Computational Intelligence for Evolving Trading Rules

Adam Ghandar[1], Zbigniew Michalewicz[2], Martin Schmidt[3], Thuy-Duong Tô[4],
Ralf Zurbruegg[5]

Abstract— **This paper describes an adaptive computational intelligence system for learning trading rules. The trading rules are represented using a fuzzy logic rule base, and using an artificial evolutionary process the system learns to form rules that can perform well in dynamic market conditions. A comprehensive analysis of the results of applying the system for portfolio construction using portfolio evaluation tools widely accepted by both the financial industry and academia is provided.**

I. Introduction

This paper describes a computational intelligence system for learning trading rules and provides a comparison of the relative performance of a portfolio managed by the system on companies that were listed as part of the MSCI Europe Index from 1990 to 2005. Using only price and volume data the system determines rules to buy and sell stocks on a regular basis. Rather than fixing these rules throughout the sample period, the rules adapt to changing market conditions, leading to an evolving rule-base that changes with time.

This paper contributes to the extant computing and finance literature in several ways. From a financial perspective, the use of learning rules to build and manage a portfolio of assets that are chosen based on fuzzy logic trading rule bases is not well documented and comprehensively examined. Emphasis is usually placed on developing buy/sell trading rules for individual stocks or indices, not a whole portfolio of stocks which then need to be managed over time. This study addresses this issue. From a computing angle, an integrated process to stock selection and portfolio management allows for a search for the best fuzzy trading rules us-

ing an evolutionary algorithm, as opposed to standard genetic algorithms, creating a highly adaptive and dynamic rule base system. The evaluation of the system is then tested using a comprehensive set of financial performance tools, including stochastic dominance tests that cater for non-normal distributions in the returns series.

The performance of the portfolio managed by the system is also compared to several other portfolios, including a portfolio that mimics a passive index fund, a portfolio that follows a price momentum strategy and a Jensen's alpha portfolio, where stocks are picked based on their alpha ranking using the single-factor Capital Asset Pricing Model (CAPM). To compare performances, a comprehensive set of portfolio evaluation tools are applied to give a detailed assessment of the ability of the system's performance.

The paper is organized as follows: Section II provides background information and a literature review. Section III explains the approach. Section IV describes the setup used in producing the results given in section V. Section VI concludes the paper.

II. Background

There are many papers which describe various applications of nature-inspired algorithms to financial modeling; in this section we survey some of this work.

One of the possible financial applications has been in the area of developing trading rules to signal when investors should buy or sell various financial instruments. Research in this area has received greater attention over recent years as an appreciation for the ease by which computational algorithms can develop and evolve complex trading strategies are further realized. Research such as described in [20] and [1] highlight the possibilities for evolutionary computation to provide trading strategies, based on pattern recognition, to profit from equity market trading. Published research in academic finance journals primarily focuses on examining how well genetic algorithms can develop specific trading rules using historical prices to test, ex post, their profitability.

This type of research is also directly related to the study of market efficiency. In an efficient capital mar-

[1] School of Computer Science, University of Adelaide, Adelaide, SA 5005, Australia, *adam.ghandar@adelaide.edu.au*

[2] School of Computer Science, University of Adelaide, Adelaide, SA 5005, Australia; also at Institute of Computer Science, Polish Academy of Sciences, ul. Ordona 21, 01-237 Warsaw, Poland, and Polish-Japanese Institute of Information Technology, ul. Koszykowa 86, 02-008 Warsaw, Poland, *zbyszek@cs.adelaide.edu.au*

[3] SolveIT Software Pty Ltd, Level 12, 90 Kings William, Adelaide 5000, Australia, *martin.schmidt@solveitsoftware.com*

[4] School of Commerce, University of Adelaide, Adelaide, SA 5005, Australia, *thuy.to@adelaide.edu.au*

[5] School of Commerce, University of Adelaide, Adelaide, SA 5005, Australia, *ralf.zurbrugg@adelaide.edu.au*

ket it would not be possible for traders to make a profit from past data as all relevant information for pricing a security today would be incorporated in today's price. Therefore, many finance papers (see [16], [21], and [12] for example) inter-relate the issue of market efficiency with the ability for genetic algorithms to literally "beat the market". Results are somewhat mixed. Although there is general consensus that financial markets do sometimes exhibit periods where certain trading rules work (see [8]), it is hard to find clear evidence that a single trading rule can function over an extended period of time. This is probably due to the fact that financial markets are ever-evolving, and in fact given the number of technical analysts that are employed in all the major financial trading institutions, when a trading rule is found to work it would not take long before it is exploited until it no longer yields a significant profit. It is therefore possibly more interesting to see if trading rules can be constructed that also continually evolve as the markets change. An adaptive trading strategy seems to be more promising than static approaches.

As well as genetic algorithms other nature-inspired search techniques have been applied to financial problems. Artificial neural networks have attracted a lot of interest over the past decade. A selection includes: [27] presents an index forecasting approach; [13] applies an ANN to currency exchange rate prediction by anticipating the direction of price change using signal processing methods for series with high noise and small sample sizes; [4] describes a neural evolutionary approach to find models of correlation between financial derivatives; [3] discusses assessing credit risk and predicting using ANNs; and [2] discusses a neural network for option pricing. More recently, numerous applications of evolutionary computation have been published: [22] describes a dynamic asset allocation system in which a model optimized using evolutionary computation determines optimal portfolio weights given trade recommendations; a genetic programming approach for combining trading rules in autonomous agents so that the rules compliment each other is given in [25]; [26] presents a linear genetic programming system for trading that uses intraday data; grammatical evolution for evolving human readable trading rules is extensively discussed in [7]; finally an application of genetic programming for discovering trading rules that are applicable in the short term is given in [23]. A significant benefit of genetic programming in expressing trading rules is the grammatical structure of phenotypes which enables expression of rules combining several input in a form that is able to be readily understood and applied.

In this paper we describe a system that forms trading rules using price and volume history of stock prices and adapts the rules to changing market conditions. The approach is essentially referred to as technical analysis. Rather than using fundamental accounting and macroeconomic data to determine which stocks to buy or sell, trading rules are developed solely applying historical data series from the previous trades of these stocks. In particular, moving average and volume indicators are employed for this purpose. We also allow for the use of a Jensen's alpha [18] which can be calculated using just historical index and stock price movements as an input for the genetic algorithm to develop profitable trading strategies from.

III. Approach

This section describes our approach in constructing a dynamic and adaptive asset allocation system that considers changing market conditions. Trading rules are represented using fuzzy logic. An evolutionary process facilitates a search for high performance trading rules. In the following subsections we describe the structure of the fuzzy rules, the evolutionary process applied to fuzzy rule bases, and the evaluation function.

A. Representation

The fuzzy rule base representation enables intuitive natural language interpretation of trading signals and implies a search space of possible rules that corresponds to trading rules a human trader could construct. An example of a typical technical trading rule such as "buy when the price of a stock X's price becomes higher than the single moving average of the stock X's price for the last, say, 20 days" (indicating a possible upward trend) could be encoded using a fuzzy logic rule such as "If *Single Moving Average Buy Signal* is *High* then *rating* is 1"; conversely we could have a trading rule such as "sell stocks with high volatility when the portfolio value is relatively low" encoded by a fuzzy rule: "If *Price Change* is *High* and *Portfolio Value* is *Extremely Low* then *rating* is 0.1".

Each fuzzy rule base consists of a set of "if - then" rules where the "If" part specifies properties of technical indicators and the "then" part specifies a rating with 10 discrete levels given a stock with these properties. The rule inputs are termed linguistic variables in the fuzzy logic component. Clearly, at least one linguistic variable must be defined to construct rules, to construct the rules used to obtain the results presented in this paper we used $V = 9$: in section IV the linguistic variables used are described. The output is interpreted as a rating of the strength of a buy recommendation given fulfillment of the If part. It is possible for the If part of a rule to refer to any combination of the technical indicators the system uses to give one output rating. A rule base may contain at least one and no more than $O = 30$ rules.
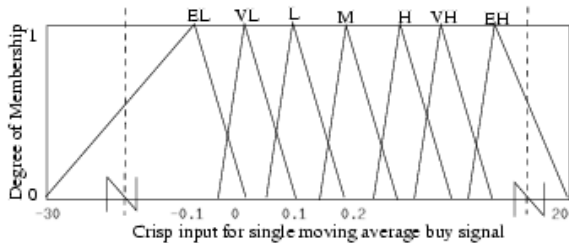
Fig. 1. Sample membership functions of Fuzzy Sets *EL*, ... *EH* for the single moving average buy signal linguistic variable (not to scale).

The value of each linguistic variable is described by one of a possible seven fuzzy membership sets. These are defined describing the relative magnitude of a particular observation: *Extremely Low* (*EL*), *Very Low* (*VL*), *Low L* (*L*), *Medium* (*M*), *High* (*H*), *Very High* (*VH*), and *Extremely High* (*EH*). Membership functions map crisp data observations to degrees of membership of these fuzzy sets.

Figure 1 shows a visualization of the membership functions for the single moving average indicator (see section IV). The membership functions are triangular: the mapping from an observation to a degree of membership for each membership function (*EL*, ..., *EH*) is fully defined by specifying a minimum, center and maximum value where the min and max values refer to the lowest and highest linguistic variable observations at the edges of the triangle that belong to the membership set to the least degree and the center belongs to the membership set to the highest degree (the top of the triangle).

The membership functions for each variable are initialized using observations of the variable derived from historical data and updated whenever new data is observed. The procedure for building a membership function for a linguistic variable comprises of first defining a membership set of observations the function applies to, and then defining the membership function by finding the minimum, maximum and center as discussed in the previous paragraph. First a series of historical data observations for the variable are sorted from lowest to highest. Then the series is divided into 7 ordered sets of equal size where each set corresponds to one of the 7 membership sets (*EL*, ..., *EH*), the lowest and highest member of each set give the minimum and maximum values that belong to the set and the center is found by taking the mean of all the observations that belong to the set. When new data is input to the system this procedure is repeated.

Any "if" part may include up to $V = 9$ linguistic variables; each linguistic variable can take one of 7 possible values; the output for each rule gives one of 10 different ratings; there can be up to $O = 30$ rules in each rule base.

