Expert Systems with Applications 42 (2015) 7684–7697

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Review

Genetic algorithms and Darwinian approaches in financial applications: A survey

Rubén Aguilar-Rivera *, Manuel Valenzuela-Rendón [1], J.J. Rodríguez-Ortiz

Tecnológico de Monterrey, National School of Engineering and Sciences, Ave. Eugenio Garza Sada 2501, C.P. 64849 Monterrey, Mexico

ARTICLE INFO

Article history:
Available online 11 June 2015

Keywords:
Genetic algorithms
Evolutionary computation
Finance
Portfolio optimization
Survey

ABSTRACT

This article presents a review of the application of evolutionary computation methods to solving financial problems. Genetic algorithms, genetic programming, multi-objective evolutionary algorithms, learning classifier systems, co-evolutionary approaches, and estimation of distribution algorithms are the techniques considered. The novelty of our approach comes in three different manners: it covers time lapses not included in other review articles, it covers problems not considered by others, and the scope covered by past and new references is compared and analyzed. The results concluded the interest about methods and problems has changed through time. Although, genetic algorithms have remained the most popular approach in the literature. There are combinations of problems and solutions methods which are yet to be investigated.

1. Introduction

Financial systems have a remarkable impact on society. For example, indexes such as the DJI (Dow Jones) or the S&P500 are commonly used to measure economic buoyancy. Financial markets provide a comfortable way to generate profit from current wealth and protect investors from adverse effects like inflation. Some authors (Cerda, 2005) have predicted an impending crisis of social security systems, a scenario where personal investment and pension funds will be of crucial importance.

On the other hand, the idea to exploit financial markets to make profit has always been in the mind of investors. For example, Lo, Mamaysky, and Wang (2000) made a description of the technical analysis (TA) charting approach, which has been used by investors for decades for this end. TA holds that security price history summarizes all the information available about a particular asset. Therefore, it is possible to find patterns, and exploit them for profit.

There are three main reasons to use evolutionary computation approaches (such genetic algorithms) in financial applications:

- The limited reasoning hypothesis (LRH) (Lakemeyer, 1994) is a realistic assumption to complete the efficient market hypothesis (EMH) (Finger & Wasserman, 2004). This means all investors will make the best decision possible, but this one will depend on their computation power and ability to process financial data.
- There is a massive amount of financial data available like never before in history.
- The computational power available is vast and increases continuously.

EMH is based on the assumption investors are rational, therefore all of them will make the optimal decision. In the definition presented by Malkiel and Fama (1970), there is no room for advantages because the information is available to all the investors at the same time. The limited reasoning hypothesis (LRH) does not contradict the rationality assumption, but realizes the investor might fail to foresight all the possible options available.

2. Genetic algorithms and Darwinian approaches

The number of approaches proposed for financial applications is vast. The scope of the survey must be delimited to find proper conclusions. Genetic algorithms (GAs) are included in the evolutionary computation (EC) field. Nevertheless, EC includes other methods inspired by nature, culture, and language. Memetic algorithms are an example of culture-inspired evolutionary algorithms. GAs are a simplified version of Darwinian evolution. The schemata theorem shows how the best solutions pass their distinctive traits to the next generation with an exponential rate (Goldberg, 1989).

* Corresponding author.
E-mail addresses: A00263824@itesm.mx (R. Aguilar-Rivera), valenzuela@itesm.mx (M. Valenzuela-Rendón), jjrodriguez@itesm.mx (J.J. Rodríguez-Ortiz).
[1] Tel.: +52 (81) 8158 2044.

http://dx.doi.org/10.1016/j.eswa.2015.06.001
0957-4174/

This is possible by the use of genetic operators like selection, crossover, and mutation over encoded solutions.

Moreover, a survey on EC imposes the problem to report comparisons between methods which are inherently different from each other. The fact the selected benchmark problem could favor a certain method is latent. For example, the difficulty of GAs to solve the traveling salesman problem (TSP) was reported in early references (Grefenstette, Gopal, Rosmaita, & Van Gucht, 1985). On the contrary, other authors reported Simulated Annealing (SA) having good performance for problem sizes up to 200 cities (Johnson & McGeoch, 1997). At that time, the use of TSP as a benchmark could lead to a biased comparison between GAs and SA. It should be noticed that recent works have reported methods to apply GAs to large instances of TSP (Nguyen, Yoshihara, Yamamori, & Yasunaga, 2007). Approaches with a similar structure were chosen to avoid these problems.

Besides GAs, there are other methods based on Darwinian evolution. Some of them have found application in finance. The methods are listed and explained below:

- Genetic Programming (GP)
- learning classifier systems (LCSs)
- multi-objective evolutionary algorithms (MOEAs)
- co-evolutionary optimization schemes
- competent evolutionary algorithms

GP was initially proposed by Koza (1992). GP evolves solutions encoded as trees instead of binary strings. This allowed to find programs, mathematical proofs, electronic circuits, etc. Ferreira (2001) proposed Gene Expression Programming (GEP). GEP format allows the use of binary strings to represent trees. Miller and Thomson (2000) presented Cartesian Genetic Programming (CGP), where programs are represented as indexed graphs. Ryan, Collins, and Neill (1998) proposed Grammatical Evolution (GE), which uses Backus Naur Form (BNF) to design a system to evolve high-level programs. BNF has the advantage to be language independent. Recently, Gandomi and Alavi (2011) presented Multi-stage Genetic Programming (MSGP). MSGP is used to model non-linear systems, considering the effect of individual prediction variables and its correlation with each other.

LCSs were envisioned by Holyoak and Holland (1989) to enable agents to find descriptions of changing environments. Further research proposed modifications to overcome LCS limitations. For example, Wilson (1994) presented and accuracy-based classifier system (XCS). Other authors, like Holmes, Lanzi, Stolzmann, and Wilson (2002), made further research on the XCS subject. The zeroth level classifier system (ZCS) was also proposed (Wilson, 1994). Valenzuela-Rendón (1991) designed the fuzzy classifier system (FCS). Stolzmann (2000) reported the anticipatory classifier system (ACS). Gerard, Stolzmann, and Sigaud (2002) proposed the latent learning classifier system (YACS).

MOEAs explore the field of multi-objective optimization (MOO). A set of solutions (the Pareto front) is obtained when two or more objectives are simultaneously optimized. The population is used to find a set of optimal solutions in parallel. The niched-Pareto GA (NPGA) (Horn, Nafpliotis, & Goldberg, 1993), and the vector-evaluated GA (VEGA) (Schaffer, 1985) are some of the earliest contributions. The Pareto archived evolutionary strategy (PAES) (Knowles & Corne, 2000), The non-dominated sorting GA (NSGA-II) (Deb, Agrawal, Pratap, & Meyarivan, 2000), and the new version of the strength-Pareto evolutionary algorithm (SPEA-2) (Zitzler, Laumanns, & Thiele, 2001) were proposed in more recent reports. The multi-objective Bayesian optimization algorithm (MO-BOA) (Laumanns & Ocenasek, 2002) pretends to find a competent MOO algorithm; this definition of competence is based on Sastry and Goldberg (2003).

Co-evolution was firstly presented by Hillis (1990). The concept of competitive co-evolution was used to evolve sorting networks in this work. Potter and De Jong (1994) used a cooperative GA (CCGA) for function optimization. CCGA showed a better performance than a traditional GA in the experiments. Chang (2010) applied co-evolution to solve supply chain network design problems.

Competent GAs are mainly related to the estimation of population probability distribution. Estimation of distribution algorithms (EDAs) do not rely directly on genetic operators. They are presented in this survey because their process is an abstraction of Darwinian evolution and genetic operators. Harik, Lobo, and Goldberg (1999) proposed the compact GA to model the population with a probability distribution; it was designed to emulate a traditional GA performance. Harik, Lobo, and Sastry (2006) presented an extended CGA (ECGA) explicitly designed to solve problems where linkage was present. Pelikan, Goldberg, and Cantú-Paz (2000a) reported the Bayesian optimization algorithm (BOA), an EDA based on Bayesian networks. Besides, an extension designed to solve hierarchical problems (HBOA) was also presented (Pelikan, Goldberg, & Cantú-Paz, 2010b).

3. Refining the scope of this work

The existence of surveys with similar approaches to this one seemed plausible. Therefore, there is the possibility to cover references already cited in similar works. For this reason, the scope of our paper was delimited by the review of surveys with related subjects than this one. The novelty of our approach comes in three different manners: this work covers time lapses not included in other review articles, it covers problems not considered by others, and the scope covered by past references and new ones is compared and analyzed.

A review to find similar surveys was conducted from 2003 to 2015. Table 1 presents a list of articles with similar scopes. The problems considered are now enlisted: abnormal noise (fraud) detection (ABN), arbitrage (ARB), bankruptcy detection (BKR), cash management (CM), credit portfolio (CP), credit scoring (CS), fundamental analysis (FA), forecasting (FC), index tracking (ITR), market simulation (MKS), procurement (PCR), portfolio selection problem optimization (PSP), trading (T), and trading execution (TX). These problems are further explained in the following section.

The reviews considered were the following ones: (Atsalakis & Valavanis, 2009; Bahrammirzaee, 2010; Chen, 2003; Giulioni, D'Orazio, Bucciarelli, & Silvestri, 2015; Hruschka, Campello, Freitas, & De Carvalho, 2009; Lahsasna, Ainon, & Teh, 2010; Ngai, Hu, Wong, Chen, & Sun, 2011; Phua, Lee, Smith, & Gayler, 2010; Ponsich, Jaimes, & Coello, 2013; Safarzyńska & van den Bergh, 2010; Tapia & Coello, 2007; Tsang & Martinez-Jaramillo, 2004;

Table 1
Summary of review articles.

Reference	Techniques	Problems
Chen (2003)	GA,GP	MKS
Vanstone and Tan (2003)	GA	T,FA
Tsang and Martinez-Jaramillo (2004)	GA,GP,LCS	MKS,FC
Tapia and Coello (2007)	GA,MO	FA,FC,MKS,PSP
Atsalakis and Valavanis (2009)	–	FC
Hruschka et al. (2009)	–	CS,BKR
Bahrammirzaee (2010)	–	CS,FA,FC,PSP
Lahsasna et al. (2010)	GA,GP	CS
Phua et al. (2010)	GA,GP	ABN
Safarzyńska and van den Bergh (2010)	GA,GP,LCS	MKS
Verikas et al. (2010)	GA	BKR,CS
Ngai et al. (2011)	–	ABN
Ponsich et al. (2013)	GA,GP,MO	BKR,CP,FA,FC,PSP,T
Giulioni et al. (2015)	GA,LCS	MKS

Table 2
Time range per technique covered by other review articles.

Problem	GA	GP	LCS	MO
ABN	1999	2000	–	–
ARB	–	–	–	–
BKR	1992–2011	2009–2010	–	2010
CM	–	–	–	–
CP	–	–	–	2002–2010
CS	1997–2007	2005–2006	–	–
FA	2001	2009	1998	1998
FC	–	1994–2004	–	1994–2009
ITR	–	–	–	–
MKS	1994–2013	1990–2003	1997–2006	1998–2011
PRC	–	–	–	–
PSP	1993–2011	–	–	1994–2010
T	1994–2002	1999–2006	–	2009–2011
TX	–	–	–	–

Vanstone & Tan, 2003; Verikas, Kalsyte, Bacauskiene, & Gelzinis, 2010).

