A Robust Genetic Programming Model for a Dynamic Portfolio Insurance Strategy

Siamak Dehghanpour, Akbar Esfahanipour
Department of Industrial Engineering and Management Systems
Amirkabir University of Technology
Tehran, Iran
Siamak.dehghanpoor@yahoo.com; esfahaa@aut.ac.ir

Abstract— **In this paper, we propose a robust genetic programming model for a dynamic strategy of stock portfolio insurance. With portfolio insurance strategy, we need to allocate part of the money in risky asset and the other part in risk-free asset. Our applied strategy is based on constant proportion portfolio insurance (CPPI) strategy. For determining the amount for investing in risky assets, the critical parameter is a constant risk multiplier which is used in traditional CPPI method so that it may not reflect the changes occurring in market condition. Thus, we propose a model in which, the risk multiplier is calculated with robust genetic programming. In our model, risk variables are used to generate equation trees for calculating the risk multiplier. We also implement an artificial neural network to enhance our model's robustness. We also combine the portfolio insurance strategy with a well-known portfolio optimization model to get the best possible portfolio weights of risky assets for insurance. Experimental results using five stocks from New York Stock Exchange (NYSE) show that our proposed robust genetic programming model outperforms the other two models: the basic genetic programming for portfolio insurance without portfolio optimization, and the basic genetic programming for portfolio insurance with portfolio optimization.**

Keywords—Robust Genetic Programming (RGP); Dynamic portfolio insurance strategy; Portfolio Optimization model; Constant Proportion Portfolio Insurance (CPPI);

I. Introduction

In the last decades countless studies tried to present a model for investors that not only would generate the best payoff, but also reduces the risk of the investment as low as possible. The theory of Markowitz [1] was the base of the most of these studies. Markowitz was the first person who suggested that investors should consider both return and risk of the investing in their decision making. But in the real world an investor encounters hundreds of stocks with different characteristics. In this situation, there are a lot of information to use for decision making which needs models to solve the problem.

One of the most important techniques in constructing a portfolio is to divide it to risk-free assets and risky assets. This is the original idea of the popular constant proportion portfolio insurance (CPPI) strategy. The CPPI strategy uses the investor's risk preference to calculate the amount that should be invested in the risky assets and clearly, the rest invested in risk-free assets [2]. Since the risk multiplier in the CPPI strategy is basically determined by investor's personal view, it may not work well in some unusual market movements. For instance, consider the situation where there is a sudden downward movement in the stock's price, then with the conventional CPPI strategy the insurance model cannot prevent the loss from the investment in risky assets in such a short time. As a result, a dynamic model should be developed that can adopt itself to such market conditions. Chen et al. proposed a dynamic model strategy called dynamic proportion portfolio insurance (DPPI) that uses genetic programming (GP) to calculate the risk multiplier according to market condition [3]. Then they compared the performance of the proposed strategy with conventional CPPI strategy which showed their model outperformed the CPPI strategy.

Chen and Benjamin introduced a goal-directed strategy based on the popular CPPI [4]. They combined this goal-directed model with regular CPPI and eventually proposed piecewise linear goal-directed CPPI strategy. Balder proposed a model for portfolio insurance that considered the discrete-time trading in the CPPI strategy [5].

In the case of comparison of the CPPI strategy with other portfolio insurance strategies Jiang proposed a VaR-based portfolio insurance strategy which is called VBPI strategy, he compared the effectiveness of this model with the other strategies such as CPPI and the buy and hold strategy [6].

In this paper, we present a new strategy using robust genetic programming for portfolio insurance and then we combine this strategy with an optimization model to obtain a model that insures a portfolio and optimizes it at the same time. Then we evaluate this model's performance by comparing it with two previous methods: basic dynamic portfolio insurance strategy using GP (the model proposed by Chen et al.), and dynamic portfolio insurance strategy using GP with simultaneous optimization. The results show that our new strategy has better outcome than the other two methods.

The rest of this paper is organized as follows. Section 2 presents research background which introduces CPPI strategy,

978-1-5090-5795-5/17/$31.00 ©2017 IEEE

Markowitz model, genetic programming and our applied approach for robustness. Section 3 describes the proposed model in this paper. The data and experimental results are stated in section 4. Finally, the conclusion is presented in section 5.

