of each crossover operator (if any) and to assess whether each operator is more inclined towards exploration or exploitation; then we perform experiments by pooling all crossover operators together in Sect. 5.3, in order to assess the benefits of Adaptive Operator Selection (AOS), which is supposed to direct the search towards the desired direction, by selecting the optimal operator for the next iteration. Finally, in Sect. 5.4 we perform an out-of-sample test, in order to evaluate future performances of portfolios obtained in Sect. 5.3, and to assess eventual benefits over plain GAs equipped with a single crossover operator.

5.1 Our data and experimental settings

We have defined our formulations imposing budget, short-selling, cardinality, floor and ceiling constraints. We have tackled five sets of data, relative to daily closing prices of assets from different stock exchanges indices, over a period of 5 years from 01/01/2009 to 01/01/2014. The summary of our sets of data is reported on Table 1.

We recall that the solution of the portfolio selection problem (PSP) is given by a vector of n variables x_1, \dots, x_n , where each x_i represents a fraction of the amount invested in asset i . In what follows, we deal with the above mentioned constraints with the following general formulation of a mixed-integer portfolio selection problem, which has been proven to be *NP-Hard* (Moral-Escudero et al. 2006), for which evolutionary approaches seem particularly suitable:

$$\begin{aligned}
 & \min_{\mathbf{x}} \quad \phi(\mathbf{x}) \\
 & \text{s.t.} \quad \sum_{i=1}^n x_i = 1 \\
 & \quad x_i \geq 0 \quad \quad \quad i = 1, \dots, n \\
 & \quad \sum_{i=1}^n \delta_i = K \\
 & \quad \delta_i l_i \leq x_i \leq \delta_i u_i \quad i = 1, \dots, n \\
 & \quad \delta_i \in \{0, 1\} \quad \quad \quad i = 1, \dots, n
 \end{aligned} \tag{32}$$

where $\phi(\mathbf{x})$ is a generic objective function, $\delta_i \in \{0, 1\}$ represent n integrality constraints, namely δ_i is a binary variable to include or exclude asset i in the portfolio; moreover, with $\delta_i l_i \leq x_i \leq \delta_i u_i$ we impose n floor and ceiling constraints (respectively, l_i and u_i) and with $\sum_{i=1}^n \delta_i = K$ we denote the cardinality constraint, i.e. we

Table 1 The sets of data used in our experimental phase

ID	Instance name	Country of origin	#	Number of observations (days)
01	Nikkei 225	Japan	210	1302
02	FTSE 100	United Kingdom	89	1306
03	Hang Seng	Hong Kong	42	1306
04	CAC 40	France	36	1306
05	FTSE MIB	Italy	32	1306

indicates the cardinality (i.e., the number of assets) that compose the different indices

impose that exactly k active position are chosen from the market index. Finally with $x_i \geq 0$ we denote the no-short selling constraint and with $\sum_{i=1}^n x_i = 1$ we ensure that the whole capital is invested in the portfolio.

In what follows, we adopt the *penalty* approach of Fletcher (2013) and Corazza et al. (2013, 2021), by defining a ℓ_1 penalty function $C(\mathbf{x}, \delta, \epsilon)$:

$$\begin{aligned}
 C(\mathbf{x}, \delta, \epsilon) = & \phi(\mathbf{x}) + \frac{1}{\epsilon} \left[\left| \sum_{i=1}^N x_i - 1 \right| + \max \left\{ 0, \sum_{i=1}^N \delta_i - K \right\} + \sum_{i=1}^N \max \{ 0, \delta_i l_i - x_i \} \right. \\
 & \left. + \sum_{i=1}^N \max \{ 0, x_i - \delta_i u_i \} + \sum_{i=1}^N \left| \delta_i (1 - \delta_i) \right| \right]
 \end{aligned}
 \tag{33}$$

With ϵ denoting a penalty parameter. Finally, our ℓ_1 penalty problem is:

$$\min_{\mathbf{x} \in \mathbb{R}^N, \delta \in \mathbb{R}^N} C(\mathbf{x}, \delta, \epsilon)
 \tag{34}$$

We have defined six different sets of experiments based on optimizing six different risk measures, that are outlined in Table 2, along with the corresponding cost functions⁴.

We have set $\lambda = 0.5$ for the *Mean-Variance* risk measure. Moreover, for the *Two-sided* risk measure of Chen and Wang (2008) we use $a = 0.5$ and $p = 2$, while for *VaR* we consider a monthly probability of losses exceeding the $VaR_{(1-\beta)}$ threshold $\beta = 0.05$.

Two performance criteria have been defined to evaluate the GAs' performances: the first is the average fitness of the population, defined as $fitness(P) = \frac{\sum_{ind \in P} eval(ind)}{|P|}$, where $eval(ind)$ denotes the individual fitness; the

second is the population entropy, defined as $entropy(P) = \frac{-\sum_{i=1}^n \sum_{j=0}^k \frac{n_{ij}}{|P|} \log \frac{n_{ij}}{|P|}}{n \log 2}$,

⁴ Please notice that, in the literature about Meta-Heuristics, the term 'objective function' and 'cost function' have different meanings: the former corresponds to the target function to be optimized, whereas the latter represents the function guiding the search algorithm in the search space (di Tollo and Roli 2008).

Table 2 Cost functions used in our experimental phase

Risk measure	Cost function
Omega ratio (Keating and Shadwick 2002)	$\Omega = \frac{\int_{-\infty}^{r_d} (r_d - r)F(r)dr}{\int_{r_d}^{\infty} (r - r_d)F(r)dr}$
Mean-variance (Markowitz 1952)	$MV_{\lambda} = \lambda r_p - (1 - \lambda)\sigma_p^2$
Two-sided (Chen and Wang 2008)	$\rho_{a,p}(r) = a\ (r_p - E(r_p))^+\ _1 + (1 - a)\ (r_p - E(r_p))^{-}\ _1 - E(r_p)$
Equal risk contribution (Maillard et al. 2010)	$f_{ERC}(\mathbf{x}) = \sum_{i=1}^n \left \frac{x_i(Cx)_i}{x^T Cx} - \frac{1}{K} \right $
$VaR_{(1-\beta)}$ (Artzner et al. 1999)	$VaR_{(1-\beta)} = F^{-1}(1 - \beta)$
$CVaR_{(1-\beta)}$ (Rockafellar and Uryasev 2000)	$CVaR = VaR + \frac{1}{\beta} \int_{VaR}^{\infty} (1 - F(r))dr$

where n is the number of individuals in the population and k is the number of genes for each individual. For these experiments, we set $P = 50$, while the number of generations g is set to 1000. Finally, we consider the set of twenty crossover operators defined in Sect. 3.

All algorithms have been implemented in Python and run on a Intel Core i7 2.4 GHz processor, with 8 GB RAM. In what follows we evaluate the crossover performances, first by using each crossover as a *stand-alone* operator in Sect. 5.2; then, we evaluate our Adaptive Operator Selection in Sect. 5.3.

5.2 Single operators' results

In the following experiments, the main idea is to isolate the influence of the optimization process in the assessment of the performance of each recombination operator. In the first group of tests, the crossover operator has the goal of retaining a significant amount of diversity, reaching simultaneously a near-optimal level of fitness. In the second group of tests, instead, the behaviour of each operator is assessed when neither selection nor mutation is allowed: the population generated with recombination is sent to the following generation as it is, so that the behaviour of each crossover strategy is singled out.