Our review concluded none of these works covers the exact approach of the present paper. Some of them are mainly focused on other areas instead of EC. Some of these works do not consider any of the techniques reviewed in this work. The exact EC contents in these works was further investigated. Table 2 shows the year span covered by the references for each solution method. Specific time ranges are provided because the coverage of a specific solution method is not always equal to the total time range covered by a reference.

Our analysis concludes co-evolution and EDAs are the techniques less used for financial applications. At least, they are not directly referenced by other review articles. They do not appear in Table 2 for this reason. GAs and GP seem to be the most studied methods, while LCSs have attained limited attention. Problems like arbitrage (ARB), cash management (CM), index tracking (ITR), procurement (PCR), and trading execution (TX) were not reported by other review articles, but references about them were found in the review conducted for the present survey. References with overlapping dates to these time ranges were eliminated from our study. The following section presents the review of uncovered references.

4. Description of financial applications

This section makes a description of each application found in the literature. The review of applications is presented classified by problem.

4.1. Abnormal noise and fraud detection (ABN)

A recent application of GAs to financial environments is abnormal noise detection in markets. Abnormal noise is believed to be correlated with illegal practices like rat trading or money laundering. Jing (2010) described a GA approach designed to detect abnormal noise. Although, experimental results were not presented in this work. Jun and He (2012) used a combination of GAs with neural networks (NNs) for money laundering detection. GAs were used to modify NNs weights only. The possibility of modifying the NN structure was suggested as future work.

4.2. Arbitrage (ARB)

Another problem of interest was Arbitrage. Arbitrage is the practice to take advantage of price differences of the same asset. This occurs when information is not simultaneously updated to all the investors. Arbitrage is more common among prices from different markets. Markose, Tsang, Er, and Salhi (2001) proposed GP to find arbitrage opportunities for the London stock index (FSTE-100) futures and options. This algorithm was trained with historical data where put-call-future parity cases are detected. GP is used to make an estimation of the longest time the arbitrage will remain profitable. The estimation is used to make earlier transactions than the traditional method where a contemporary signal of profitability is required. Financial GP-2 was developed to be an interactive tool to find arbitrage opportunities using intra-daily data. This approach attained greater returns than the textbook approach.

Tsang et al. (2000) designed the Evolutionary Dynamic Data Investment Evaluator (EDDIE), which was later used to for arbitrage applications (Tsang, Markose, Garcia, Almad Garcia, & Er, 2006). EDDIE relies on a human expert to evaluate information relevance. This information is later used by the system for the optimization process. EDDIE has proved to process financial information more efficiently than a human expert alone. An online implementation was later reported (Tsang, Yung, & Li, 2004).

Huang, Hsu, Chen, Chang, and Li (2015) proposed a GA approach for pair trading. Pair of stocks are bought and sold combined to find arbitrage opportunities. Trading pairs is based on the idea that mispriced assets will eventually reach with time their true price. The trading pairs method chooses a pair of stocks from the same industry, sells the one with relatively high price and buys the one with relatively low price. These stocks are likely to be mispriced and to converge to their real prices in the future. The method closes position to attain profit when the spread has reduced to a certain threshold value. The proposed method considered moving averages (MAs) and Bollinger bands to estimate the long term mean of stocks and thresholds, respectively. GAs are used to optimize MAs and Bollinger bands parameters, besides portfolio weights of stock pairs. The fitness value of individuals is the annualized return obtained from that specific trading pairs strategy. The method was tested using 10 stocks from the semiconductor industry of the Taiwan stock exchange and 10 stocks with the highest capitalization from the same index. The benchmark is a equal-weighted buy-and-hold portfolio of the mentioned stocks. The proposed method outperformed the benchmark in both experiments.

4.3. Bankruptcy detection (BKR)

Bankruptcy detection is the task of determining beforehand if a company is prone to bankruptcy in the near future. Accounting information and financial data are used to estimate the likelihood of the worst case.

Varetto (1998) made a comparison between GAs and traditional statistical methods for bankruptcy detection. Accounting information and financial reports were used to determine companies' health. GAs were able to obtain comparatively good results with less data than the statistical approach for the experiments presented.

Gaspar-Cunha, Recio, Costa, and Estébanez (2014) proposed an adaptive MOEA for bankruptcy detection applications. When companies grow they become complex and the estimation of their real financial situation becomes difficult. Financial reports of complex companies have a large number of features; many of them are irrelevant to estimate the actual company status. These authors proposed a MOEA to minimize features and maximize accuracy of classifiers. A feature selection algorithm is applied periodically while parameters are optimized. Each individual encodes the relevant features and the configuration parameters of the selection features algorithm, which is implemented using support vector machines (SVMs). SVMs are applied to training data and the accuracy of the specific configuration is used as the fitness value. Accuracy is computed from the confusion matrix of the test. Individuals are ranked according to their fitness value, and a fixed number of the best individuals are copied to a secondary population, which is larger then the primary population. The secondary population is completed with offspring generated from the selected individuals. A fraction of the best individuals from the secondary population substitute the worst individuals of the primary population. The method was tested with industrial French companies (DIANE database) data, Australian credit data (UCI Machine Learning Repository), and German credit data (UCI Machine Learning Repository). The results showed that the algorithm was able to significantly reduce the number of features, but better accuracy was obtained with larger features sets. Besides, the MO approach provided a set of solutions with different levels of simplicity and accuracy. Inclusion of the decision maker preferences to obtain a single solution was indicated as future work.

4.4. Cash management (CM)

Authors like da Costa Moraes and Nagano (2014) have studied the problem of cash management. Cash management is related with the accumulation and use of cash in companies and institutions; companies need cash for their operations, and a lack of liquidity will obstruct procurement and direct investment. For example, mutual funds need to plan how to invest their costumers money to provide the promised returns while providing daily liquidity. Precaution and speculation are other reasons to maintain a cash balance.

The approach proposed by da Costa Moraes and Nagano (2014) considered a stochastic model where current cash will vary along a bounded range. GAs and particle-swarm optimization (PSO) were used to find the optimal strategy. A portfolio composed of two risky assets and cash was considered. The system should decide the amount to be invested or saved while considering cash could be demanded from the portfolio. A Miller-Orr model for full cash management was implemented (i.e., cash is stochastic). Nevertheless, cash bounds were added to the original model. Cash flows were assumed to be a random variable with normal distribution. Transaction costs, interest rates for lending, bounds form risky assets, and asset liquidity are the parameters of the model. GAs and PSO searched to minimize cash management costs in a multiple-period time horizon. Different probability distribution configurations were used to simulate cash flow in the experiments. Both algorithms were capable to determine cash flow policies, but GAs outperformed PSO for most of the instances tested. Nevertheless, PSO attained lower average relative deviation than GAs. This work concluded both GAs and PSO are viable methods to further study the problem.

4.5. Credit portfolios (CP)

Credit portfolios are similar to security portfolios, but they are composed of credits instead of securities. Banks should decide which credits to approve to maximize profit and minimize loss. Profit comes in the form of interest, while loss is caused because the borrower fails to pay the debt back. This problem was mentioned by Tapia and Coello (2007) and Ponsich et al. (2013). Both references proposed MOEAs to solve the problem.

4.6. Credit scoring (CS)

Credit scoring is the problem of determining if applicants will repay their loans to banks. Credit approval is made based on this estimation. Misclassified applicants represent a loss for banks and loaners. Banks and stores invest time and resources to select the most promising applicants.

Back, Laitinen, and Sere (1996) proposed a combination of NNs and GAs to perform this task. GAs were used to select the input variables for the NN. The results showed the approach was successful for the experiments presented. It was concluded the prediction was only reliable for a one year time window. Ravi, Kurniawan, Thai, and Kumar (2008) proposed a sophisticated approach which combined NNs, SVMs, decision trees and GAs to solve this problem. GAs were used to optimize the weights of the NN. The approach showed to have less Type-I (false positive) and Type-II (false negative) errors than the individual techniques. It also performed better than a NN trained with human expert data. Hochreiter and Wozabal (2010) used a coupled Markov-chain approach to compute fail probability. In this work, maximum likelihood model estimators were found using GAs. Nikolaos and Iordanis (2010) combined a NN trained with GAs and different regression methods. The proposed method showed a higher rate of correct decisions, but its minimum squared error (MSE) was higher than the MSE of regression methods. The applications of different NNs structures was proposed in the future work section. Lin, Liang, Yeh, and Huang (2014) proposed a combination of SVMs and GAs for this task. For the experiments, the approach showed better results than the traditional analysis of discriminant method. The future work section suggested the approach needs a method to define the algorithm parameters.

4.7. Fundamental analysis (FA)

Fundamental analysis is concerned with the valuation of securities. It makes use of company's accounting and financial information for this end. A careful estimation allows profitable opportunities to be found. This occurs when an asset is over-valuated or sub-valuated at the market. The investor can take advantage of this information before the asset reaches its true value.

Jiang, Xu, Wang, and Wang (2009) proposed the use of GAs for this end. A GA selected the most significant variables to determine company value. The selected features were analyzed using discriminant analysis to estimate future financial performance of Chinese companies. It was concluded that the approach was effective to find the best selection; this selection changed according the company line of business.

Huang et al. (2012) proposed an Initial Pricing Offering (IPO) approach for PSP. An IPO contains the information used to determine asset value when it was initially introduced to the market. This approach used this data to rank securities. A GA was used to determine the most significant indicators, besides the most suitable weights for each one of them. The objective is to find an equal-weighted portfolio with the maximum return based on these indicators. The rules can be applied later to different securities. This FA approach attained positive returns in the experiments, this means the selected stocks attained higher first-day return than the average of the whole set of securities.

4.8. Forecasting (FC)

The forecasting problem is one extensively studied. It consists in the estimation of future values of securities and trends in data. Investment would be straightforward with a perfect predictions of the future. Although this is not possible, forecasting involves also an estimation of prediction error, allowing to make better decisions under uncertainty.

One of the earliest authors to use GAs to solve this task is Packard (1990). Meyer and Packard (1992) showed the same approach to find prediction rules for the Mackey–Glass equation, which shows chaotic behavior. Kingdon, Taylor, and Mannion (1997) presented applications of GAs and NNs to forecast financial time-series. Chen and Lee (1997) used GAs for option pricing; this approach was tested with the European Call Option problem, for which an exact solution is known. The experiments showed positive results, suggesting a practical implementation is attainable.

Kim and Han (2000) proposed the use of a NN for the prediction of price indexes. The continuous variables were mapped into discrete sets. A GA was used to find the optimal ranges for the input variables. Ma, Wong, Sankar, and Siu (2004) reported the use of wavelet transform and GAs to forecast financial index volatility. The obtained coefficients were processed with a GA to find patterns. The approach was tested against GARCH models and attained positive results.

Rimcharoen, Sutivong, and Chongstitvatana (2005) proposed an $(\mu + \lambda)$-ES (evolutionary strategy) to solve the problem. It was tested with Thailand stock market data. This approach showed a better performance than multiple regression. The results showed the existence of a high correlation between the Thailand and the Taiwan stock markets, which was exploited by the algorithm.

