ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/307609657

Anatomy of a portfolio optimizer under a limited budget constraint

Article *in* Evolutionary Intelligence · September 2016

DOI: 10.1007/s12065-016-0144-3

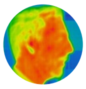
CITATIONS
0

READS
42

3 authors:

Igor Deplano
Liverpool John Moores University
2 PUBLICATIONS **1** CITATION

SEE PROFILE



Giovanni Squillero
Politecnico di Torino
232 PUBLICATIONS **1,627** CITATIONS

SEE PROFILE

Alberto Paolo Tonda
French National Institute for Agricultural Res...
75 PUBLICATIONS **152** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project

FoodMC - Mathematical and Computer Science Methods for Food Science and Industry View project

All content following this page was uploaded by Alberto Paolo Tonda on 07 September 2016.

The user has requested enhancement of the downloaded file. All in-text references underlined in blue are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

Noname manuscript No.
(will be inserted by the editor)

Anatomy of a Portfolio Optimizer under a Limited Budget Constraint

Igor Deplano · Giovanni Squillero · Alberto Tonda

DRAFT

Received: Thursday 8th September, 2016/ Accepted: date

Abstract Predicting the market's behavior to profit from trading stocks is far from trivial. Such a task becomes even harder when investors do not have large amounts of money available, and thus cannot influence this complex system in any way. Machine learning paradigms have been already applied to financial forecasting, but usually with no restrictions on the size of the investor's budget. In this paper, we analyze an evolutionary portfolio optimizer for the management of limited budgets, dissecting each part of the framework, discussing in detail the issues and the motivations that led to the final choices. Expected returns are modeled resorting to artificial neural networks trained on past market data, and the portfolio composition is chosen by approximating the solution to a multi-objective constrained problem. An investment simulator is eventually used to measure the portfolio performance. The proposed approach is tested on real-world data from New York's, Milan's and Paris' stock exchanges, exploiting data from June 2011 to May 2014 to train the framework, and data from June 2014 to July 2015 to validate it. Experimental results demonstrate that the presented tool is able to obtain a more than satisfying profit for the considered time frame.

Keywords Portfolio optimization; Multi-layer perceptron; Multi-objective optimization; Financial forecasting

Igor Deplano
Liverpool John Moores University
Byrom Street, L3 3AF, Liverpool, U.K.
E-mail: igor.deplano@gmail.com

Giovanni Squillero
Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
E-mail: giovanni.squillero@polito.it

Alberto Tonda
UMR GMPA, AgroParisTech, INRA, Université Paris-Saclay
1 av. Brétignières, 78850, Thiverval-Grignon, France
E-mail: alberto.tonda@grignon.inra.fr

1 Introduction

The recent diffusion of on-line trading platforms gives virtually anyone the possibility of investing in any stock exchange market, starting from any amount of money, thus creating an ever-growing number of small investors. These new brokerage clients wish to find the optimal portfolio composition, that is, to select the best investment policy in term of minimum risk and maximum return, given their collection of investment tools, income, budget, and convenient time frame.

The research community immediately tried to tackle the problem with mathematical tools and models, considering the investor's willingness to take higher risks for potentially higher rewards [44, 45], and, more recently, devising strategies that take into account the users' preferences for upside over downside volatility [64], or preparing mixed portfolio to represent different needs that the investors need to satisfy at the same time [57].

With the simultaneous growth of types of financial products and computers' performance, the machine learning and evolutionary computation communities also tried to approach portfolio optimization [16], with methods including *Artificial Neural Networks* (ANNs) [66], *Support Vector Machines* (SVMs) [12], *Evolutionary Algorithms* (EAs) [37], and *Multi-objective EAs* (MOEAs) [65].

In this work, we analyze in depth a decision-support tool for management strategies of small portfolios. The tool takes in input market data and, after a training phase, starts an investing simulation. During training, the performance expectation of each tradable over three days is modeled by a separate artificial neural network, a *Multi-Layer Perceptron* (MLP). Afterwards, for each day, the simulation uses these neural networks as oracles to optimize the portfolio management strategy, resorting to a multi-objective evolutionary algorithm that tries to find a balance between risk and reward, finally selecting a Pareto-optimal set of titles that requires the least amount of operations to acquire. The idea of such an architecture was first presented in [18], and this paper further enhances it, analyzing each choice in detail, and extending the experimental evaluation. Furthermore, we illustrate and motivate the design process, step by step.

The rest of the paper is organized as follows: Section 2 briefly recalls basic concepts of portfolio optimization, MLPs, and multi-objective evolutionary optimization; a high-level description of the proposed approach is given in Section 3, and its components are then detailed and discussed in Sections 4 and 5; Section 6 reports the experimental results, while Section 7 outlines future works and concludes the paper.

2 Background

Investing involves three elements: *markets*, *investors*, and *assets*. All of its rules could be summarized in a fairly simple statement: one has to buy (sell) assets that will gain (lose) value in his/her investing horizon, and the place where such a bargain happens is a specific *market*. Eventually, investors who took the correct decisions will gain money, while wrong choices will lead to losses. Despite this apparent simplicity, the task is overwhelmingly complex: for an introduction to the basic concepts of investing, we suggest [26, 58, 8].

DRAFT

How assets prices are formed and the market itself have been modeled with different theories. Fama, in the *Efficient Market Hypothesis* (EMH) [21], states that prices efficiently represent all of the available information, and his empirical work is divided in three different efficiency forms depending on the information available to the agents: in the *weak form*, current prices fully reflect their historical, and arbitrage is possible defining trading rules; in the *semi-strong form*, prices fully reflect all the publicly available information, and no arbitrage is possible; in the *strong form*, stock prices, at any point of time, represent all the available information and prices can thus be described by a random walk model. In practice, the strong form is impossible to attain, because it requires every information to be public (which is generally false): it should be seen as a superior limit to the market's efficiency. The underlying assumption of EMH is that the agents are rational (some could over-react and some under-react, but on average they are rational), they aim to maximize their expected utility, they are risk averse, and capable of processing the available information.

Despite its initial popularity, EMH has been strongly criticized by investors [33] and researchers: Grossman and Stiglitz, for example, theoretically proved that efficient information distribution is impossible [27, 28]. After these criticism, another theory emerged: *Behavioral finance* [59, 6, 60] (BF) is a sub-field of Behavioral Economics (BE) that uses *Prospect Theory* [36, 67] as the decision-making model under uncertainty, and considers agents as non-perfectly rational and subject to different biases, for example anchoring, framing, magical thinking, overconfidence, over- and under-reaction.

In 2004, Lo presented a new framework, the *Adaptive Market Hypothesis* [41] (AMH) which links EMH and BF through evolutionary adaptation. The framework solves the criticism on Simon's model of bounded rationality [61, 62] by defining the agent's optimization limit with reinforcement learning.