To estimate the number of possible phenotypes, we can first estimate the number of possible phenotypes which consist of a single rule. Note that if a single rule has one linguistic variable present, then there are $7 \times 9 \times 10$ such phenotypes (9 possible linguistic variables, 7 possible values, 10 possible outcomes). If a single rule has two linguistic variables present, then there are $7^2 \times 36 \times 10$ such phenotypes (36 possible combinations of 2 linguistic variables out of available 9, 7^2 possible values for a pair of linguistic variables, 10 possible outcomes). In general, the number of possible phenotypes for a single rule (with one, two, ..., nine linguistic variables present) is

$$p = 10 \times \sum_{i=1}^{9} 7^i \times \begin{pmatrix} 9 \\ i \end{pmatrix} \approx 10^{10.2}.$$

As the number of phenotypes with k rules can be estimated as p^k, the estimation of the total number of phenotypes is $p^{30} < 10^{306}$.

An example of a phenotype rule base that could be produced by the system is given below. It consists of three rules; each rule has one or three linguistic variables in the "if" parts:

• If *Single Moving Average Buy Signal* is *Extremely Low* then *rating* = 0.9.
• If *Price Change* is *High* and *Double Moving Average Sell* is *Very High* then *rating* = 0.4.
• If *On Balance Volume Indicator* is *Extremely High* and *Single Moving Average Buy Signal* is *Medium* and *Portfolio Value* is *Medium* then *rating* = 0.5.

Internally, each rule is represented using a sequence of slots. With reference to figure 2: column 1 contains a Boolean value to indicate whether the rule is active; columns 2 through to 10 represent the rule inputs (each corresponds to a linguistic variable) and contain (a) a Boolean value indicating whether or not the linguistic variable is active, and (b) a number from 1 to 7 representing a membership function for the variable (1 corresponds to *extremely low* and 7 to *extremely high*); finally, column 11 indicates the rule output rating and contains a single floating point value from the set $\{0.1, 0.2, ..., 1.0\}$). The internal representation for a rule base is simply a 30×11 matrix (note that columns $2 - 10$ contain a Boolean and an integer).

The number of possible genotypes is
$2^{30} \times 2^{270} \times 7^{270} \times 10^{30}$,
as there are 2^{30} possible truth assignments for the Boolean variable in column 1 of the matrix, 2^{270} possible truth assignments for all Boolean variables in columns 2–10 of the matrix (i.e., 9 columns and 30 rows), 7^{270} possible assignments for integer variables in columns 2–10 of the matrix (i.e., 9 columns and 30

1	2		3		4		5		6		7		8		9		10		11
B	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	F
B	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	F
B	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	F
B	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	F
B	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	B	I	F

Fig. 2. Internal rule base representation for a rule base with $O = 5$ and $V = 9$. B indicates a boolean value: $B \in \{T, F\}$; I an integer: $I \in \{1, 2, 3, 4, 5, 6, 7\}$; and F a float: $F \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

rows), and 10^{30} possible assignments for the variable in the 11-th column of the matrix. Note that $2^{30} \times 2^{270} \times 7^{270} \times 10^{30} \approx 10^{318}$.

As an example, the genotype representation of the phenotype given above is provided in figure 3. Note that for compactness the illustration is of a rule base with $O = 5$ rules, for $O = 30$ the rule base additional rows would have false values in the first slot.

B. Evolutionary process

The fuzzy rule bases undergo an evolutionary process. An initial population of rule bases (genotypes) is selected at random and may be seeded with some rule bases that correspond to accepted technical trading strategies, for example the seeds used in the experiments discussed in this paper are given in section IV of this paper.

The evolutionary algorithm used in our asset allocation system is summarized by the following sequence of steps:
1. Initialize population P of n solutions (each solution RB_i is a rule base):

$$P = \langle RB_1, RB_2, \ldots, RB_n \rangle,$$

2. Evaluate each solution: calculate $eval(RB_i)$ for $i = 1, \ldots, n$,
3. Identify the best solution found so far ($best$),
4. Alter the population by applying a few variation operators (tournament selection of size 2 is used),
5. Apply a repair operator to each offspring; this operator controls diversity of offspring with respect to the best solution $best_{previous}$ from the previous generation (elitism is not used),
6. Repeat steps 2-5 successively for N generations,
7. The best solution after N generations represents the final solution.

Three variation operators (one mutation and two crossovers) and one repair operator are used in the process. We discuss them in turn.

The mutation operator works by possibly modifying each gene of a single parent rule base in the process of producing an offspring. The type of gene remains the same: for instance a Boolean value cannot become an integer used to represent a membership function nor a decimal used to represent an output rating. If a gene is Boolean it is flipped. Otherwise if it is an integer or float, one of three events occur with equal probabilities:
1. The corresponding gene in the parent is incremented or decremented (equal probability for either) by a small amount, δ, to derive the offspring gene: for floats $\delta = 0.1$ and for integers $\delta = 1$. Since integers represent membership sets the change corresponds to a shift of one degree of membership (for example from low to very low).
2. The gene in the offspring is assigned a new value at random. For an integer gene the new value is selected from the domain $1, 2, \ldots 7$ and for a float from the domain $0.1, 0.2, \ldots, 1.0$.
3. The corresponding gene in the parent is passed unaltered to the offspring.

The two crossover operators combine genes from two parents to produce a single offspring. The first one, uniform crossover, assigns each gene in the offspring the value of a gene selected from one of the parents (the parent that provides the gene value is selected with equal probability). The second crossover operator assigns the rows of the offspring matrix by selecting — with equal probability — rows from both parents. In other words, the effect of this operator is to build a new rule base by choosing complete rules from each parent.

The last operator used in the system is a repair operator. It is used to maintain stability between generations. It is a binary operator with two rule bases as arguments and its effect is to modify the first genotype in such a way that it is no more than p percent different from the second genotype, which is $best$ — the best genotype found. The number p is a parameter of the method; we found this parameter to be very important in controlling the type of rules generated, in section IV the values we used for p are given.

C. Evaluation of a Fuzzy Rule Base

The evaluation process comprises of three stages: in the first stage individual stocks are evaluated according to a rule base (section III-C.1); in the second stage, the overall rule base's performance is evaluated (section III-C.2). The return on investment (ROI) is adjusted in

T	F	2	T	1	F	7	F	1	F	6	F	1	F	7	F	4	F	1	0.9
F	F	4	T	2	F	2	T	3	T	4	F	5	F	1	T	6	F	2	0.3
F	T	1	F	3	F	2	F	2	F	2	T	4	F	4	F	4	F	4	0.7
T	T	4	F	5	F	1	F	4	F	2	T	5	F	2	F	2	F	5	0.4
T	F	6	T	3	F	3	T	3	F	7	F	3	T	7	F	1	F	3	0.5

Fig. 3.   Example of the internal rule base representation. The order of the columns indicates the particular linguistic variables, both this order and the meaning of the variables is given section IV

the final stage of the evaluation process (section III-D).

C.1 Rating of individual stocks

In this section the procedure to assign a rating to stocks with respect to a rule base is explained. For any stock X a rating $RB(X)$ is defined. This mapping will be described using an example. Consider a rule base as follows:

1. If *Single Moving Average Buy Signal* is *High* then *rating* = 0.7.

2. If *Price Change is High* and *Volume Change* is *Very High* then *rating* = 0.4.

On a particular day t the following observations are made of technical indicators for stock X:

1. *Volume Change* = 0.5
2. *Single Moving Average Buy Signal* = 0.95
3. *Price Change* = 0.2

The first step of the process is to process each rule individually. First consider the single If component of the first rule:

• If *Single Moving Average Buy Signal* is H̲igh

We observed that for stock X on day t the value for Single Moving Average Buy Signal was 0.95 on day t. We must find the degree that this observation is *High* to see how much it matches the rule: the membership function for High is defined by its min, center and max which are, in this case, 0.12, 0.97 and 3.88 respectively. Using Equation 1 a membership function defined by these values maps the observed value 0.95 to a degree of membership of 0.97 in *High* or 97% *High*, a visualization of this procedure is given in figure 4.

$$ m(x) = \begin{cases} \frac{x-min}{center-min} & , & \text{if } min \le x \le center \\ 1 & , & \text{if } x = center \\ \frac{x-max}{center-max} & , & \text{if } center \le x \le max \\ 0 & , & \text{otherwise} \end{cases} \tag{1}$$

Since the first rule only has one If part we now consider the output rating part of the rule: then *rating* = 0.7. Recall from section III-A that *The output rating is interpreted as a rating of the strength of a buy recommendation given the total fulfilment of the If part.* By applying the membership function the degree that a rule fulfills the "If" part is found: the rating is adjusted proportionally to the degree of membership of
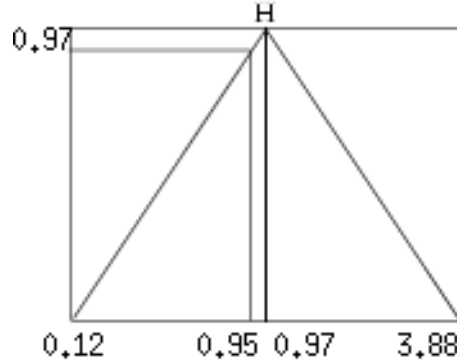

Fig. 4.   Finding the degree of membership of observed single moving average buy signal 0.95 for stock X is "High" with a degree of 0.98.

an observation to the linguistic variable specification in the If part. As the rule fulfilled the "If" part of the rule to the degree of 0.97 we adjust the output rating: $0.97 \times 0.7 = 0.679$.

The system looks at each rule in turn, the second rule in this example has two inputs:

• If $\widehat{\text{Price}}$ Change is *High* and *Volume Change* is *Very High* then *rating* = 0.4.

First each conjunction is processed separately. Assume using the process used for the first rule it is determined that the observation *Price Change* = 0.2 implies membership in the fuzzy set *High Price Change* = 0.5; and that *Volume Change* = 0.5 implies membership in *Very High Volume Change* = 1. These two values are combined using a common fuzzy a̲nd operator: multiplying the membership degrees. Hence the combined membership: $0.5 \times 1 = 0.5$. In the same way as for the first rule we adjust the output rating: $0.5 \times 0.4 = 0.2$.