Polanski (2011) used GAs for multi-dimensional time-series forecasting. This work presented an experiment with foreign exchange market (FOREX) data. de Brito and Oliveira (2012) compared the performance of GAs with a hybrid SVM-SOM (Self Organized Maps) for FOREX trading. The GA attained higher returns in financial crisis scenarios.

Goonatilake, Campbell, and Ahmad (1995) proposed the use of fuzzy systems to make trading decisions. A GA was used to find the optimal fuzzy sets configuration. The hybrid system generated meaningful rules for human experts. Kanungo (2004) proposed a combination of GAs with maximum likelihood estimation (MLE). Although the good results, it was concluded more experiments were necessary to prove its reliability. The approach proposed by de Araujo, Madeiro, de Sousa, Pessoa, and Ferreira (2006) combined Modular Morphological NNs (MMNNs) with GAs to solve the forecasting problem. It was tested with data from the S&P500 index. The experiments concluded the algorithm seemed to model time-series as a random walk. The inclusion of a correction module was suggested as future work.

Parracho, Neves, and Horta (2011) used GAs to find trend patterns and use them to predict market tendencies; it was proved with data from S&P500 index. The method showed positive results for the experiments presented. Araújo and Ferreira (2013) used GAs to optimize linear filters in forecasting applications. The obtained filters were further enhanced using MSE estimators. The approach was compared against NNs and time-delay evolutionary forecasting (TAEF) and random walk models. The proposed approach outperformed the benchmarks in the experiments. Further research on the method properties was suggested as future work. Bernardo, Hagras, and Tsang (2013) used a type-II fuzzy system adjusted with GAs for this task. This approach performed better than GP. The performance was similar to NNs, but with the advantage of producing results which are meaningful to human experts.

Ghosh and Chinthalapati (2014) proposed agent-based modeling to forecast financial markets. A non-equilibrium economics was considered and their features were included into a set of bounded-rational and heterogeneous agents. Interaction is possible among the agents. GAs were used to model them. An artificial market was implemented. Time series were binary, this means they only represent the current directional trend of prices. N agents populated the economy and each one had a number of strategies ranked according with their performance and limited memory of the past. Agents should decide to buy or sell based on current state and their own strategies. Four types of agents were considered: minority game, majority game, \$-game, and delayed minority agents. Minority agents are rewarded when their decision is opposite to the actual market. Majority agents are rewarded when their decisions are the same as the actual market. \$-game agents assume last market value is the best estimation of future and are rewarded when this occurs. Delayed agents make the opposite assumption than \$-game agents. A GA with islands is used to simulate the market. Each island is a sub-population which represents a possible optimal market. Each island is populated with agents with heterogeneous beliefs. Fitness of each individual depends of the agent type they represent. Interaction among islands is possible, but each one follows their own optimization process. The experiments presented two cases: FTSE-100 closing prices and FOREX case. Binary time series are obtained from original time-series of prices. The market returns were the ones evaluated instead of individual performance. This means islands are evaluated instead of individual agents. It is possible to obtain a forecast of future price based on the returns of each island. The forecasts were used to implement a trading strategy based on the obtained market models. Agents obtained a hit ratios in out-of-sample data of about 67%. The authors suggested using the universal information criterion in future work to better study the significance of the obtained results.

Garcia-Almanza and Tsang (2006) studied the detection of bubbles and crashes in financial markets. These events are hard to predict because they rarely occurs, nevertheless, they have heavy impact in markets. The repository method (RM) is an analysis technique which was applied to the decision trees generated by GP. This method extracts and simplifies rules encoded in each individual, adding them to a repository if they cover different possibilities than current ones. The work presented experiments designed to discover the factors required for RM good performance. It was concluded that the accumulation of rules is crucial to classify correctly the positive cases.

Wagner, Michalewicz, Khouja, and McGregor (2007) proposed a modification to traditional GP to deal with dynamic environments. This was called dynamic forecasting GP (DyFor GP). It includes an adaptive sliding-window approach chooses the best window size to describe the environment. It compares different windows sizes and chooses the one which allowed the best prediction of current data. This one is used to forecast the next future value. It was applied to estimate U.S. Gross Domestic Product. DyFor GP results were better than regular GP and other benchmarks models.

Shao, Smonou, Kampouridis, and Tsang (2014) proposed a modification to the GP financial forecasting tool EDDIE mentioned above. This version provides an extended grammar for the generated decision trees. This new grammar increased the search space. This work proposed a combination of guided local search with fast local search (GFLS) to solve this problem. Technical indicators are used as inputs. Both the rules and the forecasting time horizons are optimized. Only the latter are subject to GFLS. GFLS uses hill climbing to improve current solutions. Guided search modifies hill climbing behavior through solution classification, fitness modification and class penalization. Fast search divides the space in sub-neighborhoods and eliminates those where no improvement was found to save computational effort. The method was tested using data from different market indexes like FTSE-100, DJI, Nikkei-225 and others. The results indicated GFLS improved performance of EDDIE when comparing it to GLS alone. GLS is the same algorithm without the effort-saving capabilities.

Hamida, Abdelmalek, and Abid (2014) proposed GP for volatility forecasting. Volatility is the implied variance of return at a given time. Volatility can be estimated from historic data or from observed option prices. Option prices show the expectation about the future price that the underlying asset will reach at maturity time. Forecasting rules are optimized using GP, where both historic and option data are used as inputs. One of the problems to be solved by this approach is determining the sample size used to compute estimations. Four methods to determine sample size were proposed in this work: random subset selection (RSS), sequential subset selection (SSS), adaptive random subset selection (ARSS), and adaptive sequential subset selection (ASSS). The method used was applied each g cycles of the algorithm. RSS selects a sample size randomly with uniform distribution. SSS applies each sample size using a predetermined sequence. Adaptive methods compute the average mean squared error obtained during g cycles for the sample size used. Sequence order is rearranged according with this measure for the ASSS method. Selection probability is tuned in a similar manner for sample sizes using the ARSS method.

Karatahansopoulos, Sermpinis, Laws, and Dunis (2014) studied two different GP approaches for forecasting applications. One is regular GP, which evolves mathematical expressions represented as trees. The second one is GEP. This work proposed a one-day ahead forecasting application to model the ASE-20 Greek index. GP and GEP where tested separately against NNs, ARMA models, MA models, and a naïve strategy where present return value is taken to be the estimation of future return. Lagged index values and MAs values from ASE index were the inputs for GP and GEP. The proposed methods outperformed benchmarks. GEP showed better annualized return than GP.

Mahfoud and Mani (1996) proposed a GA for security price forecasting. The GA was similar to the one included in a LCS based on the Michigan approach. This kind of system encodes a rule per individual. A similar approach was studied by del Arco-Calderón, Vinuela, and Castro (2004), who proposed a non-generational Michigan GA for forecasting applications. Each individual tries to predict a particular case of the series. It was applied to predict random securities selected form S&P-400. This method showed the ability to detect regions which cannot be generalized. Both methods are closer to LCSs than traditional GAs. Credit assignment and conflict resolution are discussed. Both are important issues to solve when implementing LCSs.

Donate and Cortez (2014) proposed a NNs approach for forecasting applications. The difficulty of finding the best NNs model for the task is indicated. The univariate marginal distribution algorithm (UMDA) EDA is used to search for this end. This algorithm copies the best half of the current population into the new one; the rest of the individuals are randomly generated using the probability distribution computed by the method. Two design strategies for NNs are considered: sparsely connected NNs and time lag selection NNs. The former considers a binary direct encoding of NNs which can be directly mapped into a matrix of connections. The latter considers a time-lagged feed-forward NNs is designed; therefore, the lags are also searched by the algorithm. NNs are first optimized using resilient propagation before applying UMDA. DJI data series is one of the time series used to test the approach. Comparison methods used were ARIMA models, random forest (RF), echo state networks (ESN), and SVMs.The proposed method showed to attain lower mean-squared error the other ones.

4.9. Index tracking (ITR)

Market Indexes are extensively used as benchmarks in financial applications. They are also used as indicators of health of economies. Different organizations take the most representative securities in the market to built their indexes. The Dow Jones Industrial Average (DJI), the Mexican Índice de Precios y Cotizaciones (IPC), the Japanese Nikkei 225, and the FTSE-100 are examples of market indexes. Nevertheless, to build a portfolio with the exact composition of a market index is a difficult task. For example, round lots restrictions will make feasible replications to have a prohibitively high value. A portfolio based on FTSE-100 would require to buy all these securities. Information about the exact composition of IPC is quarterly published only (Bolsa Mexicana de Valores, 2015).

Index trackers are used replicate market indexes values. They are usually a fraction of true value. Index trackers are not limited to index composition. Other instruments besides securities are used to build trackers.

Chen and Chen (2011) proposed an adaptive GA for pattern recognition (AGA-PS) for ITR purposes. PS is a type of local search based on neighborhoods. Crossover and mutation probabilities are updated online. The authors selected 20 securities from Hushen-300 stock index and tried to replicate its value minimum error. The approach showed better results than a traditional GA for the same problem.

Andriosopoulos and Nomikos (2014) studied tracking of spot energy index built from New York mercantile exchange information. Commodities futures have become a mean to attain effective investment diversification. Nevertheless, these futures oblige holders to provide the commodity traded at maturity time. This work proposed to invest in commodity-related equities instead to avoid this problem. The hypothesis is that a careful selection of securities from commodity-related companies is a suitable tracker of real commodities. This work proposed to solve using the differential evolution algorithm (DEA) and GAs. The average and the standard deviation of tracking error are combined to compute a single-objective fitness value. Securities from the DJI, the Bovespa composite, and the FTSE-100 were used to build the tracker. Different number of stocks and trade-off parameter λ values were tested in the experiments. The method is applied in buy-and-hold, quarterly, and monthly re-balancing scenarios. The results showed the method is capable of tracking the index. Bovespa pool even outperformed the benchmark. The authors concluded that both equities and commodities have similar return distributions, making tracking possible. 15 stocks and $\lambda = 0.8$ were the best combination of parameters.

4.10. Market simulation (MKS)

Some studies have exploited the advantage of computational methods to investigate market properties. GAs and other techniques allow the simulation of agents with different behaviors. These agents are allowed to interact in artificial economies to observe the effect on an economy of individual behavior. These studies are concerned with validation of theoretical models. Agents which are built under the model assumptions are expected to attain the results predicted by the model. On the other hand, agents with different behaviors can be used to find more realistic models of an economy.

Kampouridis, Chen, and Tsang (2012) used GP combined with Self-Organized Maps (SOM) to simulate agents in a constantly changing economy. The hypothesis that a constant evolution of strategies is needed for the agents survival is concluded from the experiments.

Sinha, Malo, Frantsev, and Deb (2014) proposed a study about multi-period, multi-leader–follower Stackelberg game to model oligopolistic economies. They are based on Cournot games. Cournot games consider an economy where many companies compete with each other to sell the same type of product. Each agent can influence supply and market price with their own production. They should estimate future price and decide their production to maximize profit. Stackelberg games differ form Cournot games because the former ones have two types of agents: leaders and followers. Leaders have move first, and then followers, and leaders posses necessary information about followers to estimate their future actions. Followers observe leader's actions and make decision based on evidence. This is a case of bi-level optimization because both leaders and followers are part of a Cournot game among their equals. This work proposed a model and applied it to a aircraft manufacturing industry case. A steady state real-coded GA is proposed to solve the model, making decisions for number of periods. GA works first the leader part of the chromosome. Parent centered crossover (PCX) and polynomial mutation were used for this end. The follower part of the closest member of population to each child is copied into it. $n-1$ random low level individuals are generated to form a subpopulation along with the copied information. New low-level individuals are generated using crossover. The low-level individual with less violations to restrictions is finally chosen. The algorithm stops when the average normalized population variance is smaller than η. The algorithm was capable of solving the instances presented in the experiments. Parallelizations were proposed in the future work section.

