INTELLIGENT TRADING SYSTEMS

Luis F. de Mingo, Nuria Gómez Blas, Alberto Arteta

Abstract: *During last 50 years, the markets have been object of study. Technical and fundamental indicators have been used to try to predict the behavior of the market and then execute buying or selling orders. Neural networks are currently being used with good results although they can be useless after a period of time. This paper proposes an algorithm that combines bioinspired techniques to maximize the hits in the prediction rates. The proposal shown in this paper relies in an ANN to achieve these goals. The differential factors of this approach are the election of the ANN structure with grammatical swarm and the training process through the use of HydroPSO. Also a grammatical swarm algorithm is used to generate trading rules, this method shows better results than the first approach. This combination of techniques provides an automatic way to define the most suitable bioinspired model for the instrument in our analysis.*

Keywords: *Trading strategies, Technical indicators, Grammatical Swarm, Neural networks.*

ACM Classification Keywords: *F.1.1 Theory of Computation - Models of Computation, I.2.6 Artificial Intelligence.*

Introduction

Natural sciences, and especially biology, represented a rich source of modeling paradigms. Well-defined areas of artificial intelligence (genetic algorithms, neural networks), mathematics, and theoretical computer science (L systems, DNA computing) are massively influenced by the behaviour of various biological entities and phenomena. In the last decades, new emerging fields of so-called natural computing identify new (unconventional) computational paradigms in different forms [Bonabeau et al., 1999]. There are attempts to define new mathematical and theoretical models inspired by nature. Moreover, computational paradigms suggested by biochemical phenomena [Kennedy et al., 2001] are object of study.

Genetic algorithms (GAs) are problem solving methods (or heuristics) that mimic the process of natural evolution. These algorithms utilize the concepts of natural selection to determine the best solution for a problem. As a result, GA are commonly used as optimizers that adjust parameters to minimize or maximize some feedback measure, which can then be used independently or in the construction of a neural network (ANN). Their use can be helpful when optimizing the chances for a price to rise (or decrease) [Prasad, 2004] and [Lin et al., 2004].

Stock market forecasting has been implemented using different approaches in the literature. Multilayer perceptrons [de Oliveira et al., 2014], self-organizing maps [Sarlin, 2014], radial basis functions neural networks [Zhang and Liao, 2014], support vector machines [Han and Rung-Ching, 2007] or even social learning [Xiaoac et al., 2014]. Most of them are based on a data training set and the model tries to approximate/forecast the output. In this paper we proposed a model that generates a rule set that can be used in an automatic trading system to place market orders (buy/sell).

Basic Stock Market Indicators

In statistics, a moving average (rolling average or running average) is a calculation to analyze data points by creating a series of averages of different subsets of the full data set. It is also called a moving mean (MM) [Booth et al., 2006] or rolling mean and is a type of finite impulse response filter. Variations include: simple, and cumulative, or weighted forms (described below).

Given a series of numbers and a fixed subset size, the first element of the moving average is obtained by taking the average of the initial fixed subset of the number series. Then the subset is modified by "shifting forward"; that is, excluding the first number of the series and including the next number following the original subset in the series. This creates a new subset of numbers, which is averaged. This process is repeated over the entire data series. The plot line connecting all the (fixed) averages is the moving average. A moving average is a set of numbers, each of which is the average of the corresponding subset of a larger set of datum points. A moving average may also use unequal weights for each datum value in the subset to emphasize particular values in the subset.

A moving average is commonly used with time series data to smooth out short-term fluctuations and highlight longer-term trends or cycles. The threshold between short-term and long-term depends on the application, and the parameters of the moving average will be set accordingly. For example, it is often used in technical analysis of financial data, like stock prices, returns or trading volumes. It is also used in economics to examine gross domestic product, employment or other macroeconomic time series. Mathematically, a moving average is a type of convolution and so it can be viewed as an example of a low-pass filter used in signal processing. When used with non-time series data, a moving average filters higher frequency components without any specific connection to time, although typically some kind of ordering is implied. Viewed simplistically it can be regarded as smoothing the data.

HydroPSO

HydroPSO is an enhanced version of the canonical Particle Swarm Optimization (PSO) technique which was implemented and developed by [Kennedy and Ebehart, 1995]. As explained, PSO is a population-based stochastic optimization technique inspired by social behaviour of bird flocking. This is part of the of evolutionary optimization techniques such as Genetic Algorithms (GA). The main problem PSO faces is the exploration within a multi-dimensional solution space due to the lack of evolution operators which can endanger the adaptability of the neural network by falling into local optima. HydroPSO, however is capable of performing high sensitive analysis just by using the Latin Hypercube One-At-a-Time (LH-OAT) method [van Griensven et al., 2006; Iman, 2008; Singh et al., 2010; Penga and Lua, 2013], the power of HydroPSO is the increase of ability to adjust better to new models.

The LH-OAT provides a constrained sampling scheme instead of random sampling according to the direct Monte Carlo simulation. In the LH-OAT, the region is uniformly divided into N non-overlapping intervals for each random variable; where N is the number of random numbers. These need to be generated for each random variable. The N non-overlapping intervals are selected to be have the same probability of occurrence. Then, N different values in the N non-overlapping intervals are randomly chosen for each random variable. This can be accomplished by generating N random numbers in the first place. These represent the percentage position of each generated value corresponding to the variable within an interval. In the bottom layer of HydroPSO, several swarms evolve in a parallel way to avoid being trapped in local optima. The learning strategy for each swarm is the well-known comprehensive learning method with a newly designed mutation operator. As the evolution processes at the bottom layer, one particle for each swarm is selected as a candidate to construct the swarm in the top layer, which evolves by the same strategy employed in the bottom layer. The local search strategy based on LH-OAT is imposed on particles in the top layer every specified number of generations

Neural Network training using HydroPSO

The working dynamic ANN can be performed just by running these two following processes: creating a neural network and finding the right weights by using the HydroPSO package. Given a neural network architecture, every weight is coded as a genotype; then the network is trained by running the Hydroparticle swarm optimization algorithm.

First, three random particles (r_1, r_2, r_3) are selected, note that they must be different from each other. The second task is to create a mutated value for each dimension j of the particle according to the evolutionary algorithm.

New particles are created according to standard PSO formulas: Velocity and new position. A basic velocity and position clamping is performed as well. Then the new values (velocity, position) are checked to determine whether they exceed the threshold defined in the algorithm. If the newly created particle is proven to be better, it will replace the old one for the next generation. Both personal best and global best particle vectors are updated as well. This whole process occurs over again until the Halt condition is reached.

In our case, the solutions of the ANN are optimized by HydroPSO algorithm. The fitness function to be optimized is :

$$\phi(i) = \frac{|x_i - y_i|}{x_i} \tag{1}$$

here, x_i is the particle that represents the real value that a market instrument gets in the moment i and y_i is the particle representing the value that our ANN returns.

Equations used in the particle swarm optimization training process are below; c_1 and c_2 are two positive constants, R_1 and R_2 are two random numbers belonging to $[0, 1]$ and w is the inertia weight. These equations define how the genotype values change along iterations; In other words this equations show how neural network weights change.

$$x_{in}(t+1) = x_{in}(t) + v_{in}(t+1) \tag{2}$$

The equations above make the network weights gets updated until the Halt condition is reached, that is to say, either the error is minimized (fitness function is closed to zero) or a maximum number of iterations is reached.