II. RESEARCH BACKGROUND

A. Constant Proportion Portfolio Insurance (CPPI)

One of the most popular strategies for portfolio insurance is CPPI. It was introduced by Black and Jones in 1987. Investing with this strategy contains risk-free assets (usually treasury bills) and risky assets such as stocks or bonds. The amount that should be allocated to risky assets is calculated with the formula (1).

$$K = M * (A - F), \tag{1}$$

in which A is the current value of a portfolio, and F stands for floor which is the lowest value that is acceptable for portfolio. The difference between current value and floor is known as cushion. M is the risk multiplier that in this strategy is defined by investor's personal view and is commonly between 3 and 6. The most important element of this equation is M (i.e., risk multiplier), the more risk averse the investor is, the less he or she chooses the M's value. Then the amount of K should be invested in risky assets and the rest, in the risk-free assets. This strategy is easy to understand and implement for the investors that is why it is so popular around the world. For example, consider the state that current value of a portfolio is \$1000 and the floor has been determined at \$950 and the risk multiplier is 3. According to this simple strategy, the investor should invest 3*(1000-950) =\$150 of his or her money in the risky assets and the remaining \$850 in risk-free assets [2].

B. Markowitz Model

In the classic portfolio optimization problem, the objective is to reduce the risk of the investment and/or to increase the return of the investing. Variance is one of the most commonplace risk measures used in this kind of problems. Markowitz introduced variance as a risk measure for the first time in 1952, his proposed model is described as formulation of (2)-(5).

$$Min\ \sigma^2 = \sum_{j=1}^{n} \sum_{i=1}^{n} W_i * W_j * \sigma_{i,j}, \tag{2}$$

s.t.

$$\sum_{i=1}^{n} W_i = 1 \tag{3}$$

$$\sum_{i=1}^{n} W_i r_i \geq R \tag{4}$$

$$W_i \geq 0 \tag{5}$$

where W_i is the weight of the ith stock and $\sigma_{i,j}$ is the covariance between ith and jth stock. And r_i is the return of each stock and R is the investor's expected return from the portfolio. In this model the objective, as equation (2), is to choose proper weights for each of the stocks in the portfolio in a way that with a fixed expected rate of return from that portfolio, we minimize the risk of the investment. Inequality (3) states that total weights of the stocks should be equal to 1. Inequality (4) makes sure that the portfolio rate of return is equal or greater than the expected rate of return. And, inequality (5) states that short selling is not allowed, that means each stock in the portfolio must have a positive weight.

C. Genetic Programming

Genetic Programming (GP) is an extension of classic Genetic Algorithm (GA) with an important difference that offers different kind of solution candidates. In GP, solution candidates are shown in form of trees. GP was first introduced by Koza in 1990, and was then developed through different studies [7], [8]. The structure of GP is similar to that of GA: after initialization of population, random changes can occur either by changing the functions or the subtrees.

For applying a GP framework, the required elements are as follows: nodes of trees, population initialization, selection method, genetic operators such as crossover and mutation, fitness function and termination condition [3]. These elements are briefly described next.

Nodes: the nodes in the tree structure are defined in two categories: functions and terminals. Functions set usually consists of different types of mathematical or computer functions. Terminals set includes variables and constants according to the requirements of the problem on hand.

Initialization: GP starts with randomly produced initial population.

Selection: selection is a process in which it determines how to select a parent for reproduction. Usually better parents are expected to produce better offsprings. The roulette wheel selection method is the most popular selection method.

Crossover: crossover is a process that generates offsprings from chosen parents by substituting their subtrees.

Mutation: mutation occurs in a way that a limited number of parents changes randomly.

Termination condition: the most common termination conditions are: exact number of reproduction, fitness target and fitness convergence [9].

D. Robustness

Robust optimization, first introduced by Mulvey et al., has been adopted as an effective tool for optimal design and management in uncertain environments [10]. In general this term refers to the potential of a program to maintain its applicability in the presence of external or internal disturbance [11], [12]. When we are dealing with disturbance from internal sources it is called genotypic robustness, in which the purpose is to make the fitness insensitive to disturbance from genetic functions and operators. But, phenotypic robustness deals with external changes. In this paper, we are concern about the latter type of robustness [13], [14], [15], [16]. This means we are dealing with historical data of stocks that are considered external source for our robustness.