The investment process involves the concepts of risk and uncertainty. While uncertainty is not yet quantifiable, risk can be measured and managed. In 1952, Markowitz started the so-called *modern portfolio theory*, describing in a quantitative way how diversification of assets could be used to minimize, yet not completely remove, the overall risk without changing the portfolio's expected return [44]. In Markowitz's original view, choosing an optimal portfolio is a *mean-variance optimization problem*, where the objective is to minimize the variance for a given mean. The variance could be used as measure to quantify risk, while the choice of the portfolio components could be a way to manage it. Since 1952, diversification is a well known way of reduce idiosyncratic risk. More recently, several different portfolio theories were proposed: *post-modern portfolio* [64]; the very same Markowitz improved his mean-variance idea in 1968 [45]; *Capital Asset Pricing Model* (CAPM) optimal portfolios [47, 56, 22]; and the *Behavioral Portfolio Theory* [57] (BPT), that, differently from other approaches, takes into consideration investors who are not completely risk-averse and divides the portfolio composition into mental accounts. For example, an investor could have a first mental account with the objective to avoid poverty, and a second with the aim to become rich: and his/her wealth is balanced in these two portfolios.

In this work, we will focus on the issues and challenges faced by a small investor, with access to stocks, only. Nevertheless, it is important to notice how the financial field is incredibly vast, and full of potentially valuable niches.

DRAFT

2.1 Portfolio Management

Portfolio management is a complex discipline, a middle point between art and science, where the aim is to establish a “portfolio composition”, in the context of a *passive investor*, i.e., an investor that buys shares of a company without joining its management. In contrast, an *active investor* would find value in a mismanaged company, buy it, reorganize and probably sell it. Corporate portfolio management is driven by finding a set of “mispriced” assets while hedging unwanted risks: however, risk cannot be avoided completely. A good introduction to the subject can be found in Koller et al. [39].

Professional investors' investing strategies, like ones of hedge funds or portfolio managers, could be bounded by mandate, with constraints such as long-term, stocks, fixed-income investments only. More in general we have two main philosophies [33], *absolute return* and *relative return*. Absolute return [34] aims to exploit investment opportunities while avoiding losses, because they shrink the rate at which the capital compounds: the risk is managed as *total risk*, the neutral position is holding cash. In a relative return context the aim could be to replicate the performance of a benchmark (indexing) or try to beat it (benchmarking). The risk can be defined, for example, as *tracking error* or as *benchmark Value at Risk*, the risk-neutral position is to hold a portfolio which is equal to the benchmark, while holding cash will perform worse than the benchmark (increasing the risk).

Risk could be summarized as the exposure to adverse change in the investment environment. More in specific, we have the following categories of risk [5]: *Liquidity Risk* [46], *Leverage Risk*, *Market Risk*, *Counterparty Credit Risk*, *Operational Risk*, *Legal and Compliance Risk*. We are more interested in the Absolute return style, which in our context of long-only mandate involves stock picking and controlling the Market Risk. Common measure frameworks for Market Risk are Value at Risk (VaR) and conditional VaR (CVaR), with their metrics. An extensive guide to Market Risk can be found in [2].

Several techniques are used to try and forecast stock performances [52]. *Fundamental analysis* studies company reports, news and macro-economical data to estimate future earnings, defining the *fair price* [1, 17, 26], while *Technical analysis* includes a wide range of options: *Chart trading* [20, 11] (chart trading studies prices' charts, to give buy and sell signals through combinations of indicators and oscillators), Machine learning and statistical tecniques [38, 40, 55], agent-based systems [50], classifiers like EDDIE or $cAnt - miner_{PB}$ [53], and evolutionary algorithms [43, 31, 24, 42, 68, 48, 7, 32, 49]. A work on market bubbles identification and analysis has been presented by Jianga et al. [35], where the authors fit the observed price time series to a log-periodic power law model, and use Lomb spectral analysis for detecting log-periodic oscillations.

2.2 Artificial Neural Networks

Multi-layer Perceptrons (MLPs) are feed-forward artificial neural networks that are used for function approximation or for classification [29]. They consist of a layer of inputs, one or more hidden layers, and one output layer. Every layer has a set of nodes, each one fully connected to the nodes on the next layer, forming

DRAFT

a directed graph. MLPs are trained using the back-propagation algorithm, a supervised learning procedure that maps examples, each one being a pair of input and output features. Every node has as output the application of a non-linear activation function over an affine transformation of his inputs. During the back-propagation phase, starting from the output layer until the first one, the gradient descent algorithm evaluates the gradient of a distance between the desired output and the evaluated one. The gradient is back-propagated, modifying the parameters of the matrix that define the previous affine transformations. The amount of change in the network is affected by the learning rate of the algorithm used. The structure of the modifications defines the back-propagation algorithm, as example using a momentum (that can be seen as a gradient viscosity), variable learning rate, combination of the previous two, etc.

There are also different training strategies, namely *batch*, *mini-batch*, or *online*. The difference is when the gradient is applied, that is, after the whole training dataset, after a sample of the dataset, or after each example. Applying an average over the computed gradients can be useful to reduce the probability that a single outlier can make huge changes, the so-called *gradient exploding*. There are also different normalization techniques, like *gradient clipping* to prevent this problem. The distance that is used to measure the error depends on the problem, commons are *euclidean distance*, *cosine similarity* and *cross-entropy*.

A desirable activation function is *zero-centered*, does not saturate[1], and features a rapid gradient calculation. In its original version the MLP used sigmoids, that are not zero-centered. Following works exploited the *hyperbolic tangent*, which is zero-centered, but still suffers the saturation problem. Modern artificial neural networks and deep learning approaches exploit activation functions such as *Elu*, *ReLu*, *PReLu*, and *MaxOut* [25, 30].

Self-Organizing Maps (SOMs), on the other hand, are artificial neural networks with an input layer and a node matrix as output layer. The learning algorithm is unsupervised and defines the concept of neighborhood in the output layer: given a node, its neighborhood is composed by the nearby nodes, constrained by the layer topology and a distance measure. Learning in SOMs is modeled as increasing the weight of a node and distance inverse-proportionally increase the weights of the nodes in the neighborhood, the neighborhood shrinks with time [29]. A distance metric that can be used for the neighborhood is the link distance and the neighborhood topology can be hexagonal.

2.3 MOEA

A multi-objective optimization problem is characterized by several objectives that are in conflict with each other. Thus, an improvement in one objective may lead to deterioration in another. A single solution, which can optimize all objectives simultaneously, usually does not exist; a multi-objective optimization algorithm instead attempts to find the best approximation of the *Pareto frontier*, a set of *Pareto-optimal solutions*, each being an allocation of resources where it is impossible to make any one objective better without worsening at least another one. Classical

[1] Saturation of the neuron is a non linearity happens when the weights are too high in magnitude: on every input the result will be the saturation level and that means that the neuron losses memory and became useless.