This basic example reveals that the previously defined PSO algorithm can be successfully applied in the training stage in order to solve the convergence problem when working with high dimension individuals. The XOR example with dimension 9 is a good candidate to start with combining classical neural networks with Swarm intelligence.

Grammatical Swarm

Grammatical Swarm (GS) [O'Neill and Brabazon, 2006] relates Particle Swarm algorithm to a Grammatical Evolution (GE); genotype-phenotype mapping to generate programs in an arbitrary language [O'Neill and

Brabazon, 2004]. The equations for the particle swarm algorithm are updated by adding new constraints to velocity and location dimension values, such us vmax (bounded to ± 255), and search space dimensions which are bounded to the range $[0, 255]$ (This is denoted as cmin and cmax, respectively). Note that this is a continuous swarm algorithm with real-valued particle vectors. The standard GE mapping function is adopted, with the real-values in the particle vectors being rounded up or down to the nearest integer value for the mapping process. In the current implementation of GS, fixed-length vectors are used, which implies that it is possible for a variable number of dimensions to be used during the program construction genotype-phenotype mapping process. A vector's elements (values) may be used more than once if wrapping occurs, and it is also possible that not all dimensions are used during the mapping process. (This can happen whenever a program is generated before reaching the end of the vector).

Neural Network topology using GS: first approach

Previous PSO model applied to a fixed neural network is a good training solution, however it does not define any kind or topology properties as it only obtains the best weight values. Following grammars can be used with Grammatical Swarm algorithms in order to obtain a network topology for a given problem.

This grammar can specify a feed-forward neural network topology with consecutive layers, that is to say, a classical Multilayer Perceptron.

```
<layers> ::= <layer> | <layer>, <layers>
<layer> ::= <digit>
<digit> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Next grammar is able to generate feed-forward connections not only with one consecutive layer but also with more than one consecutive layer. Such connections are defined by the `<connections>` non terminal, where the `<digit>` means the n-consecutive layer.

```
<layers> ::= <layer> | <layer>, <layers>
<layer> ::= <digit> -- <connections> --
<connections> := <digit> | <digit>, <connections>
<digit> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

The whole algorithm is summarized as follows:

1. Create an initial population of genotypes.
2. For genotype i
   (a) Using genotype and grammar to obtain a neural architecture.
   (b) Compute Fitness of genotype.
      - Apply previous HydroPSO algorithm to train the genotype network.
   (c) Modified the best individual if appropiate.
3. Update velocity of genotype i.
4. Update position of genotype i.
5. If stop condition is not satisfied go to step 2.

This neural network model is a powerful one as a network topology is chosen and trained only with the input and output pattern data sets. Both tools, topology and training, are based on grammatical swarm and particle swarm optimization respectively.

Another useful approach could be to defined a grammar with the weight values of neural networks connections instead of training the topology using a PSO algorithm.

Trading Results

Technical analysis utilizes models and trading rules based on price and volume transformations, such as the relative strength index, moving averages, regressions, inter-market and intra-market price correlations, business cycles, stock market cycles or, classically, through recognition of chart patterns. Technical analysis stands for mathematical indicators (volume, prices and functions) in contrast to the fundamental analysis approach (status of the companies, government decisions, etc).

Technical analysis is widely used among traders and financial professionals and is very often used by active day traders, market makers and pit traders. In the 1960s and 1970s technicality was widely dismissed by academics; however in recent reviews, Irwin and Park reported that 56 of 95 modern studies found that Technical studies produces positive results. A more detailed explanation of the ANN achievements in the stock markets can be found in [Dase and Pawar, 2010]

Data are obtained from `tradingmotion.com`, it provides the stock price every 30 minutes from 2002 until today, and so the common trading indicators, such as RSI, MACD, BBands, etc. There are more than 50000 patterns (that is, 12 years times 16 market ticks a day (8 hours)). We have randomly taken 90% of these for training purposes and 10% for testing purposes.

Next sections will show the results obtained in our analysis by using (neural networks and grammatical swarm) when applied to the stocks market.

Neural Networks with HydroPSO learning and Grammatical Swarm architecture

This neural network has been trained to forecast the IBEX indicator. Data from 2002 to 2014, obtained every 30 minutes, are used. There are 55490 patterns. Stock market dataset is made of 176 inputs (16 trading indicators from t to $t-10$) and 2 outputs corresponding to the price change: $Price_t - Price_{t+1}$ and $Price_t - Price_{t+5}$. All data are normalized in interval $(-1, 1)$. Following list shows the indicators: *Price change, EMA, Cross average, MACD line, MACD signal, Bollinger bandwidth, Bollinger center distance, Momentum, RSI, ATR, Aroon up, Aroon down, CCI, Stochastic K, Stochastic D, EOM.*

The best performance has been achieved with a neural network (multilayer perceptron with 100 hidden neurons) trained with a HydroPSO algorithm and a grammatical swarm one, in comparison with the classical back propagation algorithm. Figures 1, 2 and tables 1, 2 show obtained results.

Trading rules by using Grammatical Swarm

In this section we display the results when using a BNF grammar to generate a evolutionary algorithm implemented in C++. At first 16 trading indicators are used; and then the ones with with a low influence in the forecasting results are discarded

Table 1: Neural network training performance

Performance results	
Main chart: IBEX 30 minute bars [27/12/2001 - 08/03/2013]	
Net P&L	25183,09 €
Gross P&L	39280,00 €
Profit factor	1,10040504
Sharpe ratio	0,215754143
Slippage per side	-2,04524309
Commission per side	5,75 €
Annual ROI	1,95%
Mathematical expectation	73,01115242
Analyzed sessions	2841
Sessions in market	771
Winning sessions	430
Winning sessions profit	275998,11 €
Winning sessions average	641,86 €
Losing sessions	341
Losing sessions profit	-250815,02 €
Losing sessions average	-735,53 €
Worst drawdown	-31216,25 €
Best session	3570,00 €
Worst session	-5140,00 €

Next list shows the trend which was detected by the grammatical swarm algorithm; After that, ADX, MACD and Bollinger Bands were used, although the only relevant indicator turned out to be the ADX. A graphical data flow is shown in figure 3.

```
if ((adx.GetADX()[0]<30) && (adx.GetADX()[1]>30)) {
  trend = false;
  high_trend = false;
}
else if ((adx.GetADX()[1]<30) && (adx.GetADX()[0]>30)) {
  trend = true;
  high_trend = false;
}
else if ((adx.GetADX()[0]<45) && (adx.GetADX()[1]>45)) {
  high_trend = false;
  trend = false;
}
else if ((adx.GetADX()[0]>45) && (adx.GetADX()[1]<45)) {
  high_trend = true;
  trend = true;
}
```