As a general rule, the crossover operator should strike a balance between quality and entropy, i.e. it is supposed to offset the impact of the selection operator by retaining a reasonable amount of variance while the optimization process is ongoing. Furthermore, as noted by di Tollo et al. (2015) and Maturana et al. (2010), the correlation of the absolute value of fitness and entropy should be generally low; ideally, an operator should be able to increase entropy when the search is stuck in local optima, displaying a self-adaptive behaviour. It is well known that these two factors are strongly related (Arabas et al. 1994): a too strong selective pressure may lead to

early convergence towards a suboptimal solution, while a weak selective pressure makes the search ineffective. As noted by Maturana et al. (2010), it is difficult to reach such a balance over all phases of the search, and this is precisely the reason that led us to implement an AOS controller, in order to select the right operator w.r.t. the wished compromise between quality and diversification at each search step. This will be discussed in Sect. 5.3.

In what follows, we analyse the behaviour of different operators during the optimization phase. Further comments on the in-sample performance and a comparison between standard GAs and the adaptive GA are available in Sect. 5.3.

Figure 2 shows the convergence of the average fitness of the population and its entropy for different configurations of the genetic algorithm. Let us mention a few intriguing facts with respect to the results of the simulation. Note that operators belonging to the family of self-adaptive crossovers display some very promising features. For example, note that the [SBX] crossover (and to a certain extent, also the [LX] crossover) manage to reduce quickly the entropy, with a considerable and fast improvement observed in the average fitness. Furthermore, when improvements become harder, the operator turns to exploration. Both [SBX] and [LX] crossover are able to increase the diversity of the population and to improve its fitness at the same time, especially in the latter stages of the search.

All the operators that by construction are designed to use fitness information ([QBX], [LNX], [DBX], [SPX]) tend to give up some diversification in order to generate fitter individuals, converging steadily to an approximated solution, due to the fact that, by design, the search region is restricted to a very limited area. An efficient crossover should instead ensure a good trade-off between quality and entropy (Lardeux et al. 2006), preventing the population from getting stuck in local optima.

In Fig. 3 we report the behaviour of a subset of operators with respect to quality and diversity. Some standard recombination operators do not yield any substantial variation ([OPX], [QBX], [AX], [GX], [HX], [AVX], [BLX], [FX], [GUX]) both in fitness and entropy, others induce just slight variation ([UX], [TPX]), whereas some self-adaptive operators ([LX], [SBX], [PNX]), as shown in Fig. 3 lean towards an explorative behaviour. Finally, the [SPX] operator tends to generate fitter individuals, as by construction it exploits fitness information in the mating process.

To recap, in this section we have proposed to test a broad range of crossover strategies for portfolio selection problems: among them, we have chosen both crossovers already in use for evolutionary-based portfolio selection and crossovers for which we have not found any financial application in the literature. Among the latter operators, let us remark again that the performance of crossovers with self-adaptive features ([LX], [SBX], [PNX], [UNDX]) stands out in our investigation: by introducing effective and flexible crossovers, which cope better with different fitness landscapes in the context of portfolio selection, we address the problem of finding high-quality solutions rapidly and in a robust way. We have not found also any financial application for other non-self-adaptive operators, i.e. [HX], [GX], [QBX] and [DBX],

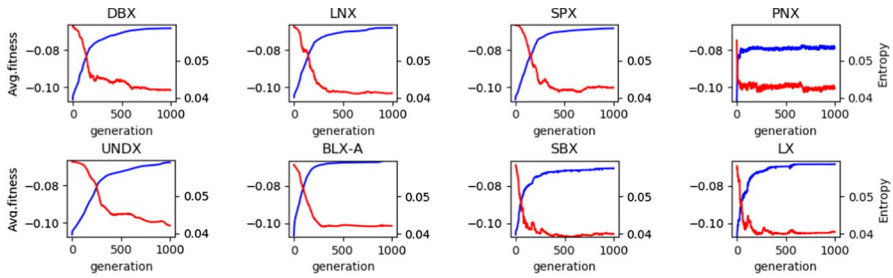


Fig. 2 Single operator performances with respect to quality and diversity. In each panel the average fitness and the entropy over the iterations is reported. The red line indicates the entropy level and the blue line indicates the fitness level. In this set of plots we report results w.r.t. a VaR-optimal portfolio, fitted on a Nikkei 225 dataset (colour figure online)

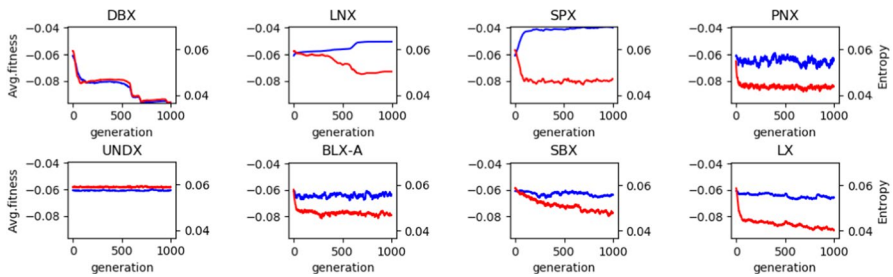


Fig. 3 Single operator performances with respect to quality and diversity. In each panel the average fitness and the entropy over the iterations is reported. The red line indicates the entropy level and the blue line indicates the fitness level. The optimization process is carried out by a GA without the selection operator. In this set of plots we report results w.r.t. a CVaR-optimal portfolio, fitted on a Nikkei 225 dataset (colour figure online)

which behave similarly to crossovers already in use in the literature, whereas the simplex crossover [SPX] is particularly efficient at finding good solutions.

5.3 Adaptive operator selection results

In this section we report the performance of the GA that resorts to AOS for operator selection, and we compare it with GAs with standard crossover operator. We also propose to evaluate the operator selection frequency for different search policies, in order to test whether the controller is able to affect indirectly fitness and entropy levels, by picking appropriate operators, which in turn should impact the exploitation or exploitation capabilities of the solver. Furthermore, we assess whether the search direction is consistent with the predefined high-level search policies described in Sect. 4. Eventually, we check the quality of the final solutions and the ability of the controller of managing efficiently an optimal quality/diversity tradeoff.

Figures 4, 5, 6 and 7 present four panels, including:

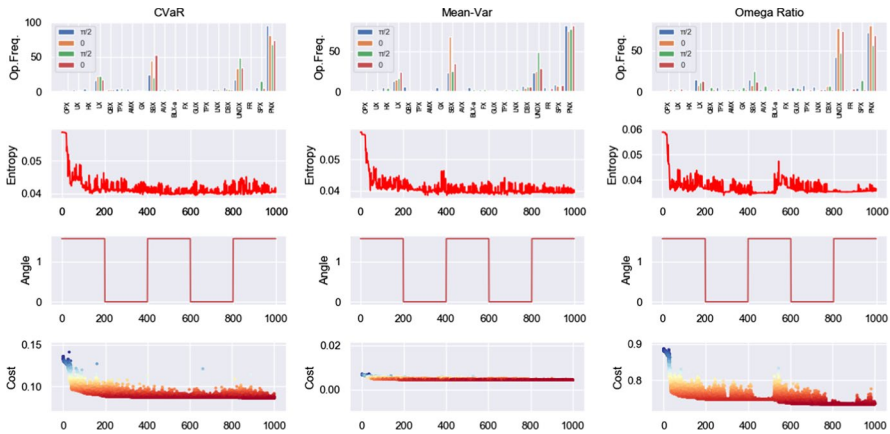


Fig. 4 Experiments with the AOS: portfolios are fitted to the Nikkei dataset. From top to bottom: histogram of the operator frequency of application (1), the entropy convergence curve (2), the always moving driving the search process (3), the individual cost scatter plot (4)

- The operators' frequency of application (top panel), needed to check whether the controller is able to discriminate operators according to different EvE tradeoffs;
- The level of population entropy over time;
- The value of the hyperparameter $\theta \in [0, \frac{\pi}{2}]$, that defines the dynamic policy guiding the search, according to the predefined (or adaptive schedule) described in Sect. 4.
- The levels of individual fitness over time (bottom panel).

