Tradinnova-LCS: Dynamic stock portfolio decision-making assistance model with genetic based machine learning

Isidoro J. Casanova

***Abstract*— This paper describes a decision system based on rules for the management of a stock portfolio using a mechanism of dynamic learning to select the stocks to be incorporated. This system simulates the intelligent behavior of an investor, carrying out the buying and selling of stocks, such that during each day the best stocks will be selected to be incorporated in the portfolio by reinforcement learning with genetic programming.**

The system has been tested in 3 time periods (1 year, 3 years and 5 years), simulating the purchase/sale of stocks in the Spanish continuous market and the results have been compared with the revaluations obtained by the best investment funds operating in Spain.

I. INTRODUCTION

Investment management consists of three phases: *strategic asset allocation, tactical asset allocation, and stock picking* [2]. *Strategic asset allocation* is a long-term allocation strategy that implies choosing the market you are going to invest in, what kind of assets you are going to buy and the distribution of these in the portfolio in accordance with the investor's objectives. *Tactical asset allocation* consists of regularly adjusting the portfolio, in a systematic or discretionary way, to take advantage of short-term opportunities. *Stock Picking* consists of selecting the best stocks to be incorporated in the portfolio. It is the most time-consuming phase and has greater impact on the return of the portfolio.

Our study focuses on *tactical asset allocation* and *stock selection* of portfolio.

To perform the *tactical asset allocation* an *intelligent system based on rules* is proposed that will dynamically invest in shares for a certain period of time. This system will simulate the behavior of any rational investor, so each day it would look for any investment opportunity, buying or not depending on how much money is available. This investor will also watch the shares bought, selling those shares in which he predicts will not obtain a good profit value, just as he will sell a share when its market price drops below a specified level.

The last phase, *stock selection*, is one of the most important phases, since it must be decided each day what shares should be bought to add to the portfolio. A simple *rule* that could be used to implement this step could be to invest in the stock that has revalued by at least a certain percentage in the last days, although we could use more sophisticated rules using technical analysis, like Relative Strength Index (RSI) or Moving Average, [14].

Isidoro J. Casanova is with the Department of Informatics and Systems, University of Murcia, Campus de Espinardo, 30100 Murcia, Spain (web: http://webs.um.es/isidoroj; email: isidoroj@um.es).

All of these rules, which serve to help us to select our investment, have some input parameters to be applied. Thus if we focus on the first rule, it is necessary to know the minimum value that a stock must have been revalued at to invest in it, the number of days to calculate the revaluation or the type of stock price that is going to be used (opening, closing, high, low) among other possible parameters. These parameters will not be fixed during all the period of the investment; they may change over time as we can find the market with bull, lateral or bear periods.

To implement this last phase we will use *genetic reinforcement learning*, similar to its use in a *Learning Classifier System, Pittsburgh-style*, [19], where for each type of selection rule, we will codify the different input parameters like a chromosome, so as the days pass, the genetic algorithm will evolve and multiple solutions (chromosomes) are combined (crossover and mutation) to form, potentially, better solutions. It is necessary to define the fitness function, that quantifies the optimality of a solution (that is, a chromosome) so that particular chromosome may be ranked against all the other chromosomes. The fitness function values will correspond with the revaluation reached by the different stocks that are selected by each chromosome and will help us to select which are the best rules to have offspring, so that during each evolution, natural selection is applied to determine which solutions (chromosomes) make it to the next evolution.

Finally, each day of the investment period, we will select a single chromosome to form an optimal rule for selecting stocks in the market, which applied in the market would select the best stocks to be passed to the tactical asset allocation phase, that has been implemented as a rules-based intelligent system, managing the real purchase/sale of stocks in the market.

The proposed system will be applied to the Spanish stock market, specifically, the IBEX 35[1], keeping in mind that we invest all the available money in stocks regardless which sector they belong to, and supporting a maximum of 4 % loss per share. We will compare our investment performance with the index itself and with the results obtained by the best equity funds invested in the Spanish continuous market in the time limits of 1, 3 and 5 years.

In Section II, we introduce the concepts of investment port-

[1]The IBEX 35 (an acronym of Iberia Index) is the benchmark stock market index of the Madrid Stock Exchange, Spain's principal stock exchange. It is a market capitalization weighted index comprising the 35 most liquid Spanish stocks traded in the Madrid Stock Exchange General Index

978-1-4244-8126-2/10/$26.00

folio and how artificial intelligence has been introduced to help in the creation of a portfolio. In Section III, we present the system proposed, Tradinnova-LCS, with all the elements that compose it. In Section IV, we show different computational results that illustrate the behavior of Tradinnova-LCS. Finally, we present our conclusions in Section V.