The following code shows the buy/sell orders depending of a given indicators that have been generated by the grammatical swarm.

```
if (this.GetOpenPosition() == 0) {
  if (trend && !high_trend) {
   if ( (adx.GetPlusDI()[1] < adx.GetPlusDI()[0]) &&
        (adx.GetMinusDI()[1] > adx.GetMinusDI()[0]) &&
        (adx.GetPlusDI()[0] > adx.GetADX()[0]) &&
        (adx.GetMinusDI()[0] < adx.GetADX()[0] ) &&
        (macd.GetSignalHistogram()[0] > 0)
    )
   this.Buy(OrderType.Market, 1, 0, "Entry Long");
```

Table 2: Neural network testing performance

Performance results	
Main chart: IBEX 30 minute bars [27/12/2012 - 07/03/2014]	
Net P&L	13009,68 €
Gross P&L	15110,00 €
Profit factor	1,712780504
Sharpe ratio	1,744592161
Slippage per side	-2,812612282
Commission per side	5,75 €
Annual ROI	54,58 %
Mathematical expectation	243,7096774
Analyzed sessions	305
Sessions in market	84
Winning sessions	51
Winning sessions profit	31261,70 €
Winning sessions average	612,97 €
Losing sessions	33
Losing sessions profit	-18252,01 €
Losing sessions average	-553,09 €
Worst drawdown	-2735,76 €
Best session	2400,00 €
Worst session	-1762,17 €

```
    if ((adx.GetMinusDI()[1] < adx.GetMinusDI()[0]) &&
        (adx.GetPlusDI()[1] > adx.GetPlusDI()[0]) &&
        (adx.GetMinusDI()[0] < adx.GetADX()[0]) &&
        (adx.GetPlusDI()[0] > adx.GetADX()[0]) &&
        (macd.GetSignalHistogram()[0] < 0)
      )
    this.Sell(OrderType.Market, 1, 0, "Entry Short");
  }
} else if (this.GetOpenPosition() == 1) {
  if (trend)
     this.ExitLong(OrderType.Stop,
     bbands.GetLowerBand()[0], "Exit Long");
  else if (!high_trend)
     this.ExitLong(OrderType.Limit,
     bbands.GetUpperBand()[0], "Exit Long");
}  else if (this.GetOpenPosition() == -1) {
  if (trend)
     this.ExitShort(OrderType.Stop,
     bbands.GetUpperBand()[0], "Exit Short");
  else if (!high_trend)
     this.ExitShort(OrderType.Limit,
     bbands.GetLowerBand()[0], "Exit Short");
}
```

Figures 4, 5 and tables 3, 4 show obtained results in training and testing.

The results reveal that grammatical approach behaves better than the neural network approach when forecasting IBEX indicator. The number of hits (winning sessions) is clearly higher when using grammatical swarm with HydroPSO. In this scenario a hit is considered a winning transaction that has been placed in the market with the estimated price that the ANN has returned. HydroPSO, as well as PSO, improve the performance when training the ANN which has been elected through grammatical swarm. The election of ANN through G.S. ensures the most suitable architecture is chosen in regards of desired outcomes. Furthermore, the use of HydroPSO, helps in the performance when training the ANN. Finding the best

Table 3: Grammatical swarm training performance

Performance results	
Main chart: IBEX 30 minute bars [27/12/2001 - 08/03/2013]	
Net P&L	63069,63 €
Gross P&L	79520,00 €
Profit factor	1,24649621
Sharpe ratio	0,623895837
Slippage per side	-2,069753347
Commission per side	5,75 €
Annual ROI	11,26%
Mathematical expectation	127,8456592
Analyzed sessions	2841
Sessions in market	866
Winning sessions	511
Winning sessions profit	318934,15 €
Winning sessions average	624,14 €
Losing sessions	355
Losing sessions profit	-255864,52 €
Losing sessions average	-720,75 €
Worst drawdown	-12503,25 €
Best session	4630,00 €
Worst session	-4425,75 €

topology to get the most accurate results and optimizing the training process contribute reveal themselves as a reliable combination to optimize the ANN functionality, as shown in the results.

Table 5 show the performance of four algorithms. The first one is the proposed grammatical swarm with a performance of 92.85%, next column shows the Neural Network based algorithm with a performance of 54.58%, then the Golden Cross algorithm (based on a basic stock market rule) with -12.7% and finally the Trend A1 algorithm with 15.2%. The last one is the best automatic system (among 39 systems), please check following web page `www.tradingmotion.com` (see figure 6).

There are a lot of indicators to evaluate a trading system, depending on the behaviour: (From high risk to low risk). `Annual ROI` (The total return divided by the number of years in the period, total return = net profit/loss divided by suggested capital), `Profit factor` (The profit factor is the ratio between profits and losses, and its calculated by dividing the sum of profits over the sum of losses), `Net P&L` (The total profit or loss (P/L) over the analysed period, net of commissions, slippage and license costs), `Worst drawdown` (The worst peak to valley loss of the system, as measured on an end of session basis, with the date of the low point listed), `Sharpe ratio` (Sharpe ratio measures the excess return per unit of deviation. It characterizes how well the return of an asset compensates the investor for the risk taken. When comparing two assets versus a common benchmark, the one with a higher Sharpe ratio provides better return for the same risk. Sortino is a modification of the Sharpe ratio but penalizes only those returns falling below a specified target, or required rate of return, while the Sharpe ratio penalizes both upside and downside volatility equally), etc...

By checking the mentioned parameters we conclude that the proposed grammatical swarm behaves, in general, better than many automatic system described in `tradingmotion.com`

Conclusion and further work

This paper analyzes a few optimization strategies in the natural computation area such as competitive and collaborative models. (ANN, Grammatical Swarm and HydroPSO) These have been described in order to extract some biological ideas and then apply them in computational models. Such bio-inspired models

Table 4: Grammatical swarm testing performance

Performance results	
Main chart: IBEX 30 minute bars [27/12/2012 - 07/03/2014]	
Net P&L	16597,88 €
Gross P&L	18580,00 €
Profit factor	3,063432494
Sharpe ratio	2,772734456
Slippage per side	-2,728527645
Commission per side	5,75 €
Annual ROI	92,85 %
Mathematical expectation	309,6666667
Analyzed sessions	305
Sessions in market	73
Winning sessions	52
Winning sessions profit	24641,71 €
Winning sessions average	473,88 €
Losing sessions	21
Losing sessions profit	-8043,82 €
Losing sessions average	-383,04 €
Worst drawdown	-1075,58 €
Best session	1560,42 €
Worst session	-1180,00 €

have proven to be an effective tool for solving non common problems; As a powerful application, neural networks can take advantage of grammatical swarm optimization models. This paper has introduced an optimized architectural neural network (built up through grammatical swarm techniques) trained by a HydroPSO algorithm. In particular this paper has shown that building an ANN with Grammatical Swarm and then trains it with HydroPSO provides a good theoretical outcome in terms of winning transactions in the stock market. The solution provided is based on the historical data of the last 2 years. Since 09/03/2014 the predictions rates behave extremely well, but of course it is impossible to know how the ANN will work from 2015 in advance. The results are promising and generalization is a solid option since the ANN architecture is optimized through the use of grammatical swarm. That ensures the optimal election of the ANN. When the ANN is not behaving well, the dynamic inherent nature of the proposed system, allows modifying the topology and restart the training process in case of a wrong prediction rates stream. That is why this process can generalize as it adapts when a pre-defined threshold is reached.