As a first remark, we note that mild improvements of diversity during the search are generally triggered by a variation of the search angle in the range $[0, \pi/2]$; this also generates an immediate variation of cost for some individuals for a few generations. We also note that operators based on exploration are usually selected when small or no improvements are possible; a changing value of the angle generally leads to a non negligible impact on selection probabilities and consequently on performance. In particular, note that [LX], [SBX] and [PNX] operators provide a good trade-off of exploration and exploitation; therefore, especially at advanced stages of search, they seem to be more effective when small or no improvements are possible. Some preliminary experiments show that the probability of selection of these exploration operators turns out to be very high whenever the solver is stagnating, so that the controller turns to exploration, by rewarding them. The selection probability of all mildly relevant operators instead tends to fall to smaller values towards later stages of the process, close to the lower bound $p_{min} = 0.01$ established before running the process, with most of the selection frequency concentrated at initial or intermediate stages of search.

As a general remark, we can say that none of the introduced search policies outperforms the others: *Always Moving* and *Reactive* policy, that led to the best results in di Tollo et al. (2015), perform similarly on our problem

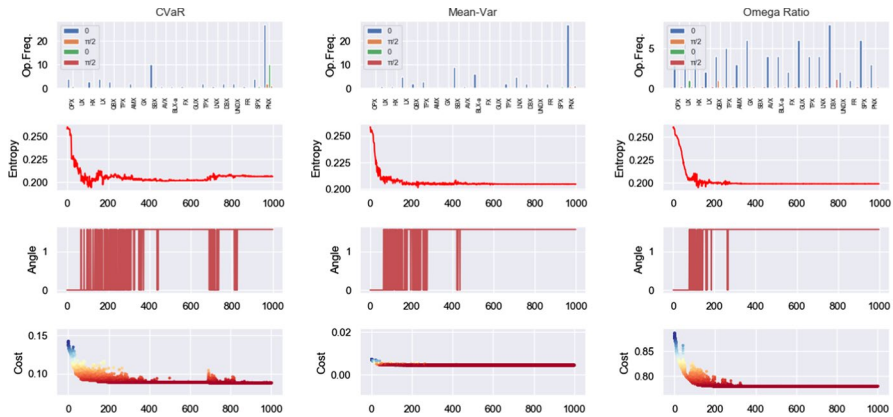


Fig. 5 Experiments with the AOS: portfolios are fitted to the CAC 40 dataset. From top to bottom: histogram of the operator frequency of application (1), the entropy convergence curve (2), the reactive policy driving the search process (3), the individual cost scatter plot (4)

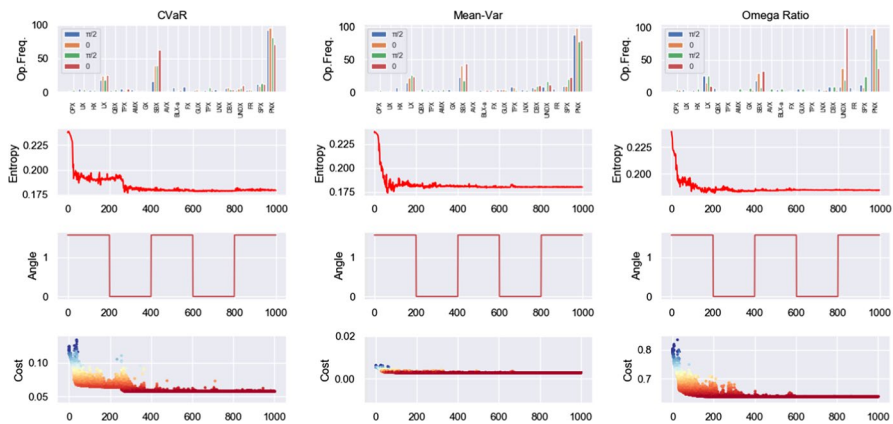


Fig. 6 Experiments with the AOS: portfolios are fitted to the Hang Seng dataset. From top to bottom: histogram of the operator frequency of application (1), the entropy convergence curve (2), the always moving policy driving the search process (3), the individual cost scatter plot (4)

instances. Anyhow, we also stress the robustness of the *Reactive* policy (Fig. 7), whose behaviour is less predictable than standard search strategies, but whose action typically succeeds in increasing the population diversity when entropy is stagnating.

In Table 3 we report the best in-sample value of the objective function obtained with a GA solver equipped with a standard operator crossover or with the controller (in bold). The solutions are reported for just one problem instance, across different objective functions.

The solver is allowed to visit infeasible solution during the search process, which comes at the price of higher penalties attributed to the cost function due

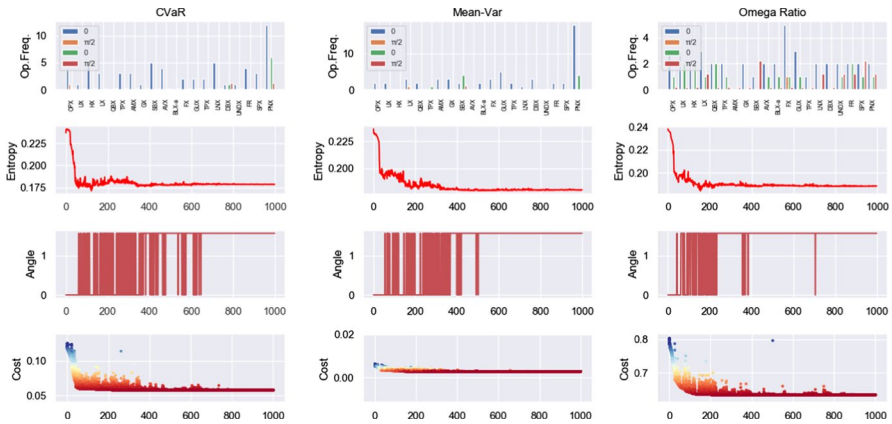


Fig. 7 Experiments with the AOS: portfolios are fitted to the Hang Seng dataset. From top to bottom: histogram of the operator frequency of application (1), the entropy convergence curve (2), the reactive moving policy driving the search process (3), the individual cost scatter plot (4)

to constraint violation. We remark that the Adaptive Operator Selection strategy has proven to be an effective approach when dealing with a variety of optimization problems. The GA equipped with AOS shows results that outperform the majority of the single-crossover based runs. Some operators still perform better than our AOS, but this is due to the computational time required to *learn*, which has to be taken into account when devising an adaptive procedure: our goal is to devise a tool that shows good performances without a-priori knowledge of the used crossovers and of the instance at hand, and to this extent, our approach is satisfactory.

Indeed, as we show in Table 3, the adaptive strategy is regularly among the top performers, whereas the results from single crossover-based genetic algorithms are less homogeneous across different risk measures, with some performing very poorly in a few cases (see in particular the ‘ERC’ column); note also that the genetic algorithm equipped with the controller is the best performing strategy for CVaR portfolio construction. Let us also mention some exceptions: the [SPX]-based genetic algorithm provide solutions that are far from optimal, whereas the quality of solutions generated by genetic algorithms equipped with [BLX], [HX] and [GUX] crossovers are generally above average. Altogether, we find that no operator can systematically outperform the others: when taking into account single-crossover genetic algorithms, the distinction between exploration and exploitation crossovers in terms of performance is less apparent.