DRAFT

optimization methods suggest converting the multi-objective optimization problem to a single-objective one, by emphasizing one particular Pareto-optimal solution at the time; for example, in portfolio optimization, Markowitz [44], chose to minimize the overall risk given a fixed rate of return. Through evolutionary computation, in particular with MOEA algorithms, it is possible to find an approximate *Pareto frontier* in a single simulation. An introduction of this field of research can be found in [14]. The state of the art in MOEA is currently represented by NSGA-II [15].

3 Proposed Approach

A professional large-scale trading system may be theoretically designed, but it would be both extremely expensive and time-consuming, and small investors would not use it, simply because of the huge running costs. Differently, we propose a tool that can be exploited for supporting investing decisions, and that can be run on a desktop computer. Since we focus on limited budget portfolios, we can disregard the constraints on book analysis and liquidity problems.

Figure 1 shows a high-level scheme of the proposed approach. In the framework, we can identify three parts: the *oracles* try to predict the performance of each stock in the near future; the *optimizer* finds a set of stock choices, representing an acceptable compromise between predicted risk and reward, on the basis of the oracles' predictions; after a Pareto front of compromises is found, the algorithm chooses the point that requires the least number of transactions to achieve, starting from its current portfolio of stocks; finally, the *simulator* computes the rewards obtained by the framework, on the basis of unseen validation data. In the following sections, we will discuss every part of the framework in more detail.

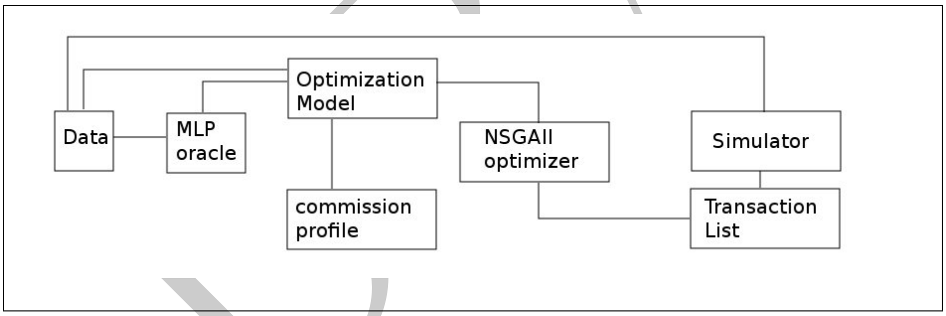


Fig. 1: High-level scheme of the proposed approach. Training data, properly clustered, are used to train the MLP oracles that will try to predict the performance of company shares. The optimizer uses a MOEA to obtain a trade-off between (predicted) risk and reward. Finally, the simulator computes the effective reward obtained, exploiting unseen validation data.

4 Discussion: Oracles

Each oracle models how the investor's *rate of return expectation* (RRe) changes in the various market conditions. Given an input, it gives a measure of what will happen in the future. MLPs are well suited to realize such trend followers, considering the good results obtained for financial forecasting in [19]. As we use MLP as function approximation, the desired output is simply the RRe codomain.

We aim at intercepting a part of the market mood over the related stocks, and include this information in the oracle. Since Fu [23] shows how SOMs are successful at discovering shape patterns, we exploit them to define *stock clusters*, that is, group of companies whose behavioral patterns are similar. Behavioral patterns are measured over the sequence of closing performances in a 10-day range. The resulting clusters are expected to change over time, because of market conditions and news, but we expect that some combinations will remain stable over time, for example with regards to companies that have strong dependencies or stable business relations.

We divide the training period in blocks of 10 consecutive days, and we train a SOM for each block. The training organizes clusters on the output layer. After the training phase, we evaluate the network on the input block, and retrieve the clusters' composition. We expect that stocks with similar behaviors will fall in the same cluster. As we have different networks for each block and the networks are sensible to the initialization weights, we see each cluster composition as a transaction for the *Apriori* algorithm [63, 69, 9] that will extract the *itemsets* with major *support* and *confidence*. The output is a set of association rules in the form of *antecedent* $\succeq$ *consequent* rules (see Equation 1 for an example) which does not mean implication, but co-presence.

$$stock_i \succeq stock_1, stock_2, \cdots, stock_n \quad (1)$$

The Oracle for $stock_i$ will use additional 10 inputs, each one representing a daily average of the values of the stocks found in the consequent set of the association rule obtained for $stock_i$.

4.1 MLP for function approximation: motivation and benchmarking

In order to tune the MLP, we compare the performance of different training techniques, structures and configurations on a benchmark problem: price forecasting for Eni[2], one of the biggest Italian companies whose core business is oil extraction and refinement. Independently from the number of hidden layers, each MLP tested has 100 input and 3 output features, predicting the future performance of the stock's price (increases, decreases, remains stable).

For each MLP we considered, we applied different input-output configurations, see Table 2 for a summary. The benchmark data includes the stocks of Eni and all companies in the oil and natural gas index, in the period ranging from January 2000 to June 2010. This dataset is divided into three parts: training, validation and testing (60%, 20%, 20% of the original data, respectively). In this work, we consider

[2] http://www.eni.com/

DRAFT

end-of-day data (EoD), the *open* and *close* price, the *minimum* and *maximum* price during the trading day, and the *number of shares* traded.

epochs	200
goal	0.0001
learning rate	0.001
learning rate increment	1.005
learning rate decrement	0.095
momentum costant	0.75
max validation fails	20
minimum gradient	10^{-10}

Table 1 Training parameters for the MLPs, common to all tests.

In Table 2, configuration 1 is the complete EoD data row; configuration 2 is a measure of volatility with the closing value; lastly, configuration 3, shows the close values, only. The close values are important because they represent the last price used during the negotiation day.

configuration	input	output
1	10 days of *min*, *max*, *close*, *open*, *vol*	next 3 days of *close*
2	10 days of $max - min$, *close*	next 3 days of *close*
3	10 days of *close*	next 3 days of *close*

Table 2 Input-output configurations. The stock input data is for Eni and oil and natural gas share index; output includes Eni stocks, only.

The I/O configuration used is reported in Table 2. The neuron activation function is a hyperbolic tangent. The initialization algorithm that chooses initial weights and bias is *Nguyen-Widrow* (NW) [51]. All training parameters are reported in Table 1.

For our purposes, we chose to exploit *adaptive learning* to perform gradient descent during back-propagation. The use of an adaptive learning is necessary to grantee acceptable performances. The preliminary experimental evaluation also suggested that the use of a *momentum* slowed down the process without being beneficial.

Training results are strongly dependent from the initial weights, and the network tends to fall into local minima. We noticed that a necessary condition for accepting the trained network is to test if the error distribution in the output layer belongs to a normal distribution family. Operatively, we can require that in all the three output nodes the error mean and variance should be lesser than e and $10e$ respectively, where e starts from 0.1 and it is incremented by 30% every 10 refusals.