II. Literature review

A. Investment portfolio

The concept behind an investment portfolio is to combine different investment targets to avoid concentrating too much risk in any one target and so disperse overall investment risk. Any combination of two or more securities or assets can be termed an investment portfolio. Over half a century, the Markowitz mean-variance model, [13], which tries to maximize return and minimize risk by carefully choosing different assets, has become a universally understood technique within the investment field. Many asset managers build on the foundation of the Markowitz mean-variance model to construct an Efficient Frontier portfolio.

Efficient-market hypothesis is an idea partly developed in the 1960s by Eugene Fama and defended by Burton G. Malkiel [12] which asserts that financial markets are "informationally efficient", or that prices on traded assets (e.g., stocks, bonds, or property) already reflect all known information, and instantly change to reflect new information. Therefore, according to theory, it is impossible to consistently outperform the market by using any information that the market already knows, except through luck.

Following this last hypothesis are the *passive managers*, who believe that it is impossible to predict which individual holdings or section of the market will perform better than another, therefore their portfolio strategy is determined at the outset of the portfolio and not varied thereafter. Many passive portfolios are index portfolio where the portfolio tries to mirror the market as a whole. Another example of passive management is the "buy and hold" method used by many traditional Unit Investment Trusts where the portfolio is fixed from the outset. Today, there is a plethora of market indexes in the world, and thousands of different index funds tracking many of them.

One of the largest equity mutual funds, the Vanguard 500, is a passive management fund. The two firms with the largest amounts of money under management, Barclays Global Investors and State Street Corp., primarily engage in passive management strategies.

In contrast to the efficient-market hypothesis are the *active managers*, who believe that by selectively buying within a financial market it is possible to outperform the market as a whole. Therefore they employ dynamic portfolio strategies, buying and selling investments with changing market conditions.

Investors or mutual funds that do not aspire to creating a return in excess of a benchmark index will often invest in an index fund that replicates as closely as possible the investment weighting and returns of that index; this is called *passive management*. *Active management* is the opposite of passive management, because in passive management the manager does not seek to outperform the benchmark index.

B. Artificial intelligence and investment portfolio

The world of artificial intelligence applied to stock market investment can be divided into two groups. The first tries to ascertain as accurately as possible the future price of a share and the second endeavors to optimize the mean-variance analysis proposed by Markowitz.

One of the most widely used techniques for predicting the price of a particular stock are neural networks, [21]. There are also genetic algorithms to optimize the parameters of these neural networks or to optimize the technical analysis of one stock [6]. In [4] a system based on rules with fuzzy logic it is proposed to select the stocks and in [1] are utilized, among others, as methods for the classification of stocks the recursive partitioning or probabilistic neural networks. There are also studies on how to predict the behavior of one stock from the news published on the Internet, [15].

With respect to Artificial Intelligence techniques to optimize the mean-variance analysis, we find [10] that deals with fuzzy optimization schemes for managing a portfolio. Kendall and Su [9] use particle swarm optimization to determine the best proportion of risk assets. This method is based on the mean-variance model and uses the Sharpe ratio as fitness function.

These articles may be helpful in investing, although they do not go so far as to perform the buying and selling of the stocks. Once stocks have been selected, the time of purchase and sale is not clearly specified, there are articles in which a notice appears in real-time on what stock or group of stock are recommended, [18], where the stocks are bought and sold annually, [20], in which is estimated if a purchase is going to be long or short term, or where it just does not say anything about what methodology has been applied to estimate when these buying and selling should be made. Also in the buying process could leave a percentage of the available money uninvested, changing this percentage depending if the market rises or falls, [16].

In most of these articles there is a vector which indicates the percentage of investment being made in each stock for each one of the subperiods. For example, in [1] there is a fixed investment horizon of 5 years, with sub-periods of 1 year. This modus operandi has not really considered the normal carrying out of a trade operation done when an investor buys or sells shares. Thus, for example all these articles have a vector which shows the percentage of investment in each stock instead of real orders to buy or sell (market orders/limit orders). Likewise, the commissions of buying and selling are not calculated in a completely real way and there is no daily monitoring of the stocks that are in the portfolio, so these stocks could fall into a downward spiral and no corrective measures would be taken in any of there cases, either because the subperiod is not rebalanced until it is over, or because the technique used to calculate the percentages of investment in each of the shares does not take into account the behavior of the stocks that have already been purchased.

C. Machine learning and genetic reinforcement learning

Machine learning is a scientific discipline that is concerned with the design and development of algorithms that allow computers to change behavior based on data, such as from sensor data or databases. A major focus of machine learning research is to learn to recognize complex patterns and make intelligent decisions based on data automatically.

