Neural Computing and Applications (2024) 36:7221–7243
https://doi.org/10.1007/s00521-024-09456-w

ORIGINAL ARTICLE

# EvoFolio: a portfolio optimization method based on multi-objective evolutionary algorithms

Alfonso Guarino[1] · Domenico Santoro[2] · Luca Grilli[3] · Rocco Zaccagnino[1] · Mario Balbi[1]

Received: 3 June 2023 / Accepted: 15 January 2024 / Published online: 19 February 2024

## Abstract

*Optimal portfolio selection*—composing a set of stocks/assets that provide high yields/returns with a reasonable risk—has attracted investors and researchers for a long time. As a consequence, a variety of methods and techniques have been developed, spanning from purely mathematics ones to computational intelligence ones. In this paper, we introduce a method for optimal portfolio selection based on multi-objective evolutionary algorithms, specifically *Nondominated Sorting Genetic Algorithm*-II (NSGA-II), which tries to *maximize* the yield and *minimize* the risk, simultaneously. The system, named *EvoFolio*, has been experimented on stock datasets in a three-years time-frame and varying the configurations/specifics of NSGA-II operators. *EvoFolio* is an *interactive* genetic algorithm, i.e., users can provide their own insights and suggestions to the algorithm such that it takes into account users' preferences for some stocks. We have performed tests with optimizations occurring quarterly and monthly. The results show how *EvoFolio* can significantly reduce the risk of portfolios consisting only of stocks and obtain very high performance (in terms of return). Furthermore, considering the investor's preferences has proved to be very effective in the portfolio's composition and made it more attractive for end-users. We argue that *EvoFolio* can be effectively used by investors as a support tool for portfolio formation.

**Keywords** Portfolio optimization · Multi-objective evolutionary algorithms · Stock market · Realistic genetic operator

**Mathematics Subject Classification** 68T07 · 68T05 · 62M45 · 93C42

## 1 Introduction

The asset's choice to include in a portfolio has always been central to financial studies. Generally, whether or not investors are experts in the financial domain, the basic idea in building a portfolio is to maximize the return, distributing one's balance "equally" among the different assets. This process, called asset allocation, has found a mathematical formalization thanks to the introduction of Markowitz's [1] *Portfolio Selection*. This theory is based on mean-variance, which aims to maximize the portfolio's expected return (mean, $\mu_{P_S}$) while maintaining a certain level of risk (variance, $\sigma_{P_S}$). In this way, under certain conditions such as risk aversion, market efficiency, and access to information, it is possible to determine, given a certain number of assets, the set of *admissible portfolios* (which can be constructed with the usable assets) and the *efficient frontier*, i.e., the set of portfolios that minimize the variance and maximize the expected return. The

✉ Rocco Zaccagnino
rzaccagnino@unisa.it

Alfonso Guarino
alguarino@unisa.it

Domenico Santoro
domenico.santoro@uniba.it

Luca Grilli
luca.grilli@unifg.it

Mario Balbi
m.balbi3@studenti.unisa.it

[1] Department of Computer Science, University of Salerno, Via Giovanni Paolo II, 130, Fisciano, SA, Italy

[2] Department of Economics and Finance, University of Bari, Largo Abbazia S. Scolastica, 70124 Bari, BA, Italy

[3] Department of Economics, Management and Territory, University of Foggia, Via da Zara, 11, 71121 Foggia, FG, Italy

fundamental role of diversification to reduce the risk, i.e., the choice of assets that are not perfectly correlated, can be seen from the Portfolio Selection. The constraints in Eq. 1 describe the optimal portfolio composition in the original version:

$$\begin{aligned}
&\mu_{P_S} = \sum_{i=1}^{n} w_i \mu_i,\\
&\sigma_{P_S} = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \sigma_i \sigma_j Cov(i,j),\\
&\sum_{i=1}^{n} w_i = 1,\\
&w_i \geq 0,
\end{aligned} \tag{1}$$

where $\mu_i$ and $\sigma_i$ represent the expected return and the variance of the $i$-th asset while $w_i$ is the number of assets in the portfolio.

Sharpe [2] extends the theory introduced by Markowitz, considering the assets' sensitivity to systematic risk (which cannot be eliminated), in the *Capital Asset Pricing Model (CAPM)* [3, 4]. The latter allows the relationship between risk and return of an asset to be expressed through a single risk factor, indicated as $\beta$ [5]. The CAPM extends Markowitz's theory by introducing an asset with a risk-free return, which allows the identification of a risky portfolio. Under certain assumptions, which include borrowing money at the risk-free rate, this model identifies an asset's return based on its degree of risk, determined using the market portfolio (an ideal portfolio comprised of all the assets present on the market). In the CAPM formulation, the expected return of the $i$-th asset $\mu_i$ is given by:

$$\mu_i = \beta(\mu_m - r_f) + r_f, \tag{2}$$

where $\mu_m$ is the market portfolio return, $r_f$ is the risk-free rate, and $\beta = \frac{Cov(r_i, r_m)}{\sigma_m}$ is the ratio of the covariance of gross returns to the variance of the market portfolio. Systematic risk $\beta$, therefore, measures the change in the asset's expected return to the change in the market return. Portfolio Selection and CAPM together make up the *Modern Portfolio Theory (MPT)*.

Subsequently, Ross [6] introduces a new method (as the evolution of the CAPM) for evaluating the return of an asset that directly considers macro-economic risk factors, the *Arbitrage Pricing Theory (APT)*. These risk factors considered can be inflation risk, economic cycle risk, interest rate risk, and any other type that can be deemed informative. Therefore, the APT, under certain assumptions such as the efficiency of markets, risk aversion of investors, and the indicativeness of systematic risk by risk factors, determines the expected returns of an asset with a multivariate model:

$$\mu_i = r_f + \sum_{k=1}^{n} \beta_{ik} Z_k, \tag{3}$$

where $r_f$ is the risk-free rate, $\beta_{ik}$ is the sensitivity of asset $i$ to the $k$-th risk factor, and $Z_k$ is the risk premium associated with that factor.

Finally, *Black and Litterman (BL)* [7] develop a model based on the Bayesian approach capable of processing the information held by the investor, known as *views*. In this way, the BL model combines the market equilibrium with the investor's views to obtain the optimal portfolio composition. Thanks to the Bayesian approach, this model can combine historical returns of an asset with views to obtain a *posterior* representing the Black-Litterman returns $\mu$ [8] in matrix form:

$$\mu = [(\tau\Sigma)^{-1} + P^T\Omega^{-1}P]^{-1}[(\tau\Sigma)^{-1}\Pi + P^T\Omega^1 Q], \tag{4}$$

where $\Sigma$ is the covariance matrix, $P$ is the matrix that connects the investor's views with the model's assets, $Q$ is the vector of views, $\Omega$ is the uncertainty matrix of views, $\Pi$ is the vector of prior expected returns, and $\tau$ is a scalar constant (defined as weight-on-views). Using the Idzorek [9] method to estimate $\Omega$ and considering $\delta$ as the risk aversion factor, the weights $w$ of the assets to be included in the portfolio are determined as:

$$w = (\delta\Sigma)^{-1}\mu. \tag{5}$$

These four models represent the cornerstones of portfolio optimization. Naturally, thanks to the development of statistical techniques that are increasingly appropriate to the domain and to the growing computational capacity, asset allocation techniques have improved exponentially over time.

*Objective and motivation* In this paper, our goal is to select the optimal portfolio given a set of stocks and a budget. In our approach, the optimal portfolio we aim at is the solution of a *multi-objective* problem, i.e., the one that *maximizes* the yield and, at the same time, *minimizes* the risk. Furthermore, our objective is to use such an optimized portfolio over time and re-optimize it in turn with market's changes. In this way, "dynamic" optimization over time allows us to tackle the market changes and extend the number of assets that can be considered. Furthermore, solving the optimization problem with evolutionary algorithms overcomes the computational problems related to matrix calculus of the classical methods and leads to superior performance from the portfolio created, also thanks to the ability of these methods to exploit latent relationships between the assets (which the classical methods do not have). Lastly, differently from previous works (more details in Sect. 3), we aim at processing the investor's interests/views, modifying the portfolio's composition with assets indicated as priorities by the end-user.

*The proposed approach.* We introduce *EvoFolio*, which is a portfolio optimization method based on Multi-Objective Evolutionary Algorithms (MOEAs), in particular, *Nondominated Sorting Genetic Algorithm* (NSGA-II) [10] (a genetic algorithm). *EvoFolio*, based on a budget and a set of stocks, can select a set of optimal portfolios determining the one(s) that provides larger yield and lower risk through the Pareto Front technique.

In Fig. 1, we sketch the visual abstract of our proposal. In general terms, *EvoFolio* performs the following steps: *(1)* creates feasible portfolios given a budget, *(2)* evaluates their yield and risk, *(3)* selects the optimal ones for *(4)* mutation and crossover operators, and iterates these phases until no more optimization can be performed or maximum steps are reached. Moreover, *EvoFolio* implements *elitism*, i.e., tends to keep track of investors' insights/suggestions, in the example to award a special consideration for <XXX> stock.

*EvoFolio* limits the number of different instruments from 10 to 14 as indicated by Pal et al. [11], and suggested by the theory of Evans and Archer [12]. It can be used to incrementally optimize portfolio selection over time to help investors keep up with the market's fluctuations, ups and downs. *EvoFolio* implements the elitism technique by enabling users to interact during the evolutionary process and highlighting their preferences for some stocks. *EvoFolio* has been tested using 47 different instruments datasets in a time-frame of three years (2020-2022) referring to companies operating in various sectors such as finance, tech, energy, and industrial and whose stocks have undergone sudden fluctuations in response to critical events such as the Covid-19 pandemic or the outbreak of the Russia-Ukraine conflict. To find the best setting for *EvoFolio*, we have experimented with a series of configurations (among which quarterly and monthly optimization) in a limited, and particularly relevant for traders, time-frame, i.e., the first period of the COVID-19 pandemic, to select the one(s) providing the highest performance. Then, the best parameters and settings were carried out for experimenting over the whole period of interest. We have evaluated *EvoFolio* outcomes with and without users' views on the market. Overall, the results were promising in all cases, not only in terms of gains from the value of the portfolio but, above all, in terms of shallow risk obtained from an equity-only portfolio, given the historical moment characterized by substantial uncertainty (due to the Covid-19 pandemic and the Russia-Ukraine conflict). Furthermore, the views lowered the risk and thus made the portfolios more appealing for an end-user (investor).

*Key points.* The main contributions of this paper are:

- The introduction of *EvoFolio*, a portfolio optimization method based on the NSGA-II algorithm, whose objectives are maximizing the yield and minimizing the risk (defined as in the Markowitz model), available at https://github.com/Balbus95/EvoFolio;
- Implementing the possibility for users to express their "views" on the market, taking inspiration from the Human-Centered AI perspective [13] that aims to keep humans at the center of the development of intelligent solutions – "[...] *empower rather than replace people* [...] *raising the value of humans* [...]" [13]; to the best

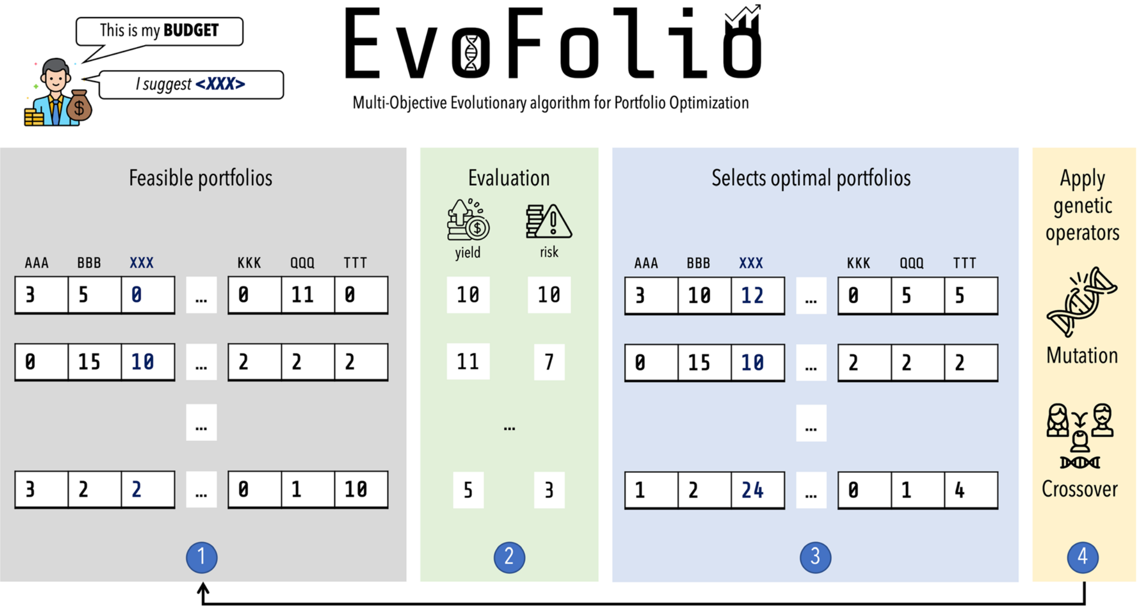


**Fig. 1** Visual abstract of our proposal. *EvoFolio* algorithm's steps

of our knowledge, this is one of the first portfolio optimization methods providing such a feature;

- Experimenting *EvoFolio* in a particularly challenging time-frame characterized by high volatility (often associated with fearful investors), i.e., the COVID-19 and the Russia-Ukraine conflict periods. In such a period, *EvoFolio* not only effectively provides adjusted portfolios but also obtains promising returns (in US Dollars).
- We compare the portfolios created by *EvoFolio* with those made by the FinRL framework [14], in terms of value in US Dollars. To the best of our knowledge, this is one of the first articles discussing such a comparison with other portfolio optimization methods. This comparison shows that *EvoFolio* portfolios in different situations outperform RL strategies, with much higher explainability given by the exact number of stocks in each portfolio.