Table 5 reports the best results achieved for MLP as function approximation, configuration 3. All networks have 20 input features and 3 output features.

Configurations 2 and 3 perform much better than configuration 1, both in final results and in training time, as they have less input features and hidden layers with less neurons. The results suggest that, for this ANN architecture, using the complete EoD data for each input day does not improve forecast capabilities.

DRAFT

	μ_1	σ_1	μ_2	σ_2	μ_3	σ_3
ff1h	0.1365	3.7334	0.2662	3.3989	0.1008	3.5990
ff2h	0.1755	3.5330	0.1996	3.7211	0.2543	3.6210
ff3h	0.3936	2.7554	0.4538	2.4185	0.2844	2.7811

Table 3 Best results for MLP as function approximation, configuration 1, mean (μ_i) and variance (σ_i) of error; randomly initialized NN have performed worst than NW. The number before the h is the number of hidden layers: 1h, 2h, or 3h. The number of neurons for each layer: ff1h(201), ff2h(200,100), ff3h(100,50,25). 100 input features, 3 output features.

	μ_1	σ_1	μ_2	σ_2	μ_3	σ_3
ff1h	0.1308	4.4511	0.0265	4.7854	0.0907	4.9236
ff2h	0.4894	2.3433	0.4656	2.4899	0.3631	2.7598
ff3h	0.3213	2.8939	0.3825	2.6646	0.4112	2.4843

Table 4 Best results for MLP as function approximation, configuration 2, number of neurons for each layer: ff1h(81), ff2h(80,40), ff3h(80,40,20). 40 input features, 3 output features.

	μ_1	σ_1	μ_2	σ_2	μ_3	σ_3
ff1h	0.0324	4.8329	0.2091	4.9226	0.0220	7.5821
ff2h	0.2132	3.7758	0.1946	3.8084	0.2032	4.0627
ff3h	0.2112	2.8737	0.3183	3.3552	0.4953	3.0535

Table 5 Best results for MLP as function approximation, configuration 3, 20 input features, 3 output features. Number of neurons for each layer: ff1h (41), ff2h (40, 20), ff3h (40, 20, 10).

5 Discussion: MOEA for portfolio optimization

For the optimization step, we assume that the investor has a very limited budget, arbitrarily set to **B**. Such an amount is small enough not to influence the overall market behavior, but still permits diversification with marginal transaction costs in the real world. No *margination* and no *short selling* are allowed.

The optimization step must choose the best composition of assets, relying upon the MLP oracles described in Section 4 to obtain a prediction of the stocks' performance in the next days.

Let q_i be the desired quantity of asset i, it determines the required action: if q_i is greater than the quantity actually owned of asset i, the investor is required to sell; if lower, the investor needs to buy; if equal, no action is required. Let also be **B** the whole budget available in the current day; γ the minimum gain for an operation to be taken into consideration; α limits the choice of assets to those that can guarantee at least a certain amount of global performance; $P_i^b(n)$ the expected buy price of asset i, $n > 0$ days in the future; and $P_i^s(n)$ the expected sell price of asset i, $n > 0$ days in the future; $P_i^b(0) = P_i^f(0) = P_i(0)$ is the closing price of asset i in the current day. The independent variables in the optimization are q_i; α and γ are constants; **B** is known for the current day; both $P_i^b(n)$ and $P_i^s(n)$ are the result of forecasts if $n > 0$, while they are known data if $n = 0$.

To obtain a suitable set of compromises between risk and reward, we benchmark different models, that all have in common the constraints of equations 2, 3, and 4.

$$\sum_{i=1}^{n} q_i \cdot P_i^b(1) \leq \mathbf{B} \tag{2}$$

DRAFT

$$\sum_{i=1}^{n} q_i \cdot P_i^s(2) \geq \alpha \cdot \mathbf{B} \tag{3}$$

$$\forall i \in [1, n] : q_i \cdot P_i^s(2) \geq \gamma \cdot P_i(0) \tag{4}$$

In more detail, equation 3 expresses the constraint of maintaining the portfolio value superior to $\alpha\mathbf{B}$, preventing unacceptable compositions; equation 2 is the budget constraint, that is, it bounds the portfolio to the available budget; equation 4 restrict the search space over the stocks that are expected to grow, as we are restricting our operativity to only long positions.

Model 1 is an adaptation of Markowitz mean-variance model discussed in [13]. It is described by equations 5 and 6. The model chooses portfolios composed by stocks whose most expected performance was uncorrelated in the past days, in order to minimize the global risk.

A single model was used in [18], and it was defined by equations 2, 3, 5, and 6. Equation 4 has been included to consider only promising assets in the search space.

$$\max \sum_{i=1}^{n} q_i \cdot (P_i^s(3) - P_i^b(1)) \tag{5}$$

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} q_i \cdot q_j \cdot \sigma_{ij} \tag{6}$$

Equation 5 is the maximization of the expected return, we use the 3-day forecast $P_i^s(3)$ because $t = 1$ (tomorrow) is the buying day, $t = 2$ (the day after tomorrow) is the first possible selling day, and $t = 3$ adds a trend-like view on the process.

Equation 6 is the global correlation minimization between portfolio stocks. σ_{ij} is the covariance between performance of stock i and j in past 10 days — it does not make sense to consider a longer interval, given our short investing horizon: decisions can be changed daily, and we assume there are no liquidity problems. The equation must be minimized to penalize portfolios composed by highly correlated stocks.

Model 2 is described by equations 5, 6, and 7. It considers one additional objective, equation 7, that represents the minimization of the number of stocks in our portfolio. The $sign(x)$ function returns 0 if $x = 0$, 1 if $x > 0$, -1 if $x < 0$. As it is not possible to have $q_i < 0$, for construction, this objective corresponds to minimizing the number of assets owned by the investor, and it could be useful in the optic of introducing transaction costs. The model with two objectives will span the operations in the whole set of stocks available, trying to find the best trade-off. Applying the same model with a simulation that takes in account a simple real (non-professional) transaction profile will result in a complete erosion of capital because of the fixed costs related to each transaction performed.

$$\min \sum_{i=1}^{n} sign(q_i) \tag{7}$$

DRAFT

Model 3 is described by equations 7 and 8, focuses on the maximization of future returns, disregarding the correlation between stocks.

$$\max \sum_{i=1}^{n} q_i \cdot (P_i^s(2) - P_i^b(1)) \tag{8}$$

Model 4, defined by equations 8 and 9 does not consider the risk as usually appears in financial literature, but takes into account the mean and variance of the error rate ϵ for each asset as reported by the neural networks. The rational for this is simple: maximizing the performance, choosing stocks that the framework is good at forecasting. We removed one term from the forecast, because the third day is usually affected by more noise.

$$\min \sum_{i=1}^{n} |sign(q_i) \cdot \mu_{\epsilon_i} \cdot \sigma_{\epsilon_i}| \tag{9}$$

Each candidate model describes a set of mixed-integer problems, including conflicting non-linear objectives and features. We have chosen them to better understand the behavior of our forecasting system in combination with the optimizer: while Model 1 and 2 have a financial meaning, Model 3 and in particular Model 4 aim to build portfolios of winners without considering exposure risk; equation 9 helps the selection of winners penalizing stocks with higher forecasting error.