III. The Proposed Model

As mentioned in previous section, the risk multiplier in CPPI strategy is determined by investor's personal view and is constant during the insurance period [17], [18]. This method could lead to a serious loss for the investors in specific market conditions such as high volatility in short period of times. Hence, in their model Chen et al. proposed a dynamic model in which the risk multiplier is calculated using GP so the multiplier would not be constant all the time. Thus, in any market conditions with risk variables being considered for GP an adapted risk multiplier will be calculated. Consider the formula (6) for this new strategy.

$$K = M(\tau) * (A - F), \tag{6}$$

In summary, our new proposed model determines the risk multiplier (M) by τ which is a set of constants and risk variables. The value of M is obtained through a robust genetic programming which has been described above. Then the amount of money which should be invested in risky assets (i.e., K in the formula (6), should be allocated to different stocks. This allocation will be determined by application of well-known Markowitz model which has been described in section II.B. Thus, our proposed model is a robust genetic programming for portfolio insurance and at the same time, we run an optimization model to gain the best possible portfolio weights. And, finally we define three sets of models for comparison: GP1: basic genetic programming for portfolio insurance without portfolio optimization. GP2: basic genetic programming for portfolio insurance with portfolio optimization and finally, our proposed model GP3: robust genetic programming for portfolio insurance with portfolio optimization.

A. The Terminal Set

Terminal set of our GP is risk variables including both market volatility factors and technical indicators. The beta in the capital asset pricing model (CAPM) usually considered constant, thus, in this study risk-free rate and market index are used instead of the beta in addition to exchange rate as market volatility factors. We also use relative strength index (RSI), stochastic oscillator (STO) and Williams %R (%R) as our three technical indicators in this study. So, for terminal set of our GP we consider market index, exchange rate (US dollar to Euro), Williams %R (%R), relative strength index (RSI), stochastic oscillator (STO) and T-bill (10-year). These risk variables and their abbreviations are listed in Table 1. Note that since the risk variables have different range of values, they are normalized to the same range before being used in GP.

TABLE I. The Risk Variables of Terminal Set and Their Abbreviations.

Variable	Abbreviation
Market index	NYSE (NYSE composite)
Exchange rate	US2Euro (US dollar to Euro)
Williams %R	(R %)
Relative strength index	RSI
Risk-free rate	T-bill (10-year)
Stochastic oscillator	STO-D

B. The Function Set

In this paper we use the four commonly used arithmetic operators as follows as used by [19]:

+, - , /, *.

C. Fitness

The fitness function for evaluating our GP is calculated by formula (7).

$$f = \frac{r+1}{\sigma+1} \quad , \tag{7}$$

In the formula (7), r is the portfolio rate of return and σ is the portfolio standard deviation. To avoid encountering negative fitness values, the rate of return is added by one in the numerator. Also, to prevent a zero denominator the standard deviation is also added by one.

D. Robustness

In this study, we use robust genetic programming to make our model's fitness, insensitive to daily changes in stock's price. The purpose of our robust model is to make adjustments in stocks historical prices. For instance, if in one period a big sudden upward change occurs in price of a stock, we cannot interpret that the stock is the better choice than the others. Thus, the use of robustness helps us to not base our decision on noise data and data that could damage the accuracy of our decision. We implement an artificial neural network (ANN) for prediction of daily stock price. Since, the objective of any investments is to make it more profitable, we should adjust the probability of up and down changes in stock's price. For this purpose, we first estimate the probability of the fluctuations and then we define 14-days periods. And, with the mean and minimum-maximum strategy we make this adjustment. Also, we use Backpropagation learning algorithm for our ANN's learning process.

With the use of artificial neural network we aim to reduce the effects of random stock price oscillations. For instance when we encounter a sudden large increase in the stock price, and this has happened only once in some period of time, then this change in price should not affect the decision about buying or selling of that stock. Thus, the following inputs are used for each stock in the specific period of time for our robust genetic programming:

- Minimum price.

- Maximum price.
- Difference between the lowest and the second lowest prices.
- Difference between the highest and the second highest prices.
- Average of stock's price.

For our prediction, we used multi-layer artificial neural network based on the historical data from our five stocks. 70 percent of our data are used for training, 15 percent for testing and 15 percent for validation. The well-known Levenberg method is implemented for the prediction. This network uses historical data and makes prediction possible. By implementing the sensitivity analysis we chose parameters of our artificial neural network as presented in Table 2.

For evaluating prediction accuracy of our ANN, we used mean absolute error (MAE), root-mean-square error (RMSE) and mean-absolute percentage error (MAPE) [20]. The results are presented in table 3 which shows the value of 15 for number of hidden layer neurons has the best accuracy (i.e. the least error).

In summary Fig. 1, shows the different steps in our proposed robust genetic programming model for a dynamic portfolio insurance strategy.