The final step of the process is to derive an output rating for the whole rule base, $RB(X)$, this rating combines the results for each rule to give a rating for stock X given some input data. Recall that for the first rule the result was 0.679 and for the second it was 0.200. To get the output rating the center of mass of the results from each individual rule is found. In the example this value is $(0.679 + 0.2) \div (0.7 + 0.4) = 0.799$.

$$ RB(X) = \frac{\sum o_i}{\sum r_i}, \tag{2}$$

where o_i is the output of rule i for stock X, and r_i is the rating of rule i.

C.2 Evaluation of rule base performance

Using the procedure explained in the previous section for stock X a rule base is applied to each stock in the market. The result is a ranking of all stocks in a market M that is ordered by rating:

$$R(M) = \langle X_{i1}, X_{i2}, \ldots, X_{in} \rangle, \tag{3}$$

where $M = \{X_1, X_2, \ldots, X_n\}$ and $RB(X_{ik}) \geq RB(X_{ik+1})$.

The performance of a rule base RB is measured through analysis of the results of applying RB to simulated trading. The ranking of stocks discussed in the previous section that is implied by each rule base contains the information used in trading. A decoder defines the interpretation of the ranking to make decisions for portfolio construction.

The simulation takes place over a set period of time — a window of historical data. In the simulated scenario an initial capital is allocated to which is the used to construct an initial portfolio on day 1 of the simulation period. This initial portfolio is updated and traded over the rest of the data window. The decoder (see figure 5) formulates buy and sell decisions given a ranking for trading the portfolio.
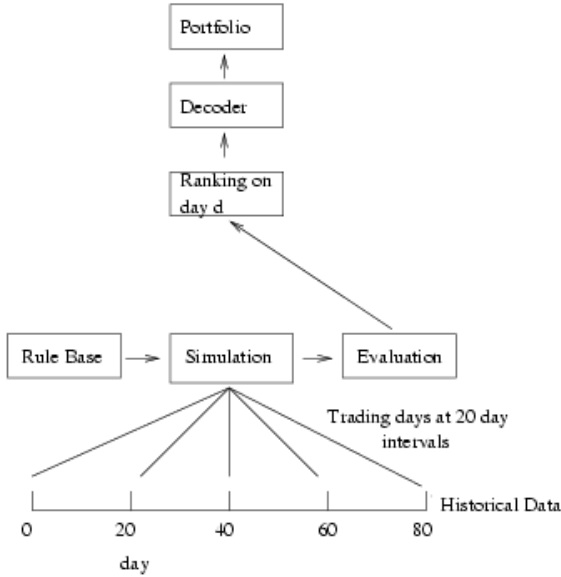

Fig. 5. The decoder takes a ranking and recommends a portfolio.

In the system, a portfolio P_t is defined as a vector of holdings of stocks in $M = (X_1, \ldots, X_n)$ at time t:

$$P = [a_1 X_{i1}, \ldots, a_k X_{im}], \tag{4}$$

where $a_1, \ldots, a_k$ are natural numbers, $\{X_{i1}, \ldots, X_{im}\} \subseteq M$, and $Value(P, t) = \sum a_j \times price(X_{ij})$, for $j = 1, \ldots, m$.

Two key parameters used in the decoder. They are *buy best stocks percentage* and *sell Worst Stocks Percentage*. *Buy best stocks percentage* is the percentage of stocks to select from top of the ranking and *sell Worst Stocks Percentage* is the percentage to sell from the bottom of the ranking. In all simulations we used a value of 10 for both parameters.

The process for updating a portfolio P_1 to get the next portfolio P_2 involves creating a new ranking on trading day 2 and selling stocks held that are at the bottom of the new ranking up to *sell worst stocks percentage*. Using the cash from selling the worst stocks as well as any unallocated cash the top buy best stocks percentage of stocks in the new ranking are bought if they are not already in the portfolio. Cash is distributed evenly over the best-ranked stocks until cash either runs out or the portfolio contains the buy best stocks percentage of all stocks in M. Trading does not usually take place every day in the simulation scenario, portfolio updates are processed at set intervals defined by the distance between trading days.

Transaction costs are accounted for in the simulation. Transaction cost is deducted from the simulated capital for every transaction. The portfolio is updated after a set number of days d (typically every 20 days), and then every d days after that.

Performance is highly influenced by the assumptions made in the simulation including the method used to interpret the ranking by the decoder and the parameters used to guide the portfolio construction during the simulation the parameters are tied to each rule base and may be subject to the evolution with the rules. Another crucial assumption is choice of the historical data window; we leave discussion of this aspect to further paragraphs of this subsection.

Rule base performance is evaluated by analysis of portfolio performance during simulation . The measure used for evaluation of portfolio performance is Return on Investment (ROI) during the whole simulation period (see Equation 5).

$$ROI = \exp\left[\frac{e^{\ln(V_{t1}) - \ln(V_{t0})}}{t_1 - t_0}\right], \tag{5}$$

where $V =$ Portfolio Value, $t_1 =$ End Time and $t_0 =$ Start Time.

The result of the simulation is the ROI during simulation for $RB_X : ROI(RB_X)$. To compare RB_X to another rule base RB_Y it is the case that if $ROI(RB_X) > ROI(RB_Y)$ then RB_X is better than RB_Y. This basic criteria is supplemented by a few additional character-

istics of performance which are considered in the final evaluation described in the next section.

D. Final Evaluation

Additional criteria are considered in addition to ROI when measuring performance. This is implemented using penalties to guide the evolutionary search away from rule bases that produce undesirable return distributions within the training period (even if the return over the whole period is good) and also to prevent over fitting solutions to training data. The final evaluation value equals the ROI in simulation minus Penalties. There are two penalties applied to modify ROI, and they are:

1. Portfolio loss penalty
2. Ockham's razor penalty

Let us discuss each penalty in turn starting with the portfolio loss penalty.

In simulation we measure the portfolio gain or loss on each trading day (see section III-C.2) as well as the final return on investment over the simulation period. Solutions that result in a reduction of portfolio value (during simulation) are penalized if they result in losses on any trading day even if at the end of the simulation period the return was high (see Equation 6). This mechanism provides a risk reduction facility and by adjustment of the penalty values that are imposed lever to focus the search for rule bases that can give particular return characteristics. The penalty becomes progressively higher for large losses.

$$m(x) = \begin{cases} 0.01, & \text{if} \quad \delta geq - 5\% \\ 0.1, & \text{if} \quad -5\% \leq \delta \geq -10\% \\ 10, & \text{if} \quad \delta \leq -10\% \end{cases} \quad (6)$$

where δ is the percentage change in portfolio value since the previous trading day.

For example if we had a 120-day training simulation with a trading interval of 60 days the penalty would be applied twice: once at 60 days and once at 120 days. In this example if a rule base had an initial value of $\$10,000,000$ on day 1 of simulation then at day 60 a value of $\$95,000,000$ and on day 120 a value of $\$99,500,000$ the penalty would be calculated on each trading day as follows:

1. On day 1 no penalty is applied as it is the first day,
2. On day 60 the penalty is incremented by 0.1 because the portfolio lost 5% of its value (Equation 6),
3. On day 120 no penalty is applied because the portfolio increased 4.5% since the previous trading day.

The second penalty, Ockham's razor (Equation 7), reduces the fitness of solutions with many rules unjustified by returns. The reason that it is better to have fewer rules is that this encourages generality rather than over fitting to training data.

$$P_{ockham} = \text{ number of rules } \times k, \quad (7)$$

where k is a penalty constant.

The penalties are added together to get an overall value for each single rule base. This value is deducted from the ROI for that rule base. Figure 6 gives an overview of the process required to determine a fitness value comprising a penalized ROI value for each rule base.
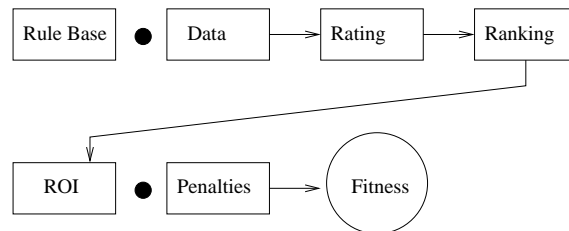

Fig. 6. Complete evaluation process.

Using the methods to set the objective of the EA the characteristics of rule bases with higher performance from both a risk and return perspective are targeted. The result is a best rule base that is able to be used for real trading. It is important that the rule base is applied to real trading in the same way as in simulation.

E. Adapting the rule base

In this section methods to cause rule bases to adapt to market conditions are discussed. Approaches towards this are through selection of data windows and by controlling the search. During the search process the performance of rule bases is evaluated based on data as described in section III-C. Rule bases that perform well during the training data window are identified by the search. We first discuss the methods to select data windows and then controlling the search.

Three methods for selecting a data window are considered:

1. Initial Window,
2. Extending Window,
3. Sliding Window.

The initial window (figure 7) uses a single initial period to evolve a rule base and then the rules from this period are used for all future trading. The extending window (figure 8) uses all the historical data available to evaluate rule bases. The sliding window (figure 9) uses a recent historical time window for evaluation. In methods 2 and 3 the rule base adapts to consider the changing market, the sliding window fits the rule base to a period in the recent past. Note that in 2 and 3 the rules are applied to trading immediately after the

last historical data period has transpired. Another approach to be tested in the future will involve identifying characteristics of the market (market regimes) during training windows and then applying rule bases when the market appears to be exhibiting these characteristics.
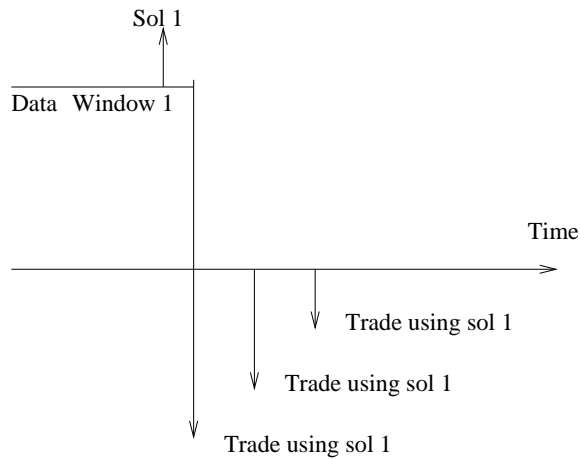

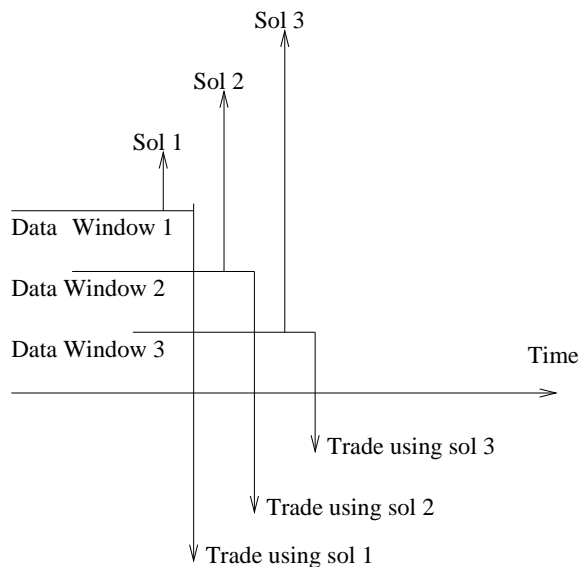Fig. 7. A static rule base approach.