*The road-map of the paper.* This article is structured as follows: Sect. 2 is devoted to sketch the basic information on MOEAs and the problem we face; Sect. 3 presents the literature review, highlighting initiative and studies that show contact points with the presented paper; Sect. 4 presents our proposal and offer details on the experiments conducted, while Sect. 5 is devoted to present the results achieved; Sect. 6 provides a discussion on our proposal with comparisons against other models; Sect. 7 concludes the paper with final remarks, an overview of the work done, its limitations and future steps for possible gaps filling, and it traces the path for future developments of this research.

## 2 Preliminaries

In this section, we offer a brief overview on Genetic Algorithms (GAs) and MOEAs (Sect. 2.1), and we provide the problem formulation (Sect. 2.2).

### 2.1 GAs and multi-objective optimization

Genetic Algorithms are based on the computer simulation of the evolution of a population according to natural selection processes and genetic laws. According to this idea, a pool of candidate solutions, known as *individuals* or *phenotypes*, corresponds to a population of abstract representations, commonly called *chromosomes*. The population is stochastically generated, and it evolves for several iterations called *generations*. In each generation, the fitness of every individual in the population is assessed and assigned with a fitness value that somehow evaluates the individual. Then, based on their fitness, multiple individuals are stochastically selected from the current population (*selection*), recombined (*recombination* or *crossover*), and randomly modified (*mutation*) to form a new population. The new population passes through the same stages of selection-mutation-crossover until a satisfactory solution is encountered or an a priori fixed maximum number of generations has been reached.

The idea of GAs was introduced by [15]. Since then, the approach has been applied to many areas. Among the others, a general discussion about the genetic approach can be found in [16–19].

In many cases, the fitness function is defined according to one particular target, which turns into a single-objective optimization (maximization or minimization) problem. However, some optimization problems involve multiple criteria or objectives to be considered. A multi-objective GA (MOGA) assesses individuals' fitness using several evaluation functions. Assuming that we have a maximization problem, this yields the problem of maximizing several functions:

$$max(f_1(x), f_2(x), \dots.., f_k(x))$$

where $k$ is the number of objectives and $f_i : X \rightarrow R$ for each $i = 1, \dots, k$ ($X$ is the problem space) is the $i$-th objective. The problem in using a multi-objective function is that it might be impossible to maximize all the objectives simultaneously (usually, trying to maximize one of the functions will inevitably lower the value of other functions). Indeed, such objectives can be conflicting, and no single solution, usually, simultaneously optimizes all of them. Hence we must find solutions that achieve reasonable compromises among the various objectives. Thus, a set of optimal solutions can be produced, instead of a single optimal solution, because no single solution could be optimal for multiple conflicting objectives. A solution $y_1$, which is better than a solution $y_2$ with respect to all the objectives, is preferable. In this case, we say that $y_1$ "dominates" $y_2$ (or $y_2$ is dominated by $y_1$). A non-dominated solution is an optimal solution. The Pareto-front [20, 21] is the set of all non-dominated solutions. MOEAs are a class of search methods including MOGAs that approximate the Pareto-front, i.e., instead of guaranteeing to find all non-dominated solutions, they aim at finding solutions that come as close as possible to the optimal ones. Once a set of such solutions is found, a higher-level decision-making strategy could be adopted to pick a single solution.

Several MOEAs have been proposed in various applications, and their performances were tested in as many comparative studies. The first MOGA, called *Vector Evaluated Genetic Algorithm* was proposed by [22]. Afterward, other major MOEAs were presented, such as the algorithm by [20]), the *Niched Pareto Genetic*

*Algorithm* by [23], the *Nondominated Sorting Genetic Algorithm* by [24], the Fast *Nondominated Sorting Genetic Algorithm* (NSGA-II) by [10]. In this work, we exploited a multi-objective genetic algorithm obtained by specialized variations of the NSGA-II strategy. One of the most interesting features of the NSGA-II algorithm is the *elitism* technique [25]. Elitism is achieved by maintaining an external population of all non-dominated individuals. Such a population is essential because its individuals have characteristics we wish to keep in our final solution. The external population will be used to preserve good genes. The evolution operators will also consider individuals in the external population to produce new chromosomes. In our algorithm, we do use such a technique.

### 2.2 Portfolio optimization: problem description

We define a *stock configuration* as a set $S = A_1, ..., A_N$ where $A_i$ is a stock, for $i = 1, ..., N$. Every (kind of) stock $A_i$ has the following characteristics:

- $c_{i,t}$, i.e., the *price* of a single unit of $A_i$ in a specific time period $t$ (day, week, month);
- $y_{i,t} = \log \frac{c_{i,t}}{c_{i,t-1}}$, i.e., the *yield* that every unit of $A_i$ is generating in a specific time period $t$ (day, week, month) with respect to a previous observation $t-1$;

$TY_{i,t}$ indicates the yields' list of $A_i$ until time $t$, that is:

$$TY_{i,t} = [y_{i,t}, y_{i,t-1}, \ldots, y_{i,0}]$$

Given a stock configuration $S = A_1, \ldots, A_N$, we define a *portfolio* $P_S$ for $S$ as a vector $P_S = [P_1, ..., P_N]$ where $P_i \geq 0$ is an integer number which indicates the quantity of $A_i$ unities purchased, for $i = 1, ..., N$.

Let $P_S$ be a portfolio for $S$ and $t$ a fixed time period, we define the *overall price* $C(P_S, t)$, the *overall average yield* $\mu_{P_S}$, and the *overall risk* $\sigma_{P_S}$, of $P_S$ at time $t$ as:

$$C(P_S, t) = \sum_{i=1}^{N} P_i \times c_{i,t} \quad //\textit{overall price}$$

$$Y(P_s, t) = \sum_{i=1}^{N} P_i \times \frac{\sum_{t=0}^{|TY_{i,t}|} y_{i,t}}{|TY_{i,t}|} \quad //\textit{overall average yield}$$

$$R(P_S, t) = \sum_{A_i, A_j | i \neq j} R(A_i, A_j, t) \quad //\textit{overall risk}$$

where $R(A_i, A_j, t) = \frac{P_i}{\sum_{k=1}^{N} P_k} \times \sigma_{TY_{i,t}} + \frac{P_j}{\sum_{k=1}^{N} P_k} \times \sigma_{TY_{j,t}} + 2 \times \frac{P_i}{\sum_{k=1}^{N} P_k} \times \frac{P_j}{\sum_{k=1}^{N} P_k} \times \sigma_{TY_{i,t}, TY_{j,t}}$.

Notice that with $\sigma_X$ (where $X$ is a generic set of values) we refer to the standard deviation of $X$, while with $\sigma_{X,Y}$ (where $X$ and $Y$ are generic sets of values) we refer to the covariance between $X$ and $Y$.

$P_S$ quality is measured in terms of $Y(P_S, t)$, that is the larger the $Y(P_S, t)$ the higher the quality, and in terms of $R(P_S, t)$, that is the larger the $R(P_S, t)$ the lower the quality. The *portfolio optimization problem* can be formalized as follows: given a stock configuration $S$ and an initial budget $B$, we have to find the "best" portfolio $S$, i.e., the portfolio $P_S$ such that $C(P_S) \leq B$, that "*maximizes*" $Y(P_S, t)$ and "*minimizes*" $R(P_S, t)$.

## 3 Literature review

The numerical methods used to solve the portfolio asset optimization problem cover many sectors, from stochastic programming to evolutionary computing and Reinforcement Learning (RL). Here, we sketch the most interesting works in such research fields and highlight some differences with our paper.

*Time series approach.* The starting point is analyzing historical time series to identify factors influencing performance. For example, Fama and French [26] use expected returns to study the link with time components, with subsequent implementation of Pesaran and Timmermann [27] to take into account macro-economic factors. From an econometric point of view, Box and Jenkins [28] develop a procedure for estimating a time series model with specific characteristics such as stationarity, the AutoRegressive Integrated Moving Average (ARIMA), a version deriving from the AutoRegressive Moving Average (ARMA) model. Many transformations, based on the formalism introduced by Hamilton [29], have been adapted to this model, such as the Seasonality AutoRegressive Integrated Moving Average (SARIMA) model with seasonality. Through the approach based on time series, many portfolio management models use these techniques to estimate the critical parameters.

As introduced by Gunjan and Bhattacharyya [30], the constraints to be respected in choosing the number of assets to include in the portfolio are based on various risk measures, where the most common are *Value-at-Risk* (VaR) [31] and Conditional VaR (CVaR) [32]. These measures indicate the potential loss of an asset in a certain period based on a level of confidence. They are calculated with various approaches (including that based on historical series and simulations). In this framework, Xu et al. [33] and Xu and Ng [34] propose a soft method for portfolio optimization in a VaR-based approach, testing the results on the New York stock market. Instead, Lim et al. [35] try to solve the optimization problem using the CVaR, demonstrating how this risk measure is coherent but sensitive to errors.

Relying on time series, many approaches use linear and stochastic programming to solve the optimization problem. For example, Geyer et al. [36] maximize expected utility in a model where asset returns follow a vector autoregression. Dupačová [37] proposes a robust stochastic programming method to solve the optimization problem in the Markowitz model. Kouwenberg [38] proposes a multi-stage stochastic programming model, comparing it with the results of a simple fix mix model and demonstrating how stochastic programming outperforms the results. Barro et al. [39] uses the Multistage Stochastic Programming (MSP) framework to shape dynamically a portfolio return composed of complex instruments such as derivatives and exchange-traded funds (ETFs).

Finally, simulation techniques are also used for portfolio management, as in the case of Greyserman et al. [40] with Markov Chain Monte Carlo (MCMC) technique, and Detemple et al. [41] in a realistic environment simulating complex dynamics. The main difference with *EvoFolio* is based on the potential of the MOEAs, through which it is possible to perform VaR-based and Programming-based models.

*Neural Networks based approach.* The increasing computational power has made it possible to use new techniques for analyzing historical series and the optimization process. For example, starting from the multidimensional analysis of the *Support Vector Machine* (SVM), it was possible to use the *Support Vector Regression* (SVR) [42] to predict the prices used in the mean-variance model. The use of neural networks has made it possible to exponentially improve the performance of the models for the allocation of assets in the portfolio. Lin et al. [43] test the effectiveness of the neural networks' use in the dynamic portfolio selection problem, using the Elman network to simulate the dynamics of assets and demonstrating how this method outperforms statistical models. Zimmermann et al. [44] use Feed-Forward Neural Networks (FNNs) for active portfolio management using a mechanism similar to the BL model, obtaining superior results. Ban et al. [45] introduce Performance-Based Regularization (PBR) that constrains the variation of risk and return of assets to include in the portfolio. Sen et al. [46] use a neural network with a Long-Short Term Memory (LSTM) cell to predict asset prices and build an efficient portfolio. Ma et al. [47] combine return prediction with Machine Learning and Deep Learning techniques on the Chinese market from 2007 to 2015, demonstrating how the best portfolio composition can be obtained using the Random Forest.