5.4 Out-of-sample performance of rebalanced portfolios

In this section we evaluate the out-of-sample performance of the adaptive policy by computing four performance criteria, namely the *annualized portfolio returns*, the

Table 3 Best in-sample solutions

	Ω	$CVaR_{95\%}$	$VaR_{95\%}$	$MV_{0.5}$	$\rho_{0.25,2}$	ERC
OPX	0.735327	0.085345	0.068228	0.004377	2.178992	0.118195
UX	0.730871	0.084446	0.066588	0.004345	2.155495	0.068394
HX	0.731128	0.085835	0.067280	0.004389	2.176885	0.092807
LX	0.748324	0.088898	0.071191	0.004534	2.282313	0.054738
QBX	0.748456	0.085283	0.067912	0.004414	2.217761	0.091158
TPX	0.732843	0.084522	0.066731	0.004413	2.160025	0.117542
AX	0.733744	0.086807	0.068348	0.004492	2.198914	0.034641
GX	0.733477	0.085961	0.067455	0.004483	2.200084	0.078996
SBX	0.736055	0.084651	0.067663	0.004428	2.193037	0.050154
AVX	0.734751	0.086744	0.067783	0.004448	2.235605	0.059664
BLX	0.730661	0.084759	0.066531	0.004332	2.156919	0.404041
FX	0.734483	0.084967	0.067769	0.004394	2.179699	0.035128
GUX	0.730787	0.083840	0.066511	0.004333	2.146576	0.048182
TPX	0.735203	0.084819	0.068793	0.004420	2.187789	0.115952
LNx	0.732824	0.086128	0.067625	0.004417	2.209383	0.064444
DBX	0.763914	0.085960	0.069733	0.004455	2.203031	0.150119
UNDX	0.732809	0.084418	0.069509	0.004468	2.207336	0.049317
FR	0.742219	0.085692	0.067988	0.004391	2.207414	0.192051
SPX	0.763846	0.097303	0.081018	0.004766	2.600863	0.556919
PNX	0.735883	0.086159	0.069102	0.004343	2.140973	0.058629
Adaptive	0.733654	0.083998	0.068996	0.004330	2.144562	0.039140

The optimal value of the objective function is referred to a portfolio constructed with assets from the Nikkei 225 index. The controller is equipped with an always moving strategy, $K = 10$ and lower/upper bound constraints

annualized standard deviation, the *out-of-sample Sharpe Ratio* and the *turnover* (see the set of Eq. 35). These four metrics are accessible to a wide audience, which makes them popular among practitioners, despite some well-known shortcomings (Biglova et al. 2004). These four criteria are also extensively used in the literature (DeMiguel et al. 2009), with the purpose of getting a first impression of the out-of-sample portfolio performance.

The performances of the adaptive strategy are compared with a set of genetic algorithms, each equipped with one of the twenty crossover operators presented in Sect. 3.

The main objective of this set of experiments is to extend our investigation on the benefits of the adaptive approach to practical portfolio management issues, by evaluating its performance across different out-of-sample performance metrics, objective functions and over different sets of data: an assessment of the out-of-sample performance is therefore crucial to validate the model from an asset management point of view, but also to deal with potential concerns about overfitting, which might be caused by the model being too strictly tailored to the features of the training data and generalizing poorly to new data.

In order to do this, we implement a rolling-window backtest for each strategy, which is applied to the adaptive model and to the set of twenty genetic algorithms as well. We describe below the structure of our backtests, which largely follows the moving-window backtest proposed in Gilli et al. (2011) for portfolio selection. The model is fitted on training data at time t_1 from $t_1 - \tau$; for this set of tests, we set the window τ equal to 2 years (about 500 data points). The portfolio is held for $F = 125$ days, i.e. until $t_2 = t_1 + F$. Therefore, the portfolio is trained on new data starting from $t_2 - \tau$ until $t_2 - 1$ and held for 125 more days, i.e. $t_3 = t_2 + F$. With T we denote the total number of returns in the dataset, while with \mathcal{I} we denote the number of holding periods of each rebalanced portfolio.

$$\begin{aligned}\hat{\mu}^i &= \frac{1}{\mathcal{I}F} \sum_{j=1}^{\mathcal{I}} \sum_{k=1}^F x_{t_j}^{iF} r_{t_j+k}^i && \text{average daily out-of-sample returns} \\ \hat{\sigma}^i &= \sqrt{\frac{1}{\mathcal{I}F-1} \sum_{j=1}^{\mathcal{I}} \sum_{k=1}^F \left(x_{t_j}^{iF} r_{t_j+k}^i - \hat{\mu}^i \right)^2} && \text{daily out-of-sample standard deviation} \\ \widehat{SR}^i &= \frac{\hat{\mu}^i}{\hat{\sigma}^i} && \text{out-of-sample Sharpe Ratio} \\ \widehat{T}^i &= \frac{1}{\mathcal{I}-1} \sum_{j=1}^{\mathcal{I}-1} \sum_{k=1}^N \left(\left| x_{k,t_{j+1}}^i - x_{k,t_j}^i \right| \right) && \text{turnover}\end{aligned}\tag{35}$$

The out-of sample tests are performed for various combinations of objective functions and problem instances. The out-of-sample performance statistics for portfolios constructed with assets from Nikkei 225 are reported in Table 4; the results are shown for Omega Ratio-optimal portfolios for two different cardinality constraints. In order to test the hypothesis that the Sharpe Ratios and the variances of the competing strategies are equal (DeMiguel et al. 2009) (that is $H_0 : \frac{\mu_i}{\sigma_i} - \frac{\mu_n}{\sigma_n} = 0$ and $H_0 : \sigma_i - \sigma_n = 0$ respectively, where i is one of the twenty portfolios construct a standard methodology and n always denotes the adaptive strategy) we use bootstrapping methods, which are suitable when portfolio returns are not independently and identically distributed. In particular, since we deal with time series data, we follow the approach of Ledoit and Wolf (2008, 2011) to generate bootstrap data and we use the circular block bootstrap (Politis and Romano 1991) and the stationary bootstrap (Politis and Romano 1994) to carry out pairwise tests of equality of the portfolio Sharpe ratios and variances, respectively. Please notice that standard tools such as the Jobson and Korkie (1981) test for the equality of Sharpe Ratios or the F-test for the equality of variances assume that the data come from a bivariate normal distribution and are dependent over time. However, these tests are not valid if the returns are correlated, have heavy tails or are dependent over time: since financial data generally violate at least one of these conditions, a different approach should be pursued, hence we compute a two-sided p value using the time series bootstrap confidence interval for the difference of the Sharpe ratios and variances, to declare the two are different if zero is not contained in the obtained interval, following the

Table 4 The table reports the average returns ($\hat{\mu}^i$), the standard deviation ($\hat{\sigma}^i$), the Sharpe ratio (\widehat{SR}^i), and the turnover \widehat{T} computed on the out-of-sample returns