MOEA [3, 10] has been shown to be effective in finding a good approximation for the Pareto frontier in such cases. In our experiments, we use NSGA-II [15], as it was already successfully exploited in [4], where it was demonstrated scalable and able to efficiently find acceptable solutions in a financial case study. No matter the model adopted, after the optimization phase ends, the framework will select the target Pareto-optimal portfolio that requires the least amount of operations to achieve.

6 Experimental Results

In this study, we consider EoD data from about one hundred large-capitalization stocks quoted in three different countries: FTSE-MIB[3], CAC 40[4], DJ30[5]. Data is taken from *Yahoo! finance*[6], ranging from June, 1st 2011 to July, 20th 2015. Some stocks presented insufficient or clearly wrong financial data, and they have been removed from the dataset: for example, Yahoo data for the MT.PA stock in the Paris market are not coherent with data found in other websites, as Bloomberg's[7].

[3] Milan stock exchange (Euro): A2A.MI, ATL.MI, AZM.MI, BMED.MI, BMPS.MI, BP.MI, BPE.MI, BZU.MI, CPR.MI, EGPW.MI, ENEL.MI, ENI.MI, EXO.MI, FCA.MI, FNC.MI, G.MI, ISP.MI, IT.MI, LUX.MI, MB.MI, MS.MI, PMI.MI, PRY.MI, SPM.MI, SRG.MI, STM.MI, TEN.MI, TIT.MI, TOD.MI, TRN.MI, UBI.MI, UCG.MI, US.MI, YNAP.MI.

[4] Paris Stock Exchange (Euro): AC.PA, ACA.PA, AI.PA, AIR.PA, BN.PA, BNP.PA, CA.PA, CAP.PA, CS.PA, DG.PA, EI.PA, EN.PA, ENGI.PA, FP.PA, GLE.PA, KER.PA, LI.PA, LR.PA, MC.PA, ML.PA, OR.PA, ORA.PA, PUB.PA, RI.PA, RNO.PA, SAF.PA, SAN.PA, SGO.PA, SU.PA, TEC.PA, UG.PA, VIE.PA, VIV.PA.

[5] Down Jones Avarage Industrial — New York Stock Exchange (US Dollars): AAPL, AXP, BA, CAT, CSCO, CVX, DD, DIS, GE, GS, HD, IBM, INTC, JNJ, JPM, KO, MCD, MMM, MRK, MSFT, NKE, PFE, PG, TRV, UNH, UTX, V, VZ, WMT, XOM.

[6] `https://finance.yahoo.com/`

[7] `http://www.bloomberg.com/markets/stocks`

DRAFT

Compared to the experimental data used in [18], related to a single market and featuring a much larger training sample, the task becomes thus harder.

The dataset is divided in two contiguous blocks: the initial 75% of the original data: June, 1st 2011 – June, 1st 2014; and the final 25%: June, 2nd 2014 – July, 20th 2015. The first block is used for training the MLPs, and for this purpose it is further divided into the canonical *training*, *validation* and *testing* sets; while the second block is used to validate the whole framework with a performance simulation.

A performance comparison between the indexes over the considered period is shown in Figure 2, with time on the horizontal axis (day 0 being June, 1st 2011), and the cumulate performance on the vertical axis. We can easily notice that the DJ index is growing, with fluctuations of small magnitude and we can say that since its minimum is channeled into a growing trend. FTSE-MIB and CAC are very similar in their behavior, but the Italian index features wider fluctuations.

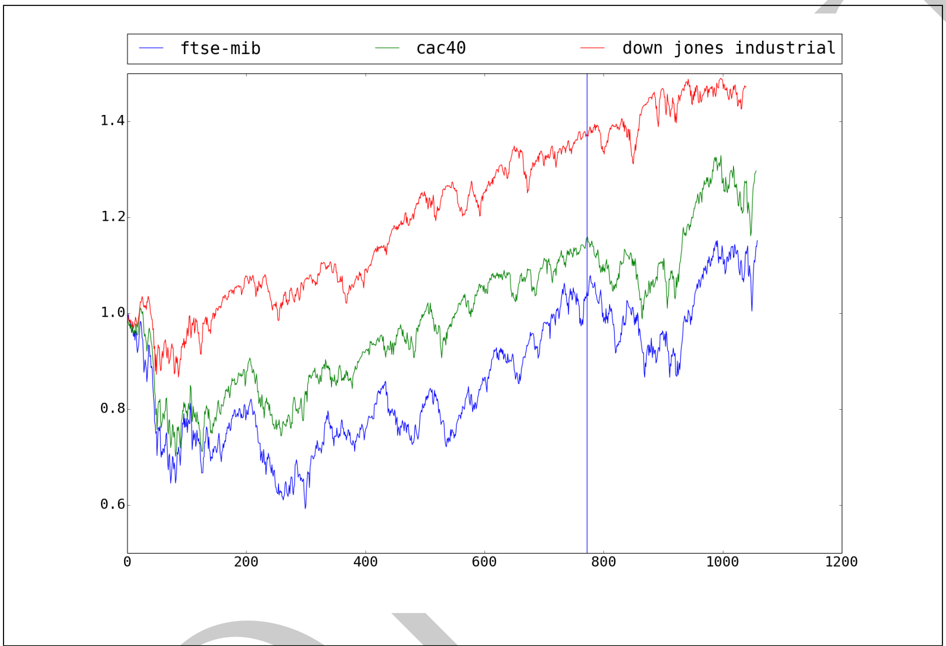


Fig. 2: Relative performances of the considered indexes, from June, 1st 2011 to July, 20th 2015. The blue vertical line is the separation between training and testing set.

As a baseline for our methodology, we compare the different models with a *random walk* and a classical *buy-and-hold* strategy. All models start from the same budget $\mathbf{B}$ = 20,000 EUR. In the random walk, we use **Model 4**, replacing the neural networks with a random oracle, and setting the model values as follows: $\alpha = 1.01$ and $\gamma = 1.01$. The random oracle simply assigns Gaussian-distributed random values with $\mu = 0$ and $\sigma = 5$ on the inputs of the optimizer. The buy-and-hold strategy consists in buying on the first day an index-representative portfolio, and hold it until the last day of the simulations. **Models 1-4** have been described in Section 5: in the following experiments, for all models we use $\alpha = 1.01$ and $\gamma = 1.01$, to drive the framework towards stocks that are forecast to gain at least 1% in the near future.