A learning classifier system, or LCS, is a machine learning system with close links to reinforcement learning and genetic algorithms. According to Robert E. Smith in [7] a "typical description of a LCS will include rules, usually taken from the common {1,0,#} syntax, that are acting as population members in a genetic algorithm. (...) Although the elements discussed above are typical to LCS descriptions, they may not define the LCS approach. In particular, specific implementation details of LCS syntax, and the system components that facilitate rule interactions with the environment, may obscure the essence of the LCS approach. Fundamentally, the LCS is defined by a population of entities that:

- act individually, responding to, and taking actions on, an external environment,
- while evolving as population members, under the action of evolutionary computation.

This definition is independent of syntax or implementation details."

Learning classifier systems can be split into two types depending upon where the genetic algorithm acts. A Pittsburgh-type LCS has a population of separate rule sets, each of which represents a potential problem solution, where the genetic algorithm recombines and reproduces the best of these rule sets. In a Michigan-style LCS there is a population of rules where the genetic algorithm operates at the level of individual rules and the solution is represented by the entire rule population.

In recent years, classifier system techniques have been used in many different fields, including financial market analysis, and have demonstrated excellent performance in all of these areas.

Thus, for example, [3] uses a LCS model to study the foreign exchange market. The empirical results of this research show that classifier systems can classify external information and generate suitable predictions, while evolving appropriate trading rules in response to environmental changes.

Furthermore, other scholars have used classifier systems to analyze the trading of individual stocks using price indicators as inputs and individual stock sell signals as outputs. For instance, [11] uses price and volume indicators including closing prices, 6-day average prices, and the OBV indicator as input factors. On the other hand [17] uses average price and volume as input factors, or in [5] there is an extended classifier system (XCSR) sub-model for each stock, which uses five technical indicators to forecast the future return of one stock in accordance with the investor adjustment cycle. The condition part of this XCSR classifier includes five technical indicators (MA, KD, MACD, RSI, and WMS%R) and the proportional increase or decrease percentage in the five indicators relative to the previous day, and the action part contains the forecast rise or fall; all these articles have obtained significantly better experimental results than both Buy & Hold and random trading strategies.

III. Research Framework

A. Intelligent system for tactical asset allocation

The proposed system for decision making is based on a policy of buying and selling the stocks that make up any stock market over a period of time.

This policy of buying and selling is based on the assumption that if a stock market is quoted from a start date ($date_{start}$) until an end date ($date_{end}$), on each one of those days all their stocks have had an opening price (p_{open}), a maximum price (p_{max}), a minimum price (p_{min}) and a closing price (p_{close}).

If we take a day d between $date_{start}$ and $date_{end}$ in which you have purchased no shares (there is no order of sale or purchase pending), then we will be able to select a set of m stocks (S_b), using technical analysis or any other technique, which would be most recommendable to buy, because it expects them to give a good return.

The technique for selecting stocks should calculate a value for each one of the stocks that make up the market on that day, d, quantifying if it would be advisable to buy the shares. Stocks are ordered from most to least according to this value, and it will have to choose the best set of stocks, defining which minimum value is considered for a stock to belong to this set of the best stocks, with the possibility that on one day no stock is recommended ($m = 0$), or that many are recommended because its analysis has been sufficiently satisfactory for all them.

Once we have this set S_b with the selection of the best stocks for a day d, we might try to buy all or some of these stocks. To simplify the algorithm we will try to buy only one of the selected stocks every day, so that after several days, we could have a portfolio of n stocks (S_a). Thus, to choose the stock to buy we would have two possibilities, to select the best or to select randomly.

Once we have chosen a stock, we will have to give the purchase order for the next day. We will limit the purchase price to any of the prices that has had the stock during that same day ($[p_{min}, p_{max}]$), or choose the opening p_{open} or closing p_{close} price of the next day. If we use a low purchase price, there are fewer possibilities to execute the purchase in the following days, but on the other hand, the stock will be bought more cheaply.

The next day ($d + 1$), one sees if the purchase of the share can be executed, whenever the purchase price of this day is between p_{min} and p_{max}. If this purchase order is not executed in W days, then we would eliminate this purchase order, and would give a new purchase order, selecting a share from the best ones of that day. On this following day, on which we can already hold purchased shares, it is necessary to calculate which of them should be kept in the portfolio and which should be sold, which is why a new analysis is performed to select p stocks (S_c), such that using again any of the previously commented techniques would tell us which stocks should be maintained in the portfolio, since it is expected that they give a good profit value. The technique used must calculate a value for each of the stocks that make up the market on that day, $d+1$, quantifying if it would be advisable to maintain this share in the portfolio.

On this new day we would check if each one of the purchased stocks continues to be among the best that are recommended to maintain in the portfolio (S_c), in which case we would not do anything, otherwise we would allow M days to pass without the stock among the best before giving a sale order. A sale order is given when M days have passed without the stock being recommended to be kept in the portfolio or when the stock has had a higher than a certain percentage $P\%$ loss regarding the price of purchase. A sale order will also have a limited price, like the purchase order. The higher the sale price the more difficult it is to execute the order to sell. If during V days it is not possible to sell at this price, probably because the stock is in a bearish period, then we would lower the price of this sale order.