	$\hat{\mu}^i$		$\hat{\sigma}^i$		\widehat{SR}^i		\widehat{T}	
	$K = 10$	$K = 20$	$K = 10$	$K = 20$	$K = 10$	$K = 20$	$K = 10$	$K = 20$
OPX	0.1901	0.1574	0.2040 (0.0146)	0.1927 (0.0004)	0.9322 (0.0310)	0.8165 (0.1198)	0.0809	0.1541
UX	0.1915	0.1599	0.2053 (0.2879)	0.1911 (0.0258)	0.9329 (0.0400)	0.8368 (0.2701)	0.0936	0.1663
HX	0.2074	0.1653	0.2043 (0.0402)	0.1928 (0.1608)	1.0154 (0.1800)	0.8574 (0.3459)	0.0740	0.1541
LX	0.1874	0.1591	0.2061 (0.4769)	0.1932 (0.0210)	0.9093 (0.0698)	0.8235 (0.1532)	0.0746	0.1765
QBX	0.1931	0.1785	0.2032 (0.0018)	0.1878 (0.7742)	0.9502 (0.0452)	0.9505 (0.9584)	0.1010	0.1906
TPX	0.1896	0.1644	0.2055 (0.5279)	0.1890 (0.2997)	0.9228 (0.0508)	0.8700 (0.3959)	0.0831	0.1777
AX	0.1890	0.1747	0.2096 (0.3771)	0.1899 (0.4479)	0.9017 (0.0188)	0.9202 (0.7946)	0.0815	0.1839
GX	0.1938	0.1644	0.2020 (0.0002)	0.1917 (0.0466)	0.9594 (0.0492)	0.8575 (0.3287)	0.0739	0.1627
SBX	0.1778	0.1535	0.2086 (0.7974)	0.1918 (0.1392)	0.8521 (0.0034)	0.8003 (0.0784)	0.1038	0.1613
AVX	0.1807	0.1580	0.2075 (0.7770)	0.1924 (0.1038)	0.8712 (0.0032)	0.8212 (0.1152)	0.0629	0.1971
BLX	0.2174	0.1533	0.1987 (0.0002)	0.1944 (0.0276)	1.0938 (0.7618)	0.7884 (0.0464)	0.0928	0.1575
FX	0.1953	0.1507	0.2021 (0.0010)	0.1925 (0.0468)	0.9663 (0.0642)	0.7829 (0.0798)	0.0876	0.1688
GUX	0.2309	0.1643	0.2025 (0.0126)	0.1895 (0.3423)	1.1401 (0.9006)	0.8671 (0.3017)	0.0844	0.1640
TPX	0.2017	0.1540	0.2082 (0.9038)	0.1928 (0.1038)	0.9689 (0.0968)	0.7989 (0.0906)	0.0693	0.1672
LNx	0.1903	0.1484	0.2130 (0.0802)	0.1971 (0.0004)	0.8934 (0.0762)	0.7531 (0.0334)	0.0755	0.1464
DBX	0.1790	0.1673	0.2180 (0.0084)	0.1897 (0.5229)	0.8212 (0.0658)	0.8817 (0.5193)	0.0726	0.1532
UNDX	0.1927	0.1467	0.2076 (0.8100)	0.1902 (0.3387)	0.9286 (0.1160)	0.7711 (0.0398)	0.1081	0.1613
FR	0.2032	0.1557	0.2077 (0.8156)	0.1942 (0.0660)	0.9782 (0.1228)	0.8014 (0.0906)	0.0956	0.2163
SPX	0.1771	0.1602	0.2166 (0.1014)	0.1908 (0.1384)	0.8175 (0.1094)	0.8393 (0.2885)	0.1192	0.1873
PNX	0.1614	0.1621	0.2132 (0.0712)	0.1909 (0.1398)	0.7570 (0.0064)	0.8495 (0.1998)	0.0706	0.2308
Adaptive	0.2136	0.1505	0.2005	0.1889	1.0653	0.7966	0.1327	0.1550

The annualised performance metrics of each portfolio are computed by optimizing the *Omega ratio*, with cardinality and upper/lower bound constraints and $\varepsilon = 1e - 001$. The results are obtained using the Nikkei 225 dataset. The numbers in the parentheses below $\hat{\sigma}^i$ and \widehat{SR}^i are the p values computed using, respectively, the Ledoit and Wolf (2011) and Ledoit and Wolf (2008) tests, by contrasting the Adaptive strategy with each of the other competing portfolios

procedure described in Ledoit and Wolf (2008) and Ledoit and Wolf (2011), with $B = 5000$ bootstrap resamples and a block size equal to $b = 5$.

Table 4 suggests that the best annualized Sharpe Ratio is obtained for $K = 10$; on average we observe larger values of the Sharpe Ratio for portfolios with a smaller number of assets, suggesting that some ‘rough’ regularization approaches, involving the number of holdings or the inclusion of weight constraints (Jagannathan and Ma 2003) are particularly useful for managing overfitting issues, because they reduce the degree of freedom of the optimization process. Note that risk and return are inevitably measured with estimation error: as pointed out by Roncalli (2013), in a dynamic framework (e.g. for semiannually-rebalanced portfolios) estimation errors

have a dramatic impact on turnover, with a significant impact on the stability of solutions. Furthermore, we empirically observe that the portfolios constructed with stricter cardinality constraints are slightly more volatile. Finally, we check whether the variance and the Sharpe Ratio for a specific strategy are statistically different from that for the adaptive portfolio, and for this purpose we have implemented the tests introduced by Ledoit and Wolf (2008, 2011). The p values are reported in parentheses in Table 4 (recall that we want to test the null of equality of the Sharpe ratios and variances of the two portfolios being compared, hence the null hypothesis is rejected when the p values are at least lower than the 10% level): it is possible to see that the variances and the Sharpe Ratios of the adaptive model with the constraint $K = 10$ record a high number of rejections. In particular, the differences are statistically significant at the 5% level for, respectively, nine and eight portfolios out of twenty, whereas a smaller, yet non-negligible number of significant differences is observed when considering a less strict cardinality constraint.

As far as the crossover performance is involved, the genetic algorithm resorting to the adaptive approach behaves well on average, with some single crossover-based genetic algorithms performing occasionally better, though. Our adaptive strategy is however more robust across different risk measures and problem instances, as we also show in Fig. 8. Among standard genetic algorithms, let us highlight the fact that neither those equipped with exploration crossovers nor those employing exploitation crossovers outperform the others: note indeed that for $K = 10$ the best Sharpe Ratio is achieved by [GUX] and [BLX] (respectively, $\widehat{SR}_{GUX} = 1.1401$ and $\widehat{SR}_{BLX} = 1.0938$); however, these two operators do not share any specific features (the former is more oriented towards exploitation, the latter towards exploration, see Sect. 3). Moreover, for $K = 20$ [QBX] and [AX] outperform the rest of the operators, though not by a wide margin.

Altogether, an inspection of the results point to the fact that operators are generally unable to stand out consistently from the rest, which might suggest that combining operators at different stages of the search process could be actually beneficial for the solver. Furthermore, the AOS-based genetic algorithm is stably among top five performers across different risk measures and problem instances, signalling that this approach is rather effective at producing consistent and robust results across various formulations of the optimization problem. Further unreported results show that the best annualized Sharpe Ratio is achieved with a two-sided risk measure across different problem instances. Mean-variance portfolios perform poorly, with some portfolios constructed from a subset of the FTSE MIB displaying even a negative Sharpe Ratio.

For what concerns transaction costs, as shown in Table 4, there are no outperforming strategies in terms of turnover: the impact of regular portfolio rebalancing is overall homogeneous across different genetic algorithms, as we do not find any strategy regularly generating less costs. For instance, note that the turnover of [AVX] and [PNX] is definitely low compared to other single crossover-based GAs; however, when turning to portfolio with $K = 20$ assets, both are among the worst strategies in terms of trading costs.