Fig. 8. A sliding window approach to adaptation by updating training data.

A new search takes place for each new window in the extending and sliding window methodologies, however instead of starting with a completely new population a memory is maintained of the best solution from previous windows. The best solution from the previous window is used in the generation of the initial population for each window. This is achieved using the repair operator (section III-B).
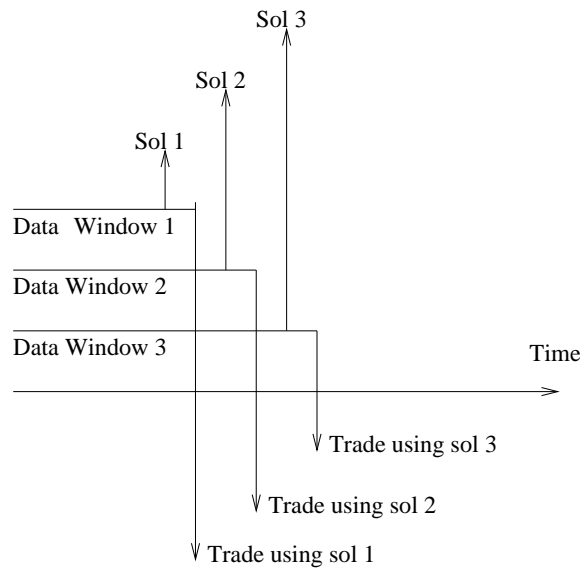

Fig. 9. An extending window approach to adaptation by updating training data.

IV. EXPERIMENTAL SETUP

In this section parameters and details used to configure the system to obtain the results discussed in the next section are provided.

A. Inputs

The system was tested using historical data for stocks in the MSCI Europe index from 1990 to 2005. The MSCI Europe index represents the largest stocks, by market capitalization, which are traded across Europe. The MSCI Europe is in fact primarily composed from the individual country indices that MSCI creates and tracks. Although the constituent stocks that make up the index change over time, between 1990 and 2005 there were at least 700 active stocks that comprised the index at any point in time, with a total of 1241 represented over the whole period.

Two input files were used: one containing series for the trading volume of each stock, the other containing price data. The linguistic variables used are based on well known technical indicators used by real traders, they were calculated solely using price and volume data. All stock data was adjusted for various company events that would alter the price of individual stocks. This would include, for example, share splits and the payment of dividends. All payments generated from a stock were assumed to have been reinvested back into the same stock. Also, share prices were converted to all be in the Euro. Where necessary, DataStream International synthetic Euro FX rates were utilized for currencies without a direct relationship with Euro or

ERM prior to it becoming a physical currency.

The risk free rate of return used to calculate the alpha of stocks plus performance evaluation statistics provided in the next section are from the 3-month Euro deposit rate series that was taken from DataStream International.

A listing and brief description of the meaning of each linguistic variable is provided below with reference to a day t when the signal applies:

1. *Price Change*: the change in price over a 20 day period before day t,
2. *Single Moving Average Buy Signal*: the difference between the price at time t and a 20 day moving average at time t when the price is greater than the moving average,
3. *Single Moving Average Sell Signal*: the difference between the price at time t and a 20 day moving average at time t when the price is less than the moving average,
4. *Portfolio Value*: the value of the portfolio at time t,
5. *Double Moving Average Buy Signal*: the difference between a 10 day moving average at time t and a second moving average based on a longer time period (20 days) at time t when the first moving average is greater than the double moving average,
6. *Double Moving Average Sell Signal*: the difference between a 10 day moving average at time t and a second moving average based on a longer time period (20 days) at time t when the first moving average is less than the double moving average,
7. *On Balance Volume Indicator (OBV) Buy Signal*: the OBV indicator compares volume to price movements. A running indicator termed the OBV indicator is constructed such that volume is added if the closing price at time t of the indicator is higher than the previous closing price (at $t-1$), subtracted if it is lower and does not change if the closing price remains static. A buy signal is produced whose strength depends on the extent of divergence between the maximum price and the maximum OBV over a period from t to $t-20$.
8. *On Balance Volume Indicator (OBV) Sell Signal*: An OBV sell signal is produced whose strength depends on the extent of divergence between the minimum price and the minimum OBV over a period from t to $t-20$.
9. *Alpha*: an indicator based on the Capital Asset Pricing Model (see section V-A).

B. Parameters for the Evolutionary Algorithm

The probabilities of applying the three offspring producing operators was for mutation there was probability 0.4, for uniform cross over 0.3 and for the rule crossover (see section III-B) the probability was again 0.3. For the repair operator in all experiments we used $p = 10\%$.

In the section III-B we mentioned the initial population can be seeded with predetermined rule bases. At the beginning of each optimization including for every window in the sliding window schema a single price momentum strategy rule base was inserted into the population, its phenotype was:

- If *Price Change* is *Extremely Low* then *rating* = 0.0
- If *Price Change* is *Very Low* then *rating* = 0.16
- If *Price Change* is *Low* then *rating* = 0.33
- If *Price Change* is *Medium* then *rating* = 0.5
- If *Price Change* is *High* then *rating* = 0.67
- If *Price Change* is *Very High* then *rating* = 0.83
- If *Price Change* is *Extremely High* then *rating* = 1.0

A sliding window methodology was used with a 120 day window with a 20 day window movement between periods. The real trading portfolio used to evaluate the results was generated using a rule base from the previous window. For the trading simulation (see section III-C.2) the parameters *buy best stocks percentage* was set to 10% and *sell worst stocks percentage* was set to 5%. An additional constraint was also set that the maximum number of companies that the portfolio could take a position in at any one time be limited to 100 stocks.

The period the evolutionary algorithm would run for each window was controlled using a max steps without improvement parameter (MSWI) which allows the evolutionary algorithm to continue iterating untill MSWI iterations passed without a better rule base being found. In these experiments $MSWI = 5000$.

V. RESULTS

In this section we present results of applying the system using the experimental setup defined in section IV. The discussion is divided into two parts: the first, section V-A, comprises an analysis using standard portfolio evaluation tools widely accepted by finance practitioners and researchers; the second, section V-B, consists of an evaluation using stochastic methods that do not make any assumptions about the characteristics of the return distributions. In the following discussion we refer to the portfolio generated using the computational intelligence system presented in this paper as the Evolutionary Algorithm (EA) portfolio.

A. Standard Performance Measures

In order to test and evaluate the performance of the EA portfolio, not only is a benchmark portfolio required, but also a comparison should be made with alternative strategies. A comparison with other traded funds would not necessarily be suitable, as the EA has been restricted to only utilize price and volume data. Traded funds in the market are obviously able to also apply a wealth of company information ranging from

cash flows, earnings and dividend behavior to name but a few, in order to determine the best buys in the market. Therefore, we instead focus our main efforts into comparing the performance of the EA portfolio to two other more traditionally constructed portfolios that use the same information set available to the EA. The first of these portfolios is constructed from a price momentum strategy. Every 20 days the portfolio is re-balanced to hold the top 10% of stocks that are the best performing, in terms of returns over the previous 120 day period. There is sufficient academic research to indicate price-momentum strategies can outperform a passive index-tracking portfolio. [17] provides a recent discussion on the profitability of price momentum strategies and the potential reasons behind it. However, by constraining the price momentum design in this way it will be congruent to utilizing the same dataset and trading constraints applied by the EA process. The results from this type of portfolio will therefore provide an indicator as to whether the EA portfolio does more than just replicate a momentum strategy.

The second portfolio is an alpha portfolio, based on the single-factor model, presented below:

$$r_{i,t} - r_{f,t} = \alpha_i + \beta_i \left[r_{m,t} - r_{f,t} \right] + e_{i,t} \qquad (8)$$

Theoretically, in an efficient market it would be possible to price stocks based solely on their risk components. Under the classical CAPM there is only one risk factor, that being the systematic risk of the stock. Therefore, excess returns of any stock, i, above the risk-free rate, r_f, can be fully explained by its level of systematic risk, β_i, and the market risk premium $(r_m - r_f)$. The alpha value of the stock, α_i, should be zero. If it is not and in fact there is a positive value then the stock in outperforming relative to its level of systematic risk and should be bought. The higher the alpha value, the better the stock is to purchase. An alpha value is calculated for each stock every 20 days using stock returns from the previous 60 days of trading data. Stocks with the highest alphas are bought and held.

We recognize that the above single-factor model is a relatively basic model of risk, and does not take into account more commonly used frameworks such as Fama and French's three-factor model [11], incorporating size and book to market value effects. One can also question the validity of calculating alphas over short periods of only 60 days, and the statistical significance of them. However, tests using various lengths of time to calculate alphas did not lead to radically different results. Moreover, the single-factor model explained above is congruent with forming a portfolio using only price and volume information It is also a subset of the

information set utilized by the EA itself, and as such can provide some measure of relative performance to the EA portfolio from its ability to deviate away from standard price momentum and alpha-based strategies.

Three further portfolios were also created. The first being a hypothetical MSCI Europe passive index. This essentially mimics the returns from the MSCI total return index itself and is set as the raw benchmark for all portfolios. The second is a buy-and-hold portfolio created by holding a selection of stocks based on optimizing the initial window, as discussed in section III-A. This will provide for a comparison of the EA performance against a static model. Finally, results from a hill climb optimization routine is also provided to compare the EA against another search-based optimization approach. The algorithm was initialized with a random rule-base of the same type as used by the EA and is based on mutation operator described in section III-B, which enables the search to avoid being trapped in local optima. The solution is progressively improved through iterations; the algorithm is terminated when no improvement is found after 5,000 iterations.