The oracles' changes are on the neural networks training acceptance: we discover that the original method suggested in [18] was insufficient and could lead to longer-than-necessary training repetitions: thus, in this tests the networks are trained for a maximum of 350 epochs, with a learning rate of 0.001. Around the 260-epoch mark, the conditions originally stated, thus after this limit if the validation failure hit the target then the network is accepted anyway and the training is stopped. If the training hits the target validation failure before the limit, it means that the initialization weights were ill-conditioned and another initialization with training will start. If the training continues until the 350 epoch does not matter as the gradients are smaller and the learning rate is reduced by the training algorithm. We modified also the number of nodes in each hidden layer, now the first hidden has 40 nodes and the second has 35 nodes. Other features are the same as the previous work.

The optimizer takes as input the oracles' forecasting and solves a constrained multi-objective problem. The first issue could arise on the Pareto front approximation, we made different trials, ranging from a population of size 50 to a population of size 350. The bigger the population, the more time and computational resources the algorithm needs, the more precise Pareto front could be. After a few trials, we found an acceptable trade-off for a population size of 200. The survival-fraction is set to 35% of the best individuals at each generation. The number of generations for N_s stocks is $200 \cdot N_s$ (i.e., $G \doteq$ 6,600). Initial population is random uniform lower-bounded by 0. The other parameters are left at default: crossover is *crossoverintermediate*, with anapplication rate of 0.8; *crowding distance* is used for measuring the distance between individuals, useful to preserve diversity; the mutation function is *mutationadaptfeasible*, with a rate of 0.2; the function to select the parents in crossover and mutation is *selectiontournament*, a tournament selection of size 2.

index	random-walk	buy-&-hold	model 1	model 2	model 3	model 4
CAC40	-5.65%	+13.88%	+15.83%	+18.10%	+16.42%	+13.15%
FTSE-MIB	+14.86%	+10.25%	+71.75%	+66.52%	+52.59%	+86.94%
DJ30	-9.21%	+8.10%	+12.18%	+11.94%	+12.02%	+12.17%

Table 6 Evaluation of the framework on the testing data, the percentage represents the cumulative performance after the last testing day.

In Table 6 we show a summary of the results. The least performing strategy is the random walk, while the proposed models are shown to be competitive with the buy&Hold strategy, often outperforming it on the Italian index. It's interesting to focus on the performances over the FTSE-MIB and CAC40 because their indexes have a similar general behaviour in both training and testing but the final results are quite different. Figure 4 shows the plots of the index-components performances-frequencies. FTSE-MIB shows wide variations and more frequently. CAC40 has a gaussian like shape, with most of the fluctuations in the interval [-5%, +5%]. The

DRAFT

components of the FTSE-MIB make wider fluctuations and more frequently than the ones of CAC40.

In the DJ index we see instead that the performances are stable for all the proposed models: in fact sometimes the optimizer does not find initial feasible points, and in these days it does not make any transaction. DJ from Figure 4 shows a more tight curve than CAC. In [18] the performances were higher, the dataset was significantly bigger and different in composition. The stocks were in large majority small-caps, which feature huge fluctuations.

We can derive some conclusions from our experiments. The system works extremely well on the Italian stock market probably because it is affected by huge fluctuations. The structure of MLPs can capture a part of information, but not enough to be as effective in markets that are characterized by a lesser variance. On the contrary, in markets that have lesser variance, we see that the error made in the regression is high enough to exclude possible good operations, like what happens in the DJ or in the CAC.

Particularly interesting is the comparison between the different proposed models. Each model is able to obtain a good performance in the Italian market, that is affected by high variance, but in the DJ they have more or less all the same behavior: from this fact, we can conclude that the problem is likely in the oracles.

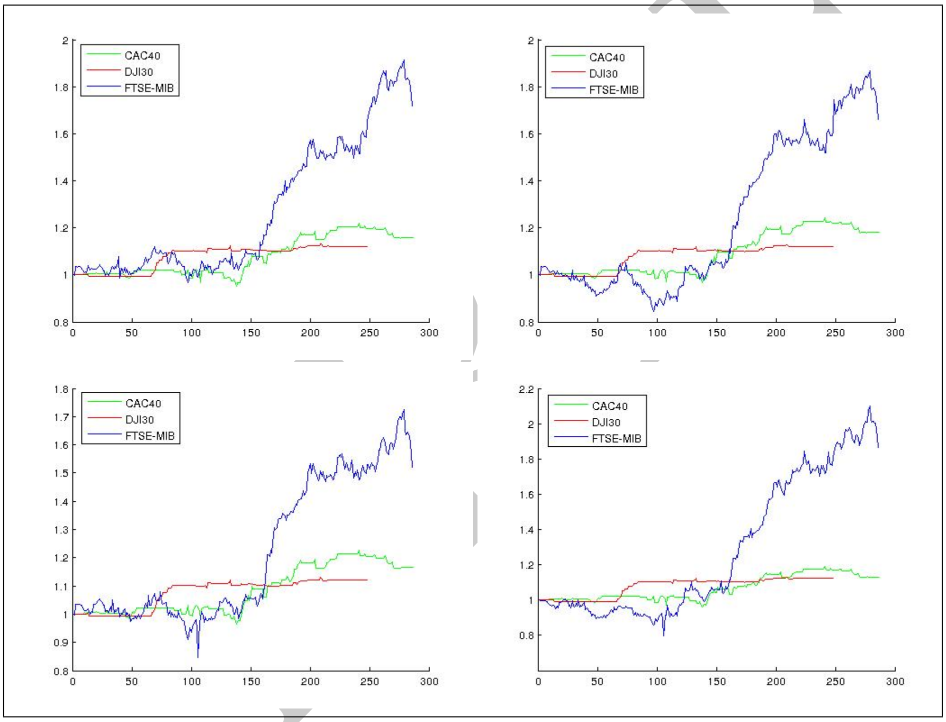


Fig. 3: Portfolio's yield curve of model 1 (top-left), 2 (top-right), 3 (bottom-left), 4 (bottom-right). There is less data for DJ30.