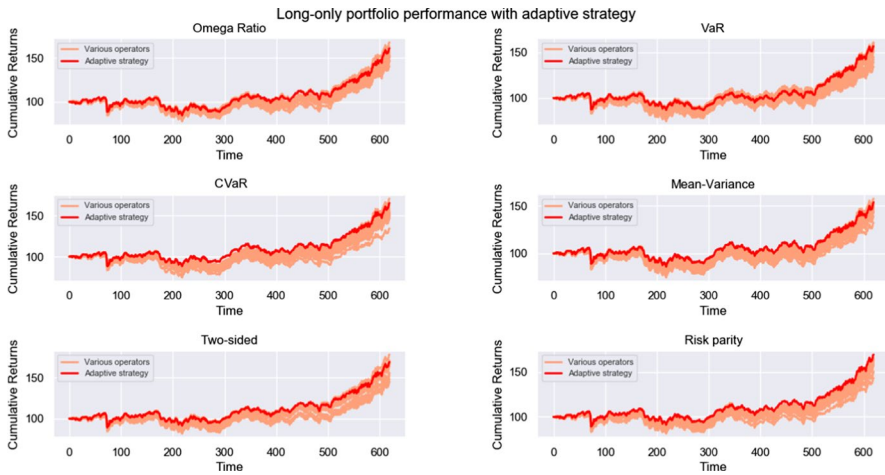


Fig. 8 Rolling-window backtest of the adaptive strategy on Nikkei 225 index, with $K = 10$, $l_i = 0.05$ and $u_i = 0.15$

Now consider a final example w.r.t. Nikkei 225 index. In Fig. 8 we present six different panels, in which both standard and adaptive strategies are backtested across different risk measures; rather than resorting to capitalization-weighted indices, we use as benchmarks the cumulative out-of-sample returns series optimized with standard genetic algorithms, which we construct with the crossover operators discussed in Sect. 3. In this set of backtests, the adaptive policy definitely performs better compared to alternative optimizers; also in terms of risk/reward tradeoff (Table 4), a small but consistent outperformance of the adaptive strategy stands out.

6 Conclusions

In this contribution we have applied an Adaptive Operator Selection (AOS) to Evolutionary Algorithms (EA) to solve Portfolio Selection Problems. First, we have identified all the crossover operators devised by the related literature regarding optimal portfolio choices; then, we have used AOS to decide which operator to use at each step of the search process, depending on several strategies.

We have seen that the introduction of AOS makes the genetic algorithm outperform the majority of the single crossover-based ones. This is due to the fact that the AOS decides, for each step of the search process, the right operator to be used with respect to the desired *Exploration vs Exploitation* balance: in our approach this balance can be either set by a deterministic strategy, or by an adaptive procedure that reacts to changes of the population's quality and diversity.

The genetic algorithm resorting to AOS show good performances over all instances taken into account. We have remarked that the genetic algorithm that relies on some specific single operators perform better than our AOS in some specific instances. This is due to the fact that AOS requires computational time to *learn* the

features of the crossover operator, hence subtracting computational time to the optimization phase itself. Single crossover-based GAs instead, do not suffer from this shortcoming. Of course, we would obtain better results by our AOS if we would run the algorithm for longer search epochs, but this is out of the scope of the current contribution: our goal is to devise a tool that show good performances without a-priori knowledge of the used crossovers and of the instance at hand, and to this extent, our approach shows good performances.

We can conclude by saying that the use of AOS improves the performances with regards to EAs that use single operators, which are the most used approaches in portfolio selection, no matter the objective function used, and over different sets of data coming from different countries and from different financial scenarios. Up to the authors' knowledge, this is the first contribution attempting to devise an approach based on all previous GA approaches for Portfolio Selection.

Moreover, we also contribute to the literature by investigating the benefits of introducing a broad range of operators for which we have not found any application to EA-based portfolio optimization: indeed, we shed light on the performance of crossovers as stand-alone operators and we find that the self-adaptive ones are able to achieve the best exploration–exploitation tradeoff, as pointed out by experiments showing both a promising behaviour and also a significantly high selection frequency.

Further work will be devoted to devise strategies that take into account external constraints and data-features. To this regard, we want to create specific variation operators that define, over the search, the investor preferences, in order to identify search spaces that corresponds to portfolios meeting the investor's preferences. Furthermore, we want to devise operators that test some of the features of the benchmark, in order to verify the investor's preferences, and to produce customized investment signals, i.e., to *stay away from the market*, or to enter into the market. Finally, a natural extension of our model should include further real-world constraints, e.g. by incorporating a restriction on tracking-error volatility or by defining a minimum lot size, so as to further challenge its capabilities in tackling an even more complex and realistic search space.

Declaration

Conflict of interest Authors declare that they have no significant competing financial, professional, or personal interests that might have influenced the performance or presentation of the work described in this manuscript.

References

- Anagnostopoulos K, Mamanis G (2011) A portfolio optimization model with three objectives and discrete variables. *Comput Oper Res* 37(7):1285–1297
- Arabas J, Michalewicz Z, Mulawka J (1994) GAVaPS-a genetic algorithm with varying population size. In: Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence. IEEE, pp 73–78

- Archetti C, Bertazzi L, Hertz A, Speranza MG (2012) A hybrid heuristic for an inventory routing problem. *INFORMS J Comput* 24(1):101–116
- Arifovic J (1994) Genetic algorithm learning and the cobweb model. *J Econ Dyn Control* 18(1):3–28
- Arnone S, Loraschi A, Tettamanzi A (1993) A genetic approach to portfolio selection. *Neural Netw World* 3(6):597–604
- Artzner P, Delbaen F, Eber J, Heath D (1999) Coherent measures of risk. *Math Finance* 9(3):203–228
- Arumugam MS, Rao M, Palaniappan R (2005) New hybrid genetic operators for real coded genetic algorithm to compute optimal control of a class of hybrid systems. *Appl Soft Comput* 6:38–52
- Ballester P, Carter J (2004) An effective real-parameter genetic algorithm with parent centric normal crossover for multimodal optimisation. In: *Genetic and evolutionary computation conference*. Springer, pp 901–913
- Beyer H, Deb K (2001) On self-adaptive features in real-parameter evolutionary algorithms. *IEEE Trans Evol Comput* 5(3):250–270
- Biglova A, Ortobelli S, Rachev S, Stoyanov S (2004) Different approaches to risk estimation in portfolio theory. *J Portf Manag* 31(1):103–112
- Birattari M, Stützle T, Paquete L, Varrentapp K (2002) A racing algorithm for configuring metaheuristics. In: *GECCO 2002: proceedings of the genetic and evolutionary computation conference*, New York, USA, 9–13 July 2002. Morgan Kaufmann, pp 11–18
- Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput Surv (CSUR)* 35(3):268–308
- Brabazon A, Keenan PB (2004) A hybrid genetic model for the prediction of corporate failure. *Comput Manag Sci* 1(3):293–310
- Brabazon A, O'Neill M (2004) Evolving technical trading rules for spot foreign-exchange markets using grammatical evolution. *Comput Manag Sci* 1(3):311–327
- Carleo A, Cesarone F, Gheno A, Ricci JM (2017) Approximating exact expected utility via portfolio efficient frontiers. *Decis Econ Finance* 40(1):115–143
- Chang TJ, Meade N, Beasley J, Sharaiha Y (2000) Heuristics for cardinality constrained portfolio optimisation. *Comput Oper Res* 27(13):1271–1302
- Chang TJ, Yang SC, Chang KJ (2009) Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Syst Appl* 36(7):10529–10537
- Chen Z, Wang Y (2008) Two-sided coherent risk measures and their application in realistic portfolio optimization. *J Bank Finance* 32(12):2667–2673
- Chen B, Zhong J, Chen Y (2020) A hybrid approach for portfolio selection with higher-order moments: empirical evidence from Shanghai Stock Exchange. *Expert Syst Appl* 145:113104
- Corazza M, Fasano G, Gusso R (2013) Particle swarm optimization with non-smooth penalty reformulation, for a complex portfolio selection problem. *Appl Math Comput* 224:611–624
- Corazza M, di Tollo G, Fasano G, Pesenti R (2021) A novel hybrid PSO-based metaheuristic for costly portfolio selection problems. *Ann Oper Res* 304(1):109–137
- Crama Y, Schyns M (2003) Simulated annealing for complex portfolio selection problems. *Eur J Oper Res* 150(3):546–571
- Črepinšek M, Liu S-H, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. *ACM Comput Surv (CSUR)* 45(3):1–33
- Deb K, Agrawal R, Bhusan R (1995) Simulated binary crossover for continuous search space. *Complex Syst* 9(2):115–148
- Deep K, Thakur M (2007) A new crossover operator for real coded genetic algorithms. *Appl Math Comput* 188(1):895–911
- DeMiguel V, Garlappi L, Nogales F, Uppal R (2009) A generalized approach to portfolio optimization: improving performance by constraining portfolio norms. *Manag Sci* 55(5):798–812
- Dempster MAH, Jones CM (2001) A real-time adaptive trading system using genetic programming. *Quant Finance* 1(4):397
- di Tollo G, Roli A (2008) Metaheuristics for the portfolio selection problem. *Int J Oper Res* 5(1):13–35
- di Tollo G, Lardeux F, Maturana J, Saubion F (2015) An experimental study of adaptive control for evolutionary algorithms. *Appl Soft Comput* 35:359–372
- Diosan L (2005) A multi-objective evolutionary approach to the portfolio optimization problem. In: *International conference on computational intelligence for modelling, control and automation and*