From this policy of buying and selling, Algorithm 1 is proposed, which we call Tradinnova-LCS, which simulates the intelligent behavior of an investor in a continuous market to form portfolios of n shares. In this algorithm the following functions are used:

- exec_ord_sell_pend(): Executes the pending sale orders that have been introduced in the system.
- exec_ord_buy_pend(): Executes the pending purchase orders.
- drop_price_ord_sell_pend_exec(): Drops the sale price of the sale orders that remain unexecuted for V days.
- delete_ord_buy_pend_exec(): Erases the purchase orders that have not been executed in W days.
- select_best_stocks_buy–LCS(date, rules$_d$): Selects the best stocks that are recommended for purchase on a certain day. This function is explained thoroughly in the following section.
- new_ord_buy(Best(S_b)): Introduces in the system a new purchase order for the best share of S_b
- select_best_stock_hold (date, rule_hold_portf): Selects which shares are better to maintain in the portfolio, because it is expected that they provide a good profit value. The rule that makes this selection "$rule\_hold\_portf$" is passed as an input parameter to the algorithm.
- new_ord_sell(A_i): Introduces in the system a new sale order for the share A_i, if this share surpasses the maximum loss permitted, or if the share is not recommended to be maintained in the portfolio during more than M days.

Algorithm 1 - Tactical asset allocation

```
FUNCTION TradInnova-LCS (date_begin, date_end, n, rule_hold_portf,
money_begin): money_end
begin
    rules_d=generate_random_rules();
    date=date_begin;
    money=money_begin;
    while (date ≤ date_end)
    begin
        money=money+exec_ord_sell_pend();
        money=money-exec_ord_buy_pend();
        if (exists(Pending sell orders during V days))
            then drop_price_ord_sell_pend_exec();
        if (exists(Pending buy orders during W days))
            then delete_ord_buy_pend_exec();
        S_b=select_best_stocks_buy–LCS(date, rules_d);
        if (S_b<>{}) and (money> 0) and (S_a<n)
            then new_ord_buy(Best(S_b));
        S_c=select_best_stock_hold (date, rule_hold_portf);
        for each A_i in S_a
            if (A_i not in S_c during M days) or (loss(A_i)> P)
                then new_ord_sell(A_i);
        date=next_day(date);
    end;
    return money;
end;
```

B. Stock picking based on the operation of a Learning Classifier System

For every day, d, along the investment period it is necessary to look for which are the best shares, S_b, to be able to introduce them in the decision support system dedicated to the tactical asset allocation commented in the previous paragraph, and this decides how it is going to invest in them.

To select the best shares of one day, d, it is necessary to choose the best rule of selection ($rule\_for\_buying$) from the existing set of rules ($rules_d$) in the system for this day. As the days pass, this group of rules evolves, adapting itself to the new conditions that the market presents. We describe below the functions that are used in the implementation of Algorithm 2, in charge of stock picking:

- evolve_parameters(rule, best_rules($rules_d$)): Evolves the different input parameters of a rule by a genetic algorithm that using mutations and crossings of a rule with others with greater fitness function value will get better generations of rules.
- execute_selection_stocks(d, rule): Applies a rule in the market, carrying out a selection of stocks according to the input parameters that include the rule ($rule$) and a date d.
- rule.update_portfolio(d, stocks): Each rule will have a portfolio associated with the shares ($stocks$) that have been recommended each day. This function is responsible for updating this portfolio, adding the stocks selected to him.
- rule.update_fitness(d, rule.portfolio): Every day the fitness of every rule will be updated, which corresponds to the revaluation that the portfolio ($rule.portfolio$) associated with every rule attains.
- best_rule($rules_d$): It selects the best rule from the set

of existing rules in the system in accordance with the fitness function.

Algorithm 2 - Stock picking

```
FUNCTION select_best_stocks_buy–LCS (d, rules_d): stocks
begin
    for each rule in rules_d
    begin
        rule = evolve_parameters(rule, best_rules(rules_d));
        stocks = execute_selection_stocks(d, rule);
        rule.update_portfolio(d, stocks)
        rule.update_fitness(d, rule.portfolio);
    end
    rule_for_buying = best_rule((rules_d));
    S_b = execute_selection_stocks(d, rule_for_buying);
    return S_b;
end;
```

As mentioned, to find the best shares S_b that are recommended to buy on one day, d, we need to choose the best rule of selection of shares inside a set of rules of selection. To find out which is the best rule we will build on the functioning of the genetic reinforcement learning used in a Learning Classifier System, where traditionally the core is a set of rules (called the population of classifiers). The desired outcome of running the LCS algorithm in the "Pittsburgh-style" is to choose classifiers to individually model an intelligent decision maker (each rule-set is a potential solution). To obtain that end, LCSs employ two biological metaphors; evolution and learning, where learning guides the evolutionary component to move toward a better set of rules. These concepts are respectively embodied by two mechanisms: the genetic algorithm, and a learning mechanism appropriate for the given problem. An early advantage of the Pitt-approach came from its credit assignment scheme, where reward is assigned to entire rule-sets, as opposed to individual rules.