Franke (1998) proposed a GA co-evolutionary scheme to simulate a cobweb economy with heterogeneous beliefs. A cobweb model explains prices fluctuation based on expectations of producers about future demand of their product. Producers should plan production based on their expectations. Changes in expectation affects supply along with price. The modified GA is similar to the one found in the LCS structure. Martinez-Jaramillo and Tsang (2009) simulated a market populated with FA-based traders, TA-based traders, and noise traders. Technical agents used a co-evolution GP approach to find their strategies. Both Single Population (SP) and Multiple Population (MP) models were studied. SP considers each individual as an independent agent. MP uses a whole population to represent a single agent. The Red Queen Principle is included in the model by taking forecasting precision as the fitness measure. The Red Queen Principle holds constant evolution is observed in competitive environments. In this case, individuals must improve constantly, lest they will be left behind by new individuals with better accuracy. This work concludes heterogeneity, learning, and the Red Queen Principle are factors that should be present in real markets.

Protopapas, Battaglia, and Kosmatopoulos (2010) used co-evolutionary GAs to simulate agents in Cournot games. They can affect prices with their own production. They should decide their production to maximize profit. The work concludes the games converges to Nash equilibria in social learning scenarios. In these scenarios the information of all the populations is used together to update the agents strategies.

4.11. Procurement (PRC)

Procurement is the systematic process companies use to purchase necessary goods and services. Procurement is important to ensure purchase satisfy the requirements at the lowest price possible.

Tezuka, Munetomo, and Akama (2007) proposed to modify GAs to solve Stochastic Programming problems. This algorithm is applied to noisy objective functions, where the mean of individual fitness is estimated using a Monte-Carlo approach. The approach tries to determine sample size of a pair of individuals which need to be discriminated using tournament selection. An F-test based method is proposed to estimate sample size which minimize variance of average. Bootstrapping is applied to avoid excessive evaluation. The experiments studied a case of procurement planning where a company should decide the amount of materials and time of purchase based on estimations of market price of products. Market prices were treated as stochastic variables.

4.12. Portfolio optimization (PSP)

Portfolio optimization is based on the concept of investment diversification. Modern portfolio theory (MPT) considers each investment decision implies risk, which is a measure of the possible loss investors could face when they make a particular decision. MPT proves a set of assets (i.e. portfolio) can attain less risk for the same expected return. Portfolio optimization searches these optimal portfolios, which is a case of Pareto optimization.

MPT proposes a quadratic optimization algorithm which solves a mean–variance description of portfolios efficiently. Nevertheless, the problem turns to be difficult when real-world restrictions are considered. Transaction costs, round-lots, composition boundaries, and non-stationary time series are examples of these restrictions.

Gupta, Mehlawat, and Mittal (2012) proposes a hybrid method based on GAs and SVMs. The first part applies a SVM to classify assets based on selected financial indicators: liquidity, high return and low risk. The second part applies a real-coded GA to build the portfolio. Preferences are included by selecting one of the sets of securities determined by SVMs. An optimal portfolio is searched using a weighted sum of these financial indicators. Boundaries of portfolio composition are considered into the process.

Wang, Hu, and Dong (2014) proposed a portfolio optimization model based on a convex risk measure called weighted expected shortfall (WES). In this measure, the cumulative probability of final portfolio value to be less than x_α is weighted by an exponential function to compute WES. A coefficient λ is included in its argument to controls risk aversion. The proposed model is a case of nonlinear optimization. GAs are chosen to optimize the model. WES is the fitness value to be minimized. 10 random stocks from the Shenzhen index were randomly chosen. Portfolios optimized with different values of risk aversion coefficient were compared in the experiments. The optimized portfolios were kept in buy-and-hold for 60 days.

Hochreiter (2014) proposed a GA combined with local search to solve the risk parity PSP. Risk parity portfolios are those where each stock weight is adjusted in a way each asset contributes equally to the total portfolio risk. The problem is trivial when long-only portfolios are allowed, but it turns difficult when short positions are possible. The objective functions is the sum of differences of average risk per stock. Elitist selection, and random addition of individuals are some of the specific properties of this algorithm. The best solution found by the GA is then optimized using local search. The method was tested with data from the DJI the and S&P-100 index. It was compared against a minimum variance portfolio and an equal-weighted portfolio for the long-case. The long–short case was compared against randomly generated portfolios. The proposed method obtained the best results.

GP has also been studied to solve PSP. Wagman (2003) proposed GP to find rules based on technical analysis indicators to design portfolios with high Return of Investment (ROI) rate. Portfolios with higher ROI than current market interest rate were found under conservative market conditions. Averages from past prices, minimum and maximum historic data were used as system inputs. Data from the DJI from 1979–1980 were used in the experiments. Future work involved considering capital adequacy into the optimization process. Krink and Paterlini (2011) used a Differential Evolution MO approach in a similar application.

Lwin, Qu, and Kendall (2014) studied mean–variance PSP with cardinality, quantity, pre-assignment, and round-lots restrictions. Cardinality restricts the number of stocks in the portfolio. Quantity refers minimum and maximum proportion of assets in the portfolio. Pre-assignment restriction forces the algorithm to include certain stocks in the portfolio. Lots of a specific number of stocks can be traded only. This work proposed a new MO algorithm for the task, and compared it against four popular MO algorithms: non dominated sorting GA (NSGA-II), Pareto envelope selection algorithm (PESA-II), strength Pareto evolutionary algorithm (SPEA-2), and Pareto archived evolutionary strategy (PAES). The proposed algorithm encodes portfolios using two vectors: one to indicate if the specific asset is part of the portfolio and another for portfolio weights. The composition of non-dominated portfolios is observed and their assets; each one of them is given a concentration value proportional to their number of occurrences. Candidate assets are selected using their concentration values. Pre-assigned stocks are compulsory included. Three random portfolio are chosen from the population. Candidate portfolios are generated from these ones using mutation operators or scaling factors. Portfolio weights of individuals are determined selecting one of these methods randomly. The resulting weights are modified to comply with quantity and round-lots restrictions. The methods were compared using typical MO measures like Δ metric or hyper-volume. The proposed method showed better performance than the benchmark in the experiments.

García, Quintana, Galván, and Isasi (2014) studied the effect of re-sampling in MO algorithms. This work solved a mean–variance PSP with cardinality and quantity restrictions. Re-sampling was implemented using a bootstrapping method where a sliding window is used to determine the sample data; expected return of assets and covariances matrix were recomputed from this new sample. Popular MO algorithms were modified to implement re-sampling. Data from the Frank Russel indexes and the S&P were used for the experiments. SPEA-2 attained the maximum improvement in quality solution by the inclusion of re-sampling.

Hochreiter and Wozabal (2010) presented PSP like a stochastic optimization problem. This work made a review about single-state PSP and multiple-period PSP. Stochastic optimization considers parameters in the model to be probability distributions instead of deterministic ones. Monte-Carlo approaches or clustering are used to estimate these distributions. EC allows to consider wide variety of risk measures and restrictions.

Adebiyi and Ayo (2015) presented the generalized differential evolution algorithm 3 (GDE3) to solve a mean–variance PSP with the following restrictions: bounded portfolio weights, cardinality, minimum transaction lots and expert opinion. Expert opinion is a weight in the range $e_i = [0,1]$ which describes the likelihood to attain its expected return. Assets with $e_i < 0.5$ are not included in the portfolio. This value is randomly initialized for this study. The model includes investor desired return to be reached. GDE3 is an extension of differential evolution (ES) to allow MO optimization and restriction handling. GDE3 generates both feasible and unfeasible offspring. Feasible individuals are always preferred. Parents should dominate offspring to be selected when both vectors are feasible or unfeasible. Crowding is used to determine which individuals are located at less populated areas of solution space. Individuals are sorted according with this measure. Non-dominated and feasible individuals are saved for the next generation. The proposed method was tested with the Hong-Kong Hang Seng index and the German DAX-100 data. The method was tested with different desired return values, and portfolio sizes. Mean variance of solutions, worst variance, standard deviation of solutions variance and mean execution time were the proposed performance measures. The method was compared against GAs, SA, taboo search (TS) and particle swarm optimization (PSO). The proposed method outperformed the benchmarks in the experiments.

Ranković, Drenovak, Stojanović, Kalinić, and Arsovski (2014) proposed the solution of a Value-at-Risk (VaR) PSP using GAs. VaR is a risk measure commonly used in finance. VaR can be understood as the α-quantile of distribution. It cannot be analytically estimated unless the probability distribution of returns is known. Besides, the Pareto front of optimal portfolios is not restricted to be connected and convex, like the cases of mean–variance PSP. This work studied the proposed problem using a single-objective GA and the MOEA called SPEA2. This worked proposed the use of portfolio weights computed from number of shares instead of asset value, which avoids dynamic effects, like portfolio unbalance, to be mixed with static portfolio optimization. The single objective GA combines both objectives using parameter λ and its complement $1-\lambda$, where $\lambda = [0,1]$. $\alpha = 0.05$ to compute VaR in the experiments. 10 exchange-trade funds (ETFs) were considered. Data were taken from February 2008 to December 2010. An equally-weighted buy-and-hold portfolio was used for comparison. Different values of λ were used to build the efficient frontier using the single-objective GA. Two methods were used to obtain the efficient frontier: the first one used a fixed set of increasing λ values. The second compute the necessary λ value to attain a particular return level. This latter method obtained the best results. These results were also better than ones obtained using SPEA2. The authors remarked solutions obtained using single-objective GA were not necessary Pareto optimal because VaR is not a coherent risk measure. SPEA2 has the advantage to compute all the solutions simultaneously. All three methods outperformed the benchmark.

4.13. Trading (T)

Trading is the practice of finding profitable investment strategies. Forecasting is related to trading because an estimation of the future is usually required to make correct decisions. Trading is concerned what to do with the forecast to make profit. For example, risk (i.e. uncertainty) can be considered to minimize loss probability along with maximizing expected return.

Lim and Coggins (2005) used GAs to find trading rules. The fitness function was based on the volume-weighted averaged price (VWAP) measure. The approach outperformed buy-and-hold, which is the strategy to keep a security until the end of time horizon. It was concluded volume information is useful for trading. A reinforcement learning approach was suggested in the future work section.

Hirabayashi, Aranha, and Iba (2009) proposed to find the best trading time instead of forecasting. The estimation is based on technical analysis indicators like relative strength index (RSI), MAs, and percent difference from MA. It was applied to FOREX data for USDs, euros, and Japanese yens. Positive returns were reported in stationary statistics time windows. A MO approach was suggested in the future work section to include risk in the optimization.

