Soft Comput (2011) 15:37–50
DOI 10.1007/s00500-009-0511-4

FOCUS

Evolving robust GP solutions for hedge fund stock selection in emerging markets

Wei Yan · Christopher D. Clack

Published online: 3 February 2010

Abstract Stock selection for hedge fund portfolios is a challenging problem for Genetic Programming (GP) because the markets (the environment in which the GP solution must survive) are dynamic, unpredictable and unforgiving. How can GP be improved so that solutions are produced that are *robust* to non-trivial changes in the environment? We explore two new approaches. The first approach uses subsets of extreme environments during training and the second approach uses a voting committee of GP individuals with differing phenotypic behaviour.

1 Introduction

In May 1994, following an increase in US short-term interest rates, tumbling bond prices, and a knock-on effect on international currencies, the financial speculator George Soros lost $650,000,000 in just 2 days (Lowenstein 2002). On 17th August 1998, Russia defaulted on its debts; 3 days later the financial markets across the world collapsed and in just 1 day, the hedge fund Long-Term Capital Management lost $553,000,000 (Lowenstein 2002).

The financial markets are highly dynamic, unpredictable and unforgiving. If Genetic Programming (GP) is used to evolve a solution to a financial trading or investment problem it must be robust to these time-varying disturbances in the markets.

It follows from the above examples that by "robust" we do not mean insensitivity of the fitness of an individual to perturbations resulting from the genetic operators (genotypic robustness, Soule 2003; Soule et al. 2002); although this form of "robustness" favours broad plateaus to sharp peaks in the search space, it does not give much indication about how the best-of-run individual will perform when the fitness function itself changes (i.e. the surface of the search space fluctuates). Relevant, though insufficient, other previous definitions of robustness include the insensitivity of the fitness of an individual to small fluctuations in an individual's parameters (sometimes known as phenotypic robustness or generalizability, Branke 1998; Tsutsui and Ghosh 1997) and the insensitivity to a noisy fitness function (Fitzpatrick and Grefenstette 1988; Hammel and Bäck 1994; Miller and Goldberg 1996). The problem with these latter two definitions is that all known work in the area assumes that the fluctuations or noise are drawn from a known and time-invariant distribution (typically uniform or Gaussian), and are small. By contrast, the financial markets undergo large, abrupt and time-varying changes.

Aragón and Esquivel (2004) model a dynamic environment as a sequence of fitness functions, each defined by changes to the previous. The model uses occasional macromutation for radical genotype shake-up ("recrudescence"), and assumes that all possible changes to the current fitness function are enumerable (and finite, and, in practice, few). It assumes that we can evolve continuously and wait several generations before adaptation to the new fitness function is achieved. Unfortunately, in the real world we cannot wait for the evolutionary system to learn from the new environment!

W. Yan
Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK
e-mail: W.Yan@cs.ucl.ac.uk

C. D. Clack (✉)
Financial Computing, Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK
e-mail: clack@cs.ucl.ac.uk

Our two approaches are substantially different to the prior work in this area.

1. The first approach is based on the assumption that examples exist of greatly differing extreme environments. If that assumption holds, then we propose to present these examples to the genetic population during training in order to select individuals that perform well in a variety of extreme training environments. We call this approach "Multiple Scenarios Training".
2. The second approach is based on the use of a "committee" structure whereby a small (odd) number of trained GP individuals offer solutions as votes, and the majority vote wins. If the individuals exhibit widely differing phenotypic behaviour, yet all have good fitness on the training data, we hope that this committee structure will be robust to large changes in the environment. We call this approach "Committee Voting".

The obvious research question these two approaches entail is, as a result of multiple-scenario exposure or committee based decision making, whether trained individuals are more likely to be robust to large disturbances in the environment?[1] There are three ways we plan to measure this kind of robustness in the context of our finance application:

1. when exposed to an out-of-sample volatile validation dataset, a more robust solution will have a lower standard deviation of returns, while the returns themselves do not decrease; or
2. when exposed to an out-of-sample volatile validation dataset, a more robust solution will have higher returns while the standard deviation does not increase; or
3. when exposed to an out-of-sample volatile validation dataset, the mean return per unit of risk of a more robust solution will not significantly reduce from that measured during training.

2 Related work

2.1 Robustness

Robustness for a biological system is a property to allow a system to maintain its functionality despite internal and external perturbations (Kitamo 2001; Wagner 2005).

Robustness is a very broad theme and it is impossible to capture all its aspects by means of a single definition. Robustness is a ubiquitously observed property of biological systems. It is considered to be a fundamental feature of complex evolvable systems (Kitamo 2001).