1) Rule representation: There are different techniques for representing the knowledge acquired in a knowledge base. In a LCS, the rules are typically represented in the form of "IF condition THEN action", but we are going to use another alternative representation, since we reach the maximum knowledge when we find the best input parameters to apply in a rule of selection of shares.

We will use 5 types of rules (RULE-TYPE) to select stocks. These rules are based on intuition and technical analysis of the shares:

1) Revaluation Period (RP): Revaluation which a stock has had in a given period of time.
2) Average Revaluation Period (ARP): Average revaluation that a stock has had in a given period of time.
3) Relative Strength Index (RSI): Relative Strength Index of a stock in a given period of time.
4) Moving Average (MA): Calculates the revaluation that a stock reaches with respect to the average price value in a given period of time.
5) Double Moving Average (DMA): Known also as double crossover method, uses a combination of long-term and short-term moving averages. When the shorter moving average rises from below to above the longer moving average, a buy signal is issued. The strength of this signal will depend of the current share price. If the current value is higher than the shorter moving average, it will be worth 0.10, and if it is also higher than the longer moving average, it will be 0.25. The initial price of the share at the opening of each period will also be considered, so if along with all the above the opening price of the share at the start of the short period is lower than the shorter moving average, then the DMA will be 0.5 and, if the price is also lower than the longer moving average at the start of the long period then DMA will be 1. If neither case occurs, DMA will be 0.

To execute the rules on the market, the following input parameters (genes) are needed:

- Period (DAYS): Number of days on which the calculation is going to be realized.
- Minimal value selection (MIN): Minimal value that must be reached for a share to be selected.
- Variation best stock selection (VAR): Percentage of variation allowed with respect to the selection value of the best stock for selecting the rest of the shares. Thus for example, if the selection value of the best stock is 4 (revaluation in a given period), and VAR = 20%, then the shares selected will be these that attain at least a 4% - 20% = 3.2% revaluation on that day.
- Type Price (PRICE): We will select the different types of prices that a stock can reach in a day to perform the calculations (open, close, maximum, average and minimum).

Therefore a rule (chromosome) has the following form (RULE-TYPE, DAYS, MIN, VAR, PRICE) and the following information associated (PORTFOLIO, FITNESS).

For each rule type (RULE-TYPE) the system will have multitude of variations of its different parameters (DAYS, MIN, VAR, PRICE) to apply the reinforcement learning.

To represent each one of the genes of the chromosomes, i.e., the parameters of each rule, we have used integer or real values, instead of realizing a binary encoding, considering the rank of values that can be reached.

The population of classifiers is randomly initialized and the fitness function value (FITNESS) is set to the initial value of 0, before the algorithm is run.

Possible values that the input parameters can have when they are created randomly at the beginning are:

- DAYS: 1, 2, 3, 4, 5, 10, 15, 20, 30, 45, 60 and 180 days.
- MIN: The minimum value depends on RULE-TYPE
  - RP: 0.5, 1, 1.5, 2, 2.5, 3, 4, 5.5, 7 and 9%.
  - ARP: 0.01, 0.02, 0.03, 0.04, 0.06, 0.07, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 and 0.9%.
  - RSI: 10, 25, 35, 45, 50, 55, 60, 70, 75, 80, 85, 90, 95 and 99.
  - MA: 3, 4, 5, 6, 8, 12, 15 and 20%.
  - DMA: 0.10, 0.25, 0.50 and 1.

- VAR: 0, 2, 8, 20, 40, 100 and 1000%.
- PRICE: close = 0, average = 1, maximum = 2, open= 3 and close = 4.

As the DMA requires two periods and DAYS only contains one number of days, DAYS has been codified as x =(c,l), so that depending on the value x which takes DAYS, we will carry out the following correspondence to a short period c and to another long l: 1=(2,5), 2=(5,10), 3=(5,15), 4=(10,15), 5=(10,30), (5,10]=(10,40), (10,15]=(20,60), (15,20]=(20,70), (20,30]=(25,75), (30, 45]=(25,100), (45,60]=(30,90) and (60,180]=(30,120).

It is necessary to consider that unlike a Learning Classifier System, the rules defined in this system do not have a condition part, which is why a "match set" of rules whose condition matches the input string with the current state of the environment at each position does not exist. However it is assumed that any one of the rules pertaining to the population of classifiers can be applied to the system at any time.