Matsui and Sato (2009), proposed GAs using binary and integer representation of rules. Technical indicators like MAs, exponential MAs and Bollinger bands were used to conform these trading rules. In the presented experiments, the integer representation obtained higher profits at less computational cost. Matsui and Sato (2010) extended this work suggesting the use of neighborhood evaluation in GAs for trading applications. The idea is to reduce over fitting by considering the average of neighbors and individual fitness. The experiments showed the method presented less over-fitting than the former approach, although the computational cost increased proportionally to neighborhood size. Further investigation of this approach was recommended in the future work section.

Lipinski (2012) described a similar approach with technical indicators. A GA was proposed besides SA. Both were improved with local search operators. A parallel processor architecture is described and tested in this work. Chen, Hou, Wu, and Chang-Chien (2009) combined the concept of portfolio optimization with trading to create investment strategies. This is possible because a portfolio change implies a trading action. Chen defined portfolio optimization as a combination optimization problem. A new combination GA was suggested to solve the problem. Custom operators for combination problems were proposed.

Sarijaloo and Moradbakloo (2014) proposed a study to investigate the efficiency of traditional approaches for trading. GP was proposed to solve a mean–variance portfolio problem with maximum and minimum limits in their composition. The best 50 securities from the Tehran stock market index (2006–2009) were chosen for this study. Optimal portfolios were chosen yearly and compared against equal-weight portfolios and random search.

The approach showed better performance for the time range used in the experiments.

Yaman, Lucci, and Gertner (2014) used evolutionary programming (EP) to generate trading agents with different investment strategies. Agents were modeled using echo-state network models (ESNMs). ESNMs are a type of NN with three layers: input, hidden, and output layer. Feedback from the output layer to the hidden layer and from the hidden layer to itself are present. All connection but the ones going to the output layer are randomly initialized and fixed form the beginning. EP is used to optimize the free connections. EP individuals have two parts, the objective vector and the variance vector. Only the first one is evaluated, but both of them evolve with time. Mutation is the only operator used. This work applied an EP($\mu + \mu$) type. This means μ new individuals are generated from μ original individuals. The half of the population with the lowest fitness values is discarded. Resulting agents have different behavior depending on their initialization. Exchange rate from different currencies and technical indicator were used as inputs. The experiments concluded that to have a set of different behavior agents is more profitable than one single type of agents.

Lohpetch and Corne (2009) stated some works were unable to repetitively outperform buy-and-hold, although they report profitable strategies. This work proposed a GP algorithm to attain this objective. Technical analysis indicators are used for the rules. Some of the modifications to ensure solution quality were the following: the use of monthly data, reduced function set, over-complexity penalization, and an objective function which penalized lower than buy-and-hold performance. It was concluded, although the approach was successful in the presented experiments, tests with different time windows should be conducted.

Chen and Hirasawa (2010) proposed robust genetic network programming (R-GNP) to find trading strategies. A genetic relation algorithm (RGA) was proposed to find optimal portfolio to be managed using the R-GMP model. RGA has the property to work with graphs instead of trees. R-GNP encodes trading strategies using graphs. Judgment nodes, processing nodes and delays are part of this graph. Judgment nodes represent conditions, processing represent trading actions and delays represent the technical indicator used. This work used the β parameter as risk measure. β has its origins in the capital allocation pricing model (CAPM). β is the fraction of market risk borne by the portfolio. The initial portfolio was build using a relational GA (RGA). The paper concluded that further tests are necessary to evaluate R-GNP performance.

Hochreiter (2015) proposed a method based on GP and sentiment indicators to find trading rules. Sentiment indicators were extracted from a social media service to determine the expectations of the online trading community. The estimations are based on the number and contents of messages in the service about DJI stocks. If–then type rules were obtained with this method. DJI data from 2010–2013 were used for training. The classical portfolio optimization model was used to build buy-and-hold portfolios. Equal weighted portfolios were also used for comparison. The proposed approaches outperformed both benchmarks in the experiments. The inclusion of transaction costs was left as future work.

Radeerom (2014) proposed a trading system based on technical analysis indicators and GAs. The system generates a trading signal to indicate the best moments to buy or sell stock. The process consists of two phases: the first one is the selection of the most suitable stocks, the second is the trading of selected stocks. The technical indicators used are the following: relative strength index (RSI), MA convergence/divergence. The algorithm maximizes last-day Sharpe ratio. This risk-weighted measure allowed to treat the multi-objective problem as single objective. Stocks with negative Sharpe ratio or negative shareholder's equity value are eliminated. Trading rules based on the mentioned indicators and their combinations are optimized with training data. The approach was tested with the Thai-100 stock index. The results showed the proposed method outperformed buy-and-holds.

Chen and Chang (2005) proposed the use of XCSs for trading. The rules were encoded to process sentiment indicators. Sentiment indicators are variables which measure general expectation of investors about market trend. A bull market means investors expect a rise in prices, while bear markets indicates price drops. Volatility index, put-call ratio, and trading index were used in this work. XCS found rules to decide when to sell or by futures. It was compared against buy-and-hold (the future was kept until last day), a trend-based strategy and a mean-reversion strategy. This last strategy made transactions when sentiment indicators reached a threshold. XCS showed better performance than the other strategies.

Huang, Pasquier, and Quek (2009) proposed a hybrid system which combines a hierarchical co-evolutionary fuzzy System (HiFECS) and a hierarchical co-evolutionary GA (HCGA) to forecast stock prices. A prudent strategy based on the price percentage oscillator (PPO) is run using these forecasts. The experiments show HiFECS outperforms buy-and-hold, and other predictive models like evolving NNs (EFuNNs), dynamic neural-fuzzy inference system (DENFIS), and rough set-based pseudo outer-product fuzzy NN (RSPOP).

Lipinski (2007) made a comparative study about ECGA and BOA. Some modifications were necessary to perform online trading. In the experiments, both algorithms attained better returns than static strategies like buy-and-hold. Although, ECGA proved to be time-consuming and not suitable for an online application. BOA was faster, but with less returns.

Matsumura and Kakinoki (2014) proposed a MO-GP approach for PSP-based trading. The algorithm first determines the 10 most suitable stocks to be traded. A multi-objective GA is proposed for this end. The algorithm is a regular GA where fitness value is the number of individuals which dominate the one to be evaluated. Average return along the time horizon and covariance are used in this step. Elitism and an end-cutting operator ensures solutions to be non-dominated and well spread. This approach is used in a second step for portfolio selection. Investment ratio is the fitness value to be maximized. This step uses real-number representation individuals. A third step uses GP to optimize strategy trading trees. In this case, an optimal initial portfolio is selected and buy/sell operations rules are determined based on technical indicators. Experiments for the Nikkei-225 index and comparison against buy-and-holds were presented. The performance of the technique varied with market current scenario.

Hu, Feng, Zhang, Ngai, and Liu (2015) proposed the use of an XCS to find trading rules based on trend following strategies (TF). Technical indicators are computed from data to input the system. MAs and volume MAs were used. Their state is binary encoded for each stock. XCS estimates short-term trading only, while long-term trend is estimated using MAs only. Both signals are combined to generate the trading signal. Buying occurs only when XCS recommends it at bull market condition. Selling occurs when XCS recommends a sell at bear market condition. No new position is opened while there is one already on course. Stop loss was implemented to avoid heavy loss. Data from the Shangai stock exchange from January 1, 2001 to July 31, 2013 were used for the tests. Performance was measured using the Sortino ratio, which is a risk-weighted return measure where rise fluctuation is distinguished form fall fluctuation. Buy-and-holds, neural networks and decision tree models were implemented for comparison purposes. The XCS method attained higher return then the benchmarks. Besides, the generated rules were analyzed to reach some conclusion about the best behavior at different market conditions. The conclusion was that the generated rules were consistent with general financial knowledge.

Schmidbauer, Rösch, Sezer, and Tunalioğlu (2014) studied the effectiveness reduction suffered for trading rules-based systems when they are tested with out-of-sample data. The authors indicated this is a consequence of data-snooping bias. An a priori robustness strategy is proposed to reduce this adverse effect. This approach used GP, which encodes rules using a Backus–Naur form grammar. Fitness of individuals is computed from their profit and their hit-miss ratio. The approach requires opening, minimum, maximum, and closing prices to be input to the algorithm. A-priori robustness is ensured generating simulated realizations of these time series. Closing time series realizations are created using maximum entropy bootstrapping. Minimum and maximum signals are generated from ARMA models. The opening signal realizations are generated using kernel density estimation. The method was tested for FOREX trading of Euro/USD exchange with intra-day data from February to June 2011. A-priori robustness GP was tested against regular GP. The method was tested with different out-of-sample data, and different sets of GP parameters. The results indicated the implementation of a priori robustness and test time period were the main sources of variation in final profit. In-sample data robustness was curbed by the method, but enhanced robustness of out-of-sample tests. Profit seemed to be increased, but further work is needed to ensure repeatably profitable rules. Fridays proved to be a day with higher uncertainty than others where the a priori robustness method seemed to be more effective.

4.14. Trading execution (TX)

Trading execution problem is concerned with the methods used to fulfill trading orders efficiently. For example, a trading method could make the order to sell a stock which price has dropped dramatically, but the probability the actual order could be executed is low until stock price seems to stabilize. Another case occurs when transaction volume is high enough to impact prices. Brokers should schedule partial orders to avoid adverse effects.

Almgren and Chriss (2001) suggested the problem of execution schedules. This approach considers investor wishes to liquidate a specific asset before some fixed time limit. Prices are affected by volatility, drift and market impact. Market impact depends on transactions volume. The authors used stochastic dynamic programming to find optimal execution strategies, and Monte-Carlo simulations to investigate their performance. Nevertheless, this work did not made use of EC approaches.

5. Discussion and open problems

Results of the review are summarized in Table 3, and Figs. 1 and 2. The surveys used to define the scope of this review were not considered. Table 3 presents the percentage of references which treated a particular problem and used a particular solution method. Open research areas can be identified when a white space is present. White spaces indicate the specific problem has not been solved with a specific solution method. Fig. 1 summarizes total percentage of references per solution method. Fig. 2 summarizes total percentage of references per problem. Table 4, and Figs. 3 and 4 provide the same information about the references covered by other similar surveys. References found in other surveys are also called past references in this discussion. Care was taken to avoid counting repeated references. Most of the references of this review were published later in time than the scopes of other surveys. Some references which were published before the scope of other surveys were included for completion purposes.

Table 4 shows GA is the solution approach which was the most applied to solve different financial problems in the past. Table 3 shows this tendency has been continued. Fig. 3 shows GAs references are about 45% of total. GAs seemed to have become even more popular through time. Fig. 1 shows GAs references augment to about 60% of the total. The rise of interest in GAs applications is probably due GAs popularity has spread outside of the evolutionary computation community. This means people from other fields

Table 3
Summary of open problems. Percent of references per problem and approach.