- international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06), vol 2. IEEE, pp 183–187
- Drenovak M, Ranković V, Urošević B, Jelic R (2022) Mean-maximum drawdown optimization of buy-and-hold portfolios using a multi-objective evolutionary algorithm. *Finance Res Lett* 46:102328
- Eiben AE, Smith JE (2015) Introduction to evolutionary computing, 2nd edn. Natural computing series. Springer, Berlin
- Eshelman L, Schaffer J (1993) Real-coded genetic algorithms and interval-schemata. In: Foundations of genetic algorithms, vol 2. Elsevier, pp 187–202
- Fialho Á (2010) Adaptive operator selection for optimization. Ph.D. thesis, University of Paris-Sud, Orsay, France
- Fletcher R (2013) Practical methods of optimization. Wiley, Hoboken
- Gilli M, Küllezi E, Hysi H (2006) A data-driven optimization heuristic for downside risk minimization. Swiss Finance Institute Research Paper (06-2)
- Gilli M, Schumann E, di Tollo G, Cabej G (2011) Constructing 130/30-portfolios with the Omega ratio. *J Asset Manag* 12(2):94–108
- Goldberg DE (1989) Genetic algorithms in search optimization and machine learning. Addison-Wesley Professional, Boston
- Gunjan A, Bhattacharyya S (2022) A brief review of portfolio optimization techniques. *Artif Intell Rev* 1–40
- Herrera F, Lozano M, Sánchez A (2005) Hybrid crossover operators for real-coded genetic algorithms: an experimental study. *Soft Comput* 9(4):280–298
- Hochreiter R (2008) Evolutionary stochastic portfolio optimization. In: Natural computing in computational finance. Springer, pp 67–87
- Hochreiter R (2015) An evolutionary optimization approach to risk parity portfolio selection. In: European conference on the applications of evolutionary computation. Springer, pp 279–288
- Holland J (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence. MIT Press, Cambridge
- Hoogs B, Kiehl T, Lacombe C, Senturk D (2007) A genetic algorithm approach to detecting temporal patterns indicative of financial statement fraud. *Intell Syst Account Finance Manag Int J* 15(1–2):41–56
- Huang X (2007) A new perspective for optimal portfolio selection with random fuzzy returns. *Inf Sci* 177(23):5404–5414
- Huang X (2008) Mean-semivariance models for fuzzy portfolio selection. *J Comput Appl Math* 217(1):1–8
- Huang C, Li Y, Yao X (2020) A survey of automatic parameter tuning methods for metaheuristics. *IEEE Trans Evol Comput* 24(2):201–216
- Jagannathan R, Ma T (2003) Risk reduction in large portfolios: why imposing the wrong constraints helps. *J Finance* 58(4):1651–1683
- Jeong SH, Lee HS, Nam H, Oh KJ (2021) Using a genetic algorithm to build a volume weighted average price model in a stock market. *Sustainability* 13(3):1011
- Jobson JD, Korkie BM (1981) Performance hypothesis testing with the Sharpe and Treynor measures. *J Finance* 36:889–908
- Kalayci CB, Polat O, Akbay MA (2020) An efficient hybrid metaheuristic algorithm for cardinality constrained portfolio optimization. *Swarm Evol Comput* 54:100662
- Kar MB, Kar S, Guo S, Li X, Majumder S (2019) A new bi-objective fuzzy portfolio selection model and its solution through evolutionary algorithms. *Soft Comput* 23(12):4367–4381
- Kaucic M (2019) Equity portfolio management with cardinality constraints and risk parity control using multi-objective particle swarm optimization. *Comput Oper Res* 109:300–316
- Keating C, Shadwick K (2002) A universal performance measure. *J Perform Meas* 6(3):59–84
- Kizys R, Doering J, Juan AA, Polat O, Calvet L, Panadero J (2022) A simheuristic algorithm for the portfolio optimization problem with random returns and noisy covariances. *Comput Oper Res* 139:105631
- Kolm P, Tütüncü R, Fabozzi F (2014) 60 years of portfolio optimization: practical challenges and current trends. *Eur J Oper Res* 234(2):356–371
- Krink T, Paterlini S (2011) Multiobjective optimization using differential evolution for real-world portfolio optimization. *Comput Manag Sci* 8(1–2):157–179

- Lardeux F, Saubion F, Hao J-K (2006) GASAT: a genetic local search algorithm for the satisfiability problem. *Evol Comput* 14(2):223–253
- Ledoit O, Wolf M (2008) Robust performance hypothesis testing with the Sharpe ratio. *J Empir Finance* 15(5):850–859
- Ledoit O, Wolf M (2011) Robust performances hypothesis testing with the variance. *Wilmott* 2011(55):86–89
- Lin C-C, Liu Y-T (2008) Genetic algorithms for portfolio selection problems with minimum transaction lots. *Eur J Oper Res* 185(1):393–404
- Lin D, Wang S, Yan H (2001) A multiobjective genetic algorithm for portfolio selection problem
- Liu B (2010) Uncertainty theory. In: *Uncertainty theory*. Springer, pp 1–79
- Liu S, Stefek D (1995) A genetic algorithm for the asset paring problem in portfolio optimization. In: *Proceedings of the first international symposium*, pp 441–450
- Loraschi A, Tettamanzi A (1995) An evolutionary algorithm for portfolio selection in a downside risk framework. *Work Papers Financial Econ* 6:8–12
- Loraschi A, Tomassini M, Tettamanzi A, Verda P (1995) Distributed genetic algorithms with an application to portfolio selection problems. In: *Artificial neural nets and genetic algorithms*. Springer, pp 384–387
- Lwin K, Qu R, MacCarthy BL (2017) Mean-VaR portfolio optimization: a nonparametric approach. *Eur J Oper Res* 260(2):751–766
- Maillard S, Roncalli T, Teiletche J (2010) The properties of equally weighted risk contribution portfolios. *J Portf Manag* 36(4):60–70
- Maringer DG (2005) Distribution assumptions and risk constraints in portfolio optimization. *Comput Manag Sci* 2(2):139–153
- Maringer D, Winker P (2003) Portfolio optimization under different risk constraints with modified memetic algorithms. *Universität Erfurt Staatswissenschaften Fakultät, Erfurt*
- Markowitz H (1952) Portfolio selection. *J Finance* 7(1):77–91
- Markowitz H (1959) *Portfolio selection: efficient diversification of investments*. Wiley, Hoboken
- Markowitz H (2014) Mean-variance approximations to expected utility. *Eur J Oper Res* 234(2):346–355
- Maturana J, Saubion F (2008) A compass to guide genetic algorithms. In: *International conference on parallel problem solving from nature*. Springer, pp 256–265
- Maturana J, Fialho Á, Saubion F, Schoenauer M, Sebag M (2009) Extreme compass and dynamic multi-armed bandits for adaptive operator selection. In: *2009 IEEE congress on evolutionary computation*, pp 365–372
- Maturana J, Lardeux F, Saubion F (2010) Autonomous operator management for evolutionary algorithms. *J Heuristics* 16(6):881–909
- Mazraeh NB, Daneshvar A, Roodposhti FR et al (2022) Stock portfolio optimization using a combined approach of multi objective grey wolf optimizer and machine learning preselection methods. *Comput Intell Neurosci* 2022
- Mehlawat MK, Gupta P, Khan AZ (2021) Multiobjective portfolio optimization using coherent fuzzy numbers in a credibilistic environment. *Int J Intell Syst* 36(4):1560–1594
- Michalewicz Z (1992) *Genetic algorithms + data structures = evolution programs, artificial intelligence*. Springer, Berlin
- Michalewicz Z, Schoenauer M (1996) Evolutionary algorithms for constrained parameter optimization problems. *Evol Comput* 4(1):1–32
- Michalewicz Z, Nazhiyath G, Michalewicz M (1996) A note on usefulness of geometrical crossover for numerical optimization problems. In: Fogel LJ, Angeline PJ, Bäck T (eds) *Proceedings of the fifth annual conference on evolutionary programming, EP 1996, San Diego, CA, USA, February 29–March 2, 1996*. MIT Press, pp 305–312
- Mittal SK, Srivastava N (2021) Mean-variance-skewness portfolio optimization under uncertain environment using improved genetic algorithm. *Artif Intell Rev* 54(8):6011–6032
- Moral-Escudero R, Ruiz-Torrubiano R, Suárez A (2006) Selection of optimal investment portfolios with cardinality constraints. In: *2006 IEEE international conference on evolutionary computation*. IEEE, pp 2382–2388
- Neri F, Cotta C, Moscato P (2012) *Handbook of memetic algorithms*. Studies in computational intelligence
- Ono I, Kobayashi S (1999). A real-coded genetic algorithm for function optimization using unimodal normal distribution. In: *Proceedings of international conference on genetic algorithms*, pp 246–253