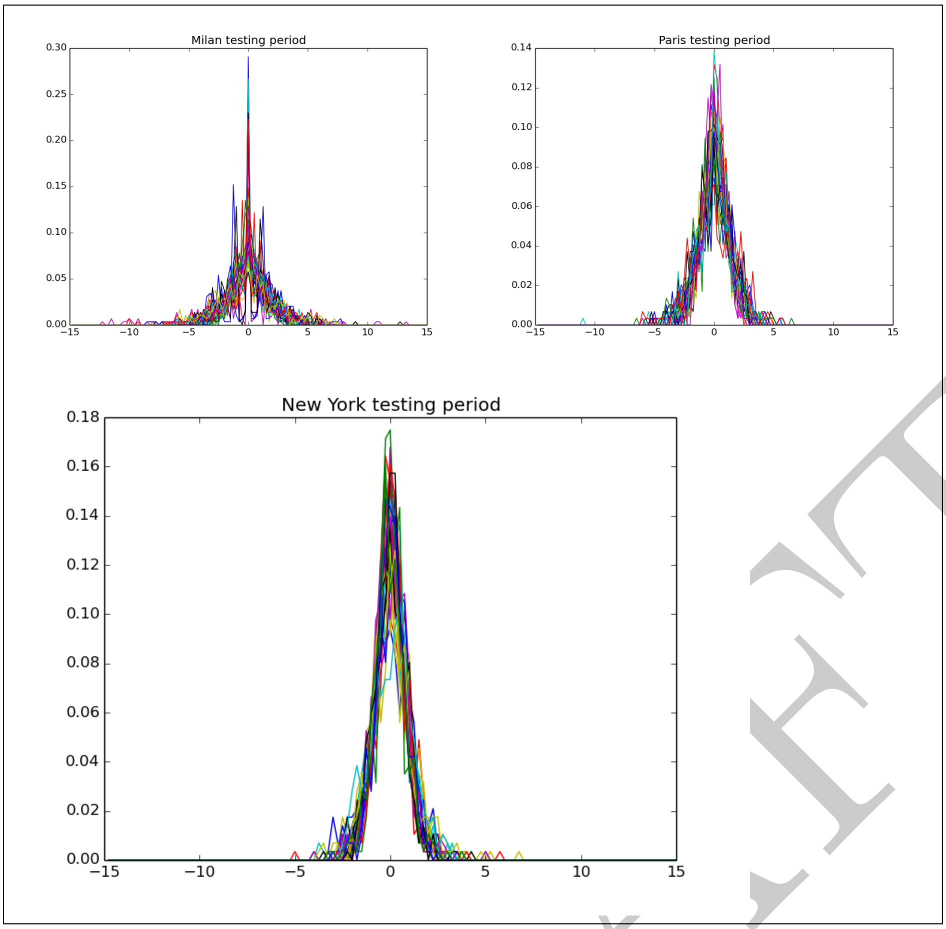


Fig. 4: Starting from top-left we have FTSE-MIB, CAC-40, DJI-30. In each of these plots we have all the frequencies for every index stock component. The frequencies has been measured in classes of 0.125 radius, the central point in each class is a multiple of 0.25 starting from -15 to +15. Note that the scales in the three graphs are different. The most high frequency in the DJ is 18%, in FTSE-MIB is 30% and in CAC is 14%.

7 Conclusions and Future Works

In this paper, we presented a detailed description of a portfolio optimizer under a limited budget constraint, previously presented in [18]. The approach exploits a set of neural networks to predict the behavior of each stock in a 3-day window; it subsequently makes use of a multi-objective evolutionary algorithm to find a suitable compromise over objectives compromising risk and rewards, described by four different models; and finally selects one of the compromises, on the basis of the least number of transactions needed to achieve the target portfolio, starting from its current one. The framework is tested on real-world data from three different

DRAFT

stock markets (Milan, New York, Paris), and it is proven able to outperform both a classical buy-and-hold strategy and a random walk.

No matter the model used for the multi-objective optimization step, the experimental analysis suggest that the performance of the framework is heavily relying upon the oracles' ability to predict future trends. The tool has shown good performances in high variance markets, while to improve its performance in markets characterized by small variance, the oracles' structure has to be modified. We are currently working on improved oracles, based on recurrent deep networks [54], because this architecture is able to model the dependency of a temporal sequence, and using different time-frame forecast regressions could improve the general forecasting ability of the framework.

References

1. Abarbanell JS, Bushee BJ (1997) Fundamental analysis, future earnings, and stock prices. Journal of Accounting Research pp 1–24
2. Alexander C (2009) Market Risk Analysis, Value at Risk Models, vol 4. John Wiley & Sons
3. Anagnostopoulos K, Mamanis G (2010) A portfolio optimization model with three objectives and discrete variables. Computers & Operations Research 37(7):1285–1297
4. Anagnostopoulos K, Mamanis G (2011) The mean–variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. Expert Systems with Applications 38(11):14,208–14,217
5. Association MF, et al (2009) Sound practices for hedge fund managers. URL `http://www.managedfunds.org`
6. Barberis N, Thaler R (2003) A survey of behavioral finance. Handbook of the Economics of Finance 1:1053–1128
7. Beasley JE, Meade N, Chang TJ (2003) An evolutionary heuristic for the index tracking problem. European Journal of Operational Research 148(3):621–643
8. Bodie Z, Kane A, Marcus AJ (2014) Investments. McGraw-Hill Higher Education
9. Borgelt C, Kruse R (2002) Induction of association rules: Apriori implementation. In: Compstat, Physica-Verlag HD, pp 395–400
10. Branke J, Scheckenbach B, Stein M, Deb K, Schmeck H (2009) Portfolio optimization with an envelope-based multi-objective evolutionary algorithm. European Journal of Operational Research 199(3):684–693
11. Bulkowski TN (2011) Encyclopedia of chart patterns, vol 225. John Wiley & Sons
12. Cao LJ, Tay FE (2003) Support vector machine with adaptive parameters in financial time series forecasting. Neural Networks, IEEE Transactions on 14(6):1506–1518
13. Chang TJ, Meade N, Beasley JE, Sharaiha YM (2000) Heuristics for cardinality constrained portfolio optimisation. Computers & Operations Research 27(13):1271–1302
14. Coello CAC, Lamont GB, Van Veldhuisen DA (2007) Evolutionary algorithms for solving multi-objective problems. Springer