At a first glance it is noticeable that the EA portfolio has performed exceedingly well when examined from an investors point of view who would have held the portfolio from inception until the end of the sample period. In fact, the EA provides an excess holding period return of 782.98%, this being more than four times the excess holding period return generated from an investor that had simply bought into a passive fund that tracked the market index (earning a return of 187.25%). To illustrate this, figure 10 tracks the value of each portfolio for the 13 year holding period. Annualized excess returns for the EA were more than double (at 19.09%) to the market index. Interestingly, this higher return performance was not at the expense of higher risk, with annualized standard deviations below that of the MSCI index.

From a visual inspection of figure 10 it is interesting to observe that the alpha and price momentum strategies seemed to perform quite well from 1992 to 2000 when for the most part the MSCI index followed an upward drift. The bearish market conditions thereafter did not help either portfolio perform as they did in the past. This is to be somewhat expected as the two strategies are more aligned for working with bull runs. There is a substantial body of research analyzing the potential reasons for the success of such simple strategies as that of a price momentum (see [17]), particularly during the 1990s. However, it is also interesting to note that despite the change in market sentiment, the EA portfolio did not lose anywhere close to the same amount of money that the alpha and price momentum portfolios declined by, as it would seem that

	MSCI Europe	EA	Alpha	Buy and Hold	Price Momentum	Hill Climber
Excess Holding Period Returns	187.25%	782.98%	258.72%	224.48%	175.94%	78.02%
Excess Annualized Geometric Returns	8.61%	19.09%	10.78%	10.25%	8.46%	4.797%
Annualized Volatility	18.96%	18.07%	25.81%	19.69%	25.63%	16.979%
Sharpe Ratio	0.5307	1.063	0.5251	0.5948	0.4452	0.2825
Jensen Alpha	NA	16.62%	9.16%	9.85%	7.93%	5.173%
Modified Alpha	NA	16.57%	9.09%	9.76%	7.86%	5.055%
Information Rank	NA	0.9524	0.3734	0.5066	0.3189	0.3061
Net Selectivity Measure	NA	9.62%	-0.01%	1.26%	-2.12%	-2.079

TABLE I

STANDARD PORTFOLIO PERFORMANCE MEASURES. ALL FIGURES ARE FOR PORTFOLIOS THAT WERE ORIGINALLY CREATED ON 16TH NOVEMBER 1992 AND HELD UNTIL 19TH SEPTEMBER 2005 USING EURO AS THE BASE CURRENCY. THE SHARPE RATIO IS CALCULATED FROM ANNUALIZED ARITHMETIC RETURNS. EXCESS RETURNS ARE BASED ON COMPARISON WITH THE 3-MONTH EURO DEPOSIT RATE. THE QUOTED ALPHAS, INFORMATION RANK AND NET SELECTIVITY MEASURES HAVE BEEN ANNUALIZED.
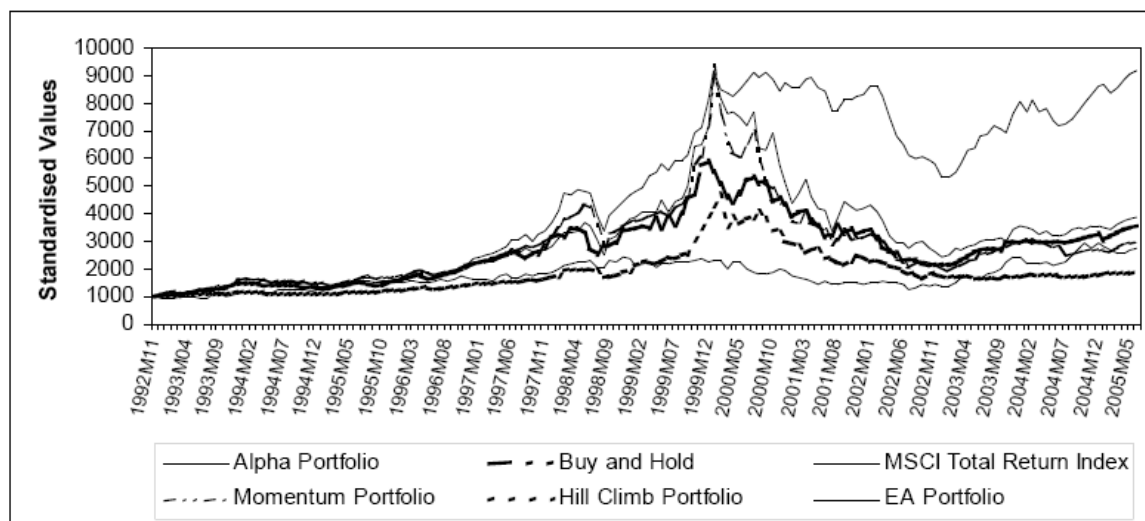

Fig. 10.   Portfolio Values from 1992 to 2005. Each portfolio starts at a value of 1000 on 16th November 1992.

the adaptive trading rules utilized by the EA were able to evolve to the bearish phase in the financial market. This highlights well the importance of having an evolving rule-base to adapt to new market conditions. The Buy and Hold rule-base portfolio declines in value with the alpha and price momentum portfolios, suggesting further that the initial set of trading rules were suited for a bullish market and not suitable for bear runs. Interestingly, the hill climb approach also falls in portfolio value, suggesting the rule-bases were not as adaptive to changing market conditions as the EA method.

As one of the most popular and easily recognizable methods to compare portfolios is through their Sharpe ratios [24], table I also tabulates these results. The Sharpe measure is calculated as the returns of the portfolio, r_p, above the risk-free rate, r_f, divided by the portfolio standard deviation:

$$Sharpe = \frac{(r_p - r_f)}{\sigma_p}$$

As a measure of total risk adjusted return performance, only the EA and buy and hold portfolios were able to beat the market index. The slightly higher returns from the alpha portfolio did not sufficiently compensate for the far higher level of risk (a standard deviation of 25.81%). What is also of interest to note is the relatively high sharpe ratio for the EA portfolio at 1.063. Once total risk, as measured by the standard deviation of portfolio returns, is taken into account the EA portfolio stands out amongst all of the alternatives.

The next four measures are all based on the single-factor model and relate the performance of the portfolios to the benchmark, MSCI Europe index. The first two measures tabulate the portfolio alphas. These are similar to a stock's alpha, but relate to how much better the portfolio has performed relative to the systematic risk of the portfolio and performance of the benchmark index. All of the portfolios show some degree of overperformance, having positive alphas. However, only the alpha statistic from the EA portfolio was found to be significant at the 1% confidence level. Modified (see [19]) alpha values are also tabulated. These alpha values have been computed to take into account the fact that the returns series may not be normally distributed. [1] However, there is actually no significant difference in

<hr>

[1] The modified alpha is calculated as:

$$B_p = \frac{Cov\left[r_p, -(1 + r_m)^{-b}\right]}{Cov\left[r_m, -(1 + r_m)^{-b}\right]},$$

where

$$b = \frac{ln(E\left[1 + r_m\right]) - ln(1 + r_f)}{Var\left[ln(1 + r_m)\right]}.$$

the figures presented. The robustness of these alpha values can also be measured through the information performance rank that is presented. Sometimes also known as the appraisal ratio, it measures the portfolios average return in excess of the benchmark portfolio over the standard deviation of this excess return. Essentially, it evaluates the active stock-picking skills of the strategy, once unsystematic risk generated from the investment process is accounted for. As we are comparing each of our portfolio's with the MSCI Europe total return index, the information ratio is calculated as:

$$\text{Annualized Information Ratio} = \frac{\sqrt{T}\alpha}{\sigma_e},$$

where T is the period multiple to annualize the ratio and σ_e is the standard error of equation 8. Compared to other funds in the market, an appraisal ratio of 95% for the EA portfolio is indicative of a very strong and consistent performance. Grinold and Kahn, [15], have argued good information ratios should be between 0.5 and 1, with 1 being excellent. Goodwin [14] examined over 200 professional equity and fixed income managers over a ten year period and found that although the median information ratio was positive, it never exceeded 0.5. Of all the alternative portfolios, only the Buy and Hold portfolio comes close to beating the 0.5 value.

The final row in table I presents the results from Fama's Net Selectivity measure [10]. It provides a slightly more refined method to analyze overall performance for an actively managed fund. Overall performance, measured as the excess returns of the portfolio over the risk-free rate can be decomposed into the level of risk-taking behavior of the strategy and security selection skill. This security selection skill, or Selectivity, can be measured as a function of the actual return of the portfolio minus the return that the benchmark portfolio would earn if it had the same level of systematic risk. This selectivity value, however, can be broken down still further to calculate Net Selectivity. given that a portfolio's strategy may not be limited to simply track the benchmark portfolio – which would be the case for our portfolios under examination – it is also necessary to take into account the fact that the portfolios are not fully diversified, relative to the chosen benchmark. In fact, for the EA and buy and hold portfolios the maximum number of stocks that it is allowed to have is restricted to 100, far less than the MSCI index. To account for this, net selectivity is the value of selectivity that the strategy adds to the portfolio minus the added return required to justify the loss of diversification from the portfolio moving away from the benchmark. This effectively means any returns that the portfolio earns above the risk free rate

must be adjusted for *both* the returns that the benchmark portfolio would earn if it had the same level of systematic risk and the same level of total risk to the benchmark.

The net selectivity figures quoted will, by default, all be less than the alpha values previously examined. However, even when the differences in total risk are accounted for, the EA portfolio provides a very positive result. In fact, the only other portfolio to show a positive net selectivity figure is, again, from the Buy and Hold rule base.

Table II shows general distribution characteristics of the portfolios under examination. A Jarque-Bera test [6] shows that none of the constructed portfolios are normally distributed with the exception of the MSCI index. The EA portfolio shows evidence of negative skewness, implying from an investor's perspective that the majority of returns are generally above the mean, although large negative returns can be expected on an irregular basis. With the exception of the MSCI index all series demonstrate fat tails, implying there will be more than expected large price changes than under a normal distribution. In particular, the hill climb exhibits far more excess kurtosis and skewness than the other portfolios. The excess kurtosis would lead to more regular, larger swings away from the mean in investor returns when compared to the EA portfolio. From an investors perspective, this is not particularly desirable and is investigated further in the stochastic dominance tests that are conducted.