The definition of robustness in evolutionary systems varies from author to author, but in broad terms, it can be divided into two categories:

1. Robust to internal changes (genotypic robustness): Robustness as the resistance to changes from variation operators such as crossover and mutation. Soule (Soule 2003; Soule et al. 2002) observes that the most outstanding evidence of pressure towards this type of robustness is the phenomenon of code growth (or bloat) in GP. Code bloat is a rapid increase in code size that does not result in fitness improvements. It is proposed that GP trees grow this extra code ("introns") as a means of protecting the useful code within good solutions. By adding introns the useful code is less likely to be affected by crossover or other similar operators. The robustness in this sense can be drawn parallel to gene redundancy in biosystems and to the existence of "neutral networks"[2] which enable a population to maintain a dominant phenotype required for adaptation despite random genotype changes during the evolution (Huynen et al. 1996).
2. Robust to external changes (phenotypic robustness):
   (a) Robustness as the generalisation ability of the programs evolved using GP (Bersano-Begey and Daida 1997; Kuscu 1998, 2000; Moore and Garcia 1997; Panait and Luke 2003). The concept of generalisation is originated from connectionist or symbolic learning research, and it is defined as the desired successful performance of the solution when it is applied to an environment similar to the one it was evolved for. In the context of evolutionary systems, the ability to generalise is defined as "the predictive accuracy of the learner in mapping unseen input cases to outputs with a satisfactory degree of correction" (Kushchu 2002). In this respect, robustness is in line with though opposite to the definition of overfitting. Overfitting happens when the computational effort spent on obtaining a more precise fit of the sample results in an increased error on other data.
   (b) Robustness as the ability to cope with non-constant noise (Jordaan et al. 2004; Nissen and Propach 1998): Practical optimisation problems often require the evaluation of solutions through experimentation, stochastic simulation, sampling, or even interaction with the user. Thus, most

[1] An alternative approach is to look for an *adaptive* solution, i.e. one that detects changes in the environment and responds by modifying its internal structure and the way that it operates. However, a similar question arises: in an unforgiving environment, would it have *time* to adapt and survive without prior exposure to extreme environments?

[2] Connected networks of RNA sequences with identical structure.

practical problems involve noise. Jordanne et al. (2004) investigated this particular aspect of robustness when noise is added to the deterministic objective function values.

(c) Robustness as the sensitivity of performance quality in the presence of external environmental perturbations. For example, Hermann (1999) defines robust solutions as the one that has the best worst-case performance.
This aspect of robustness is the most consistent with phenotypic robustness in nature. Although a biological system exhibits robustness in terms of genes, structures, etc. from an evolution point of view, ultimately robustness of only one feature matters: fitness is the ability to survive and reproduce (which in evolutionary systems means the performance quality of a solution).

(d) Robustness as the ability for self-repair when subject to severe phenotypic damage (Miller 2004; Bowers 2005). This behaviour is reminiscent of autonomous regeneration of the pond organism hydra, which can reform itself when its cells are dissociated and then re-aggregated in a centrifuge (Gierer et al. 1972).

2.2 Structured training sets for robustness

The way in which training data are presented to the population is central to our work yet has received little prior attention. The techniques that have been proposed are twofold: the use of random noise in the training data; and the use of randomly generated environments (fitness cases). However, the experimental methodology of the prior work is not entirely helpful in giving confidence that robust solutions have actually been evolved.

Ito et al. (1996) and Reynolds (1992) use noise and modify initial conditions in order to promote robustness of the programs produced by GP—robustness both to changes in the initial conditions and to changes in the environmental stimuli. The use of noise can be helpful in reducing the brittleness of programs and increasing the likelihood of robustness (Reynolds 1992). In Ito et al. (1996) both changing initial coordinates and the direction of the robot, together with the introduction of noisy sensors and actuators, are tried to produce robust programs for robot behaviour. Separate training and testing sets are used, but there is no discussion of how training and test cases should be chosen. Furthermore, training and testing comparisons are done on a generation basis and the measure of robustness in testing on a different environment after the training (i.e., after evolution has stopped) is not reported. The results of the experiments do not make it clear whether a robust behaviour has been reached and if so, how it is reached.

Haynes et al. (1995) use GP to evolve an agent that can survive in a hostile environment. Rather than using a fixed environment for a particular run, a new environment is generated randomly at the end of each generation. In this way, it is hoped that the agent can handle "any" environment. Since the agent does not seem to be tested in a new environment after the evolution has stopped, the nature and the degree of robustness to new environments (that might have been obtained by variable training environment during the evolution) remains unexplained.

A good example of experiments attempting to reduce the brittleness of individuals generated by GP is presented in Moore and Garcia (1997). The system in this paper evolves optimised manoeuvres for a pursuer/evader problem. The results of this study suggest that use of a fixed set of training cases may result in brittle solutions due to the fact that such fixed fitness case may not be representative of possible situations that the agent may encounter. It is shown that use of randomly generated fitness cases at the end of each generation can reduce the brittleness of the solution when the solution is tested against a set of large representative situations after the evolution. However, a proper selection method for training and testing cases is not provided.