Particle Swarm Optimization often fails when searching the global optimal solution when the objective function has a large number of dimensions. The reason of this phenomenon is not just existence of the local optimal solutions but also the degenerative process of the particles velocities (this means that particles stay in a subregion within the search area)[Rapaic et al., 2009]. This is a sub-plane which is defined by a finite number of particle velocities. local optima problem in PSO is object of study. New proposals of modifications on the basic particle driven equation [Parsopoulos et al., 2001; Hendtlass, 2005] try to overcome this. They use a randomized method (e.g. mutation in evolutionary computations) for either to maintain particles velocities or to accelerate them. Although such improvements work well and have ability to avoid falls in the local optima, the problem of early convergence by the degeneracy of some dimensions still exists, even when local optima do not exist. Hence the PSO algorithm does not always work well for the high-dimensional function. That is the reason why this proposal counts on HydroPSO as the optimal trainer for ANNs. The obtained results suggest HydroPSO might be the right procedure [Zambrano-Bigiarini and Rojas, 2013].

HydroPSO allows the modeller to perform a standard modelling work flow including, sensitivity analysis, parameter calibration, and assessment of the calibration results, using a single piece of software. HydroPSO implements several state-of-the-art enhancements and fine-tuning options to the Particle Swarm Optimisation

Table 5: Results using different algorithms, note that the proposed grammatical swarm one obtains the best performance. Data obtained from `www.trading motion.com` in which the best IBEX automatic system is the Trend A1 IBEX 11.

Performance results	**Gram. Swarm**	Neural Net	Gold Cross	Trend A1 Ibex'11
Main chart: IBEX 30 minute bars [27/12/2012 - 07/03/2014]				
Net P&L	16597,88 €	13009,68 €	-55766 €	5149 €
Gross P&L	18580,00 €	15110,00 €	-57766 €	7021 €
Profit factor	3,063432494	1,712780504	0.76	1.12
Sharpe ratio	2,772734456	1,744592161	-2.0072	0.6420
Slippage per side	-2,728527645	-2,812612282	-2.1687	-2.2632
Commission per side 5,75 €				
Annual ROI	92,85 %	54,58 %	-12.7 %	15.2 %
Mathematical expectation	309,6666667	243,7096774	74,3984991	124.23984
Analyzed sessions 305				
Sessions in market	73	84	161	168
Winning sessions	52	51	42	80
Winning sessions profit	24641,71 €	31261,70 €	27541,14 €	57584.92 €
Winning sessions average	473,88 €	612,97 €	665.45 €	720.22 €
Losing sessions	21	33	119	88
Losing sessions profit	-8043,82 €	-18252,01 €	-83307.14 €	-52345.92 €
Losing sessions average	-383,04 €	-553,09 €	-700.06 €	-594.84 €
Worst drawdown	-1075,58 €	-2735,76 €	-64444 €	-5525 €
Best session	1560,42 €	2400,00 €	2362 €	2181 €
Worst session	-1180,00 €	-1762,17 €	-1783 €	-1513 €

(PSO) algorithm to meet specific user needs. HydroPSO easily interfaces the calibration engine to different model codes through simple ASCII files and/or R wrapper functions for exchanging information on the calibration parameters. Then, optimises a user-defined goodness-of-fit measure until a maximum number of iterations or a convergence criterion are met. Finally, advanced plotting functionalities facilitate the interpretation and assessment of the calibration results. The current HydroPSO version allows easy parallelization and works with single-objective functions, with multi-objective functionalities being the subject of ongoing development. Although the application of hydroPSO is mainly focused to hydrological models, flexibility of the package can be implemented in a wider range of models requiring some form of parameter optimisation, such as proposed algorithms in this paper.

Although the proposed ANN outputs are considered as a relevant indicator to obtain buying or selling signals, there is a lot more work to be done as the combination with other known indicators might contribute with an even stronger strategy for succeeding with market operations. Thus, this ANN could be an interesting part of a strategy that combines human trading experience, dynamic ANN output and conventional indicators.

Acknowledgements

The research was supported by the Spanish Research Agency projects TRA2010-15645 and TEC2010-21303-C04-02.

Bibliography

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.

Booth, E., Mount, J., and Viers, J. H. (2006). Hydrologic variability of the cosumnes river floodplain. *San Francisco Estuary and Watershed Science*, 4(2).

Dase, R. K. and Pawar, D. D. (2010). Application of artificial neural network for stock market predictions: A review of literature. *International Journal of Machine Intelligence*, 2(2):14–17.

de Oliveira, F. A., Nobre, C. N., and Zárate, L. E. (2014). Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index - case study of petr4, petrobras, brazil. *Expert Systems with Applications*, 40(18):7596–7606.

Han, S. and Rung-Ching, C. (2007). Using svm with financial statement analysis for prediction of stocks. *Communications of the IIMA*, 7(4):63–72.

Hendtlass, T. (2005). A particle swarm algorithm for high dimensional, multi-optima problem spaces. In *Proceedings of Swarm Intelligence Symposium*, pages 149–154.

Iman, R. L. (2008). *Latin hypercube sampling*. Wiley Online Library.

Kennedy, J. and Ebehart, R. (1995). Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, volume IV, pages 1942–1948.

Kennedy, J., Eberhart, R., and Shi, Y. (2001). *Swarm Intelligence*. Morgan Kauff-man.

Lin, L., Cao, L., Wang, J., and Zhang, C. (2004). *The Applications of Genetic Algorithms in Stock Market Data Mining Optimization*. WIT Press.

O'Neill, M. and Brabazon, A. (2004). Grammatical swarm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 163–174.

O'Neill, M. and Brabazon, A. (2006). Grammatical swarm: The generation of programs by social programming. *Natural Computing*, 5(4):443–462.

Parsopoulos, K., Plagianakos, V. P., and Magoulas, G. D. (2001). Stretching technique for obtaining global minimizers through particle swarm optimization. In *Proceedings of Particle Swarm Optimization Workshop*, pages 22–29.

Penga, Y. and Lua, B.-L. (2013). A hierarchical particle swarm optimizer with latin sampling based memetic algorithm for numerical optimization. *Applied Soft Computing*, 13(5):2823–2836.

Prasad, R. (2004). *Genetic Algorithms: Genesis of Stock Evaluation*. Aston University.

Rapaic, R. M., Kanovic, Z., and Jelicic, Z. D. (2009). A theoretical and empirical analysis of convergence related particle swarm optimization. *WSEAS Transactions on Systems and Control*, 4:541–550.

Sarlin, P. (2014). A weighted som for classifying data with instance-varying importance. *International Journal of Machine Learning and Cybernetics*, 5(1):101–110.

Singh, G., Grandhi, R. V., and Stargel, D. S. (2010). Modified particle swarm optimization for a multimodal mixed-variable laser peening process. *Structural and Multidisciplinary Optimization*, 42(5):769–782.

van Griensven, A., Meixner, T., Grunwald, S., Bishop, T., Diluzio, M., and Srinivasa, R. (2006). A global sensitivity analysis tool for the parameters of multi-variable catchment models. *Journal of Hydrology*, 324:10–23.