DRAFT

15. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation, IEEE Transactions on 6(2):182–197
16. Deboeck G (1994) Trading on the edge: neural, genetic, and fuzzy systems for chaotic financial markets, vol 39. John Wiley & Sons
17. Dechow PM, Hutton AP, Meulbroek L, Sloan RG (2001) Short-sellers, fundamental analysis, and stock returns. Journal of Financial Economics 61(1):77–106
18. Deplano I, Squillero G, Tonda A (2016) Portfolio optimization, a decision-support methodology for small budgets. In: Applications of Evolutionary Computation, Springer, pp 58–72
19. Devadoss AV, Ligori TAA (2013) Forecasting of stock prices using multi layer perceptron. International Journal of Computing Algorithm 2:440–449
20. Edwards RD, Magee J, Bassetti W (2007) Technical analysis of stock trends. CRC Press
21. Fama EF (1970) Efficient capital markets: A review of theory and empirical work*. The journal of Finance 25(2):383–417
22. Fama EF, French KR (2004) The capital asset pricing model: Theory and evidence. Journal of Economic Perspectives 18:25–46
23. Fu Tc, Chung Fl, Ng V, Luk R (2001) Pattern discovery from stock time series using self-organizing maps. In: Workshop Notes of KDD2001 Workshop on Temporal Data Mining, Citeseer, pp 26–29
24. Gabrielsson P, König R, Johansson U (2013) Evolving hierarchical temporal memory-based trading models. Springer
25. Goodfellow I, Bengio Y, Courville A (2016) Deep learning, URL `http://www.deeplearningbook.org`, book in preparation for MIT Press
26. Graham B, Dodd DL (2008) Security analysis. McGraw-Hill New York
27. Grossman S (1976) On the efficiency of competitive stock markets where trades have diverse information. The Journal of finance 31(2):573–585
28. Grossman SJ, Stiglitz JE (1980) On the impossibility of informationally efficient markets. The American economic review 70(3):393–408
29. Haykin S, Lippmann R (1994) Neural networks, a comprehensive foundation. International Journal of Neural Systems 5(4):363–364
30. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1026–1034
31. Hochreiter R (2010) Evolutionary multi-stage financial scenario tree generation. In: Applications of Evolutionary Computation, Springer, pp 182–191
32. Høyland K, Wallace SW (2001) Generating scenario trees for multistage decision problems. Management Science 47(2):295–307
33. Ineichen A, Silberstein K (2008) Aimas roadmap to hedge funds. Alternative Investment Management Association, London
34. Ineichen AM (2002) Absolute returns: The risk and opportunities of hedge fund investing, vol 195. John Wiley & Sons
35. Jiang ZQ, Zhou WX, Sornette D, Woodard R, Bastiaensen K, Cauwels P (2010) Bubble diagnosis and prediction of the 2005–2007 and 2008–2009 chinese stock market bubbles. Journal of economic behavior & organization 74(3):149–162

DRAFT

36. Kahneman D, Tversky A (1979) Prospect theory: An analysis of decision under risk. Econometrica: Journal of the econometric society pp 263–291
37. Kim D, Kim C (1997) Forecasting time series with genetic fuzzy predictor ensemble. Fuzzy Systems, IEEE Transactions on 5(4):523–535
38. Kimoto T, Asakawa K, Yoda M, Takeoka M (1990) Stock market prediction system with modular neural networks. In: Neural Networks, 1990., 1990 IJCNN International Joint Conference on, IEEE, pp 1–6
39. Koller T, Goedhart M, Wessels D (2015) Valuation: measuring and managing the value of companies, 6th edn. Wiley Finance, john Wiley and sons
40. Laboissiere LA, Fernandes RA, Lage GG (2015) Maximum and minimum stock price forecasting of brazilian power distribution companies based on artificial neural networks. Applied Soft Computing 35:66–74
41. Lo AW (2004) The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. Journal of Portfolio Management, Forthcoming
42. Loginov A, Heywood MI (2013) On the utility of trading criteria based retraining in forex markets. Springer
43. Lohpetch D, Corne D (2010) Outperforming buy-and-hold with evolved technical trading rules: Daily, weekly and monthly trading. In: Applications of Evolutionary Computation, Springer, pp 171–181
44. Markowitz H (1952) Portfolio selection*. The journal of finance 7(1):77–91
45. Markowitz HM (1968) Portfolio selection: efficient diversification of investments, vol 16. Yale university press
46. Matz L, Neu P (2006) Liquidity risk measurement and management: a practitioner's guide to global best practices, vol 408. John Wiley & Sons
47. Merton RC (1973) An intertemporal capital asset pricing model. Econometrica: Journal of the Econometric Society pp 867–887
48. Michalak K (2015) Selecting best investment opportunities from stock portfolios optimized by a multiobjective evolutionary algorithm. In: Proceedings of the 2015 on Genetic and Evolutionary Computation Conference, ACM, pp 1239–1246
49. Michalak K, Filipiak P, Lipinski P (2013) Usage patterns of trading rules in stock market trading strategies optimized with evolutionary methods. Springer
50. Neri F (2011) Learning and predicting financial time series by combining natural computation and agent simulation. In: Applications of Evolutionary Computation, Springer, pp 111–119
51. Nguyen D, Widrow B (1990) Improving the learning speed of 2-layer neural networks by choosing. In: Initial Values of the Adaptive Weights, International Joint Conference of Neural Networks, pp 21–26
52. Oberlechner T (2001) Importance of technical and fundamental analysis in the european foreign exchange market. International Journal of Finance & Economics 6(1):81–93
53. Otero FE, Kampouridis M (2014) A comparative study on the use of classification algorithms in financial forecasting. In: Applications of Evolutionary Computation, Springer, pp 276–287
54. Pascanu R, Gulcehre C, Cho K, Bengio Y (2013) How to construct deep recurrent neural networks. arXiv preprint arXiv:13126026
55. Rather AM, Agarwal A, Sastry V (2015) Recurrent neural network and a hybrid model for prediction of stock returns. Expert Systems with Applications 42(6):3234–3241

DRAFT

56. Ross SA (1976) The arbitrage theory of capital asset pricing. Journal of economic theory 13(3):341–360
57. Shefrin H, Statman M (2000) Behavioral portfolio theory. Journal of financial and quantitative analysis 35(02):127–151
58. Sheppard K (2010) Financial econometrics notes. University of Oxford
59. Shiller RJ (1999) Human behavior and the efficiency of the financial system. Handbook of macroeconomics 1:1305–1340
60. Shiller RJ (2003) From efficient markets theory to behavioral finance. The Journal of Economic Perspectives 17(1):83–104
61. Simon HA (1955) A behavioral model of rational choice. The quarterly journal of economics pp 99–118
62. Simon HA (1982) Models of bounded rationality: Empirically grounded economic reason, vol 3. MIT press
63. Srikant R, Vu Q, Agrawal R (1997) Mining association rules with item constraints. In: KDD, vol 97, pp 67–73
64. Swisher P, Kasten GW (2005) Post-modern portfolio theory. JOURNAL OF FINANCIAL PLANNING-DENVER- 18(9):74
65. Tapia MGC, Coello CAC (2007) Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In: IEEE congress on evolutionary computation, vol 7, pp 532–539
66. Trippi RR, Turban E (1992) Neural networks in finance and investing: Using artificial intelligence to improve real world performance. McGraw-Hill, Inc.
67. Tversky A, Kahneman D (1992) Advances in prospect theory: Cumulative representation of uncertainty. Journal of Risk and uncertainty 5(4):297–323
68. Vassiliadis V, Thomaidis N, Dounias G (2011) On the performance and convergence properties of hybrid intelligent schemes: application on portfolio optimization domain. In: Applications of Evolutionary Computation, Springer, pp 131–140
69. Ye Y, Chiang CC (2006) A parallel apriori algorithm for frequent itemsets mining. In: Software Engineering Research, Management and Applications, 2006. Fourth International Conference on, IEEE, pp 87–94

DRAFT

View publication stats