2) Reward allocation: Classifier systems are rule-based systems. Each classifier has its fitness strength (FITNESS) that shows its usefulness in the current system. After a classifier has been chosen, it receives a reward in the case of successful prediction; otherwise, it pays compensation for an incorrect prediction.

In this research, each rule or classifier will have a portfolio of shares (PORTFOLIO) associated, such that every day all the existing rules in the system will be selected and their application in the market will be simulated. When running a rule with certain parameters, this rule returns a selection of the best stocks, which are incorporated into the portfolio of shares (PORTFOLIO) associated with the rule in question, keeping the date and the price at which a share has been incorporated in the portfolio.

The fitness each rule will have will be the total average profit value that each stock of the portfolio reaches since its incorporation. So, if all the stocks that are held in the portfolio associated with a rule are revalued over time, then this rule will reach a greater fitness that other rule than that of all its stocks with losses in that same period.

This fitness, or total average revaluation that the stocks of the portfolio associated to a rule reach, will be updated every day of the investment period and will be initialized to 0 (just like the portfolio associated) when a rule changes, either by mutation or by crossing

3) The genetic algorithm: The genetic algorithm is a computational search technique which manipulates (evolves) a population of individuals (rules) each representing a potential solution (or piece of a solution) to a given problem.

Additionally, the genetic algorithm needs to be initialized with a population of rules generated randomly to broadly cover the range of possible solutions (the search space), as explained above.

The algorithm implemented has the following parameters:

- Maximum Population Size (MPS): specifies the maximal number of rules in the population. A larger population size means more potential solutions to choose from and more genetic diversity, but it also means more work to evolve that population. Ultimately, it is necessary to settle on a value that balances the need for a nice selection of potential solutions against how much time one is willing to spend waiting for the population to evolve.
- Mutate Rate (MR): specifies what percentage of MPS will mutate.
- Mutate Probability (MP): specifies what probability each one of the different input parameters of a rule has of mutating.
- Crossover Rate (CR): specifies what percentage of MPS will cross, using a Uniform Crossover.
- Crossover Probability (CP): specifies what probability two genes have of crossing calculating the average value, otherwise the gene with more fitness would dominate.

The following steps will guide the reader through a single iteration of the genetic algorithm:

1. Evaluate the fitness of all rules in the current population. The fitness of a rule corresponds to the average revaluation that its associated portfolio has. It is calculated daily.
2. Descendant arrangement of the rules according to their fitness, such that a percentage of the first rules will correspond with elitist population, and therefore will be used like "parent" rules from the population. Using the parameters of the genetic algorithm, this ratio would calculate as 100% - (MR + CR).
3. Crossover and/or mutate "parent" rules to form "offspring" rules. A (MR + CR)% of new individuals would be generated in this way. A uniform crossover will be used, so it will be possible to combine any of the genes of the two progenitors of the same type. In this combination, a gene would be calculated as the arithmetic mean of the values that have the genes of the parents or it would be a copy of the gene whose parent has a higher fitness
4. Add "offspring" rules to the next generation and remove enough rules from the next generation (with probability of being removed inversely proportional to fitness) to restore the number of rules to (MPS), by carrying out a partial change of the population.

IV. Experiments and Results

We are going to verify the operation of the system in 3 periods of time: 5 years (2005-2009), 3 years (2007-2009) and 1 year (2009). Before each test, a learning of the system will be carried out using daily dates during a year prior to the start of each period of simulation, so learning will be performed during years 2004, 2006 and 2008. The rule sets established during the training period are used during the testing period as initial rule sets.

Before running the training algorithm, 200 rules for each type of rule are randomly created, such that for each day, the operation of 1000 possible combinations of rules will be verified.

The market in which we will operate is the Spanish stock market, but restricted to the shares that made up the IBEX35 in 2009. The historic prices have been corrected from dividends, splits and increases in capital. Short selling is not allowed. The cost of each trade has been taken into consideration, so we assume that the financial intermediary charges a fee of 0.2% and we are going to consider the transaction fees published by the Madrid market.

The genetic algorithm has been configured with the following parameters: MR = 20%, MP = 60%, CR = 30% and CP = 70%. The mutation probability has been defined as high because we want the genetic algorithm to adapt quickly to the changes that the market has, there being no unique optimum rule for the whole investment period.

The portfolio will be formed by 14 shares maximum and the rule which is responsible for defining whether a stock should remain in the portfolio ($rule\_hold\_portf$) has been defined such that the Relative Strength Index of the shares for 28 days must be worth at least 45 for a sale order not to be issued if the stock has remained in the portfolio at least 14 days.

Table I gives the results for each of the three periods, changing in each the maximum loss allowed (P=4% or P=2%) before giving an immediate sale order and these are compared with the revaluation of IBEX35 at that time. The variance of the daily revaluation throughout each of the periods is also given.