Xiaoac, Y., Jin, X., Fengbin, L., and Shouyang, W. (2014). Ensemble anns-pso-ga approach for day-ahead stock e-exchange prices forecasting. *International Journal of Computational Intelligence Systems*, 7(2):272.

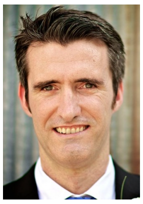
Zambrano-Bigiarini, M. and Rojas, R. (2013). A model-independent particle swarm optimisation software for model calibration. *Environ. Model. Softw.*, 43:5–25.

Zhang, F. and Liao, Z. (2014). Gold price forecasting based on rbf neural network and hybrid fuzzy clustering algorithm. *Lecture Notes in Electrical Engineering*, 241:73–84.

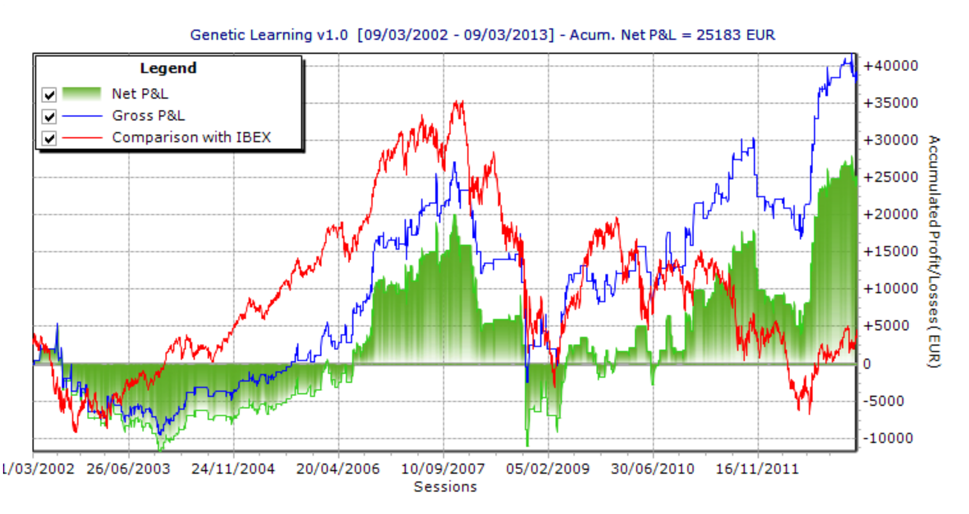
Authors' Information

***Nuria Gómez Blas** - Dept. Organización y Estructura de la Información, Escuela Univesitaria de Informática, Universidad Politécnica de Madrid, Crta. de Valencia km. 7, 28031 Madrid, Spain; e-mail: ngomez@eui.upm.es*
Major Fields of Scientific Research: Bio-inspired Algorithms, Natural Computing

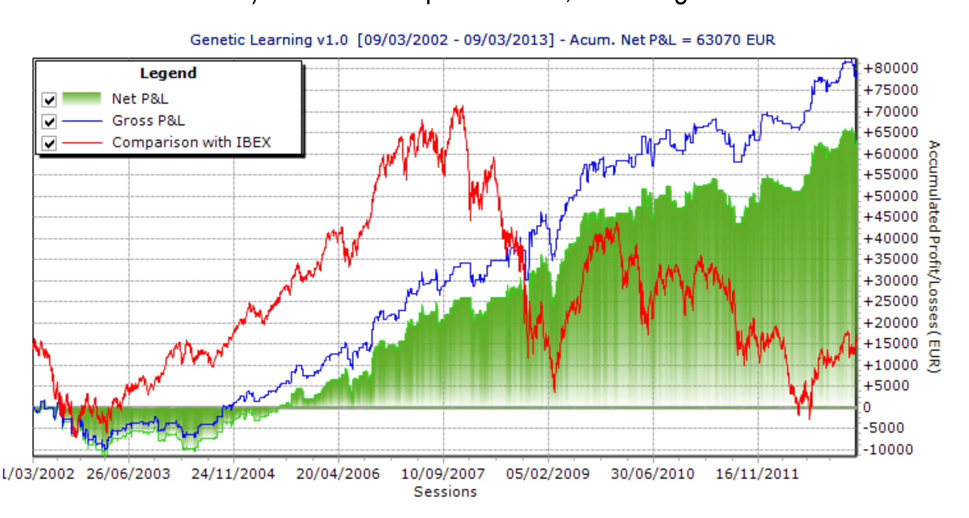
***Luis Fernando de Mingo López** - Dept. Organización y Estructura de la Información, Escuela Univesitaria de Informática, Universidad Politécnica de Madrid, Crta. de Valencia km. 7, 28031 Madrid, Spain; e-mail: lfmingo@eui.upm.es*
Major Fields of Scientific Research: Artificial Intelligence, Social Intelligence



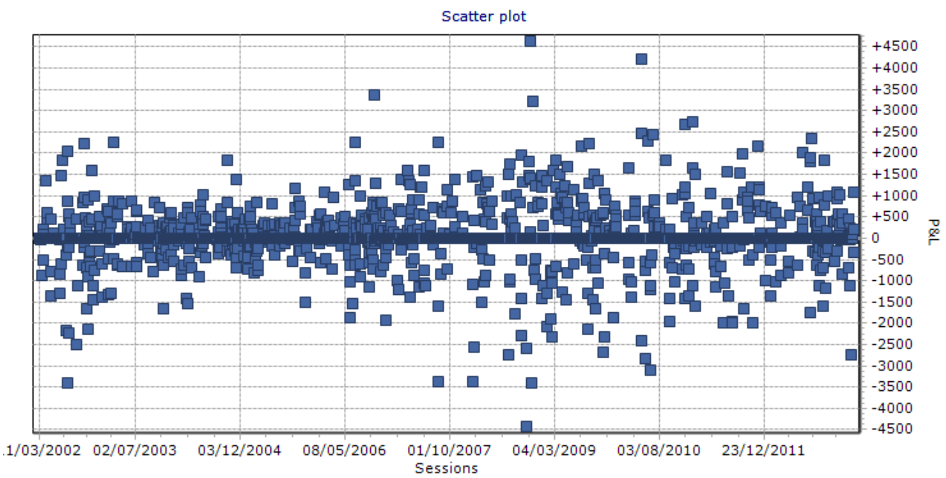
***Alberto Arteta** - Dept. Tecnología Fotónica, ETSI Telecomunicación, Universidad Politécnica de Madrid, Avenida Complutense 30, Ciudad Universitaria, 28040 Madrid, Spain; e-mail: m.muriel@upm.es*
Major Fields of Scientific Research: Theoretical Computer Science, Microwave Photonics