One of the reasons for the shape of the distributions that have arisen from the EA strategies could be due to the specific fitness and penalty functions imposed upon the system. To investigate the tail ends of the returns distribution for the portfolios the table also reports some basic probability statistics. Specifically, the probability of experiencing in any given month a gain or loss greater than 10%. From these figures it is noteworthy that it is the EA portfolio that has the greatest chance of producing a monthly return in excess of 10%, and the smallest chance of producing undesirable negative returns greater than 10%. The probability of these occurring in any given month is 4.58% and 2.61%, respectively. These results may be indicative of the penalty function correctly discarding the choice of stocks that are more likely to experience a large decline. Although the penalty function can be viewed as a means to ensure the fitness function is geared more closely towards being a risk-adjusted return, it is not the same as employing a Sharpe ratio or other standard deviation measure. The difference being that the penalty function only penalizes for large downside risk, rather than both up and downside risk.

The table also provides a simple measure of how often a negative monthly return can be expected for an investor holding the relevant portfolios. Once again, it is the EA portfolio that performs the best out of the alternative strategies, experiencing a negative return only once every 3 months.

To cater for the fact that the returns distributions are non-normal, the following section, V-B, evaluates the relative performance of each of the constructed portfolios with the MSCI index using non-parametric, distribution free stochastic dominance tests. These will go someway to deal with the fact that upside and downside movements in the above portfolios are not symmetric.

B. Stochastic Dominance Portfolio Evaluation

The concept of stochastic dominance (SD) gives a systematic framework to analyze investment choices under uncertainty, utilizing only some general assumptions on an investor's utility function. The attractiveness of the method is therefore on it not requiring any knowledge of the statistical distribution of the investment alternatives. Nevertheless, it provides a statistical comparison between portfolios using the whole distribution, rather than just point estimates.

In CAPM analysis, the efficiency criterion uses only mean and variance of the returns, based on the underlying assumption that returns are distributed normally. As discussed in section V-A, none of the return distributions of our portfolios are normally distributed. SD efficiency criteria do not require this distributional assumption. The three most general SD efficiency criteria are

1. First degree stochastic dominance (FSD) rule. This is the smallest efficiency criterion which produces the smallest possible efficient set for all rational investors — individuals with an increasing utility function.

2. Second degree stochastic dominance (SSD) rule. This is the smallest efficiency criterion which produces the smallest possible efficient set for all risk averse investors.

3. Third degree stochastic dominance (TSD) rule. This is the smallest efficiency criterion which produces the smallest possible efficient set for all rational investors, who are risk-averse and have decreasing absolute risk aversion.

To answer the question of whether the EA portfolio is a superior investment choice for any of the above three types of investors, we need to test whether the return distribution generated by the EA rules dominates the alternative strategies. This is achieved by conducting pair-wise tests of stochastic dominance:

H_0^j : G dominates F stochastically at order j,

H_1^j : G does not dominate F

stochastically at order j,

	MSCI Europe Total Return Index	EA	Alpha	Buy and Hold	Price Momentum	Hill Climber
Average monthly return	0.8280%	1.6629%	1.2095%	1.0152%	1.0272%	0.5196%
Median monthly return	1.1648%	2.0782%	1.7063%	1.3757%	1.5490%	0.7591%
Largest positive monthly return	14.0260%	17.0171%	28.6675%	22.3981%	29.8286%	18.0645%
Largest negative monthly return	-12.8476	-17.0083	-19.5658	-16.1300	-24.3551	-25.2463%
Average monthly volatility	5.4607%	5.1996%	7.4242%	5.6528%	7.3720%	4.9016%
Skewness	-0.0247	-0.1863	0.2633	-0.0322	0.0552	-0.667352
Kurtosis	3.0383	4.3629	5.3485	4.2690	5.8977	8.907055
Jarque-Bera	0.02491	12.7255[a]	36.9289[a]	10.2929[a]	53.6072[a]	226.1609[a]
Probability of a loss greater than 10% in any given month	4.58%	2.61%	11.76%	5.88%	16.99%	4.70%
Probability of a gain greater than 10% in any given month	3.92%	4.58%	7.84%	0.65%	1.31%	4.00%
Number of months before a negative monthly return	2.4	3.1	2.7	2.5	2.7	2.3

TABLE II

STANDARD PORTFOLIO PERFORMANCE MEASURES. MONTHLY RETURNS ARE CALCULATED ON A DISCRETE BASIS AS A PERCENTAGE CHANGE FROM ONE DAY TO THE NEXT. THE JARQUE-BERA STATISTIC IS A CHI-SQUARE DISTRIBUTED TEST FOR NORMALITY WITHIN THE SERIES. [a] SIGNIFIES REJECTION OF THE NULL HYPOTHESIS OF A NORMAL DISTRIBUTION AT THE 1% SIGNIFICANCE LEVEL.

where G and F are two cumulative return distributions generated from two different technical strategies. The hypotheses can be written compactly as

$$H_0^j : J_j(z;G) \leq J_j(z;F)$$
$$\text{for all } z \in [0, \bar{z}],$$
$$H_1^j : J_j(z;G) > J_j(z;F)$$
$$\text{for some } z \in [0, \bar{z}],$$

where $[0, \bar{z}]$ is the common domain of F and G and $J_j(.;G)$ is the function that integrates the function G to order $j-1$ so that, for example:

$$J_1(z;G) = G(z),$$

$$J_2(z;G) = \int_0^z G(t)dt$$
$$= \int_0^z J_1(t;G)dt,$$
$$J_3(z;G) = \int_0^z \int_0^t G(s)dsdt$$
$$= \int_0^z J_2(t;G)dt,$$

and so on.

Recently, Barret and Donald [5] proposed a set of Kolmogorov-Smirnov type tests (KS tests) for SD of any order. The KS tests compare two distributions at all points in the domain range, therefore having the

potential to be consistent tests of the full restrictions implied by SD. The tests also allow for different sample sizes, and the p-values are generated via a variety of simulation and bootstrap methods. Table III and table IV report the p-values for various tests of pair-wise dominance between all five portfolios under consideration. In table III, the p-values are calculated via 2 different Monte Carlo simulation methods, whereas in table IV the p-values are calculated using 3 different bootstrapping procedures. The null hypothesis of G dominance over F is rejected at 95% level of confidence if the p-value is smaller than 0.05.

This table reports the p-values of tests for the dominance of portfolio *G* over portfolio *F*

Panel 1. P-values calculated by Monte Carlo simulation method 1

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						*G* portfolio						*G* portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.33	0.33	0.39	0.15	0.45	-	0.01	0.01	0.04	0.01	0.71	-	0.00	0.00	0.01	0.00	0.67
PM	0.52	-	0.45	0.39	0.19	0.23	0.69	-	0.64	0.70	0.47	0.02	0.65	-	0.66	0.65	0.67	0.09
Al	0.52	0.59	-	0.45	0.45	0.12	0.69	0.43	-	0.55	0.30	0.03	0.64	0.31	-	0.66	0.67	0.14
BH	0.94	0.45	0.59	-	0.23	0.19	0.65	0.12	0.16	-	0.29	0.16	0.58	0.04	0.12	-	0.26	0.41
Idx	0.9	0.59	0.59	0.85	-	0.02	0.52	0.09	0.15	0.46	-	0.11	0.49	0.11	0.18	0.56	-	0.38
Hc	0.00	0.00	0.00	0.01	0.05	-	0.00	0.21	0.11	0.11	0.24	-	0.00	0.59	0.52	0.16	0.36	-

Panel 2. P-values calculated by Monte Carlo simulation method 2

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						*G* portfolio						*G* portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.33	0.33	0.39	0.15	0.45	-	0.01	0.01	0.04	0.01	0.69	-	0.00	0.00	0.01	0.00	0.65
PM	0.52	-	0.45	0.39	0.19	0.23	0.72	-	0.65	0.72	0.47	0.02	0.67	-	0.66	0.67	0.65	0.11
Al	0.52	0.59	-	0.45	0.45	0.12	0.73	0.44	-	0.57	0.30	0.03	0.69	0.33	-	0.68	0.66	0.13
BH	0.94	0.45	0.59	-	0.23	0.19	0.66	0.13	0.17	-	0.30	0.15	0.58	0.05	0.13	-	0.28	0.41
Idx	0.90	0.59	0.59	0.85	-	0.02	0.57	0.11	0.17	0.49	-	0.08	0.52	0.14	0.21	0.49	-	0.36
Hc	0.00	0.00	0.00	0.01	0.05	-	0.00	0.22	0.12	0.10	0.29	-	0.00	0.63	0.55	0.15	0.41	-

Legend — EA: evolutionary algorithm; PM: price momentum; Al: alpha; BH: buy and hold; Idx: msci total return index; Hc: hill climbing

TABLE IV

KS TEST FOR SD – BOOTSTRAP METHODS

This table reports the p-values of tests for the dominance of portfolio G over portfolio F

Panel 1. P-values calculated by bootstrap method 1

F	1st order dominance G portfolio						2nd order dominance G portfolio						3rd order dominance G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.33	0.33	0.39	0.15	0.45	-	0.00	0.00	0.00	0.00	0.20	-	0.00	0.00	0.00	0.00	0.20
PM	0.52	-	0.45	0.39	0.19	0.23	0.60	-	0.40	0.60	0.40	0.00	0.20	-	0.20	0.20	0.20	0.00
Al	0.52	0.59	-	0.45	0.45	0.12	0.80	0.60	-	0.80	0.60	0.00	0.80	0.20	-	0.80	0.80	0.00
BH	0.94	0.45	0.59	-	0.23	0.19	0.40	0.00	0.00	-	0.00	0.00	0.40	0.00	0.00	-	0.00	0.00
Idx	0.90	0.59	0.59	0.85	-	0.02	1.00	0.40	0.40	0.80	-	0.00	0.80	0.40	0.40	0.80	-	0.00
Hc	0.00	0.00	0.00	0.01	0.05	-	0.00	0.00	0.20	0.00	0.60	-	0.00	0.20	0.80	0.00	0.40	-