On the other hand, portfolio optimization lends itself well to a Reinforcement Learning task, where agents change the number of assets based on a reward. Park et al. [48] overcome the limitation of Deep Neural Networks (DNN) by introducing a new framework called Risk-Sensitive MultiAgent Network (RSMAN), which includes Risk-Sensitive Agents (RSAs), demonstrating the feasibility of this approach. Liang et al. [49] use some RL agents, such as Policy Gradient (PG) and Deep Deterministic Policy Gradient (DDPG), to test the best combination of portfolio assets. Alternatively, Koratamaddi et al. [50] propose an RL system for automatic trading on a portfolio that relies on the market sentiment extracted from social media analysis. Guarino et al. [51] propose a neuro-fuzzy agent to decide the best trading strategy, testing it in markets such as cryptocurrency. Betancourt and Chen [52] use a Proximal Policy Optimization (PPO) agent for portfolio management characterized by a market with a dynamic number of assets, such as that of cryptocurrencies, developing an architecture adaptive to the introduction of a new asset.

Differently from our tool, RL agents cannot create a particular portfolio (based on the previous reward) but only a strategy that can replicate the portfolio performance. *EvoFolio* can modify the portfolio composition directly. Furthermore, differently from any of these methods, *EvoFolio* implements the so-called elitism, allowing users to express their views on the market, incorporating their know-how. Lastly, a direct comparison of the performance of these kinds of methods against *EvoFolio* is available in Sect. 6.

*Evolutionary approach.* This approach is based on adapting an individual (belonging to a population) to achieve their goals. Several nature-inspired approaches to portfolio management are used not only to solve the problem of the number of assets to include in the portfolio but also the definition of the constraints to be respected and the choice of which are the best assets to consider. For example, on the MOEAs side, Guennon et al. [53], starting from an initial portfolio, use a GA to classify the assets in the portfolio, obtaining sub-portfolios from which the one that maximizes the return is chosen, using a dynamic optimization algorithm MinVaRMaxVaL. Ranković et al. [54] propose a mean-VaR optimization algorithm in which VaR is estimated using a Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) volatility model, using the NSGA-II algorithm on a sample of 40 US stocks. Lwin et al. [55] still in the mean-VaR framework, propose a learning-guided hybrid Multi-Objective Evolutionary Algorithm (MODE-GL) under real constraints such as the pre-allocated amount of assets in the portfolio, demonstrating the effectiveness in very reasonable calculation times. Pal et al. [11] propose a method for forming initial portfolios based on automatic vertical and horizontal clustering by exploiting a single metric, the returns per unit of risk. Through these portfolios, the authors use the

NSGA-II algorithm to dynamically determine the assets' weights, obtaining Pareto optimal portfolios in a Markowitz-based process and limiting themselves to only one type of market. Mehlawat and Gupta [56] use fuzzy parameters to optimize the portfolio assets based on classic indicators such as return, risk, and liquidity. To solve the problem, the authors use a Real Coded Genetic Algorithm (RCGA) hybridized with fuzzy rules, demonstrating the effectiveness of this approach. Kaucic et al. [57] analyze different portfolio selection strategies according to the investor's risk appetite, comparing them with a modified version of NSGA-II and SPEA II to solve the asset choice problem. The results show how the GAs can outperform the used metrics and determine the Pareto efficient portfolio. Liagkouras [58] solves the problem of the dependence between the required time and size of the test to be examined in GA for combinatorial problems, developing a three-dimensional encoding MOEA for portfolio optimization and testing it on the optimal allocation of portfolio assets. Dreżewski and Doroz [59] use a co-evolutionary algorithm for multi-objective portfolio optimization, comparing it with the classic GA version on the Warsaw Stock Exchange, demonstrating its strengths and weaknesses. Macedo et al. [60] propose incorporating some financial indicators generally used by investors in the NSGA-II and SPEA II algorithms in a framework characterized by only adverse return variations instead of overall variations. With these introductions, the authors demonstrate the effectiveness of the different algorithms. Meghwani and Thakur [61] modify the classic mean-variance problem to incorporate investor preferences, complicating the solution search space and solving the problem with multi-criteria algorithms. The authors propose four MOEAs, such as NSGA-II, SPEA II, GWASFGA, and PESA-II, adapted to the new problem, highlighting their performance.

Inspired, however, by natural mechanisms such as the ants and swarms' movement, Babaei et al. [62] defined the optimization problem as a multi-objective mixed integer programming, in which VaR represents the measure of risk, solving it with two variants of Multi-Objective Particle Swarm Optimization (MOPSO) and comparing the results with NSGA-II and SPEA II. Dangi [63] combines a series of agents to solve the optimization problem by introducing the Super-Agent, based on stochastic optimization mechanisms and swarm solvers. Rezani et al. [64] use the K-Means algorithm as a clustering method to diversify the assets in the portfolio and the ant colony optimization (ACO) method to determine the weight to be attributed to each asset. El-Shorbagy and Hassanien [65] use Particle Swarm Optimization (PSO) to solve various unconstrained optimization problems, including portfolio management, to minimize the risk constraint. Haqiqi and Kazemi [66] test the performance of the ACO in the portfolio optimization problem on the Tehran Stock Exchange, using monthly closing prices as a test set. Steven et al. [67] use the K-Means algorithm to cluster the assets to be included in the portfolio based on some financial statement indicators (converted into scores) and, to optimize the quantity of these in the portfolio, they use an ACO algorithm, highlighting how the choice of the fitness function is a fundamental step.

The main difference with the previous works on the application of GAs to the portfolio optimization problem concerns the optimization timing since, in this paper, we do not limit ourselves only to optimizing the portfolio once[1]. Still, the modification of the assets in the portfolio is done dynamically over time, proposing a system optimized over the weeks/months. Furthermore, to do this, the *EvoFolio* system is equipped with a minimal trading capacity since – in addition to carrying a residual budget – between one optimization and another, it "sells" all the stocks in its possession and re-create one's portfolio. To the best of our knowledge, the techniques used in the literature are different from ours, also having a different problem definition. Moreover, *EvoFolio* pays particular attention to the "feasibility" from a financial point of view of the portfolio composition. The portfolios generated, in fact, respect objectives and constraints that are achievable by the financial investor based on his starting portfolio (understood as budget constraints) and considering the desired level of risk, in addition to the maximum number of stocks that can be regarded as in a typical financial portfolio and any personal preferences in selecting assets. Instead, in the surveyed works, the focus was more on achieving, for example, a shallow risk. Furthermore, unlike any of these evolutionary approaches, *EvoFolio* implements the so-called elitism, allowing users to express their views on the market, incorporating their know-how. Finally, the previous works do not find any comparison with the related works but limit themselves to measuring their portfolio management capabilities. Here, *EvoFolio* was compared to the portfolios composed by FinRL [14] agents.

*Our proposal against gaps of literature* In this paragraph, we provide the highlights of our proposal, *EvoFolio*, with respect to the gaps in the literature discussed in previous paragraphs. From the analysis carried out, it emerges that *(i)* some of the approaches found just provide the strategy to create a portfolio, while others just provide a method to create a portfolio once. *(ii)* Moreover, there is no chance for the final user to interact somehow with the systems to add his/her know-how. *(iii)* Furthermore, most of the

[1] Though, some other approaches that allow multiple and successive portfolios optimizations over time exist [68].

previous works did not consider in their method budget constraints, which are always present in real-world scenarios. *(iv)* In addition, we have not found works sharing their source code for complete reproducibility. *(v)* Lastly, most of the surveyed works did not provide any comparison with their method against others.

In light of these gaps, our highlights are as follows: *(i) EvoFolio* can form and continuously modify the composition of a specific portfolio. *(ii)* Differently from any other previous work, it implements the so-called elitism, which allows the algorithm to give importance to particular portfolios having selected features. The elitism in *EvoFolio* relays on the domain expert's feedback on specific stocks, thus providing a simple way to interact with the machine. *(iii)* Our method can create feasible portfolios based on the user's budget and desired risk. *(iv) EvoFolio*'s outcomes have been compared with those achieved by the FinRL library with its several RL-based methods for portfolio optimization, and checked against the mean-variance model (i.e., the cornerstone method). *(v)* We have released the code of EvoFolio; the link can be found in Sects. 1 and 7.

## 4 EvoFolio

In this section, we provide all information about *EvoFolio*. We start by discussing high-level motivations and an overall description, and then we explain the details (Sect. 4.1). Next, we show the datasets used to conduct the experiments (Sect. 4.2), and offer details on the pseudocode of *EvoFolio* (Sect. 4.3). Moreover, we describe the experiments carried out and the settings tested (Sect. 4.4). Lastly, we show and discuss the results obtained with the best settings (Sect. 5).

### 4.1 Choices and motivations

To develop *EvoFolio*, first, we have chosen a reference MOEA, and then we have customized the strategy to work with specific choices regarding the chromosome representation, the operators, and the fitness function. Thus, the evolutionary process (optimal portfolio selection) is obtained by running the selected algorithm, with the addition of our "*ad hoc*" technical choices.

As explained above, the strategy employed as reference is NSGA-II [10]. We made such a choice for several reasons. From a *practical* point of view, these reasons are: *(i)* its popularity among practitioners and the vast available body of peer-reviewed material about it, *(ii)* its proven efficiency on both benchmarks and real-world engineering problems, and *(iii)* its straightforward algorithmic design and ease of implementation. NSGA-II has the advantage of incorporating techniques that could naturally reflect some financial investment habits: the *elitism* and *diversity maintenance*. Elitism consists of maintaining an external population of all non-dominated solutions; these are individuals with characteristics we wish to keep in the final solution. As we will see, in the specific context of the optimal portfolio selection problem, it is crucial to preserve solutions having specific financial properties and use them during the process. Diversity is a significant aspect of evolutionary multi-objective optimization since it improves the coverage of the search space and allows for the exploration of different evolutionary paths leading to the trade-off surface. Furthermore, in the specific context, the diversity of portfolios used for producing strategies and configurations allows to avoid investment flattening and, so, trying to obtain always new ideas during the process.

Moreover, to better shape the optimal portfolio selection process as an evolutionary process, we have defined a chromosome representation that reflects a portfolio's real-world structure. Then, we have added specialized operators guided by a fitness function built using realistic financial aspects of the portfolio, such as overall yield and overall risk (see Sect. 2.2).

### 4.2 Data collection

To create and experiment with *EvoFolio*, we considered 47 of the most attractive stocks for investors. These shares, sketched in Table 1, are listed on various stock exchanges such as New York Stock Exchange (NYSE, United States), NASDAQ (United States), and Borsa Italiana (Italy, whose stock codes have the wording *".MI"*) and have been selected to avoid excessive correlation between stocks and represent companies operating in different sectors such as automotive, financial, food, tech, aerospace, chemical, pharmaceutical.

The datasets collected refer to the daily stock prices recorded in the three years 2020-2023, processed through the *yfinance*[2] package. In Table 2, there is an extract of the Apple Inc (AAPL) stock dataset. This dataset consists of five fundamental features relating to Opening, High, Low, and Close prices and an indication of the Volume of transactions carried out on that day, and researchers and domain experts usually refer to it with the acronym OHLCV.

### 4.3 Pseudocode

Here, we offer details on the NSGA-II algorithm we have customized for *EvoFolio* (based on the DEAP framework [69]), and in particular, the initialization function, the

[2] https://pypi.org/project/yfinance/.

chromosome and gene representation, the mutation and the crossover operators, and the function implemented for elitism. The description of each component will be carried out by observing the specific use within the algorithm step by step. At the same time, therefore, the pseudocode will also be technically commented, in order to provide details on the entire evolutionary process as well.