a) Accumulated profit/losses, net and gross

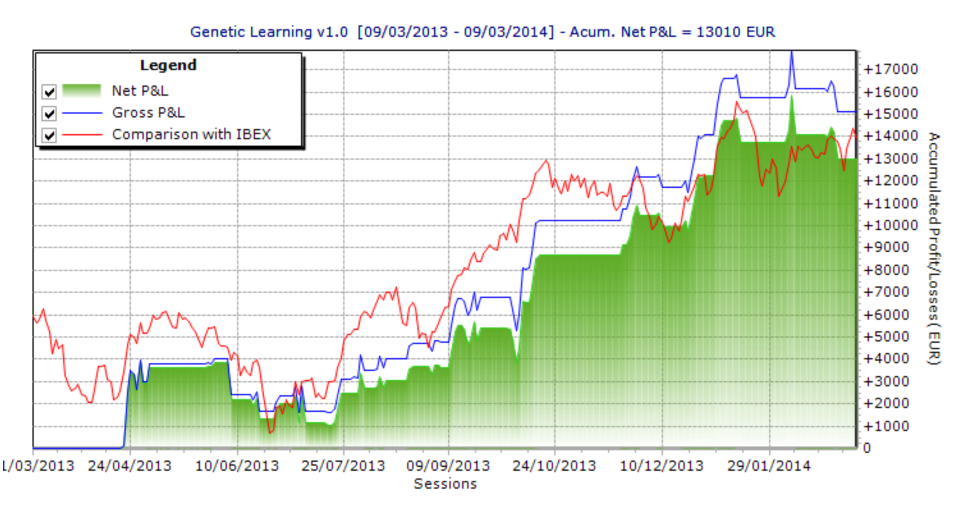


b) Scattered plot of winning/losing sessions

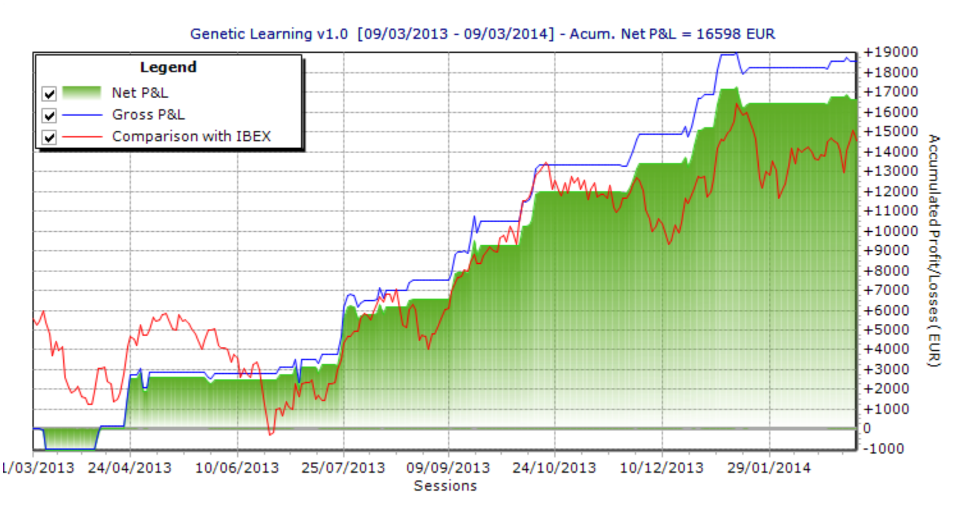


c) Maximum drawdown (worst session losses)

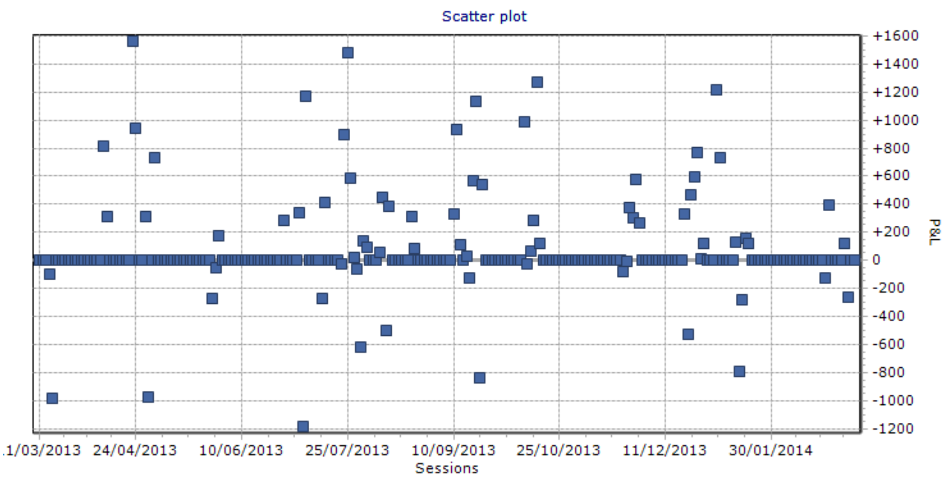
Figure 1: Neural network training performance. The training data set is obtained from `www.tradingmotion.com` and it starts from 2002 till present with a period of 30 minutes.



a) Accumulated profit/losses, net and gross



b) Scattered plot of winning/losing sessions



c) Maximum drawdown (worst session losses)

Figure 2: Neural network testing performance. The system has been cross-validated with real stock market indicators