TABLE II. THE PARAMETERS OF THE MULTI-LAYER ARTIFICIAL NEURAL NETWORK IMPLEMENTED IN THIS STUDY

parameters	values
Activation function	sigmoid
Number of neurons	15
Number of layers	2
Training set	70%
Validation set	15%
Testing set	15%
Maximum number of epoch	100

TABLE III. THE RESULTS OF PREDICTION ACCURACY FOR OUR ANN.

Number of hidden layer neurons	RMSE	MSE	MAPE
10	1.23	1.6	0.99
15	0.95	1.04	0.98
20	0.99	1.23	0.98
30	1.24	1.33	0.99

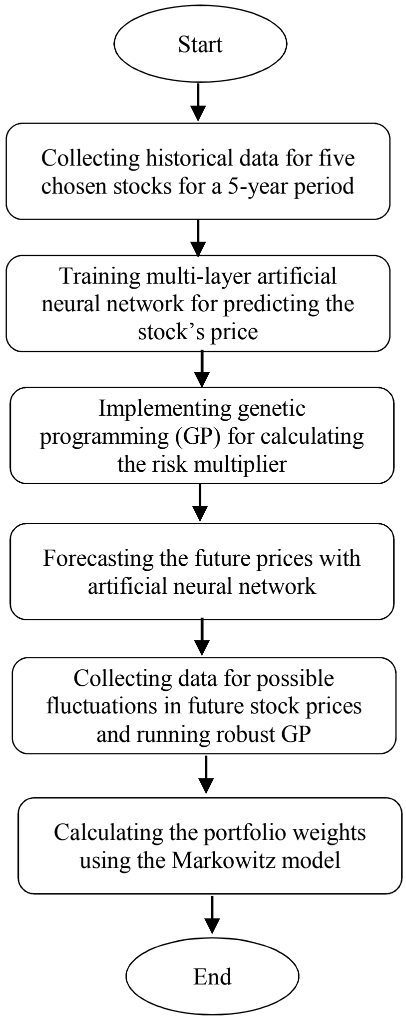

Fig. 1. Our proposed robust genetic programming model for a dynamic portfolio insurance strategy

IV. EXPERIMENTAL RESULTS

A. Data

We choose five stocks from different activity sectors of New York Stock Exchange (NYSE). These five stocks are IBM, MSI, LEE, UPS and XOM. Table 4 presents these stocks with details.

TABLE IV. THE FIVE STOCKS THAT USED IN THIS STUDY.

Symbol	Company	Industry
IBM	Int. Business machines	Computer hardware
MSI	Motorola Solutions, Inc.	Communications & Networking
LEE	Lee enterprises	Printing and publishing
UPS	United parcel service	Trucking
XOM	Exxon mobil	Oil and Gas

The testing and training periods used in our study are shown in Table 5. As we used historical data for our five chosen stocks from 2012 to the end of 2016, similar to [3], we divided this 5 years to 10 six-month periods and we assign them for our training and testing periods as shown in Table 5. We divided our data set into training and testing periods in order to evaluate the fitness of our robust GP in different periods.

TABLE V. THE TESTING AND TRAINING PERIODS OF OUR ROBUST GENETIC PROGRAMMING.

Training period	Testing period
2012.01-06	2012.07-12
2012.07-12	2013.01-06
2013.01-06	2013.07-12
2013.07-12	2014.01-06
2014.01-06	2014.07-12
2014.07-12	2015.01-06
2015.01-06	2015.07-12
2015.07-12	2016.01-06
2016.01-06	2016.07-12

B. *Parameters*

In table 6, the parameters of the robust genetic programming are described.

TABLE VI. PARAMETERS OF ROBUST GENETIC PROGRAMMING.

Parameter	Value
Population size	500
Maximum tree depth	8
Selection method	Roulette wheel
Crossover method	Subtree crossover
Mutation method	Subtree replacement
Number of iterations	10
Crossover rate	0.7
Mutation rate	0.3

C. *Results*

In this section, we present the fitness values for testing periods of our robust genetic programming and we compare them with fitness values obtained from basic genetic programming without optimization and basic genetic programming with portfolio optimization. In order to avoid different results due the random nature of the GP, we run the three models 30 times. The reported results in this section are the average fitness values of these 30 runs. The result of this comparison shows this new model's fitness values are better than the other two. This implies our proposed model yields superior results in comparison with the other two models. From now on, in this study we consider our proposed robust genetic programming with optimization as GP3, the basic genetic programming with optimization as GP2 and the basic genetic programming without optimization as GP1. The fitness values for the testing periods for each of these three models are presented in table 7.