**Algorithm 1** The main multi-objective evolutionary algorithm, *EvoFolio*.

```
Input  : $MU$, $TOURNPARAM$, $CXPB$, $indp$,
         $ELITEPARAM$, $SELPARAM$,
         $NGEN$
Output: $pop$
dfstocks ← getAllstocksdatasets();
date ← getDate();
BUDGET ← 1000000;
// initial budget
size ← len(dfstocks);
UserPreferences ← inputGUI();
// get user-preferred stocks through Graphic
   User Interface
pop ← initPop(MU);
// get initial population of portfolios
pop ← ParetoFront(pop, len(pop)*SELPARAM);
// Pareto Front selection based on fitnesses
   values and front rank of individuals in the
   starting population.
for $gen=1$ to $NGEN$ do
    offspring ← TournamentDCD(pop, len(pop) × TOURNPARAM);
    // Tournament selection based on dominance
       between two individuals, if the two
       individuals do not interdominate the
       selection is made based on crowding
       distance.
    for $ind1, ind2$ in $zip(offspring[::2], offspring[1::2])$ do
        if $random.random() \le CXPB$ then
            cxOnePoint($ind1$, $ind2$);
        mutUniformIntAdaptive($ind1$);
        mutUniformIntAdaptive($ind2$);
    elitepop ← genElite(pop, UserPreferences);
    if $1 \le len(elitepop)$ then
        for $ind1$ in $elitepop$ do
            for $ind2$ in $offspring$ do
                if $random.random() \le ELITEPARAM$ then
                    cxOnePoint($ind1$, $ind2$);
        pop ← ParetoFront(pop + offspring + elitepop, MU);
        // Pareto Front selection based on
           fitnesses values and front rank of
           MU individuals.
```

*Chromosome and gene representation* The population in *EvoFolio* is made up of individuals (*chromosomes*) that are portfolios of stocks (see Sect. 2.2). A chromosome, that is an individual of the population, is represented as a

**Table 1** List of stocks considered in the environment

| Company name | Stock code |
|---|---|
| A2A S.p.A | A2A.MI |
| Apple Inc | AAPL |
| Adidas AG | ADS.MI |
| Airbus SE | AIR.MI |
| Amgen Inc | AMGN |
| Amplifon S.p.A | AMP.MI |
| Amazon.com Inc | AMZN |
| American Express | AXP |
| The Boeing Company | BA |
| Banca Mediolanum S.p.A | BMED.MI |
| Caterpillar Inc | CAT |
| CNH Industr. N.V | CNHI.MI |
| Davide Campari-Milano S.p.A | CPR.MI |
| Salesforce Inc | CRM |
| Cisco Systems Inc | CSCO |
| Chevron Corp | CVX |
| The Walt Disney Company | DIS |
| Dow Inc | DOW |
| EssilorLuxottica Société An | EL.MI |
| ENEL S.p.A | ENEL.MI |
| ENI S.p.A | ENI.MI |
| Generali S.p.A | G.MI |
| Alphabet Inc | GOOG |
| Goldman Sachs Group Inc | GS |
| The Home Depot Inc | HD |
| Honeywell International Inc | HON |
| Intern. Business Machine | IBM |
| Intel Corporation | INTC |
| Johnson & Johnson | JNJ |
| JPMorgan Chase & Co | JPM |
| The Coca-Cola Company | KO |
| McDonald’s Corp | MCD |
| Meta Platforms Inc | META |
| 3 M Company | MMM |
| Merck & Co. Inc | MRK |
| Microsoft Corp | MSFT |
| NIKE Inc | NKE |
| The Procter & Gamble Co | PG |
| Stellantis N.V | STLA |
| The Travelers Companies Inc | TRV |
| Tesla Inc | TSLA |
| UniCredit S.p.A | UCG.MI |
| UnitedHealth Group Inc | UNH |
| Visa Inc | V |
| Verizon Communications Inc | VZ |
| Walgreens Boots Alliance Inc | WBA |
| Walmart Inc | WMT |

**Table 2** Excerpt of AAPL dataset

| Date | Open | High | Low | Close | Volume |
|---|---|---|---|---|---|
| 2/1/2020 | 74.06 | 75.15 | 73.80 | 75.09 | 135480400 |
| 3/1/2020 | 74.29 | 75.15 | 74.12 | 74.36 | 146322800 |
| 6/1/2020 | 73.45 | 74.99 | 73.19 | 74.95 | 118387200 |
| 7/1/2020 | 74.96 | 75.22 | 74.37 | 74.60 | 108872000 |
| 8/1/2020 | 74.29 | 76.11 | 74.29 | 75.80 | 132079200 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

sequence of *genes*. Each gene represents the quantity of units purchased of a specific stock. Let $S = A_1, \ldots, A_N$ be the input stocks configuration. Formally, a chromosome $C$ is a portfolio for $S$, i.e., a sequence $C = [G_1, \ldots, G_N]$ where each gene $G_i$ is the quantity of units purchased of the stock $A_i$, for $i = 1, \ldots, N$.

*Main algorithm* The main function's pseudocode is available in Algorithm 1. It takes as input all the parameters impacting on the optimization, i.e. the size of the population *MU*, the percentage of portfolios to keep after the tournament *TOURNPARAM*, the probability of a crossover, mutation, and elitism *CXPB*, *indpb*, and *ELITEPARAM*, respectively. Lastly, it also takes as input the percentage of individuals to keep after the ParetoFront, *SELPARAM*, and the number of generations, *NGEN*. First, the function retrieves all the data from the datasets folder, gets the first date for tests, and sets the budget; then, it asks the user for his/her views on the market (lines 1:6). Next, initialize the population and select the top *SELPARAM%* elements for further processing (lines 8:10). For every generation, it selects the best individuals with the TournamentDCD function putting them in the offspring population. Such a new population will be used for the mutation and crossover operations (lines 13:19), as well as for the selection of elite individuals (lines 20:25). Lastly, the new population is then chosen by the Pareto Front method and composed of the best *MU* individuals among the initial population, the offspring one, and the elite one (line 26).

For initializing the population, we employ the Algorithm 2. With regards to crossover and mutation, we refer the reader to Algorithm 4 and Algorithm 3, respectively. For elitism, we employ the approach in Algorithm 5.

*The initialization algorithm.* This function's pseudocode is available in Algorithm 2. It takes as input the number of stocks to handle (which is the size of portfolios/individuals), the date (to find the exact row in the stocks datasets), and the budget. It creates a valid individual (i.e., a portfolio) within the budget constraints (see line 3) and follows the suggestions of [11, 12] with 10 to 14 diverse instruments (see line 6). The individual is formed by buying a randomly chosen quantity (line 22) of one kind of stock (randomly chosen – see line 10) at a time. The price of the stock is computed through the subfunction `costOfindividual()` as the average price between the "low" and the "high" (see an example of dataset in Table 2) registered for the specific date. Notice that the `costOfindividual()` function can also be used to compute the price considering only the "low" column or only "high" one.

**Algorithm 2** Initialization function pseudocode (`InitPop`).

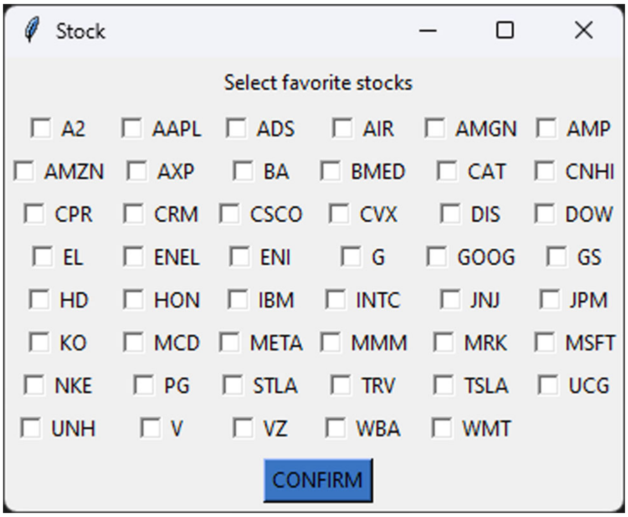
```
   Input  : size, date, BUDGET
   Output: ind
 1 // Customized generator of individual. Checks
      for its validity based on budget's
      constraints.
   // :param size: max portfolio size (number of
      stocks).
   // :param date: date.
   // :param BUDGET: money available.
   // :returns: one individual.
 2 maxbudg ← BUDGET+1;
 3 while maxbudg > BUDGET do
 4     ind ← [0 for i in range(size)];
 5     // initialization of portfolio
 6     numstock ← random.randint(10,14);
 7     stockused ← [];
 8     currentbudg ← BUDGET;
 9     for i=1 to numstock do
10         stock ← random.randint(0,size-1);
11         // select random no. of stocks to buy
12         while stock in stockused do
13             stock ← random.randint(0,size-1);
14             // 'stock' has already been used, so
                  try again with another random
                  stock
15         stockused.append(stock);
16         // 'stock' has been used, so it adds
              stock to the list of used stock
17         h ← stockdf[stock]["High"][date];
18         l ← stockdf[stock]["Low"][date];
19         upperbound ←
            int(currentbudg/avg(h,l));
20         // max no. of stocks of type 'stock'
              purchasable with the current budget
21         if upperbound>1 then
22             ind[stock] ←
                random.randint(low+1,upperbound);
23             // random number of 'stock' to buy
24             currentbudg ←
                BUDGET-costOfindividual(ind);
25             // update current money available
26     maxbudg ← costOfindividual(stockdf,ind);
27     // checks whether the portfolio value is
          too high
28 return ind;
```

*The mutation operator.* This function's pseudocode is available in Algorithm 3. It takes as input the individual (i.e., the portfolio) to mutate, the lower and upper bound for the range from which to draw the new integer value (which is the quantity of specific stock to buy), the probability for each stock in the portfolio to be mutated, the dataframe comprising all stocks information, and the date when the mutation occurs. It changes the portfolio given as input by buying a feasible number of stocks (line 9) for a series of stocks chosen based on the parameter *indpb* (line 8). The upper bound is adapted to the price of each stock (see lines 4-6). The price is computed with `costOfStock()` function as the average between the "low" and the "high" (see an example of the dataset in Table 2) registered for the specific date. Notice that the `costOfStock()` function can also be used to compute the price considering only the "low" column or only "high" one.

**Algorithm 3** Mutation function's pseudocode. Function is named *mutUniformIntAdaptive*.

```
   Input  : $individual, low, up, indpb, dfstocks, date$
   Output: $individual$
1  // :param individual: a portfolio to be
      mutated.
   // :param low: The lower bound of the range
      from which to draw the new integer.
   // :param up: The upper bound of the range
      from which to draw the new integer.
   // :param indpb: Independent probability for
      each attribute to be mutated.
   // :param stockdf: List of stocks and stock
      prices (classic csv data from
      yahoo!finance)
   // :param date: date when the mutation occurs
   // :returns: one individual.
2  size ← len(individual);
3  for $i = 1$ to $size$ do
4      price ← costOfStock($dfstocks$[i], $date$);
5      up1 ← int($up$/price);
6      uplist.append(up1);
7  for $i$, $xl$, $xu$ in $zip(range(size), low, uplist)$ do
8      if $random.random() < indpb$ then
9          individual[i] ← random.randint($xl$,
            $xu$);
10 return individual;
```

*The crossover operator.* This function's pseudocode is available in Algorithm 4. It takes two individuals as input, randomly selects the crossover point (line 3), and then crosses the two individuals (line 4), generating two children which will be returned as a result. One child will be created, including the first part of the first individual (from an element in position 0 to the one preceding the crossover point) and the second part of the second individual (from an element in the crossover point to the end of the portfolio).



□ A2 □ AAPL □ ADS □ AIR □ AMGN □ AMP
□ AMZN □ AXP □ BA □ BMED □ CAT □ CNHI
□ CPR □ CRM □ CSCO □ CVX □ DIS □ DOW
□ EL □ ENEL □ ENI □ G □ GOOG □ GS
□ HD □ HON □ IBM □ INTC □ JNJ □ JPM
□ KO □ MCD □ META □ MMM □ MRK □ MSFT
□ NKE □ PG □ STLA □ TRV □ TSLA □ UCG
□ UNH □ V □ VZ □ WBA □ WMT

**Fig. 2** *EvoFolio*: User Interface to select favorite stocks

**Algorithm 4** Crossover function's pseudocode. Function is named *cxOnePoint*

```
  Input  : $ind1, ind2$
  Output: $ind1, ind2$
1 // Executes a one point crossover on the input
     individuals / portfolios. The two
     individuals are modified in place. The
     resulting individuals will respectively
     have the length of the other.
  // :param ind1: The first individual
     participating in the crossover.
  // :param ind2: The second individual
     participating in the crossover.
  // :returns: two individuals.
2 size ← min(len(ind1), len(ind2));
3 cxpoint ← random.randint(1, size - 1);
4 ind1[cxpoint:], ind2[cxpoint:] ←
   ind2[cxpoint:], ind1[cxpoint:];
5 return ind1, ind2;
```

*Elite population.* In *EvoFolio*, we have implemented the possibility for users to "share" their views on the market with the tool (somehow imbuing the perspectives disclosed by the Human-Centered AI [13]). In this first version, we provide the User Interface available in Fig. 2 implemented with Python `tkinter` library.[3] All the stocks checked here are the users' "preferred" ones. We assume that users' views are like "predictions" on the market, therefore

[3] https://docs.python.org/3/library/tkinter.html.