Problem	GA	GP	LCS	MO	CO	EDA
ABN	2.77					
ARB	1.38	4.16				
BKR	1.38			1.38		
CS	6.94					
CM	1.38					
CP						
FA	2.77					
FC	20.83	6.94	2.77			1.38
ITR	2.77					
MKS	1.38				2.77	
PRC	1.38					
PSP	6.94	1.38		4.16		
T	8.33	8.33	1.38	2.77	2.77	1.38
TX						

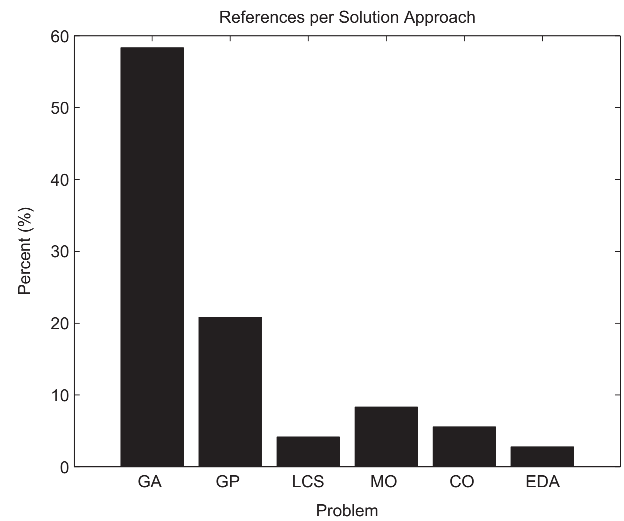

Fig. 1. Summary of references per solution approach.

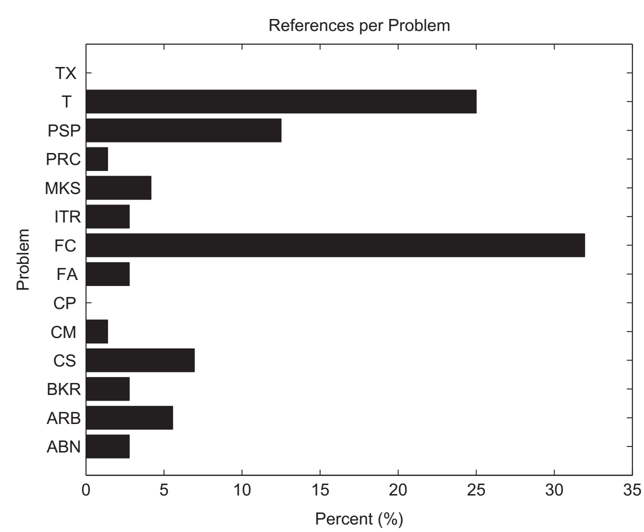

Fig. 2. Summary of references per problem.

Table 4
Summary of other surveys. Percent of references per problem and approach.

Problem	GA	GP	LCS	MO	CO	EDA
ABN	3.75	0.07	0.07			
ARB	0.07					
BKR	15.78	3.00				
CS	2.25	1.50				
CM						
CP				2.25		
FA	0.07			2.25		
FC	3.00			2.25		
ITR						
MKS	15.03	10.52	0.07	4.51		
PRC						
PSP	0.07			18.04		
T	3.00	2.25		3.75		
TX						

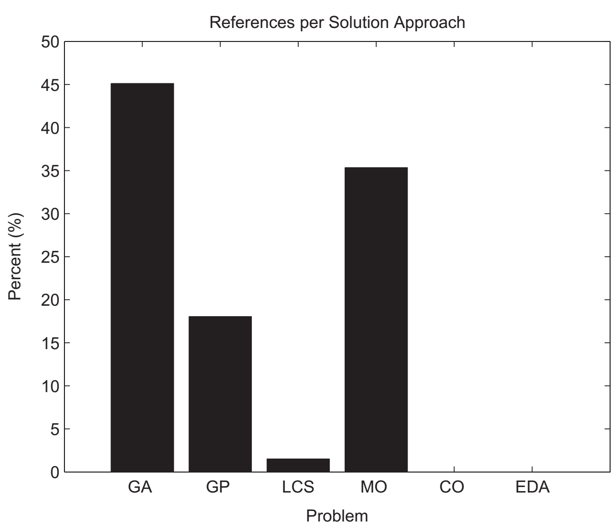

Fig. 3. Summary of other surveys per solution approach.

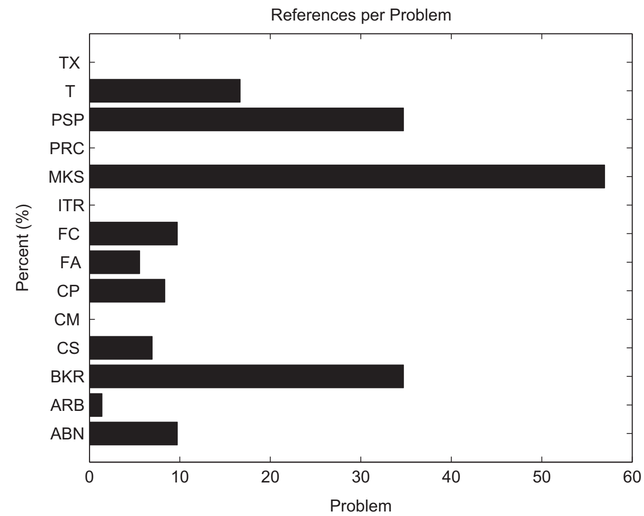

Fig. 4. Summary of other surveys per problem.

has adopted GAs to solve their respective problems. Other Darwinian approaches are yet in process to reach mainstream popularity outside the field. GAs applications have also changed with time. MKS and BKR were the most popular applications, according with past references, but new references indicate that FC, T, and PSP are now more popular than MKS. This could be explained probably MKS was the first link among the economics and computation community. Further research is needed to confirm it.

MO and GP references were at second and third place, respectively. Nevertheless, while the number of GP references seems to remain unchanged (about 20%), MO references percentage has dropped significantly. Although, the drop of percentage of references for MO case does not mean the community has lost interest in MOEAS. This drop seems a relative effect to the rise of interest in GAs. On the other hand, The number of problems approached with GP and MO seemed to have decreased with time. FC and T problems seemed to have substituted MKS for the GP case. This could be explained by the reported advantages of GP's ability to build explainable rules. GP has been applied to problems where new knowledge about markets and investors is desired. MKS has benefited from this properties, but the idea to have several GP agents working simultaneously appears to be costly. Island GAs allow a similar effect with a single population. Islands could be implemented in GP or other approaches.

LCS is the fourth approach with more references. Table 4 showed LCS have been used in the past for ABN and MKS. New references show the interest has changed to FC and T applications. The references only mentioned LCS and XCS types. Other variations of LCS have yet to be applied to financial problems. Nevertheless, Fig. 1 shows the interest for LCS in financial applications have augmented with time.

Fig. 4 shows the percentage of total references per problem. MKS, PSP, and BKR were the three most addressed problems in past references. Newer references indicate the interest has changed to FC and T applications. One hypothesis about this change is that probably PSP applications were defined to be trading problems in new references. Changes in portfolio are equivalent to trading decisions. The difference between PSP and T is the factor of risk, which is explicitly stated in PSP. Trading surely benefits from a return/risk model because it allows to include uncertainty into the optimization process. FC seemed to have attained higher levels of attention because better estimations of the future has direct impact in profit. Therefore, trading applications try to manage risk in the most profitable manner while FC applications try to reduce risk. In other words, T and FC interest raised together because they are closely related with each other.

Figs. 1 and 3 show new references have payed more attention to co-evolutionary approaches and EDAs. There is the possibility some of the referenced works has made use of co-evolution without explicitly report it. Co-evolution is more a concept than a concrete technique. This means the could be co-evolutionary approaches using GAs, GP, or any other evolutionary computation technique. The co-evolution term was found in past references, but these cases did not associated it to a EC approach. Therefore, it could not be counted as such. Other cases did not clearly stated their EC approach was co-evolutionary.

On the other hand, EDAs are a relatively new EC approach. It seems natural they are yet to find application at different fields. EDAs are specially concerned with deception, linkage and hierarchical problems. A methodology to characterize real-world problems using these concepts seems an open question. Nevertheless, both co-evolution and EDAs seem solution methods with high potential to be exploited in financial applications.

Figs. 2 and 4 show there are problems which have received relatively few attention in the literature. TX, PRC, ITR, and CM are the problems with less references. Most of these problems seemed to be neglected also in past references. Darwinian approaches are a promising option to find solution to them.

Finally, Tables 3 and 4 indicate financial applications of Darwinian approaches is yet open to further research. Both tables show many combinations of problems and approaches where no references were found. This occurs in both tables. Moreover, references in both tables are concentrated at few problem-approach pairs. This indicate even the problems already addressed are still open to further investigation.

6. Conclusions

This work presented a review of evolutionary computation to financial applications. The scope was limited to Darwinian approaches. Darwinian approaches are population-based approaches where selection of the fittest is implemented to find solutions. Genetic algorithms, genetic programming, learning classifier systems, multi-objective evolutionary algorithms, co-evolutionary algorithms and evolutionary estimation of distribution algorithms were the solution methods considered in this review.

The scope of this work was further refined through the revision of surveys related to applications of evolutionary computation to finance. It was found that no other review has exactly the same scope than the presented work. Scope intersection from past references was identified to exclude already covered references and time slots from the review. This ensured our survey covered novel references and reached conclusions form them.

References from each of the solution methods proposed for the defined scoped were presented. It was found that the application of Darwinian approaches to financial applications has been widely studied, although, there are still opened questions and approaches to be tested against specific financial problems.

Results were summarized in tables and figures which showed the percentage of references dedicated to each combination of problem and solution method. The figures summarized the percentage of references dedicated to specific approaches and problems separately. A similar analysis was conducted to the references addressed by related surveys to make a comparison between the scope and interests of past time references and the new ones. It was found that genetic algorithms are the most used approach for financial applications and the one applied to the wider set of problems. Genetic Programming and multi-objective evolutionary algorithms are the second and third places, respectively. The popularity of genetic algorithms seemed to have augmented with time, while Genetic Programming has remained stable. MOEAS references seemed to have diminished, but this seems a relative effect of the raise of genetic algorithm applications. The raise of references using GAs is probably caused by the wider acceptance this approach has found in other areas outside evolutionary computation. References studying other methods will increase when they become widely known in other fields.

The results indicated the interest of different problems and solutions approaches has changed through time. The combinations of problems and approaches is different from past to new references. Also, the problems addressed with a particular method have changed in the new references. Forecasting, portfolio optimization, and trading have been favored in recent works. This raise seems indicate a connection among these three problems. Relationship among problems needs to be further investigated.

The review indicated approaches like learning classifier systems, co-evolutionary algorithms and EDAs are promising opportunities to be applied to solve financial problems. The number of references found about these methods was low in both past references and recent works. Application of LCS has changed with time and the number of references has raised. The application of variations of the LCS to financial problems has been limited to LCS and XCS.

Some open problems were found in the literature. This means the references about them into the scope of this work were few. Trading execution, procurement, cash management, and credit portfolios management have still open questions to be addressed using Darwinian approaches.

Financial applications have been widely studied in the literature. Nevertheless, this study showed further investigation is possible. The analysis showed references tended to concentrate at certain combinations of problem and solution method, while the rest of combinations attain relatively low attention in the literature. Bankruptcy detection using GAs, market simulation using GAs and market simulation using Genetic Programming were the most popular pairs in past references. New references favored forecasting applications using GAs, and trading using GAs and GP. There is still a number of combinations of problems and approaches to be addressed together. Nevertheless, the likelihood of certain combinations should be further evaluated and investigated.

Acknowledgements

The first and second authors are with the Research Group with Strategic Focus on Intelligent Systems of the National School of Engineering and Sciences at the Tecnológico de Monterrey, and gratefully acknowledge its support.