Panel 2. P-values calculated by bootstrap method 2

F	1st order dominance G portfolio						2nd order dominance G portfolio						3rd order dominance G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.33	0.33	0.39	0.15	0.45	-	0.20	0.20	0.20	0.00	0.80	-	0.20	0.20	0.20	0.00	0.80
PM	0.52	-	0.45	0.39	0.19	0.23	0.80	-	0.80	0.80	0.80	0.20	0.80	-	0.80	0.80	0.80	0.40
Al	0.52	0.59	-	0.45	0.45	0.12	0.80	0.40	-	0.40	0.60	0.20	0.80	0.40	-	0.80	0.80	0.20
BH	0.94	0.45	0.59	-	0.23	0.19	0.80	0.20	0.20	-	0.60	0.20	0.80	0.20	0.20	-	0.60	0.80
Idx	0.90	0.59	0.59	0.85	-	0.02	0.80	0.20	0.20	0.60	-	0.40	0.80	0.20	0.20	0.40	-	0.60
Hc	0.00	0.00	0.00	0.01	0.05	-	0.00	0.40	0.00	0.20	0.00	-	0.00	0.60	0.60	0.20	0.20	-

Panel 3. P-values calculated by bootstrap method 3

F	1st order dominance G portfolio						2nd order dominance G portfolio						3rd order dominance G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.33	0.33	0.39	0.15	0.45	-	0.20	0.20	0.20	0.00	0.80	-	0.40	0.40	0.00	0.00	0.80
PM	0.52	-	0.45	0.39	0.19	0.23	1.00	-	1.00	1.00	0.80	0.60	0.80	-	1.00	0.80	0.80	0.60
Al	0.52	0.59	-	0.45	0.45	0.12	1.00	0.80	-	1.00	0.60	0.40	1.00	0.80	-	1.00	1.00	0.60
BH	0.94	0.45	0.59	-	0.23	0.19	0.80	0.60	0.60	-	0.80	0.60	0.80	0.60	0.60	-	0.80	0.80
Idx	0.90	0.59	0.59	0.85	-	0.02	0.80	0.80	0.80	0.80	-	0.20	0.80	0.80	0.80	0.80	-	0.60
Hc	0.00	0.00	0.00	0.01	0.05	-	0.00	0.20	0.40	0.40	0.80	-	0.20	0.80	0.80	0.60	0.80	-

Legend — EA: evolutionary algorithm; PM: price momentum; Al: alpha; BH: buy and hold; Idx: msci total return index; Hc: hill climbing

For each pair of portfolios we run the tests of SD in both ways. Portfolio A is concluded as dominant over portfolio B if (1) the hypothesis that A dominates B is not rejected and (2) the hypothesis that B dominates A is rejected. It can be seen from table III and table IV that there is no clear dominance patten among the five portfolios: Hill Climbing, MSCI Index, Buy and Hold, Price Momentum, and Alpha strategies. However, the EA portfolio is found to dominate the Hill Climbing in all orders, and dominate the other four portfolios in the second and third orders, implying that all risk-averse investors will favor the EA portfolio compared to the others.

One important assumption underlying the KS tests is the independence of the two samples coming from the two return distributions to be compared. In our case, even though the EA procedures do have some links with other portfolio generation rules, the correlation is between the return distributions themselves, rather than between the samples generated. However, we perform an additional SD test, as proposed by [9], which allows for interdependency between the samples tested. The test is basically a Maximal-T test, which compares two return distributions at a fixed number of points only. Conservative p-values (based on the widely applicable conservative critical values) and simulated p-values are reported in table VI.

TABLE V

MAXIMAL T TEST FOR SD

This table reports the p-values of tests for the dominance of portfolio G over portfolio F

Panel 1. 10-point evaluation, simulated p-values

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.05	0.14	0.50	0.05	0.29	-	0.01	0.02	0.15	0.07	0.70	-	0.02	0.02	0.17	0.13	0.82
PM	0.64	-	0.67	0.49	0.27	0.14	0.95	-	0.66	0.71	0.60	0.04	0.99	-	0.71	0.87	0.81	0.09
Al	0.34	0.34	-	0.34	0.34	0.14	0.92	0.52	-	0.65	0.51	0.05	0.98	0.44	-	0.83	0.75	0.13
BH	0.94	0.80	0.63	-	0.44	0.30	0.96	0.05	0.09	-	0.49	0.28	0.81	0.07	0.06	-	0.54	0.53
Idx	0.74	0.42	0.42	0.74	-	0.19	0.89	0.04	0.04	0.46	-	0.20	0.64	0.03	0.02	0.39	-	0.53
Hc	0.00	0.00	0.01	0.01	0.03	-	0.05	0.44	0.37	0.36	0.52	-	0.12	0.74	0.68	0.44	0.34	-

Panel 2. 10-point evaluation, conservative p-values

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.09	0.21	0.42	0.09	0.50	-	0.09	0.11	0.19	0.33	1.00	-	0.13	0.15	0.20	0.53	1.00
PM	0.90	-	0.91	0.77	0.46	0.21	1.00	-	1.00	1.00	1.00	0.21	1.00	-	1.00	1.00	1.00	0.47
Al	0.58	0.58	-	0.58	0.58	0.21	1.00	0.99	-	1.00	0.99	0.23	1.00	0.97	-	1.00	1.00	0.57
BH	1.00	0.99	0.89	-	0.70	0.50	1.00	0.28	0.41	-	0.98	0.81	1.00	0.33	0.31	-	0.99	0.99
Idx	0.95	0.69	0.69	0.95	-	0.32	1.00	0.16	0.21	0.97	-	0.66	1.00	0.19	0.16	0.96	-	0.99
Hc	0.00	0.00	0.01	0.01	0.05	-	0.26	0.98	0.92	0.91	0.99	-	0.53	1.00	1.00	0.97	0.94	-

Panel 3. 5-point evaluation, simulated p-values

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.13	0.12	0.69	0.04	0.21	-	0.01	0.02	0.63	0.06	0.71	-	0.02	0.02	0.72	0.10	0.77
PM	0.63	-	0.50	0.49	0.26	0.21	0.93	-	0.63	0.68	0.56	0.04	0.98	-	0.67	0.83	0.76	0.08
Al	0.49	0.54	-	0.49	0.36	0.12	0.89	0.52	-	0.61	0.48	0.05	0.97	0.45	-	0.77	0.70	0.10
BH	0.84	0.80	0.60	-	0.35	0.21	0.93	0.12	0.19	-	0.45	0.25	0.87	0.05	0.06	-	0.52	0.50
Idx	0.85	0.89	0.95	0.75	-	0.12	0.95	0.06	0.11	0.50	-	0.17	0.72	0.02	0.02	0.39	-	0.51
Hc	0.00	0.00	0.01	0.01	0.05	-	0.26	0.98	0.92	0.91	0.99	-	0.53	1.00	1.00	0.97	0.94	-

TABLE VI

Maximal T test for SD - (continued)

Panel 1. 5-point evaluation, conservative p-values

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.15	0.15	0.67	0.04	0.26	-	0.05	0.07	0.39	0.20	0.99	-	0.07	0.08	0.50	0.30	1.00
PM	0.81	-	0.67	0.66	0.34	0.26	1.00	-	0.97	0.98	0.93	0.14	1.00	-	1.00	1.00	1.00	0.25
Al	0.66	0.74	-	0.66	0.48	0.15	1.00	0.92	-	0.95	0.86	0.16	1.00	0.89	-	1.00	0.99	0.33
BH	0.97	0.95	0.80	-	0.48	0.26	1.00	0.31	0.47	-	0.85	0.62	1.00	0.17	0.21	-	0.93	0.93
Idx	0.97	0.98	1.00	0.91	-	0.15	1.00	0.20	0.29	0.88	-	0.43	0.99	0.10	0.10	0.81	-	0.93
Hc	0.00	0.01	0.01	0.01	0.05	-	0.04	0.41	0.33	0.31	0.48	-	0.09	0.69	0.63	0.45	0.41	-

Legend — EA: evolutionary algorithm; PM: price momentum; Al: alpha; BH: buy and hold;
Idx: msci total return index; Hc: hill climbing

Panel 4. 5-point evaluation, conservative p-values

F	1st order dominance						2nd order dominance						3rd order dominance					
	G portfolio						G portfolio						G portfolio					
	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc	EA	PM	Al	BH	Idx	Hc
EA	-	0.15	0.15	0.67	0.04	0.26	-	0.05	0.07	0.39	0.20	0.99	-	0.07	0.08	0.50	0.30	1.00
PM	0.81	-	0.67	0.66	0.34	0.26	1.00	-	0.97	0.98	0.93	0.14	1.00	-	1.00	1.00	1.00	0.25
Al	0.66	0.74	-	0.66	0.48	0.15	1.00	0.92	-	0.95	0.86	0.16	1.00	0.89	-	1.00	0.99	0.33
BH	0.97	0.95	0.80	-	0.48	0.26	1.00	0.31	0.47	-	0.85	0.62	1.00	0.17	0.21	-	0.93	0.93
Idx	0.97	0.98	1.00	0.91	-	0.15	1.00	0.20	0.29	0.88	-	0.43	0.99	0.10	0.10	0.81	-	0.93
Hc	0.00	0.01	0.01	0.01	0.05	-	0.04	0.41	0.33	0.31	0.48	-	0.09	0.69	0.63	0.45	0.41	-

Legend — EA: evolutionary algorithm; PM: price momentum; Al: alpha; BH: buy and hold;
Idx: msci total return index; Hc: hill climbing

The Maximal-T test results confirm the previous finding that there is no clear dominance pattern amongst the Price Momentum, Alpha, Buy & Hold and Hill Climbing portfolios. However, the tests give some support to the hypothesis that the Price Momentum and Alpha portfolios outperform the MSCI Index in the third order, ie. those investors who are risk averse and have decreasing absolute risk aversion will not choose to invest in the Index portfolio.

Similar to the KS tests, the EA portfolio is still found to be the best performing one. It dominates both the Price Momentum and the Alpha strategies in the second and third orders, and therefore is still the preferred choice for risk-averse investors.

VI. Conclusions and Future Work

This paper has provided a discussion of the framework in which an evolutionary algorithm manages a portfolio of selected stocks chosen using only price and volume information of individual company shares. Using fuzzy logic rules to characterize stocks into various membership sets and then ranking them according to their fit as a buy recommendation, a portfolio is initially constructed and then re-balanced every 20 days with the best buy recommended stocks bought and held. As stocks fall in ranking from the top buy recommendations, they are sold off and replaced with higher ranked stocks. The fuzzy logic rules are designed to optimize towards a specific fitness function. The fitness function describes the ultimate objective of the portfolio. In this EA code the fitness function is a simple return on investment measure. A specific financial penalty function is also incorporated to penalize solutions that select a portfolio of stocks that experiences significant losses. Effectively, it is a penalty for downside risk.