*EvoFolio* will likely build portfolios having the indicated instruments. The algorithm designed for elitism (see Algorithm 5) selects and returns the portfolios that have the highest number of preferred stocks bought. These will be those that the main algorithm will use for subsequent crossover operations.

**Algorithm 5** Function to select preferred individuals in the population to implement elitism.

```
Input  : pop, pref
Output: indliked
// Returns a list of individuals (elite
   population) having multiple preferred
   stocks
// :param pop: population of portfolios
// :param pref: list of user preferred stocks
// :returns: list of preffered individuals in
   pop (the ones having maximum quantity of
   preferred stocks)
indliked ← [];
for  i=1 to len(pref) do
    maxind ← [];
    if pref[i]==1 then
        indmaxazioni ← 0;
        for ind in pop do
            if ind[i] > indmaxazioni then
                indmaxazioni ← ind[i];
                maxind ← [];
                maxind ← ind;
            if ind[i] == indmaxazioni then
                maxind.append(ind);
        indliked.extend(maxind);
return indliked;
```

## 4.4 Methods and settings

We recall that *EvoFolio* has been implemented through the Python DEAP[4] library [69].

We have performed a series of experiments to find the best configuration for our *EvoFolio*. We have used *(i)* standard operators for NSGA-II, *(ii)* DEAP suggested ones, and *(iii)* customized ones (for which we refer the readers to Sect. 4.3). For what concerns the standard operators we refer to mutation with a random uniform function (named `mutUniformInt`), and crossover with two points (named `cxTwoPoints`). Instead, for DEAP suggested ones we refer to polynomial mutation (named `mutPolynomialBounded`) and simulated binary crossover (named `cxSimulatedBinaryBounded`).

Furthermore, we have searched in a grid of parameters for such algorithms. The parameters, the range of values experimented, and the best ones found are available in Table 3.

To search for the best parameters, we have performed experiments on a 6 months time-window. We have found that the optimal population size was 250, the optimal probability for crossover (`CXPB`) was 80%, while the one for mutation (`indpb`) was 2%. The best number of generations may vary in different periods but in general terms 250 generation was often the finding. `SELPARAM`, `ELITEPARAM` and `TOURNPARAM` were found at 80%, 30%, and 90%, respectively.

Lastly, the best portfolios were found with the operators seen in the previous sections, i.e., Algorithms 3-4, with Algorithm 5.

All experiments have been carried out on an Intel i7 quad-core machine equipped with 16GB RAM.

# 5 Results

Once defined the key parts of *EvoFolio* method, in this section, we test it and show the results achieved on the set of stocks previously indicated (Table 1) as a universe, so that *EvoFolio* can determine the optimal composition of portfolio based on the defined objectives, testing it in two timings: stock prices on a monthly (Sect. 5.1) and quarterly basis (Sect. 5.2) with and without preferences. In particular, the method is constructed in such a way that preferences can be expressed in any period (every month in the monthly and every quarter in the quarterly); but, to test its capabilities, in these experiments, we have opted for a constant choice of stocks (the views are provided "*a priori*" and held for all the experimentation time-frame). The underlying idea is that the system, with correct views, can provide even better results. In both cases, the starting budget is 1,000,000 US Dollars ($).

As a default choice, *EvoFolio* performs incremental optimizations over time referring to the value, in US Dollars, of the first ranked portfolio in the Pareto Front. Note that this choice allows real usability of *EvoFolio* because it ensures the investor changes the assets in the portfolio using only the value produced by the sale of those currently owned without borrowing money from external sources. This condition, coming from the literature and always preferable for the investor, is verified with this algorithm and is based on the absence of transaction costs (on the purchase and, especially, on the sale of portfolio stocks for the subsequent acquisition of new ones).

## 5.1 Monthly optimization

This section analyzes the results obtained when forming the optimal portfolio every month. With this timing, *EvoFolio*, like a real financial investor, updates the assets in the portfolio monthly, possibly changing the stocks to be

[4] https://deap.readthedocs.io.

**Table 3** Parameters tuned and the search range

| Parameter | Range | Step | Best |
|---|---|---|---|
| `MU`* | [100, 500] | 50 | 250 |
| `TOURNPARAM` | [0.2, 0.9] | 0.1 | 0.9 |
| `SELPARAM` | [0.2, 0.9] | 0.1 | 0.8 |
| `CXPB`** | [0.3, 0.9] | 0.1 | 0.8 |
| `ELITEPARAM` | [0.1, 0.5] | 0.1 | 0.3 |
| `indpb`*** | [0.02, 0.1] | 0.02 | 0.02 |
| `NGEN`**** | [50, 500] | 50 | 250 |

* Population size. ** Crossover probability. *** Mutation probability. **** Number of generations

included in the portfolio and the quantity of these (in proportion to the gain (in US Dollars) obtained during the previous period).

*Without preferences.* In the views' absence on the assets, the initial combination of stocks is shown in Table 4, corresponding to the portfolio ranked first in Pareto Front to be optimized in the following months. However, *EvoFolio* has preferred to keep this portfolio constant in all periods. The first evidence concerns the presence of four Italian stocks in the portfolio, which should not be underestimated given the minority in the universe of stocks known to *EvoFolio*. In particular, almost all of the capital was invested in the A2A.MI stock that, for example on 02/01/2020 had a price of 1.48, corresponding to an investment of almost 795, 500 US Dollars ($).

The second evidence, on the other hand, can be obtained by observing the trend over time of the portfolio value generated. In particular, Fig. 3 represents the average risk and return of the portfolio, while Fig. 4 its value (in monetary terms) over time.

The average monthly yield (green line), although in absolute value dependent on the logarithmic scale (generally used for the calculation), has settled over time, settling at around 5,130. On the other hand, the average portfolio risk (red line) is mainly used as an indicator by investors. It shows a substantial reduction over time, despite the decision to leave the assets in the portfolio unchanged. However, this portfolio's characteristic is the risk value achieved: 0.6% on average. The optimal choice of shares has allowed *EvoFolio* to generate a portfolio with a monthly risk far lower than that generally recorded by individual stocks. This risk gradually decreases over time, starting from 2% until stabilizing at the average value.

Observing Fig. 4, on the other hand, it is evident how the indications on risk and yield have found a counterpart in the value of the portfolio. In fact, for the first few months, decreasing yield (corresponding to an increase in risk) led to a loss in USD until around May 2020 (a year characterized by the pandemic outbreak and critical situations on all markets).

Subsequently, stabilizing risk and yield meant that the portfolio's value depended exclusively on market variations, causing months of losses to follow months of gain. The characteristic of this result (very popular among investors), however, is that in the months following the initial ones characterized by higher volatility (including yields), the changes in the portfolio's value were all very similar, regardless of whether these were positive or negative.

*With preferences.* Now let us consider the investor's preferences in choosing assets. To test *EvoFolio*, these views were entered considering stocks whose positive trend was known prior. In particular, the selected assets are AAPL, AMGN, AXP, CNHI.MI, GS, HON, MSFT. As in the previous case without preferences, *EvoFolio* maintains the portfolio ranked first in Pareto Front, also shown in Table 4 in row "*With*", constantly over time which contains three out of the seven preferred stocks (i.e., AAPL, GS, HON). In this way, the custom NSGA-II chose the stocks that, together with the others in the known universe, would best optimize the constraints based on the preference. We can see how using a preference has changed the risk and yield of the portfolio in Fig. 5 and the corresponding value in Fig. 6.

**Table 4** Different portfolios (ranked first in Pareto Front) optimized throughout the time-window from 2020 to end of 2022 (monthly optimization) with and without domain expert's preferences

| Portfolios | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Expert's preferences | | | | | | | | | | | |
| Without | 1st month | A2A.MI | AIR.MI | AMP.MI | AXP | GOOG | MMM | MRK | TSLA | UCG.MI | VZ |
| | | 537533 | 1 | 1 | 1 | 1 | 1 | 484 | 1 | 1 | 1 |
| With | 1st month | AAPL | AMP | CAT | CVX | GS | HON | IBM | MRK | PG | TRV |
| | | 1 | 1 | 1 | 2226 | 1 | 1 | 1 | 1 | 1 | 1 |

The preferences expressed have been: AAPL, AMGN, AXP, CNHI.MI, GS, HON, MSFT

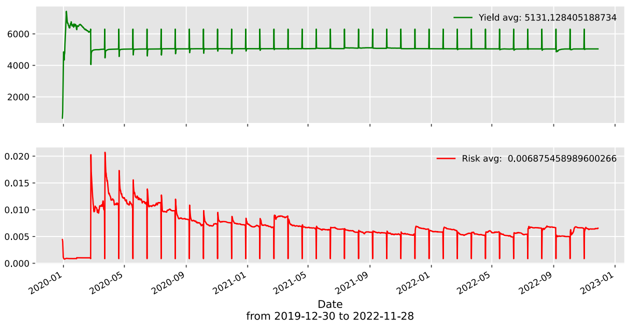


**Fig. 3** Average yield and risk (fitness values) obtained by the portfolios throughout the time-window from 2020 to 2022 with a monthly optimization. Here, no stocks preferences have been provided

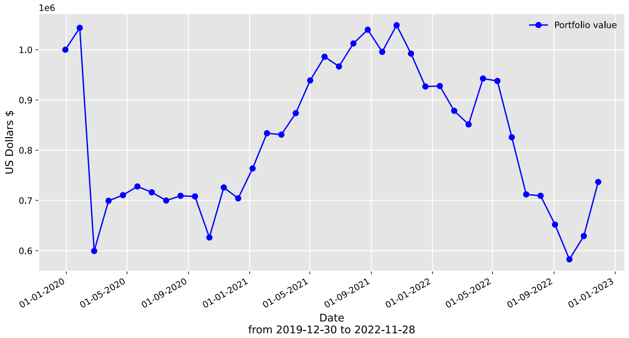


**Fig. 4** Value (in USD) of the first ranked (Pareto Front ranking) portfolio in the time-frame from 2020 to 2022 (monthly optimization). Here, no stock preferences have been provided

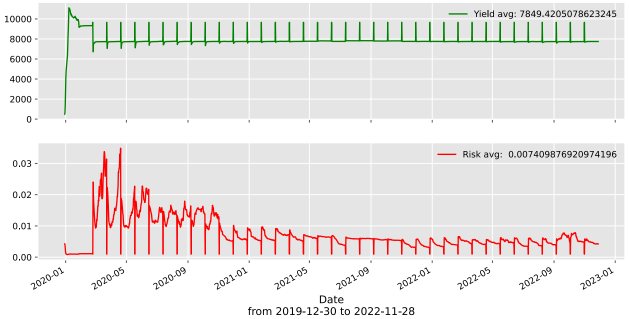


**Fig. 5** Average yield and risk (fitness values) obtained by the portfolios throughout the time-window from 2020 to 2022 with a monthly optimization. Here, the domain expert has expressed the following preferences: AAPL, AMGN, AXP, CNHI.MI, GS, HON, MSFT

For example, as shown in Fig. 5, the average monthly risk is higher than in the portfolio built by *EvoFolio* without the preferences (going from 0.6% to 0.7%) and, consequently, also the average yield stands at a higher absolute value (for the risk-return relationship). This portfolio maintained a high monthly risk for several months longer than in the previous case, settling toward the end of 2020 instead of in the middle of the same year, and a higher maximum reached (above 3%).