We found that in general the system outperforms the IBEX35 revaluation in all the periods and also achieves a lower variance than produced by the IBEX35 and therefore offers less risk.

Variance in general decreases on restricting the maximum loss allowed (from 4% to 2%). This is because the intelligent system for tactical asset allocation controls the behavior of the shares, immediately selling those shares that surpass the established loss allowed ($P\%$).

It is also observed that greater restrictions of loss allowed usually lead to lower profitability.

TABLE I

RESULT OF THE SIMULATION (P=4% AND P=2%) VS. IBEX35 REVALUATION

System	Period	Result	Variance
Tradinnova-LCS (-4%)	1-2-2009/12-30-2009	33.70%	1.78
Tradinnova-LCS (-2%)	1-2-2009/12-30-2009	30.35%	1.56
IBEX35	1-2-2009/12-30-2009	26.21%	2.45
Tradinnova-LCS (-4%)	1-2-2007/12-30-2009	2.37%	1.57
Tradinnova-LCS (-2%)	1-2-2007/12-30-2009	-6.16%	1.33
IBEX35	1-2-2007/12-30-2009	-16.92%	3.33
Tradinnova-LCS (-4%)	1-3-2005/12-30-2009	103.53%	1.21
Tradinnova-LCS (-2%)	1-3-2005/12-30-2009	108.46%	1.06
IBEX35	1-3-2005/12-30-2009	30.85%	2.23

Figure 1 graphically shows a comparison of daily results of the intelligent system (with a 4% maximum loss allowed) with the behavior of the IBEX35 index in the period 2005-2009, with an initial investment of 100,000 Euros.

Fig. 1. Daily evolution of the investment ($P = 4\%$) and the IBEX35

To evaluate the importance of the results obtained we will consider only those results obtained with a maximum loss of 4%, so we compare them with the results of a report elaborated by INVERCO (Spanish Association of Investment and Pension Funds). Table II shows the ranking (R) by annual equivalent return (APR) in periods of 1, 3 and 5 years of each one of the Spanish equity funds, until 31 December 2009.

TABLE II

RANKING OF FUNDS FROM SPANISH EQUITY INVESTMENT([8])

	2009		2007-2009		2005-2009	
Equity funds & Simulation	APR	R	APR	R	APR	R
Foncaixa Bolsa España 150	51.9	1	−8.9	79	–	–
BBVA Bolsa Ibex Quant	48.7	2	−13.0	86	–	–
Bankinter Bolsa España 2	34.5	32	4.4	1	11.7	1
CC Borsa 11	19.1	88	2.4	2	4.1	72
Venture Bol. Española	34.9	24	0.0	6	10.6	2
TRADINNOVA-LCS (P=4%)	33.7	39	0.8	3	15.3	1

In the INVERCO report we can see that the most profitable fund in 2009 was the *Foncaixa Bolsa España 150*, with 51.94% revaluation, although this fund was ranked in position 79 by return of -8.9% APR that it obtained in the 3 previous years. Our system obtains in 2009 a yield of 33.7%, so if we could participate in this ranking we would be included in position 39 by yield at one year.

At three years the stock market crisis continues nevertheless to cause losses, since almost all the funds register red numbers, except the first funds, like *Bankinter Bolsa España 2*, that with a 4.41% APR would remain first in this ranking of the best investment funds in Spain for 3 years. Our system achieves a revaluation of 0.8% APR in this period, so we would be third in the ranking for profitability for 3 years.

With a horizon of five years, the situation is different: some funds, such as the mentioned *Bankinter Bolsa España 2* (11.7% APR) or *Venture Bol. Española* (10.6% APR) obtain notable performances, although our system outperforms all of these funds with a profit value of 15.3% APR.

V. Conclusions

This article has proposed an intelligent system that solves investment in shares forming a portfolio quite successfully. This system has two main parts: the first is responsible for buying and selling shares, managing a portfolio and monitoring the purchased shares, and the second is responsible for selecting which are the best shares to incorporate into the portfolio.

The part entrusted to the tactical asset allocation, corresponds to a decision system based on rules and the part entrusted to selecting shares has been implemented by reinforcement learning using genetic programming. This last part is inspired by the operation that has a Learning Classifier System, Pittsburgh-style.

The proposed system solves the process of investment in shares in an integral way, since usually all the research papers focus on stock selection and leave out portfolio management, not taking into account the way in which an investor operates normally when he carries out the purchase or sale of shares. Moreover, the implementation of stock picking is also novel, inspired by a Learning Classifier System, but without fully implementing it as such.

In the results, the revaluation of the reference index is surpassed (IBEX35) in all the periods and we can even place the intelligent system in the first positions of the ranking by profit value if it is compared with commercial investment funds that invest in Spanish equities.

Acknowledgment

Supported by the project TIN2008-06872-C04-03 of the MICINN of Spain and European Fund for Regional Development.