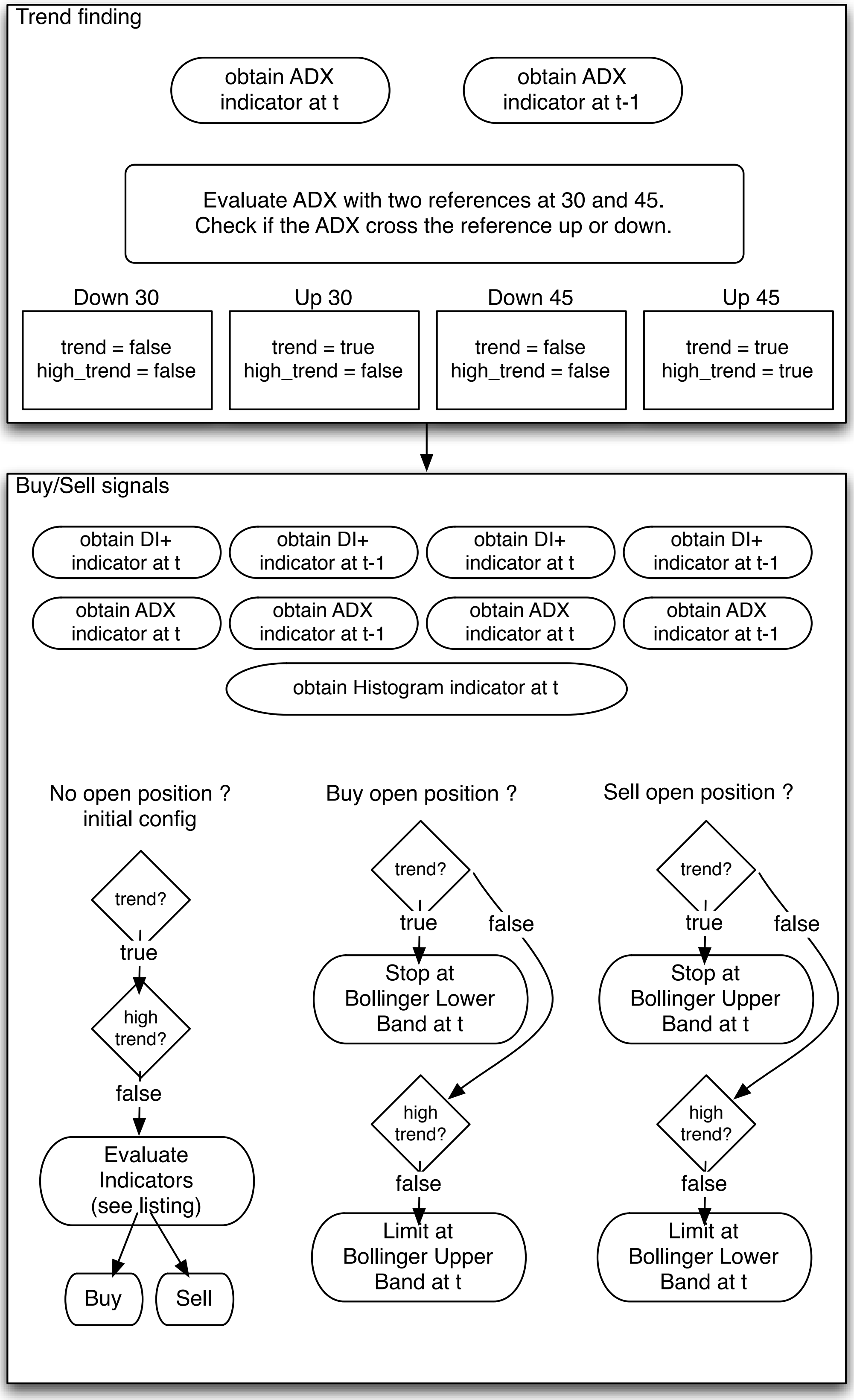
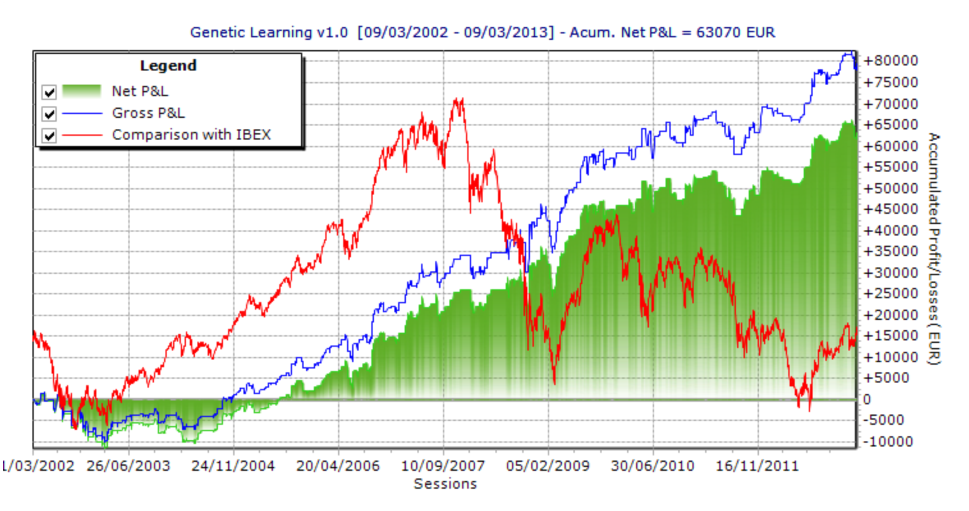
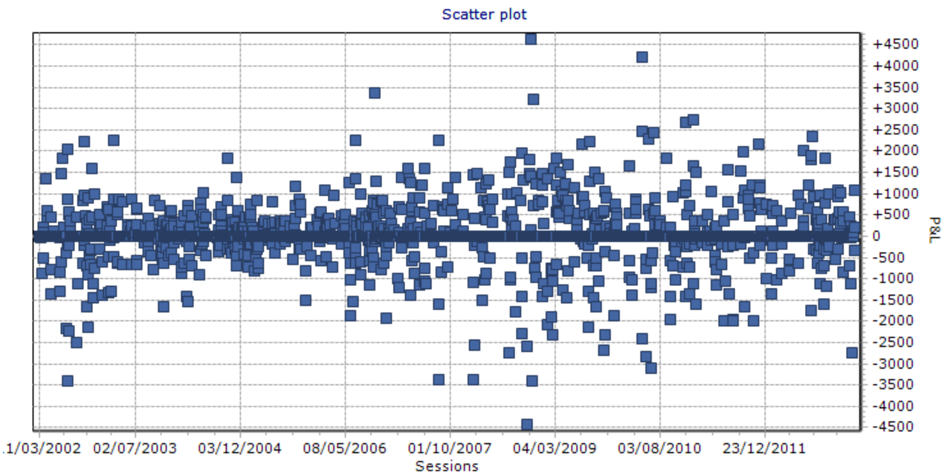


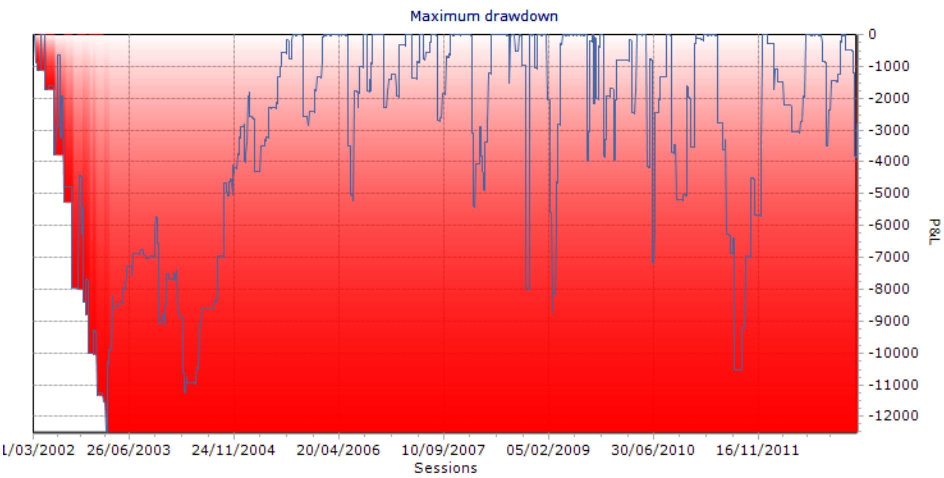
Figure 3: Data flow corresponding to the obtained algorithm using grammatical swarm, see section to find the C++ listing.