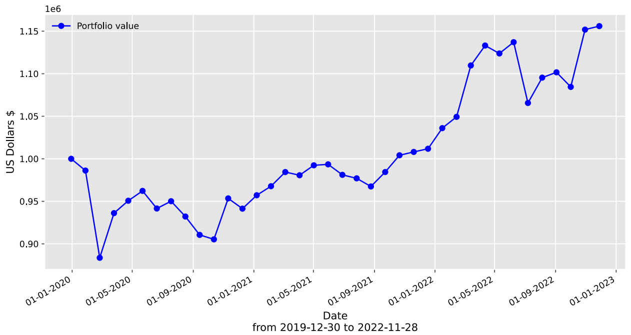


**Fig. 6** Value (in USD) of the first ranked (Pareto Front ranking) portfolio in the time-frame from 2020 to 2022 (monthly optimization). Here, the domain expert has expressed the following preferences: AAPL, AMGN, AXP, CNHI.MI, GS, HON, MSFT

Analyzing the equivalent value in USD, however, as shown in Fig. 6, the investor's views made it possible to obtain a return of $150,000$$ at the beginning of 2023 (compared to the loss obtained without preferences equal to approximately $250,000$$). Furthermore, the optimal allocation obtained by "forcing" *EvoFolio* to "keep an eye" the preferential stocks followed an upward trend until March 2022, as in the competitor case. Still, beyond this period, it was not affected by the downward trend, obtaining a higher return on capital invested at the beginning of the period.

## 5.2 Quarterly optimization

This section analyzes the results obtained when forming the optimal portfolio every quarter. Also, in this case, the assets are updated in proportion to the profit recorded in each quarter.

*Without preferences.* The first case considered is without investor preferences. This time, again in 2020-2023, *EvoFolio* changed the composition of the ranked first portfolios to Pareto Front, as shown in Table 5. In particular, the changes were made to the 5th quarter until May 2021, when the portfolio remained unchanged until the last optimization. Characteristic was the shift of the most significant weight in the portfolio from a biotech company (AMGN), due to high volatility for vaccine news, to a tech company (TSLA) characterized by a substantial increase from the beginning of 2022, beyond the elimination of JNJ stock from the portfolio.

Fig. 7 depicts the development of risk and yield during this type of quarterly optimization. It would not be correct to directly compare these average values with those obtained through monthly optimization, as the ranges of

**Table 5** Different portfolios (ranked first in Pareto Front) optimized throughout the time-window from 2020 to end of 2022 (quarterly optimization)

| Portfolios | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $1^{st}$ quarter | AAPL | AMGN | AMZN | CPR.MI | CVX | INTC | JNJ | META | UNH | WMT |
| | 1 | 1475 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| $2^{st}$ quarter | AAPL | AMGN | AMZN | CPR.MI | CVX | INTC | JNJ | META | UNH | WMT |
| | 1 | 1475 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| $3^{st}$ quarter | AAPL | AMGN | AMZN | CPR.MI | CVX | INTC | JNJ | META | UNH | WMT |
| | 1 | 1475 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| $4^{st}$ quarter | AAPL | AMGN | AMZN | CPR.MI | CVX | INTC | JNJ | META | UNH | WMT |
| | 1 | 1475 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| $5^{st}$ quarter | AMZN | BA | CRM | CSCO | CVX | HD | INTC | TSLA | UNH | WMT |
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4474 | 1 | 1 |

Here, no stock preferences have been provided

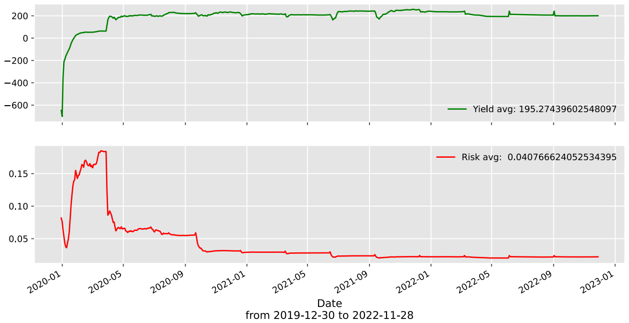


**Fig. 7** Average yield and risk (fitness values) obtained by the portfolios throughout the time-window from 2020 to 2022 with a quarterly optimization. Here, no stock preferences have been provided

variation here are much narrower for yield and wider for volatility.

It is evident how, in absolute value, *EvoFolio* managed to obtain a lower return (compared to the monthly one) but has proved to be constant since the first modification of the assets in the portfolio, reaching an average value of 195. On the other hand, the risk moves on a higher range of variation since the prices of shares can fluctuate more efficiently over a quarter. This peaked at almost 20% toward the end of the first quarter, decreasing significantly and remaining constant. On average, the portfolio's risk built by *EvoFolio* reached a value of 4%.

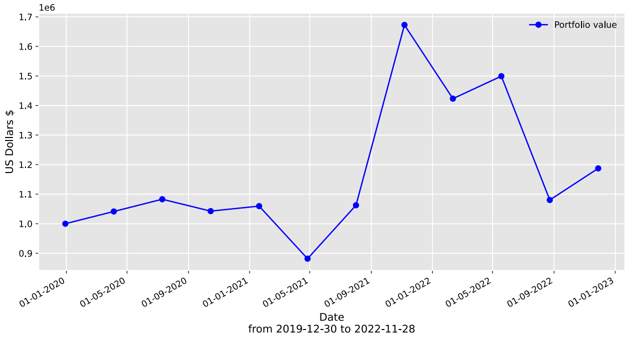


**Fig. 8** Value (in USD) of the first ranked (Pareto Front ranking) portfolio in the time-frame from 2020 to 2022 (quarterly optimization). Here, no stocks preferences have been provided

Regarding the portfolio's value in USD, Fig. 8 illustrates its trend. Since this is an optimization over a more extended period (3 months), the fluctuations in the value are certainly more extensive and frequent than in the monthly case, as shown by the average risk level of 4%. However, *EvoFolio* managed to keep the value of the portfolio reasonably stable over time until September 2021, a period from which it obtained a very high excess return which was gradually lost. This is undoubtedly due to the news that affected the various companies during the pandemic, which led (over the 3 months of observation) to high variations in the recorded price levels. At the end of the management period (January 2023), *EvoFolio* still brought home a return of $200,000\$$ on the initial investment.

*With preferences.* Finally, also for the quarterly case, the investor's preferences were AAPL, AMGN, AXP, CNHI.MI, GS, HON, MSFT stocks. Table 6 shows the changes that occurred in the portfolios ranked first in Pareto Front, from which it is evident that the last change was made in the 6th quarter (August 2021), to then keep the portfolio unchanged. Compared to the first quarter, as early as the third quarter, there were changes in the quantities invested in tech sector stocks (AMZN and TSLA), to the advantage of the biotech sector (AMGN) also according to the views entered by the investor. In the latest change, however, *EvoFolio* concentrated almost the entire portfolio on stocks of the Italian market (ENEL.MI and CPR.MI) which, in those periods, were driven by strong growth.

Looking instead at the average risk and yield of the portfolios in Fig. 9, compared to the case without preferences, the average yield is higher in absolute value (299

**Table 6** Different portfolios (ranked first in Pareto Front) optimized throughout the time-window from 2020 to the end of 2022 (quarterly optimization)

| Portfolios | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1st quarter | AMGN | AMZN | CNHI.MI | CRM | GS | HD | MRK | MSFT | PG | TSLA |
| | 1 | 5893 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 15747 |
| 2nd quarter | AMGN | AMZN | CNHI.MI | CRM | GS | HD | MRK | MSFT | PG | TSLA |
| | 1 | 5893 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 15747 |
| 3rd quarter | AMGN | AMZN | BMED.MI | CRM | EL.MI | INTC | PG | TSLA | VZ | WMT |
| | 2071 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4th quarter | AMGN | AMZN | BMED.MI | CRM | EL.MI | INTC | PG | TSLA | VZ | WMT |
| | 2071 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5th quarter | G | GS | HON | INTC | JNJ | MMM | MSFT | PG | VZ | WMT |
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 17764 | 1 |
| 6th quarter | AMGN | AXP | CPR.MI | EL.MI | ENEL.MI | GS | KO | PG | UNH | VZ |
| | 1 | 1 | 61618 | 1 | 135341 | 1 | 1 | 1 | 1 | 1 |

Here, the domain expert has expressed the following preferences: AAPL, AMGN, AXP, CNHI.MI, GS, HON, MSFT

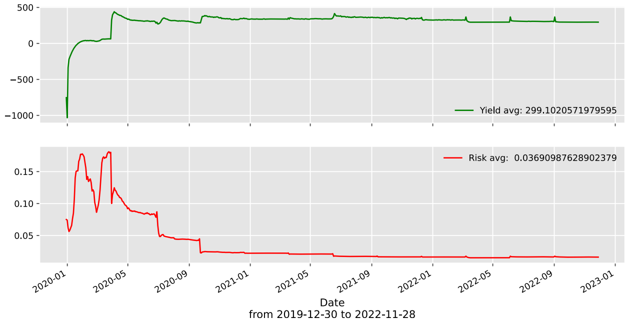


**Fig. 9** Average yield and risk (fitness values) obtained by the portfolios throughout the time-window from 2020 to 2022 with a quarterly optimization. Here, the domain expert has expressed the following preferences: AAPL, AMGN, AXP, CNHI.MI, GS, HON, MSFT

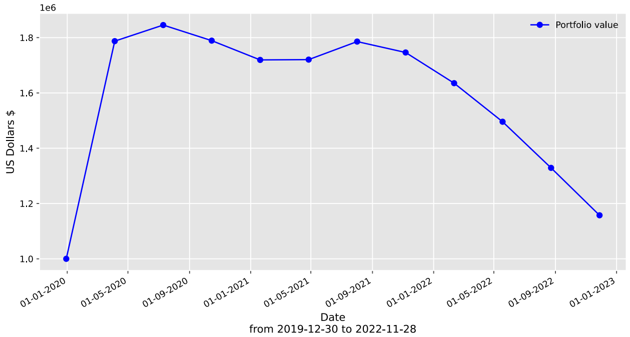


**Fig. 10** Value (in USD) of the first ranked (Pareto Front ranking) portfolio in the time-frame from 2020 to 2022 (quarterly optimization). Here, the domain expert has expressed the following preferences: AAPL, AMGN, AXP, CNHI.MI, GS, HON, MSFT

now), and the moderate risk is lower (3%, compared with 4% of the previous case). However, the suggestion of preferred stocks led to the achievement of two very high peaks in the portfolio's overall riskiness, both at around 15% between January and May 2020. Despite the reduction over time, the same is repeated between July and October 2020, with risk values around 10% and 5%. Since the last change in portfolio stocks, the average risk has dropped to a constant level.

Finally, the situation is different for the USD equivalent of the portfolio. Figure 10 shows how the first combination of stocks paid off, earning *EvoFolio* over 800, 000$ after two quarters (likely related to high risk).

Subsequently, the situation worsened, bringing the portfolio value, at the beginning of 2023, with a profit of 200, 000$ characterized by the gradual reduction in value. Also, in this case, stabilizing risk and return at a certain value has made it possible to obtain fairly constant decreases in value without peaks.

## 6 Discussion

Analyzing the behavior of *EvoFolio* from the monthly and quarterly optimization, it is clear how the strategies pursued have led to excellent results. Not only from the point of view of the equivalent value in USD but also in terms of risk reduction of the entire portfolio. For example, in the quarterly case (more common among investors), a 4% risk on an all-equity portfolio is optimal, especially with regard to the historical period crossed by the Covid-19 pandemic.

The advantage of having limited movements in the value of the portfolio from one period to another is related to the choices that the investor can make. Even if the yield (portfolio value) has decreased from one period to the next, the investor can imagine how much his potential exposure could be, possibly covering the portfolio with other forms of investment (e.g., options, futures, ...). In this way, the knowledge of the average risk in the period considered allows the creation of a financial strategy.

In the remaining part of this section, we provide:

- a test of the *EvoFolio* effectiveness compared to the mean-variance model (Sect. 6.1);
- a comparison of *EvoFolio* with FinRL agents [14] (Sect. 6.2).