As it is shown in Table 7, our proposed robust genetic programming model (GP3) has the better performance compare to the other two models. Fig. 2, presents this comparison.

In Fig. 3, the changes in portfolio value are being presented over time. As it is shown, the robust genetic programming (GP3) yields the better values for the portfolio.

TABLE VII. COMPARISON OF FITNESS VALUES OF THE THREE MODELS. EACH MODEL HAS BEEN RUN 30 TIMES. THE REPORTED VALUES ARE THE AVERAGE FITNESS VALUES OF 30 TIMES.

Testing period	GP1's fitness	GP2's fitness	GP3's fitness
2012.07-12	1.0314	1.0315	1.0323
2013.01-06	1.0316	1.0319	1.0322
2013.07-12	1.0316	1.0319	1.0323
2014.01-06	1.0317	1.0319	1.0324
2014.07-12	1.0315	1.0318	1.0325
2015.01-06	1.0315	1.0317	1.0323
2015.07-12	1.0316	1.0319	1.0324
2016.01-06	1.0315	1.0318	1.0325
2016.07-12	1.0317	1.0319	1.0325

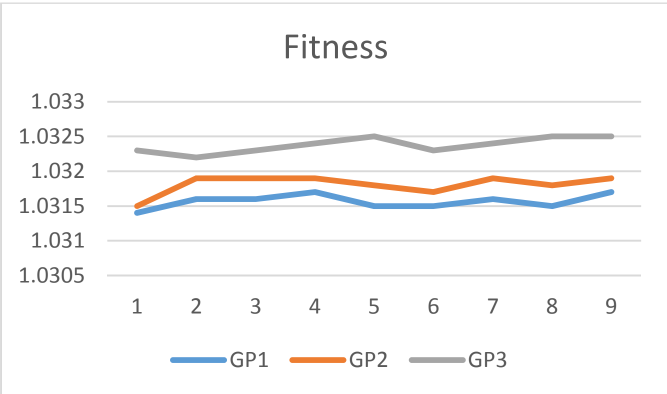

Fig. 2. Comparison of fitness values for three models. GP1: portfolio insurance strategy using genetic programming, GP2: portfolio insurance strategy using genetic programming and portfolio optimization, GP3: portfolio insurance strategy using robust genetic programming with portfolio optimization.

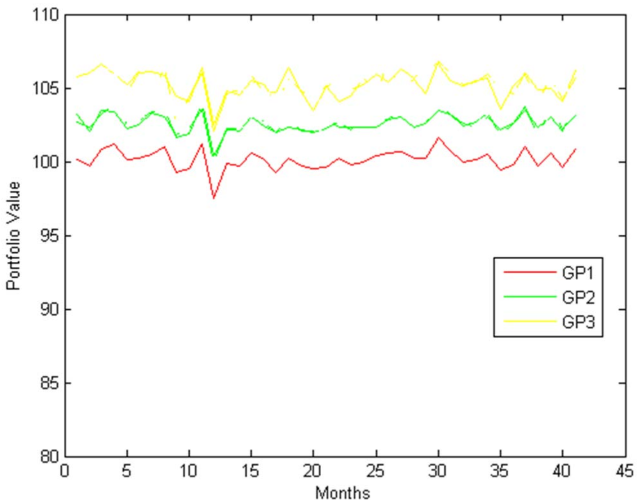

Fig. 3. Changes in portfolio value with floor of 95 and initial value of 100. GP1: portfolio insurance strategy using genetic programming, GP2: portfolio insurance strategy using genetic programming and portfolio optimization, GP3: portfolio insurance strategy using robust genetic programming with portfolio optimization.

We implement t-test on fitness values of our optimized robust genetic programming model and the other two models. The outcome of the t-test shows that our model has the best performance of the three models. This means that combining optimization methods with robust genetic programming model is a better choice for investors. The t-test is calculated using MATLAB t-test function, with 95% confidence level, and the null hypothesis is that the mean fitness values of our proposed model is less than the mean fitness values of the other two models. The outcome indicates that the t-test does reject this null hypothesis.