a) Accumulated profit/losses, net and gross

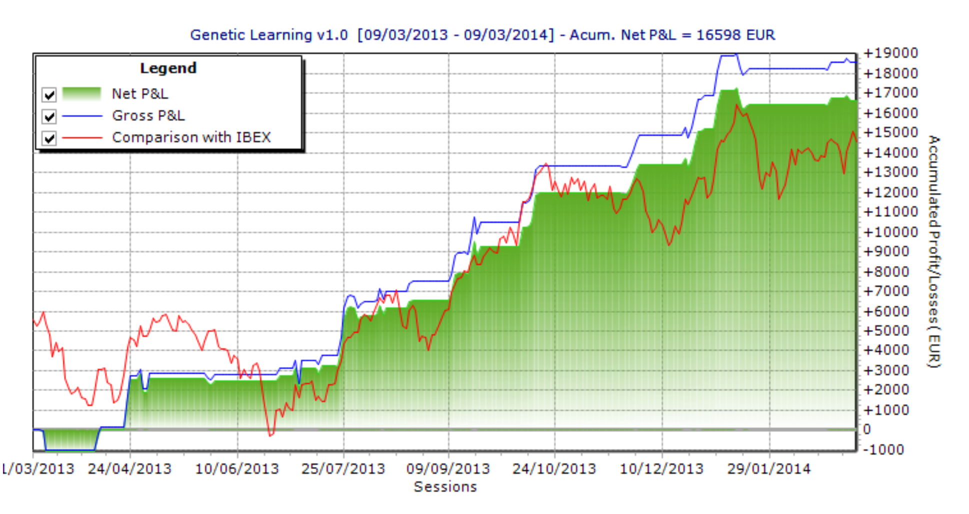


b) Scattered plot of winning/losing sessions

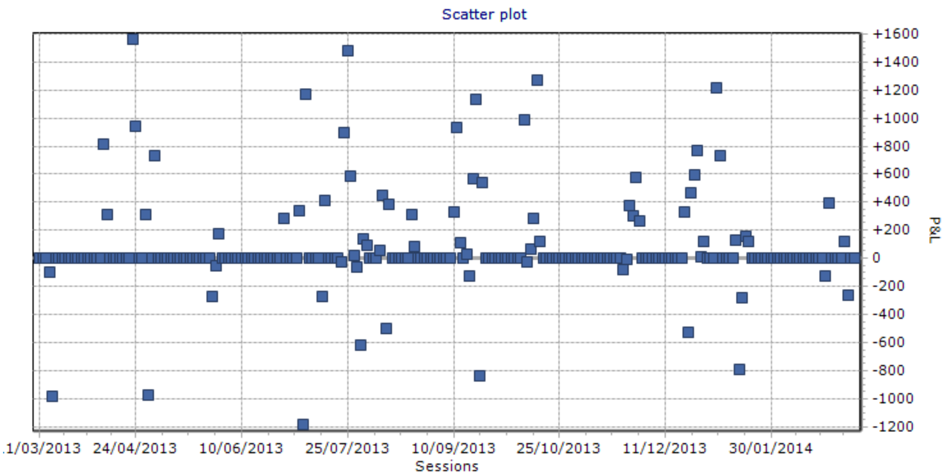


c) Maximum drawdown (worst session losses)

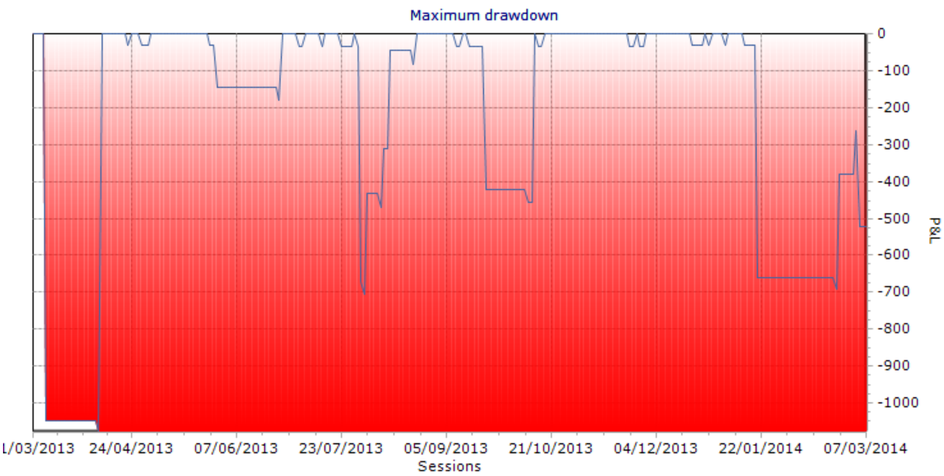
Figure 4: Grammatical swarm training performance. The system has been trained with historical data available at `www.tradingmotion.com` with a period of 30 minutes, corresponding to real stock market indicators



a) Accumulated profit/losses, net and gross



b) Scattered plot of winning/losing sessions



c) Maximum drawdown (worst session losses)

Figure 5: Grammatical swarm testing performance. Data testing is performed with non trained data set corresponding to real values.

Search: ibex Show 25 entries First Previous 1 2 Next Last EUR

★	System	Product	ılı	Developer	Type	Start	Total P/L	Annual ROI	Winning Sessions	Profit Factor	Worst Drawdown	Suggested / Required
☆	Trend A1 Ibex 11'	IX		AutoTradingBot	I	1/2001	258091 €	+63.7%	50.3%	1.65	-9442 €	30000 € 2200 €
☆	Innova 221 Ibex	IX		TodoBolsa	I	1/2006	90985 €	+53.4%	56.2%	1.70	-5288 €	20000 € 2200 €
☆	Hercules_1151 Ibex	IX		TodoBolsa	I	1/2001	214284 €	+52.8%	49.9%	1.44	-9794 €	30000 € 3000 €
☆	LAG_IBEX_3	IX		Luis Antón	I	1/2001	301176 €	+49.5%	48.5%	1.37	-14242 €	45000 € 2200 €
☆	Intr Break 10' Ibex p.p. E	IX		TradingMotion	I	1/2001	198760 €	+49.0%	49.7%	1.38	-9989 €	30000 € 2700 €
☆	Intr Break 10' Ibex p.p. A	IX		TradingMotion	I	1/2001	197545 €	+48.7%	49.6%	1.38	-10598 €	30000 € 2700 €
☆	Cibeles 1051 Ibex	IX		SistemasTrading	I	1/2001	208018 €	+43.9%	46.6%	1.44	-11800 €	35000 € 2700 €
☆	MagicBreak Ibex 15' V2	IX		Joel Galí	I	1/2001	156234 €	+38.5%	49.4%	1.62	-9346 €	30000 € 3300 €
☆	Intr Break 10' Ibex p.p. B	IX		TradingMotion	I	1/2001	204243 €	+37.7%	49.8%	1.40	-15123 €	40000 € 2700 €
☆	NinjaCrea 10' Ibex	IX		CreaTrading	S	4/2006	149685 €	+36.2%	56.6%	1.65	-8929 €	50000 € 11200 €
☆	Intr Tauro 9' Ibex	IX		SistemasTrading	I	1/2001	236609 €	+35.0%	49.0%	1.38	-16876 €	50000 € 2200 €
☆	Velazquez Ibex 17'	IX		Javier González-Barros	S	1/2001	234830 €	+34.7%	58.5%	1.43	-13191 €	50000 € 11200 €
☆	Barrix Ibex 17'	IX		Javier González-Barros	S	1/2001	255280 €	+34.3%	59.0%	1.45	-14352 €	55000 € 11200 €
☆	Univer Ibex 15'	IX		Javier González-Barros	S	1/2001	202590 €	+27.2%	47.5%	1.45	-17426 €	55000 € 11200 €

Figure 6: Ranking of best automatic systems forecasting Spanish stock IBEX index. Information can be check it at `www.tradingmotion.com`. Note that are sorted by annual Return On Investment (ROI).