### 6.1 Effectiveness of EvoFolio

The Markowitz model (or mean-variance) is still one of the most used methods to determine the optimal portfolio composition. We can test the different portfolios created by *EvoFolio* with the portfolio optimization cornerstone model by measuring performance in terms of Sharpe ratio, considering the portfolios obtained through the Markowitz model as benchmarks. We have maintained the same periodicity expected for the *EvoFolio* portfolios (monthly and quarterly optimization with and without preferences) and, in the different cases, the only value represented both for the benchmark and *EvoFolio* represents the average of the different Sharpe, annual volatility (different from the volatility of the previous sections because based on returns with different portfolio constructions), and expected return values obtained. Notice that the mean-variance model returns a different portfolio composition for each optimization period. In this case, we do not want to underline which stocks have been chosen but rather the profitability (measured by the Sharpe ratio) generated by these portfolios and compared with *EvoFolio*'s ones. Furthermore, for the Markowitz model, we should consider the entire universe of stocks (as indicated in Table1) from which to choose for the composition of the different portfolios. However, since EvoFolio has the characteristic of choosing a maximum of 10 stocks, there will be two benchmarks for comparison: $Benchmark_U$ containing reference values for the entire universe of stocks and $Benchmarks_S$ (or small) in which the mean-variance model has been applied considering only the assets chosen by *EvoFolio* at each optimization. All information is obtained using the Python PyPortfolioOpt package,[5] leaving the main parameters unchanged, such as the risk-free rate $r_f = 0.02$.

Table 7 highlights comparing portfolios in the case of monthly optimization, with and without investor preferences. The first evidence (also present in the subsequent comparison) is that the values of the $Benchmark_U$ remain unchanged in the two cases because the benchmark is constructed by drawing directly on the entire universe of stocks. In the monthly case, *EvoFolio*'s performance is very close to the mean-variance benchmark while still remaining slightly below, while the volatility of GA-based portfolios is higher, evidence also confirmed by the performance (in monetary terms) that the *EvoFolio* portfolios have achieved.

Table 8, however, comparing portfolios in the case of quarter optimization, also with and without investor preferences. Here, it is evident how the different portfolio combinations created by *EvoFolio* outperform both benchmarks. In particular, considering the investor's preferences, *EvoFolio*'s Sharpe ratio is significantly higher than competing portfolios while maintaining very low volatility.

### 6.2 EvoFolio vs FinRL

Finally, in this section, we compare the performance of *EvoFolio* with respect to FinRL [14] agents.[6] in terms of portfolio value. The framework includes several reinforcement learning-based agents, namely A2C [70], DDPG [71], TD3 [72], PPO [73], SAC [74]. Notice that other works in the literature did not compare their proposal with others.

The results of FinRL agents are reported in Fig. 11. Such agents have been experimented with in the same time-frame (2020-2022) of *EvoFolio*, on the same dataset (i.e., 47 instruments in Table 1). However, in this case, we had to offer a training set to the agents, and we selected the instruments' trends in 2019. In the figure, we have reported the portfolio values for A2C in blue, PPO in red, TD3 in teal, DDPG in plum, and SAC in yellow color. Lines' style (solid, dotted, dashed, etc.) has been diversified between the agents. Graphically, FinRL's agents performed very similar strategies, differing only in the absolute value recorded in USD.

We observe that the RL agents have obtained a viable strategy based on a previous training period which, albeit small to minimize the gap between FinRL and *EvoFolio*, is present, unlike *EvoFolio* which, instead, is based only on MOEAs. Furthermore, differently from *EvoFolio*, these agents cannot choose which stocks used for training are preferable (to build a winning trading strategy) but only allow them to obtain a strategy that leads to a specific type of income. In addition, notice that the use of these agents "mattifies" the risk associated with the adopted strategy. It is not known by an actual user, unlike *EvoFolio* where the risk is one of the objectives and it is shown in association

[5] https://pypi.org/project/pyportfolioopt/.

[6] Financial reinforcement learning (FinRL) is the first open-source framework for financial reinforcement learning.

**Table 7** Comparison (in terms of Sharpe ratio) of different portfolios based on monthly optimization

| Monthly optimization | | | | |
|---|---|---|---|---|
| Without preferences | | Sharpe ratio | Annual volatility | Expected return |
| | *EvoFolio* | 0.34 | 26.8% | 9.2% |
| | $Benchmark_U$ | 0.27 | 23.1% | 8.3% |
| | $Benchmark_S$ | 0.39 | 24.5% | 11.6% |
| With preferences | | Sharpe ratio | Annual volatility | Expected return |
| | *EvoFolio* | 0.52 | 39.5% | 20.7% |
| | $Benchmark_U$ | 0.27 | 23.1% | 8.3% |
| | $Benchmark_S$ | 0.59 | 24.1% | 16.1% |

$Benchmark_U$ and $Benchmark_S$ refer to the set from which to extract the portfolio stocks

**Table 8** Comparison (in terms of Sharpe ratio) of different portfolios based on quarterly optimization

| Quarterly optimization | | | | |
|---|---|---|---|---|
| Without preferences | | Sharpe ratio | Annual volatility | Expected return |
| | *EvoFolio* | 0.28 | 27.1% | 9.9% |
| | $Benchmark_U$ | 0.20 | 23.0% | 6.3% |
| | $Benchmark_S$ | 0.26 | 29.8% | 8.6% |
| With preferences | | Sharpe ratio | Annual volatility | Expected return |
| | *EvoFolio* | 1.64 | 18.1% | 29.5% |
| | $Benchmark_U$ | 0.20 | 23.0% | 6.3% |
| | $Benchmark_S$ | 1.06 | 29.6% | 28.2% |

$Benchmark_U$ and $Benchmark_S$ refer to the set from which to extract the portfolio stocks

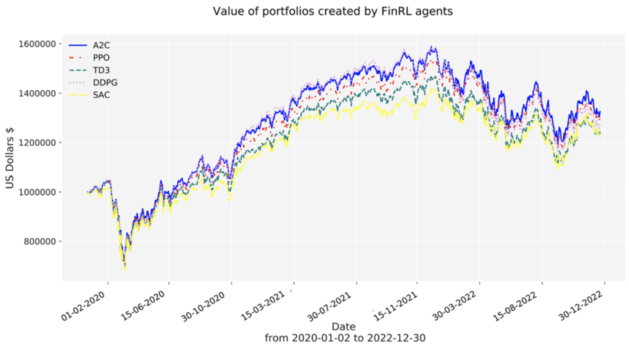


**Fig. 11** Value (in USD) of the portfolios composed by the different FinRL agents, i.e., A2C, TD3, SAC, PPO, DDPG, in the time-frame from 2020 to 2022

with the portfolio composed, what is the risk taken to obtain such a (viable) strategy.

In light of the above considerations, we can compare FinRL's trading strategies with the USD value of the portfolios created by *EvoFolio*, with respect to the time intervals 2020-2022 and 2022-2023.

In the first period, RL agents suffered the effect of the Covid-19 pandemic (losing more than 20% of the initial investment value), to then stabilize toward the fourth quarter of 2020. In the same period, however, the portfolios built by *EvoFolio* and Monthly-optimized fared similarly, with attempts to recover into 2021. Those optimized every quarter, instead, completely outperformed the FinRL traders, obtaining a yield that proved far superior in the presence of views. In the following months of 2021, until the end of this year, RL agents significantly increased the value of the strategy, reaching peaks of $1,600,000$\$. *EvoFolio* portfolios, on the other hand, have followed the same growth trend, most evident in the case of monthly and quarterly no-preference optimization. In the quarterly case with views, the trend was slightly down while maintaining the portfolio's value at around $1,800,000$\$, higher than that recorded by FinRL agents. The situation was reversed in the second time frame, starting in 2022. In this period, FinRL's agents recorded a trend that alternates positive and negative changes, in which the strategy, on average, allowed them to earn $1,300,000$\$. Conversely, the *EvoFolio* portfolios had a steep downward trend in the case of the quarterly optimization with preferences. The USD portfolio value could not overcome the RL agents' strategy in the remaining cases despite a positive trend.

In conclusion, *EvoFolio* has proved to be up to expectations and capable of outperforming (over several periods) agents whose behavior is provided by very deep training strategies, especially with portfolios built based on investor preferences, making it a potentially ideal tool not only for the ability to indicate to the investor the quantity of stocks to include in the portfolio, but also able to process his/her views.

Last but not least, the experience made here allowed us to reflect on the adequate trade-off between *exploration* and *exploitation* to solve the multi-objective optimization problem. The way FinRL and *EvoFolio* tackle it is profoundly different since, on the one hand, RL agents try to learn/estimate during the training process the value of a specific action in a particular state (*"how much do I gain if I do action-x in this state-s?"*), while, on the other hand, GAs bypass such an aspect by stochastically applying operators and selecting the best individuals/solutions to keeping in throughout the generations. Therefore, a question emerges: *"What is the best approach for the type portfolio optimization?"* In a realistic context, approaches like the one proposed here are less complex and do not overweight the model significantly in step with the complexity's increase. By adding more stocks to take into account, more portfolios, and so on, the RL model will overgrow (think about Q-learning methods that, in turn with the increasing of possible states/actions, become Deep Q-learning since a Q-table is unfeasible), while our *EvoFolio* will remain substantially the same.

## 7 Conclusion

In this paper, we have proposed a method based on Multi-Objective Evolutionary Algorithms (and in particular NSGA-II) for optimal portfolio selection, named *EvoFolio*. The method aims at maximizing the returns/yield and minimizing the risk. It has been tested in a very challenging period for investors, i.e., from 2020 to the end of 2022 (including the COVID-19 pandemic outbreak and the Russia-Ukraine conflict), with monthly and quarterly optimization, with a budget constraint on 47 diverse instruments (including tech companies, energy companies, and so on). *EvoFolio*, inspired to the perspective of Human-Centered AI [13], provides the user to express his/her views on the market, implementing the so-called elitism and giving "weight" to the preferences of humans. This kind of feature is the most distinctive from prior works in the literature.

The results of the experiments carried out show that *EvoFolio* can reduce the portfolio risk to an average of 0.07% in the case of the monthly optimization and 4% on average for the quarterly optimization, great values considering the portfolio consisting only of stocks. Furthermore, exploiting users' views significantly improves performance and makes the portfolio much more attractive to the investor. Looking at the equivalent value in USD, on the other hand, the portfolios achieved returns in line with their constitution (pure equity), albeit with peaks in some high points (both positive and negative). Lastly, we remark that *EvoFolio* and the used datasets are available at https://github.com/Balbus95/EvoFolio.

Currently, the tool here developed is mainly a Python-based framework with a minimal GUI. In future, as first step, we aim at making *EvoFolio* an easy-to-use production tool for investors. Our future goal is to embed its source code into a cross-platform application with a natural user interface, and insightful dashboards with a monitoring toolbox to enlarge the audience of possible users (economists, financial experts, investors, and researchers) of this tool. This will enable/make actionable the so-called Interactive Genetic Algorithm [75] (where the investors, besides providing his/her views, select the best portfolio every time as well as when to re-optimize it). As it is, *EvoFolio* already has this feature, but its use, efficacy, and satisfaction level should be measured with end-users in an operational environment (and in real time). Moreover, some future directions are foreseen in terms of experiments and scenarios to take into account, on both the computer science and finance/economics levels:

- Experimenting other kinds of crossover operators, e.g., some taking into account the differences between the "parents portfolios";
- Experimenting solutions within operators such that they give priority to instruments with high volumes exchanged to exploit in advance the bid-ask spread that could arise between the prices of an asset so that the investor does not miss the investment opportunity;
- From the experiments carried out, it emerges that *EvoFolio* tends to compose portfolios where one to three instruments are purchased in high quantities, while the others (within the constraints given) are purchased in minimum quantities. Our idea for the next developments is the following: experimenting penalty mechanisms to improve the distributions of instruments and number thereof within portfolios to determine the number of assets that are strategic and linked to the markets in which the investor generally carries out transactions;
- In this paper, we have built *EvoFolio* and carried out all experiments assuming no transaction costs. Our future efforts will surely be put toward implementing actual fees that financial intermediaries impose in asset trading transactions. In this way, not only does one try to overcome the assumption of transaction costs' absence,

but it is also possible to consider additional costs linked to the exchange rate (e.g., Euro-USD for the purchase of assets in foreign currency).

- Currently, *EvoFolio* performs market operations only with the incomes based on the given initial budget and the value of portfolios. In future, we can implement in the tool a method for making loans to try to maximize returns net of the interests applied.