- Politis DN, Romano JP (1991) A circular block-resampling procedure for stationary data. Department of Statistics, Purdue University, West Lafayette
- Politis DN, Romano JP (1994) The stationary bootstrap. *J Am Stat Assoc* 89(428):1303–1313
- Prins C (2004) A simple and effective evolutionary algorithm for the vehicle routing problem. *Comput Oper Res* 31(12):1985–2002
- Procacci PF, Aste T (2022) Portfolio optimization with sparse multivariate modeling. *J Asset Manag* 23:1–21
- Radcliffe N (1991) Equivalence class analysis of genetic algorithms. *Complex Syst* 5(2):183–205
- Ranković V, Drenovak M, Orošević B, Jelić R (2016) Mean-univariate GARCH VaR portfolio optimization: actual portfolio approach. *Comput Oper Res* 72:83–92
- Renders J, Bersini H (1994) Hybridizing genetic algorithms with hill-climbing methods for global optimization: two possible ways. In: Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence, vol 1, pp 312–317
- Rockafellar R, Uryasev S (2000) Optimization of conditional value-at-risk. *J Risk* 2:21–42
- Roncalli T (2013) Introduction to risk parity and budgeting. CRC Press, Boca Raton
- Salehpour IB, Molla-Alizadeh-Zavadehi S (2019) A constrained portfolio selection model at considering risk-adjusted measure by using hybrid meta-heuristic algorithms. *Appl Soft Comput* 75:233–253
- Schaerf A (2002) Local search techniques for constrained portfolio selection problems. *Comput Econ* 20(3):177–190
- Simon D (2013) Evolutionary optimization algorithms. Biologically inspired and population-based approaches to computer intelligence. Wiley, Hoboken
- Skolpadungket P, Dahal K, Harnpornchai N (2007) Portfolio optimization using multi-objective genetic algorithms. In: 2007 IEEE congress on evolutionary computation. IEEE, pp 516–523
- Smit SK, Eiben AE (2009) Comparing parameter tuning methods for evolutionary algorithms. In: 2009 IEEE congress on evolutionary computation. IEEE, pp 399–406
- Soleimani H, Golmakani HR, Salimi MH (2009) Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm. *Expert Syst Appl* 36(3):5058–5063
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359
- Streichert F, Ulmer H, Zell A (2004a) Comparing discrete and continuous genotypes on the constrained portfolio selection problem. In: Genetic and evolutionary computation conference. Springer, pp 1239–1250
- Streichert F, Ulmer H, Zell A (2004b) Evaluating a hybrid encoding and three crossover operators on the constrained portfolio selection problem. In: Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753), vol 1. IEEE, pp 932–939
- Streichert F, Ulmer H, Zell A (2004c) Evolutionary algorithms and the cardinality constrained portfolio optimization problem. In: Operations research proceedings 2003. Springer, pp 253–260
- Subbu R, Bonissone PP, Eklund N, Bollapragada S, Chalermkraivuth K (2005) Multiobjective financial portfolio design: a hybrid evolutionary approach. In: 2005 IEEE congress on evolutionary computation, vol 2. IEEE, pp 1722–1729
- Suksonghong K, Boonlong K (2021) A new encoded scheme GA for solving portfolio optimization problems in the big data environment. In: Advances in computer, communication and computational sciences. Springer, pp 551–560
- Sung H (2007) Queen-bee and mutant-bee evolution for genetic algorithms. *J Fuzzy Logic Intell Syst* 17:417–422
- Syswerda G (1993) Simulated crossover in genetic algorithms. In: Foundations of genetic algorithms, vol 2. Elsevier, pp 239–255
- Tapia MGC, Coello Coello CA (2007) Applications of multi-objective evolutionary algorithms in economics and finance: a survey. In: 2007 IEEE congress on evolutionary computation. IEEE, pp 532–539
- Voigt H, Mühlenbein H, Cvetkovic D (1995) Fuzzy recombination for the breeder genetic algorithm. In: Proceedings of the 6th international conference on genetic algorithms. Citeseer
- Wang S, Chen J, Wee H, Wang K (2006) Non-linear stochastic optimization using genetic algorithm for portfolio selection. *Int J Oper Res* 3(1):16–22
- Wang Z, Zhang X, Zhang Z, Sheng D (2022) Credit portfolio optimization: a multi-objective genetic algorithm approach. *Borsa Istanbul Rev* 22(1):69–76

- Wright AH (1991) Genetic algorithms for real parameter optimization. In: Foundations of genetic algorithms. Morgan Kaufmann, pp 205–218
- Xia Y, Liu B, Wang S, Lai K (2000) A model for portfolio selection with order of expected returns. *Comput Oper Res* 27(5):409–422
- Yaman I, Dalkılıç TE (2021) A hybrid approach to cardinality constraint portfolio selection problem based on nonlinear neural network and genetic algorithm. *Expert Syst Appl* 169:114517
- Yu Y, Deng X, Chen C, Cheng K (2020) Research on fuzzy multi-objective multi-period portfolio by hybrid genetic algorithm with wavelet neural network. *Eng Lett* 28(2)
- Yusuf R, Handari BD, Hertono GF (2019) Implementation of agglomerative clustering and genetic algorithm on stock portfolio optimization with possibilistic constraints. In: AIP conference proceedings, vol 2168. AIP Publishing LLC, p 020028
- Zadeh LA (1965) Information and control. *Fuzzy Sets* 8(3):338–353

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.