The first author thanks the Consejo Nacional de Ciencia y Tecnología (CONACyT) for its financial support through the PNPC program.

References

Adebiyi, A., & Ayo, C. (2015). Portfolio selection problem using generalized differential evolution 3. *Applied Mathematical Sciences, 9*, 2069–2082.
Almgren, R., & Chriss, N. (2001). Optimal execution of portfolio transactions. *Journal of Risk, 3*, 5–40.
Andriosopoulos, K., & Nomikos, N. (2014). Performance replication of the spot energy index with optimal equity portfolio selection: Evidence from the UK, US and Brazilian markets. *European Journal of Operational Research, 234*, 571–582.
Araújo, R. d. A., & Ferreira, T. A. (2013). A morphological-rank-linear evolutionary method for stock market prediction. *Information Sciences, 237*, 3–17.
Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques–part II: Soft computing methods. *Expert Systems with Applications, 36*, 5932–5941.
Back, B., Laitinen, T., & Sere, K. (1996). Neural networks and genetic algorithms for bankruptcy predictions. *Expert Systems with Applications, 11*, 407–413.
Bahrammirzaee, A. (2010). A comparative survey of artificial intelligence applications in finance: Artificial neural networks, expert system and hybrid intelligent systems. *Neural Computing and Applications, 19*, 1165–1195.
Bernardo, D., Hagras, H., & Tsang, E. (2013). A genetic type-2 fuzzy logic based system for financial applications modelling and prediction. In *IEEE international conference on fuzzy systems (FUZZ)* (pp. 1–8). IEEE.
Bolsa Mexicana de Valores (2015). Notas sobre Índices.
Cerda, R. A. (2005). On social security financial crisis. *Journal of Population Economics, 18*, 509–517.
Chang, Y.-H. (2010). Adopting co-evolution and constraint-satisfaction concept on genetic algorithms to solve supply chain network design problems. *Expert Systems with Applications, 37*, 6919–6930.
Chen, S.-H. (2003). Agent-based computational macro-economics: A survey. In *Meeting the challenge of social problems via agent-based simulation* (pp. 141–170). Springer.
Chen, A.-P., & Chang, Y.-H. (2005). Using extended classifier system to forecast S&P futures based on contrary sentiment indicators. *The 2005 IEEE congress on evolutionary computation* (Vol. 3, pp. 2084–2090). IEEE.
Chen, G., & Chen, X. (2011). A hybrid of adaptive genetic algorithm and pattern search for stock index optimized replicate. In *2nd International conference on artificial intelligence, management science and electronic commerce (AIMSEC)* (pp. 4912–4915). IEEE.
Chen, Y., & Hirasawa, K. (2010). Generating trading rules on the stock markets with robust genetic network programming using variance of fitness values. In *Proceedings of SICE annual conference* (pp. 3095–3102). IEEE.
Chen, J.-S., Hou, J.-L., Wu, S.-M., & Chang-Chien, Y.-W. (2009). Constructing investment strategy portfolios by combination genetic algorithms. *Expert Systems with Applications, 36*, 3824–3828.
Chen, S.-H., & Lee, W.-C. (1997). Option pricing with genetic algorithms: A second report. *International conference on neural networks* (Vol. 1, pp. 21–25). IEEE.
da Costa Moraes, M. B., & Nagano, M. S. (2014). Evolutionary models in cash management policies with multiple assets. *Economic Modelling, 39*, 1–7.
de Araujo, R., Madeiro, F., de Sousa, R. P., Pessoa, L. F., & Ferreira, T. (2006). An evolutionary morphological approach for financial time series forecasting. In *IEEE congress on evolutionary computation, CEC 2006* (pp. 2467–2474). IEEE.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Lecture Notes in Computer Science, 1917*, 849–858.

de Brito, R. F., & Oliveira, A. L. (2012). Comparative study of FOREX trading systems built with SVR+ GHSOM and genetic algorithms optimization of technical indicators. *IEEE 24th international conference on tools with artificial intelligence (ICTAI)* (Vol. 1, pp. 351–358). IEEE.

del Arco-Calderón, C. L., Vinuela, P. I., & Castro, J. C. H. (2004). Forecasting time series by means of evolutionary algorithms. In *Parallel problem solving from nature-PPSN VIII* (pp. 1061–1070). Springer.

Donate, J. P., & Cortez, P. (2014). Evolutionary optimization of sparsely connected and time-lagged neural networks for time series forecasting. *Applied Soft Computing, 23*, 432–443.

Ferreira, C. (2001). Gene expression programming: A new adaptive algorithm for solving problems. *Complex Systems, 13*, 87–129.

Finger, M., & Wasserman, R. (2004). Approximate and limited reasoning: Semantics, proof theory, expressivity and control. *Journal of Logic And Computation, 14*, 179–204.

Franke, R. (1998). Coevolution and stable adjustments in the cobweb model. *Journal of Evolutionary Economics, 8*, 383–406.

Gandomi, A. H., & Alavi, A. H. (2011). Multi-stage genetic programming: A new strategy to nonlinear system modeling. *Information Sciences, 181*, 5227–5239.

Garcia-Almanza, A. L., & Tsang, E. P. (2006). Forecasting stock prices using genetic programming and chance discovery. In *12th International conference on computing in economics and finance* (p. 489).

García, S., Quintana, D., Galván, I. M., & Isasi, P. (2014). Multiobjective algorithms with resampling for portfolio optimization. *Computing and Informatics, 32*, 777–796.

Gaspar-Cunha, A., Recio, G., Costa, L., & Estébanez, C. (2014). Self-adaptive MOEA feature selection for classification of bankruptcy prediction data. *The Scientific World Journal, 2014.* Article ID 314728 (20 pages). http://dx.doi.org/10.1155/2014/314728.

Gerard, P., Stolzmann, W., & Sigaud, O. (2002). YACS: A new learning classifier system using anticipation. *Soft Computing, 6*, 216–228.

Ghosh, P., & Chinthalapati, V. (2014). Financial time series forecasting using agent based models in equity and FX markets. In *6th Computer science and electronic engineering conference (CEEC)* (pp. 97–102). IEEE.

Giulioni, G., D'Orazio, P., Bucciarelli, E., & Silvestri, M. (2015). Building artificial economies: From aggregate data to experimental microstructure. A methodological survey. In *Advances in artificial economics* (pp. 69–78). Springer.

Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning* (Vol. 412). Reading, Menlo Park: Addison-Wesley.

Goonatilake, S., Campbell, J. A., & Ahmad, N. (1995). Genetic-fuzzy systems for financial decision making. In *Advances in fuzzy logic. Neural networks and genetic algorithms* (pp. 202–223). Springer.

Grefenstette, J., Gopal, R., Rosmaita, B., & Van Gucht, D. (1985). Genetic algorithms for the traveling salesman problem. In *Proceedings of the first international conference on genetic algorithms and their applications* (pp. 160–168). New Jersey: Lawrence Erlbaum.

Gupta, P., Mehlawat, M. K., & Mittal, G. (2012). Asset portfolio optimization using support vector machines and real-coded genetic algorithm. *Journal of Global Optimization, 53*, 297–315.

Hamida, S. B., Abdelmalek, W., & Abid, F. (2014). Applying dynamic training-subset selection methods using genetic programming for forecasting implied volatility. *Computational Intelligence*.

Harik, G. R., Lobo, F. G., & Goldberg, D. E. (1999). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation, 3*, 287–297.

Harik, G. R., Lobo, F. G., & Sastry, K. (2006). Linkage learning via probabilistic modeling in the extended compact genetic algorithm (ECGA). In *Scalable optimization via probabilistic modeling* (pp. 39–61). Springer.

Hillis, W. D. (1990). Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D: Nonlinear Phenomena, 42*, 228–234.

Hirabayashi, A., Aranha, C., & Iba, H. (2009). Optimization of the trading rule in foreign exchange using genetic algorithm. In *Proceedings of the 11th annual conference on genetic and evolutionary computation* (pp. 1529–1536). ACM.

Hochreiter, R. (2014). An evolutionary optimization approach to risk parity portfolio selection. arXiv preprint <arXiv:1411.7494>.

Hochreiter, R. (2015). Computing trading strategies based on financial sentiment data using evolutionary optimization. arXiv preprint <arXiv:1504.02972>.

Hochreiter, R., & Wozabal, D. (2010). Evolutionary estimation of a coupled Markov chain credit risk model. In *Natural computing in computational finance* (pp. 31–44). Springer.

Holmes, J. H., Lanzi, P. L., Stolzmann, W., & Wilson, S. W. (2002). Learning classifier systems: New models, successful applications. *Information Processing Letters, 82*, 23–30.

Holyoak, K. J., & Holland, J. H. (1989). *Induction: Processes of inference, learning, and discovery*. MIT press.

Horn, J., Nafpliotis, N., & Goldberg, D. E. (1993). Multiobjective optimization using the niched Pareto genetic algorithm. *IlliGAL report* (pp. 61801–2296).

Hruschka, E. R., Campello, R. J. G. B., Freitas, A. A., & De Carvalho, A. P. L. F. (2009). A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 39*, 133–155.

Huang, C.-F., Chang, C.-H., Kuo, L.-M., Lin, B.-H., Hsieh, T.-N., & Chang, B.-R. (2012). A genetic-search model for first-day returns using IPO fundamentals. *International conference on machine learning and cybernetics (ICMLC)* (Vol. 5, pp. 1662–1667). IEEE.

Huang, C.-F., Hsu, C.-J., Chen, C.-C., Chang, B. R., & Li, C.-A. (2015). An intelligent model for pairs trading using genetic algorithms. *Computational Intelligence and Neuroscience, 501*, 939606.

Huang, H., Pasquier, M., & Quek, C. (2009). Financial market trading system with a hierarchical coevolutionary fuzzy predictive model. *IEEE Transactions on Evolutionary Computation, 13*, 56–70.

Hu, Y., Feng, B., Zhang, X., Ngai, E., & Liu, M. (2015). Stock trading rule discovery with an evolutionary trend following model. *Expert Systems with Applications, 42*, 212–222.

Jiang, Y., Xu, L., Wang, H., & Wang, H. (2009). Influencing factors for predicting financial performance based on genetic algorithms. *Systems Research and Behavioral Science, 26*, 661–673.

Jing, L. (2010). Data modeling for searching abnormal noise in stock market based on genetic algorithm. In *2010 International symposium on computational intelligence and design* (Vol. 2, pp. 129–131).

Johnson, D. S., & McGeoch, L. A. (1997). The traveling salesman problem: A case study in local optimization. *Local Search in Combinatorial Optimization, 1*, 215–310.

Jun, T., & He, L. (2012). Genetic optimization of BP neural network in the application of suspicious financial transactions pattern recognition. In *International conference on management of e-commerce and e-government (ICMeCG)* (pp. 280–284). IEEE.

Kampouridis, M., Chen, S.-H., & Tsang, E. (2012). Microstructure dynamics and agent-based financial markets: Can dinosaurs return? *Advances in Complex Systems, 15*.

Kanungo, R. P. (2004). Genetic algorithms: Genesis of stock evaluation. *Economics WPA Working Paper*.