Besides, with regards to the whole project, meanwhile carrying experiments before mentioned, we primarily aim at developing a cross-platform (Web, Mobile) application for end-users (domain experts, researchers, investors, and so on) to make *EvoFolio* available to everyone "just a click/tap away." The basic idea will be to prototype the application centering the design on users [76, 77], involving a group of experts for pre-development questionnaires, requirements elicitation, and another group of users for a post-development survey, exploiting well-known questionnaires (like in [78]). The ultimate direction is that of building an actual application (enabling/making actionable Interactive Genetic Algorithm [75]) to perform portfolio formation on the markets directly, made available directly by the credit institution where the user holds the account. With a simple click, the user can accept the portfolio proposed by *EvoFolio* and send the order directly to the market via the App-Bank-Market link.

**Funding** Open access funding provided by Università degli Studi di Salerno within the CRUI-CARE Agreement.

**Data availability** All data analyzed during this study are included in the GitHub repository https://github.com/Balbus95/EvoFolio and available through Yahoo!Finance as well.

## Declarations

**Conflict of interest.** The authors declare that they have no conflict of interest.

## References

1. Markowitz Harry (1952) Portfolio selection. The. J Finance 7(1):77–91
2. Sharpe William F (1964) Capital asset prices: A theory of market equilibrium under conditions of risk. J Finance 19(3):425–442
3. Lintner J (1965) The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. Rev Econ Stat 47(1):13–37
4. Mossin J (1966) Equilibrium in a capital asset market. Econometrica 34(4):768–783
5. Fama EF, French KR (2004) The capital asset pricing model: theory and evidence. J Econ Perspect 18(3):25–46
6. Ross Stephen A (1976) The arbitrage theory of capital asset pricing. J Econ Theory 13(3):341–360
7. Black F, Litterman R (1992) Global portfolio optimization. Financ Anal J 48(5):28–43
8. Colasanto F, Grilli L, Santoro D, Villani G (2022) Bert's sentiment score for portfolio optimization: a fine-tuned view in black and litterman model. Neural Computing Appl 34:17507–1752
9. Idzorek TM (2004) A step-by-step guide to the black and litterman model. Zephyr Associates Inc, Incorporating User-Specified Confidence Intervals
10. Kalyanmoy D, Amrit P, Sameer A, Meyarivan TAMT (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans Evol Comput 6(2):182–197
11. Pal R, Chaudhuri TD, Mukhopadhyay S (2021) Portfolio formation and optimization with continuous realignment: a suggested method for choosing the best portfolio of stocks using variable length nsga-ii. Exp Syst Appl 186:115732
12. Evans JL, Archer Stephen H (1968) Diversification and the reduction of dispersion: An empirical analysis. J Finance 23(5):761–767
13. Shneiderman B (2022) Human-centered AI. Oxford University Press, Oxford
14. Liu X-Y, Yang H, Gao J, Wang CD (2021) FinRL: Deep reinforcement learning framework to automate trading in quantitative finance. In: ACM international conference on AI in finance (ICAIF 2021)
15. Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press, Michigan
16. Goldberg David E (1989) Genetic algorithms in search, 1st edn. Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc, Boston, MA, USA
17. Goldberg DE (2002) The design of innovation: lessons from and for competent genetic algorithms. Kluwer Academic Publishers, Norwell, MA, USA
18. Davis L editor (1991) Handbook of genetic algorithms. Van Nostrand Reinhold,
19. de Jong E, Watson R, Pollack J (2001) Reducing bloat and promoting diversity using multi-objective methods
20. Fonseca CM, Fleming PJ et al (1993) Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In: Icga, vol 93, pp 416–423. Citeseer
21. Fonseca CM, Fleming Peter J (1995) An overview of evolutionary algorithms in multiobjective optimization. Evol Comput 3(1):1–16
22. Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. In: Proceedings of the 1st international conference on genetic algorithms, pp 93–100, Hillsdale, NJ, USA. L. Erlbaum Associates Inc
23. Horn Jeffrey, Nafpliotis Nicholas, Goldberg David E (1994) A niched pareto genetic algorithm for multiobjective optimization. In: Proceedings of the first IEEE conference on evolutionary

computation. IEEE world congress on computational intelligence, pp 82–87. Ieee,
24. Srinivas N, Deb K (1994) Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evol Comput 2(3):221–248
25. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195
26. Fama E, French KR (1988) Dividend yields and expected stock returns. J Financ Econ
27. Hashem Pesaran M, Timmermann A (1995) Predictability of stock returns: robustness and economic significance. J Finance 50(4):1201–1228
28. Box GEP, Jenkins GM (2015) Time series analysis: forecasting and control. Holden-Day, 2015. ISBN: 978-1118675021
29. Hamilton JD (1994) Time series analysis. Princeton University Press, Princeton. 978-0691042893
30. Gunjan A, Bhattacharyya S (2023) A brief review of portfolio optimization techniques. Artif Intell Rev 56:3847–3886
31. Holton GA (2003) Value-at-risk. Academic press San Diego, CA
32. Rockafellar RT, Uryasev S (2002) Conditional value-at-risk for general loss distributions. J Bank Finance 26(7):1443–1471
33. Xu C, Wang J, Shiba N (2007) Multistage portfolio optimization with var as risk measure. Int J Innov Comput Inf Control 3(3):709–724
34. Xu C, Ng P (2006) A soft approach for hard continuous optimization. Eur J Oper Res 173(1):18–29
35. Lim AEB, Shanthikumar JG, Vahn G-Y (2011) Conditional value-at-risk in portfolio optimization: coherent but fragile. Oper Res Lett 39(3):163–171
36. Geyer A, Hanke M, Weissensteiner A (2009) A stochastic programming approach for multi-period portfolio optimization. Comput Manage Sci 6:187–208
37. Dupacová J (1999) Portfolio optimization via stochastic programming: methods of output analysis. Math Methods Oper Res, pp 245–270,
38. Kouwenberg R (2001) Scenario generation and stochastic programming models for asset liability management. Eur J Oper Res , 134(2):279–292. Financial Modelling
39. Barro D, Consigli G, Varun V (2022) A stochastic programming model for dynamic portfolio management with financial derivatives. J Bank Finance 140:106445
40. Greyserman A, Jones DH, Strawderman WE (2006) Portfolio selection using hierarchical bayesian analysis and mcmc methods. J Bank Finance, 30(2):669–678. Risk Management and Optimization in Finance
41. Detemple J, Garcia René, Rindisbacher M (2003) A monte carlo method for optimal portfolios. J Finance 58(1):401–446
42. Drucker H, Burges C, Kaufman L, Smola A, Vapnik Vladimir (1996) Linear support vector regression machines. Adv Neural Inf Process Systems 9(9):155–161
43. Lin C-M, Huang J-J, Gen M, Tzeng G-H (2006) Recurrent neural network for dynamic portfolio selection. Appl Math Comput 175(2):1139–1146
44. Zimmermann H-G, Neuneier R, Grothmann R (2001) Active portfolio-management based on error correction neural networks. In: Dietterich T, Becker S, and Ghahramani Z (Eds) Adv Neural Inf Process Syst, volume 14. MIT Press
45. Ban G-Y, El Karoui N, Lim AEB (2018) Machine learning and portfolio optimization. Manage Sci 64(3):1136–1154
46. Sen J, Dutta A, Mehtab S (2021) Stock portfolio optimization using a deep learning lstm model. In: 2021 IEEE Mysore sub section international conference (MysuruCon), pp 263–271,
47. Ma Y, Han R, Wang W (2021) Portfolio optimization with return prediction using deep learning and machine learning. Exp Syst Appl 165:113973
48. Park K, Jung H-G, Eom T-S, Lee S-W (2022) Uncertainty-aware portfolio management with risk-sensitive multiagent network. IEEE Trans Neural Netw Learn Syst , pp 1–14,
49. Liang Z, Chen H, Zhu J, Jiang K, Li Y (2018) Adversarial deep reinforcement learning in portfolio management. arXiv:1808.09940v3,
50. Koratamaddi P, Wadhwani K, Gupta M, Sanjeevi SG (2021) Market sentiment-aware deep reinforcement learning approach for stock portfolio allocation. Eng Sci Technol Int J 24(4):848–859
51. Guarino A, Grilli L, Santoro D, Messina F, Zaccagnino R (2022) To learn or not to learn? Evaluating autonomous, adaptive, automated traders in cryptocurrencies financial bubbles. Neural Comput Appl
52. Betancourt C, Chen Wen-Hui (2021) Deep reinforcement learning for portfolio management of markets with a dynamic number of assets. Exp Syst Appl 164:114002
53. Guennoun Z, Hamza F, El hachloufi M (2012) Stocks portfolio optimization using classification and genetic algorithms. Appl Math Sci 6(94):4673–4684
54. Ranković V, Drenovak M, Urosevic B, Jelic R (2016) Mean-univariate garch var portfolio optimization: actual portfolio approach. Comput Oper Res 72:83–92
55. Lwin KT, Qu R, MacCarthy BL (2017) Mean-var portfolio optimization: a nonparametric approach. Eur J Oper Res 260(2):751–766
56. Mehlawat MK, Gupta P (2014) Fuzzy chance-constrained multiobjective portfolio selection model. IEEE Trans Fuzzy Syst 22(3):653–671
57. Kaucic M, Moradi M, Mirzazadeh M (2019) Portfolio optimization by improved nsga-ii and spea 2 based on different risk measures. Financ Innov, 5(26),
58. Liagkouras K (2019) A new three-dimensional encoding multiobjective evolutionary algorithm with application to the portfolio optimization problem. Knowl Based Syst 163:186–203
59. Drezewski R, Doroz K (2017) An agent-based co-evolutionary multi-objective algorithm for portfolio optimization. Symmetry, 9(9)
60. Macedo LL, Godinho P, Alves M (2017) Mean-semivariance portfolio optimization with multiobjective evolutionary algorithms and technical analysis rules. Exp Syst Appl 79:33–43
61. Meghwani SS, Thakur M (2017) Multi-criteria algorithms for portfolio optimization under practical constraints. Swarm Evol Comput 37:104–125
62. Babaei S, Sepehri MM, Babaei E (2015) Multi-objective portfolio optimization considering the dependence structure of asset returns. Eur J Oper Res 244(2):525–539
63. Dangi A (2013) Financial portfolio optimization: computationally guided agents to investigate, analyse and invest!? arXiv:1301.4194v1
64. Rezani MA, Hertono GF, Handari BD (2020) Implementation of iterative k-means-+ and ant colony optimization (ACO) in portfolio optimization problem. AIP Conf Proc 2242(1):06
65. El-Shorbagy MA, Hassanien AE (2018) Particle swarm optimization from theory to applications. Int J Rough Sets Data Anal (IJRSDA) 5(2):1–24
66. Haqiqi KF, Kazemi T (2012) Ant colony optimization approach to portfolio optimization—a lingo companion. Int J Trade Econ Finance 3(2):148–153
67. Steven A, Hertono GF, Handari BD (2018) Clustered stocks weighting with ant colony optimization in portfolio optimization. AIP Conf Proc 2023(1):10
68. Wang Y, Aste T (2023) Dynamic portfolio optimization with inverse covariance clustering. Exp Syst Appl 213:118739

69. Fortin F, De Rainville F-M, Gardner M-A, Parizeau M, Gagné Christian (2012) Deap: evolutionary algorithms made easy. J Mach Learn Res 13(1):2171–2175
70. Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. In:Balcan MF, WeinbergerKQ (Eds), Proceedings of The 33rd international conference on machine learning, volume 48 of proceedings of machine learning research, pp 1928–1937, New York, New York, USA, 20–22. PMLR
71. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2019) Continuous control with deep reinforcement learning. arXiv:1509.02971v6
72. Fujimoto S, van Hoof H, David M (2018) Addressing function approximation error in actor-critic methods. In: ICML, pp 1582–1591
73. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. arXiv:1707.06347
74. Haarnoja T, Zhou A, Abbeel P, Levine S (2018) Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv:1801.01290v2

75. Quiroz JC, Louis SJ, Shankar A, Dascalu SM (2007) Interactive genetic algorithms for user interface design. In: 2007 IEEE congress on evolutionary computation, pp 1366–1373. IEEE
76. Xu W (2019) Toward human-centered ai: a perspective from human-computer interaction. interactions, 26(4):42–46
77. Auernhammer J (2020) Human-centered ai: The role of human-centered design research in the development of ai
78. Guarino A, Lettieri N, Malandrino D, Zaccagnino R (2021) A machine learning-based approach to identify unlawful practices in online terms of service: analysis, implementation and evaluation. Neural Comput Appl 33(24):17569–17587

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