V. Conclusion

In this paper, we propose a robust genetic programming model for portfolio insurance. The proposed model can determine the risk multiplier of Constant Proportion Portfolio Insurance (CPPI) strategy using Genetic Programming (GP) to consider different market situations. To be more practical, we use Markowitz's portfolio optimization theory to determine the best portfolio weights for investors. Our strategy is to generate the risk multiplier of the popular CPPI strategy with genetic programming by using some risk variables in order to use the best risk multiplier according to different market conditions. Then with this dynamic strategy, we determine how much of the investor's money should be invested in risky assets and how much of it in risk-free assets. We implement robust GP, in order to make our model more effective and reliable for future stock price prediction. The next step that we used is to run the well-known Markowitz optimization model in order to determine the portfolio weights to be invested in risky assets. To show the performance of the proposed model, we ran three sets of model using data of five stocks from New York Stock Exchange (NYSE). The experimental results show that our new model outperforms the other two models: the basic genetic programming for portfolio insurance without portfolio optimization, and the basic genetic programming for portfolio insurance with portfolio optimization.

References

[1] H. Markowitz, Portfolio selection, Wiley, New York, 1959.

[2] F. Black, R. Jones, "Simplifying portfolio insurance," Journal of Portfolio Management, 14(1), 48–51, 1987.

[3] JS. Chen, CL. Chang, "Dynamical proportion portfolio insurance with genetic programming," Advances in Natural Computation, 3611, 2005.

[4] JS. Chen, PL. Benjamin, "Piecewise nonlinear goal-directed CPPI strategy," Journal of Expert Systems with Applications 33, 857-869, 2007.

[5] S. Balder, M. Brandl, A. Mahayani, "Effectiveness of CPPI strategies under discrete-time trading," Journal of Economic Dynamics & Control, 33, 204-220, 2009.

[6] C. Jiang, Y. Ma, Y. An, "The effectiveness of the VaR-based portfolio insurance strategy: An empirical analysis," International Review of Financial Analysis, 18(4), 185-197, 2009.

[7] JR. Koza, Introduction to genetic programming. In K. J. Kinnear (Ed.), Advances in Genetic Programming (pp. 21–42). Cambridge, MA: The MIT Press, 1994b.

[8] W. Banzhaf, P. Nordin, RE. Keller, and FD. Francone, "Genetic programming – an introduction : On the automatic evolution of computer programs and its applications," Morgan Kaufmann, 1998.

[9] M. Brameier, W. Banzhaf, "A Comparison of Linear Genetic Programming and Neural Networks in Medical Data Mining," IEEE Trans. Evol. Comput. 5, 17–26, 2001.

[10] A. Esfahanipour, S. Mousavi, "A genetic programming model to generate risk-adjusted technical trading rules in stock markets," Expert Systems Applications, 38, pp. 8438-8445, 2011.

[11] S. Mousavi, A. Esfahanipour, MHF. Zarandi, "A novel approach to dynamic portfolio trading system using multitree genetic programming," Knowledge Based Systems, 66:68–81, 2014.

[12] JM. Mulvey, RJ. Vanderbei, SA. Zenios, "Robust optimization of large-scale systems," Operational Research, 43, 264–281, 1995.

[13] H. Guo, LB. Jack, and AK. Nandi, "Feature generation using genetic programming with application to fault classification," IEEE Trans. Syst., Man, Cybern. B, vol. 35, no. 1, pp. 89–99, Feb. 2005.

[14] Y. Lin and B. Bhanu, "Evolutionary feature synthesis for object recognition," IEEE Trans. Syst., Man, Cybern. C, vol. 35, no. 2, pp. 156–171, May 2005.

[15] JM. Berutich, F. López, F. Luna, D. Quintana, "Robust technical trading strategies using GP for algorithmic portfolio selection," Expert Systems Applications, 46:307–315,2016.

[16] C. Chen, RH. Kwon, "Robust portfolio selection for index tracking," Computers& Operations Research, 39:829–837, 2012.

[17] E. Hakano, R. Kopprasch, and E. Roman, "Constant proportion portfolio insurance for fixed-income investment," Journal of Portfolio Management, 15(4), 58–66, 1989.

[18] T. Estep, M. Kritzman, " TIPP: Insurance without complexity," Journal of Portfolio Management, 14(4), 38–42, 1988.

[19] C. Fyfe, JP. Marney, and HFE. Tarbert, "Technical analysis versus market efficiency: A genetic programming approach," Appl. Finance Econ., 9(2), 183–191, 1999.

[20] JS. Armstrong, Evaluating forecasting methods. In J. S. Armstrong (Ed.), Principles of Forecasting: A Handbook for Researchers and Practitioners. New York: Kluwer, 2001.