Karatahansopoulos, A., Sermpinis, G., Laws, J., & Dunis, C. (2014). Modelling and trading the Greek stock market with gene expression and genetic programing algorithms. *Journal of Forecasting, 33*, 596–610.

Kim, K. J., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications, 19*, 125–132.

Kingdon, J., Taylor, J., & Mannion, C. (1997). *Intelligent systems and financial forecasting*. New York: Springer-Verlag, Inc.

Knowles, J. D., & Corne, D. W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation, 8*, 149–172.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection* (Vol. 1). MIT press.

Krink, T., & Paterlini, S. (2011). Multiobjective optimization using differential evolution for real-world portfolio optimization. *Computational Management Science, 8*, 157–179.

Lahsasna, A., Ainon, R. N., & Teh, Y. W. (2010). Credit scoring models using soft computing methods: A survey. *International Arab Journal of Information Technology, 7*, 115–123.

Lakemeyer, G. (1994). Limited reasoning in first-order knowledge bases. *Artificial Intelligence, 71*, 213–255.

Laumanns, M., & Ocenasek, J. (2002). Bayesian optimization algorithms for multi-objective optimization. In *Parallel problem solving from nature-PPSN VII* (pp. 298–307). Springer.

Lim, M., & Coggins, R. J. (2005). Optimal trade execution: An evolutionary approach. *The 2005 IEEE congress on evolutionary computation* (Vol. 2, pp. 1045–1052). IEEE.

Lin, F., Liang, D., Yeh, C. C., & Huang, J. C. (2014). Novel feature selection methods to financial distress prediction. *Expert Systems with Applications, 41*, 2472–2483.

Lipinski, P. (2007). ECGA vs. BOA in discovering stock market trading experts. In *Proceedings of the 9th annual conference on genetic and evolutionary computation* (pp. 531–538). ACM.

Lipinski, P. (2012). Parallel evolutionary algorithms for stock market trading rule selection on many-core graphics processors. In *Natural computing in computational finance* (pp. 79–92). Springer.

Lohpetch, D., & Corne, D. (2009). Discovering effective technical trading rules with genetic programming: Towards robustly outperforming buy-and-hold. In *World congress on nature & biologically inspired computing, NaBIC* (pp. 439–444). IEEE.

Lo, A. W., Mamaysky, H., & Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The Journal of Finance, 55*, 1705–1770.

Lwin, K., Qu, R., & Kendall, G. (2014). A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing, 24*, 757–772.

Mahfoud, S., & Mani, G. (1996). Financial forecasting using genetic algorithms. *Applied Artificial Intelligence, 10*, 543–566.

Malkiel, B. G., & Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance, 25*, 383–417.

Markose, S., Tsang, E., Er, H., & Salhi, A. (2001). Evolutionary arbitrage for FTSE-100 index options and futures. *Proceedings of the 2001 congress on evolutionary computation* (Vol. 1, pp. 275–282). IEEE.

Martinez-Jaramillo, S., & Tsang, E. P. (2009). An heterogeneous, endogenous and coevolutionary GP-based financial market. *IEEE Transactions on Evolutionary Computation, 13*, 33.

Matsui, K., & Sato, H. (2009). A comparison of genotype representations to acquire stock trading strategy using genetic algorithms. In *International conference on adaptive and intelligent systems, ICAIS'09* (pp. 129–134). IEEE.

Matsui, K., & Sato, H. (2010). Neighborhood evaluation in acquiring stock trading strategy using genetic algorithms. In *International conference of soft computing and pattern recognition (SoCPaR)* (pp. 369–372). IEEE.

Matsumura, K., & Kakinoki, H. (2014). Portfolio strategy optimizing model for risk management utilizing evolutionary computation. *Electronics and Communications in Japan, 97*, 45–62.

Ma, I., Wong, T., Sankar, T., & Siu, R. (2004). Forecasting the volatility of a financial index by wavelet transform and evolutionary algorithm. *IEEE international conference on systems, man and cybernetics* (Vol. 6, pp. 5824–5829). IEEE.

Meyer, T. P., & Packard, N. H. (1992). Local forecasting of high-dimensional chaotic dynamics. *Santa Fe Institute studies in the sciences of complexity-proceedings* (Vol. 12). Addison-Wesley Publishing Co, p. 249.

Miller, J. F., & Thomson, P. (2000). Cartesian genetic programming. In *Genetic programming* (pp. 121–132). Springer.

Ngai, E., Hu, Y., Wong, Y., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems, 50*, 559–569.

Nguyen, H. D., Yoshihara, I., Yamamori, K., & Yasunaga, M. (2007). Implementation of an effective hybrid GA for large-scale traveling salesman problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 37*, 92–99.

Nikolaos, L., & Iordanis, E. (2010). Default prediction and bankruptcy hazard analysis into recurrent neuro-genetic-hybrid networks to adaboost M1 regression and logistic regression models in finance. In *7th WSEAS international conference on engineering education* (pp. 22–24).

Packard, N. H. (1990). A genetic learning algorithm for the analysis of complex data. *Complex Systems, 4*, 543–572.

Parracho, P., Neves, R., & Horta, N. (2011). Trading with optimized uptrend and downtrend pattern templates using a genetic algorithm kernel. In *IEEE congress on evolutionary computation (CEC)* (pp. 1895–1901). IEEE.

Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (2000). Bayesian optimization algorithm, population sizing, and time to convergence. In *GECCO* (pp. 275–282).

Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (2000). Hierarchical problem solving and the bayesian optimization algorithm. In *GECCO* (pp. 267–274).

Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. arXiv preprint <arXiv:1009.6119>.

Polanski, A. (2011). Genetic algorithm search for predictive patterns in multidimensional time series. *Complex Systems, 19*, 195.

Ponsich, A., Jaimes, A. L., & Coello, C. A. C. (2013). A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Transactions on Evolutionary Computation, 17*, 321–344.

Potter, M. A., & De Jong, K. A. (1994). A cooperative coevolutionary approach to function optimization. In *Parallel problem solving from nature-PPSN III* (pp. 249–257). Springer.

Protopapas, M. K., Battaglia, F., & Kosmatopoulos, E. B. (2010). Coevolutionary genetic algorithms for establishing Nash equilibrium in symmetric Cournot games. *Advances in Decision Sciences, 2010*.

Radeerom, M. (2014). Automatic trading system based on genetic algorithm and technical analysis for stock index. *Change, 1*, 100.

Ranković, V., Drenovak, M., Stojanović, B., Kalinić, Z., & Arsovski, Z. (2014). The mean-value at risk static portfolio optimization using genetic algorithm. *Computer Science and Information Systems, 11*, 89–109.

Ravi, V., Kurniawan, H., Thai, P. N. K., & Kumar, P. R. (2008). Soft computing system for bank performance prediction. *Applied Soft Computing, 8*, 305–315.

Rimcharoen, S., Sutivong, D., & Chongstitvatana, P. (2005). Prediction of the stock exchange of Thailand using adaptive evolution strategies. In *17th IEEE international conference on tools with artificial intelligence, ICTAI 05* (pp. 5–pp). IEEE.

Ryan, C., Collins, J., & Neill, M. O. (1998). Grammatical evolution: Evolving programs for an arbitrary language. In *Genetic programming* (pp. 83–96). Springer.

Safarzyńska, K., & van den Bergh, J. C. (2010). Evolutionary models in economics: A survey of methods and building blocks. *Journal of Evolutionary Economics, 20*, 329–373.

Sarijaloo, A., & Moradbakloo, A. (2014). Asset management using genetic algorithm: Evidence from Tehran stock exchange. *Management Science Letters, 4*, 221–226.

Sastry, K., & Goldberg, D. E. (2003). Scalability of selectorecombinative genetic algorithms for problems with tight linkage. In *Genetic and evolutionary computation GECCO 2003* (pp. 1332–1344). Springer.

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st international conference on genetic algorithms* (pp. 93–100). L. Erlbaum Associates Inc.

Schmidbauer, H., Rösch, A., Sezer, T., & Tunalioğlu, V. S. (2014). Robust trading rule selection and forecasting accuracy. *Journal of Systems Science and Complexity, 27*, 169–180.

Shao, M., Smonou, D., Kampouridis, M., & Tsang, E. (2014). Guided fast local search for speeding up a financial forecasting algorithm. In *IEEE conference on computational intelligence for financial engineering & economics (CIFEr)* (pp. 325–332). IEEE.

Sinha, A., Malo, P., Frantsev, A., & Deb, K. (2014). Finding optimal strategies in a multi-period multi-leader–follower Stackelberg game using an evolutionary algorithm. *Computers & Operations Research, 41*, 374–385.

Stolzmann, W. (2000). An introduction to anticipatory classifier systems. In *Learning classifier systems* (pp. 175–194). Springer.

Tapia, M. G. C., & Coello, C. A. C. (2007). Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In *IEEE congress on evolutionary computation* (Vol. 7, pp. 532–539).

Tezuka, M., Munetomo, M., & Akama, K. (2007). Genetic algorithm to optimize fitness function with sampling error and its application to financial optimization problem. In *Evolutionary computation in dynamic and uncertain environments* (pp. 417–434). Springer.

Tsang, E., Markose, S., Garcia, A., Almad Garcia, & Er, H. (2006). EDDIE for discovering arbitrage opportunities. In *International conference on numerical methods for finance*.

Tsang, E. P., Li, J., Markose, S., Er, H., Salhi, A., & Iori, G. (2000). EDDIE in financial decision making. *Journal of Management and Economics, 4*.

Tsang, E. P., & Martinez-Jaramillo, S. (2004). Computational finance. *IEEE Computational Intelligence Society Newsletter, 3*.

Tsang, E., Yung, P., & Li, J. (2004). EDDIE-Automation, a decision support tool for financial forecasting. *Decision Support Systems, 37*, 559–565.

Valenzuela-Rendón, M. (1991). The fuzzy classifier system: A classifier system for continuously varying variables. In *Fourth international conference on genetic algorithms (ICGA)* (pp. 346–353).

Vanstone, B., & Tan, C. (2003). A survey of the application of soft computing to investment and financial trading. *Information technology papers* (p. 13).

Varetto, F. (1998). Genetic algorithms applications in the analysis of insolvency risk. *Journal of Banking & Finance, 22*, 1421–1439.

Verikas, A., Kalsyte, Z., Bacauskiene, M., & Gelzinis, A. (2010). Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: A survey. *Soft Computing, 14*, 995–1010.

Wagman, L. (2003). Stock portfolio evaluation: An application of genetic-programming-based technical analysis. *Genetic Algorithms and Genetic Programming at Stanford, 2003*, 213–220.

Wagner, N., Michalewicz, Z., Khouja, M., & McGregor, R. R. (2007). Time series forecasting for dynamic environments: The DyFor genetic program model. *IEEE Transactions on Evolutionary Computation, 11*, 433–452.

Wang, W., Hu, J., & Dong, N. (2014). A convex-risk-measure based model and genetic algorithm for portfolio selection. *Mathematical Problems in Engineering, 2014*.

Wilson, S. W. (1994). ZCS: A zeroth level classifier system. *Evolutionary Computation, 2*, 1–18.

Yaman, A., Lucci, S., & Gertner, I. (2014). Evolutionary algorithm based approach for modeling autonomously trading agents. *Intelligent Information Management, 2014*.

Zitzler, E., Laumanns, M., Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-report*.