The empirical results from testing the EA on historical data show that it can not only beat traditional, fixed rule-based strategies (such as the price momentum and alpha portfolios) buy also beat the market index. This is shown for the case of MSCI Europe listed stocks spanning a period from 1990 until the end of 2005. Given that we impose both costs to trading and restrictions on how trades can occur, it is a relatively impressive result. This is even more particularly true when considering that only price and volume information is used to generate buy recommendations.

We believe the success of the EA portfolio lies in its ability to adapt its rule-base to new market conditions. This is a significant advantage over fixed rule-base strategies and as such can also successfully pin-point technical trading patterns that allow it to select stocks that are likely to outperform.

In the future the system will incorporate several new features to allow it to evolve better rules based on specific market conditions. This includes, but not restricted to:
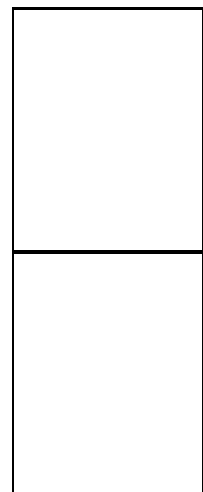
1. The level of volatility within the market
2. Whether it is a bull or bear market
3. Refined techniques to minimize downside risk within the portfolio via changes to the fitness function and design of portfolio construction using higher moment analysis

In addition, the system will be tested on other markets and with shorter term data (intraday), as well as applied to other derrivative classes. Finally, the system will incorporate additional financial accounting and economic information to increase the information content of the rules found. This will also allow for the incorporation of several multi-factor risk model approaches to further increase the asset selection performance of the system.

References

[1] F. Allen and R. Karjalainen. Using genetic algorithms to find technical trading rules. *Journal of Financial Economics*, 51:245–271, 1999.

[2] Panayiotis C. Andreou, Chris Charalambous, and Spiros H. Martzoukos. Robust artificial neural networks for pricing of european options. *Comput. Econ.*, 27(2-3):329–351, 2006.

[3] Amir Atiya. Bankruptcy prediction for credit risk using neural networks: a survey and new results. *IEEE Transactions on Neural Networks*, 12(4):929–935, 2001.

[4] Antonia Azzini and Andrea G.B. Tettamanzi. A neural evolutionary approach to financial modeling. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1605–1612, New York, NY, USA, 2006. ACM Press.

[5] Garry F. Barrett and Stephen G. Donald. Consistent Tests for Stochastic Dominance. *Econometrica*, 71(1):71–104, 2003.

[6] A. Bera and C. Jarque. Efficient tests for normality, heteroscedasticity, and serial independence of regression residuals. *Economics Letters*, 6:255–259, 1980.

[7] Anthony Brabazon and Michael O'Neill. *Biologically Inspired Algorithms for Financial Modelling (Natural Computing Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[8] W. Brock, Lakonishok, J., and B. LeBaron. Simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance*, 47:1731–1764, 1992.

[9] Russell Davidson and Jean-Yves Duclos. Statistical Inference for Stochastic Dominance and for the Measurement of Poverty and Inequality. *Econometrica*, 68(6):1435–1464, 2000.

[10] E. F. Fama. Components of Investment Performance. *Journal of Finance*, 27(3):551–567, 1972.

[11] E.F. Fama and K. French. Common risk factors in the returns of stocks and bonds. *Journal of Financial Economics*, 33(1):3–56, 1993.

[12] C. Fyfe, J. Marney, and H. Tarbert. Technical analysis versus market efficiency a genetic programming approach. *Applied Financial Economics*, 9:183–191, 1999.

[13] C. Lee Giles, Steve Lawrence, and Ah Chung Tsoi. Noisy time series prediction using recurrent neural networks and grammatical inference. *Mach. Learn.*, 44(1-2):161–183, 2001.

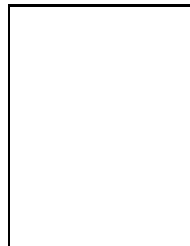[14] T. H. Goodwin. The Information Ratio. *Financial Analysts Journal*, 54(4):34–43, 1998.

[15] R.C. Grinold and R.N.Kahn. *Active Portfolio Management.* McGraw-Hill: New York, 2 edition, 2000.

[16] P. Hsu and C. Kuan. Reexamining the profitability of technical analysis with data snooping checks. *Journal of Financial Econometrics*, 3(4):606–628, 2005.

[17] N. Jegadeesh and S. Titman. Profitability of Momentum Strategies: An Evaluation of Alternative Explanations. *Journal of Finance*, 56(2):699–720, 2001.

[18] M.C. Jensen. The performance of mutual funds in the period 1945-1964. *Journal of Finance*, 23(2):389–416, 1968.

[19] H. Leland. Beyond Mean-Variance: Performance Measurement in a Nonsymmetrical World. *Financial Analysts Journal*, pages 27–36, 1999.

[20] A.W. Lo, Mamaysky, H., and J. Wang. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *Journal of Finance*, 55(4):1705–1765, 2000.

[21] C. Neely, Weller, P., and R. Dittmar. Is technical anlaysis in the foreign exchange market profitable? a genetic programming approach. *Journal of Financial and Quantitative Analysis*, 32(4):405–426, 1997.

[22] Jangmin O, Jongwoo Lee, Jae Won Lee, and Byoung-Tak Zhang. Dynamic asset allocation for stock trading optimized by evolutionary computation. *IEICE - Trans. Inf. Syst.*, E88-D(6):1217–1223, 2005.

[23] Jean-Yves Potvin, Patrick Soriano, and Maxime Valle. Generating trading rules on the stock markets with genetic programming. *Comput. Oper. Res.*, 31(7):1033–1047, 2004.

[24] W.F. Sharpe. Mutual fund performance. *Journal of Business*, 39(1):119–138, 1966.

[25] Harish Subramanian, Subramanian Ramamoorthy, Peter Stone, and Benjamin J. Kuipers. Designing safe, profitable automated stock trading agents using evolutionary algorithms. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1777–1784, New York, NY, USA, 2006. ACM Press.

[26] Nils Svangard, Stefan Lloyd, Peter Nordin, and Clas Wihlborg. Evolving short-term trading strategies using genetic programming. In David B. Fogel, Mohamed A. El-Sharkawi, Xin Yao, Garry Greenwood, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors, *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, pages 2006–2010. IEEE Press, 2002.

[27] Shaun-Inn Wu and Ruey-Pyng Lu. Combining artificial neural networks and statistics for stock-market forecasting. In *CSC '93: Proceedings of the 1993 ACM conference on Computer science*, pages 257–264, New York, NY, USA, 1993. ACM Press.

Adam Ghandar is currently working towards the Ph.D degree in the school of Computer Science at the University of Adelaide — Australia. His main research interests currently include Evolutionary Computation and Financial Modelling.
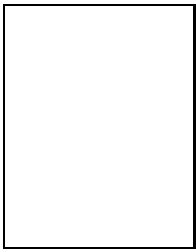
Zbigniew Michalewicz is Professor of Computer Science at the University of Adelaide in Australia. He completed his Masters degree at Technical University of Warsaw in 1974 and he received Ph.D. degree from the Institute of Computer Science, Polish Academy of Sciences, in 1981. He also holds a Doctor of Science degree in Computer Science from the Polish Academy of Science. Zbigniew Michalewicz also holds Professor positions at the Institute of Computer Science, Polish Academy of Sciences, the Polish-Japanese Institute of Information Technology, and the State Key Laboratory of Software Engineering of Wuhan University, China. He is also associated with the Structural Complexity Laboratory at Seoul National University, South Korea. Zbigniew Michalewicz has published over 200 articles and 15 books on subjects related to heuristic methods. These include the most recent *Adaptive Business Intelligence*, and the scientific bestseller *How to Solve It: Modern Heuristics*. Other books include a monograph *Genetic Algorithms + Data Structures = Evolution Programs* (3 editions, a few translations), *Handbook of Evolutionary Computation*, and quite recent *Winning Credibility: A guide for building a business from rags to riches*, where he described his business experiences over the last years.
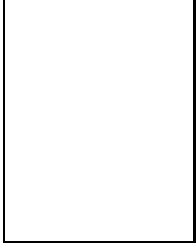
Martin Schmidt is an expert in the commercialization of software solutions that are based on evolutionary algorithms, simulated annealing, large-scale simulations, artificial neural networks, fuzzy logic, and Hybrid Systems. His scientific expertise in Artificial Intelligence methods is centered around real-life constraint optimization, multi-objective optimization, co-evolutionary systems, simulations, as well as classification and prediction systems. Before co-founding SolveIT Software, Martin was a Researcher at the Institute of Political Science at the University of Aarhus, where he used a supercomputer to develop large-scale simulation systems. Later, he worked for various software companies in Europe and the United States of America, where he managed the development and deployment of prediction and optimization solutions for clients such as General Motors, Honda, Ford Motor Company, and Bank of America. Martin is also an Adjunct at the University of Adelaide at the School of Computer Science where he supervises Ph.D. students. In addition to being the author of 25+ scientific research publications and several newspaper articles, Martin also possesses expert knowledge of object oriented development in Java and C++, and data mining/modelling. Martin Schmidt holds a Master of Science degree in Computer Science from the University of Aarhus, Denmark, and a Ph.D. degree in Computer Science from the Polish Academy of Sciences in Warsaw, Poland. He coauthored a recent book *Adaptive Business Intelligence* (Springer 2007).

Thuy-Duong Tô is a Lecturer in Finance at the University of Adelaide and holds a PhD in Finance and Economics from the University of Technology, Sydney. She has published papers in well respected international journals. Her research interests include interest rate modelling, credit risk modelling, market microstructure, issues in risk management and financial econometrics.

Ralf Zurbruegg is a Professor of Finance at the University of Adelaide. He holds an undergraduate degree from the London School of Economics and a Master's degree and PhD from the University of Manchester. Ralf has published over 70 articles, with the majority appearing in leading finance journals. His primary research interests lie in risk management issues, particularly as they relate to insurance, derivatives pricing and securitised real estate assets. He has consulted for numerous financial institutions over the years on risk management topics and is also the co-editor of the International Journal of Managerial Finance.