References

[1] G. Albanis and R. Batchelor, "Combining Heterogenous Classifiers for Stock Selection," *International Journal of Intelligent Systems in Accounting and Finance Management*, vol. 15(1-2), pp. 1–21, 2007.

[2] N. Amenc and V. Le Sourd. *Portfolio Theory and Performance Analysis*, Wiley Finance, 2003.

[3] L. Beltrametti, R. Fiorentini, L. Marengo and R. Tamborini, "A learning-to-forecast experiment on the foreign exchange market with a classifier system," *Journal of Economic Dynamics and Control*, vol 21(8-9), pp. 1543-1575, 1997.

[4] M.-C. Chan, C.-C.. Wong, W.F. Tse, B. K.-S. Cheung and G. Y.-N. Tang, "Artificial Intelligence in Portfolio Management," *Lecture Notes in Computer Science*, vol. 2412, pp. 159–166, 2002.

[5] M.-C. Chen, C.-L. Lin and A.-P. Chen. "Constructing a dynamic stock portfolio decision-making assistance model: using the Taiwan 50 Index constituents as an example," *Soft Computing*, vol. 11(12), pp. 1149–1156, 2007.

[6] A.E. Drake and R.E. Marks. Genetic Algorithms in Economics and Finance: Forecasting Stock Market Prices and Foreign Exchange - A Review. In S-H. Chen (ed.), Genetic algorithms and genetic programming in comp. finance. Springer. 2002.

[7] J.H. Holland, L.B. Booker, M. Colombetti, M. Dorigo, D.E. Goldberg, S. Forrest, R.L. Riol, R.E. Smith, P.L. Lanzi, W. Stolzmann and S.W. Wilson, "What Is a Learning Classifier System?," *Lecture Notes In Computer Science (Learning Classifier Systems)*, vol. 1813, pp. 3–32, 2000.

[8] Inverco: Asociación de Instituciones de Inversión Colectiva y Fondos de Pensiones, Ranking Fondos Inversión a 31-12-2009, http://www.inverco.es/documentos/estadisticas/fondos_inversion/0912_Diciembre2009/0912_24-FIM-RVNnal.pdf

[9] G. Kendall and Y. Su, "A particle swarm optimization approach in the construction of optimal risky portfolios," *Proc. 23rd IASTED International Multi-conference Artificial Intelligence and Applications*, Innsbruck, Austria, Feb. 2005, pp. 140-145.

[10] T. León, V. Liern and E. Vercher, "Viability of infeasible portfolio selection problems: A fuzzy approach," *European Journal of Operational Research*, vol. 139(1), pp. 178–189, 2002.

[11] P.-Y. Liao and J.-S. Chen, "Dynamic trading strategy learning model using learning classifier systems," *Proc. Congress on Evolutionary Computation*, Seoul, South Korea, May 2001, vol. 2, pp. 783-789.

[12] B. Malkiel, *A Random Walk Down Wall Street*, W. W. Norton & Company, 1973

[13] H. Markowitz, "Portfolio selection," *Journal of Finance*, vol. 7(1), pp. 77–91, 1952.

[14] J.J. Murphy, *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*, New York Institute of Finance. Prentice Hall Press, 1999.

[15] M.A. Mittermayer. "Forecasting Intraday Stock Price Trends with Text Mining Techniques," *Proc. of the 37th Annual Hawaii International Conference on System Sciences*, Big Island, Hawaii, USA, January 2004, track 3, vol. 3, pp. 10.

[16] A. F. Perold and W. F. Sharpe, "Dynamic Strategies for Asset Allocation," *Financial Analysts Journal*, vol. 51(1), pp. 149–160, 1995.

[17] S. Schulenburg and P. Ross, "Explorations in LCS Models of Stock Trading," *Lecture Notes in Computer Science*, vol. 2321, pp. 309–331, 2002.

[18] C.-C. Tseng and P. J. Gmytrasiewicz, "Real Time Decision Support System for Portfolio Management," *Proc. 35th Annual Hawaii International Conference on System Sciences - HICSS'02*, vol. 3, Big Island, Hawai, Jan. 2002, pp. 79.

[19] R. J. Urbanowicz and J. H. Moore, "Learning Classifier Systems: A Complete Introduction, Review, and Roadmap," *Journal of Artificial Evolution and Applications*, vol. 2009, Article No. 1, pp. 1–25, 2009.

[20] M. R. Zargham and M. R. Sayeh, "A Web-based Information System for Stock Selection and Evaluation," *Proc. International Conference on Advance Issues of E-Commerce and Web-Based Information Systems*, Santa Clara, CA, USA, Aug. 1999, pp. 81–83.

[21] M. Zekic, "Neural Network Applications in Stock Market - A Methodology Analysis," *Proc. 9th International Conference on Information and Intelligent Systems*, 1998, pp. 255–263.