Rosca (1996) addresses issues of size and generality in GP; however, the degree of overlapping between training instances and the testing instances does not seem to be explicitly controlled. In such a case, an objective and direct comparison using a common basis between training and testing may be difficult.

2.3 Committees

The use of a committee or "voting pool" is well known in the area of machine-learning (ML) classifier systems. In particular, a multiple-classifier system (MCS) (Kittler and Roli 2001) would utilise a number of different classifiers that run simultaneously and their results combined in a second stage or master classifier. The master prediction algorithm can either be another classification algorithm or a voting committee. The MCS may utilise classifiers that each provide a confidence estimate together with their classification—the committee may then choose a subset of results to be used for voting (Stefano et al. 2003; Ranawana and Palade 2006).

Where possible, complementary classifiers are chosen, whose errors are partially or fully uncorrelated. However, this is not always possible and so a second approach is to search for combinations of classifiers whose performance lies outside the ROC (Egan 1975) of the constituent classifiers. There is no guarantee of improvement with the

MCS approach, but Buxton et al. (2001) have demonstrated impressive results using a GP to identify an optimal second-stage classifier. Similarly, Herbster has developed successful master prediction algorithms that can optimally combine sub-prediction algorithms (Herbster and Warmuth 1998, 2001; Herbster 2001).

Zhang and Joung (2000) have presented a mechanism for determining the constituents of a committee for GP classification problems. Ensemble systems are "learning algorithms that construct a set of classifiers and then classify new data points by taking a weighted vote of their predictions" (Dietterich 2000). Dietterich (2000) provides an informative review of these systems; see also Brown et al. (2005), Liu et al. (2000) and Imamura et al. (2003).

Whilst the concept of a committee structure with majority voting has been established for many years in the research area of ML classifiers, it is rarely reported in the implementation of optimisers. Soule (1999, 2000) is an exception; he has investigated the evolution of co-operating teams that vote on solutions, but the proposed technique is complex, and it is not clear whether this work could be extended to problems in financial time-series analysis. See also Zhu and Chipman (2006).

Several researchers have specifically investigated the advantages of robustness and the minimisation of solution risk that accrue from using a committee of solutions instead of a single model in a changing environment. The advantages that have been previously reported are

1. First since the final decision is a combination of a number of problem solvers, one obtains a more consistent estimate of the output. The performance of the system is more robust as the outcome does not depend on the accuracy of one single model anymore, but on the outcome of several models (Imamura et al. 2003; Soule 1999).
2. Second, the spread or variance of the different outcomes can be used to derive a measure of confidence, called model disagreement indicator. A small difference in behaviour gives the users more certainty about the decision (Zhang and Joung 2000).
3. Another advantage of a committee is that it enables redundancy. If the committee consists of models that behave differently on different environmental inputs, there will be at least one model available for a particular type of environment (Dietterich 2000; Brown et al. 2005).

3 Description of the algorithm

3.1 Multiple scenarios approach

We are concerned with not only the performance or fitness of the GP-evolved solutions but also the performance volatility of the GP-evolved solutions across a range of environment dynamics; for example, in a scenario where market prices are rising ("bull market"), a scenario where market prices are falling ("bear market"), and a scenario where market prices are fluctuating with large amplitude ("volatile market").

We therefore consider the training data to be a set of fitness cases—a vector of environments—representing a possible range of different environments and then adjust the fitness with its perceived volatility to obtain a whole picture of an individual's performance. Let S be the training environment vector and s_n be the nth possible type of environment, which we call a "scenario"; then $S = \{s_1 \ldots, s_n\}$. We also hold a separate out-of-sample validation vector V.

We consider three ways in which the GP population should be exposed to these scenarios:

1. "Standard GP" (SGP): use the entire vector S, treated as a single unit, throughout all generations;
2. "Multiple-scenario Evaluation in the Last Generation" (MELG): use a variant of the three-dataset methodology (Panait and Luke 2003; Gagné et al. 2006), where the entire vector S (treated as a single unit) is used for $n - 1$ generations, and in the final generation individuals are tested against a subset of environments $\{s_i\}$ drawn from S. The "best-of-run" individual used in the validation on set V is that which has, in the final generation, the highest Volatility-Adjusted Fitness (see below);
3. "Multiple-scenario Evolution" (MEVO): in each generation, use a subset of environments $\{s_i\}$ drawn from S, and ascribe to each individual a Volatility-Adjusted Fitness (see below). Evolution proceeds as normal on the basis of this adjusted fitness measure. The "best-of-run" individual from the final generation is used in the out-of-sample validation on set V.

3.1.1 *Volatility-adjusted fitness*

We have previously introduced S as the training environments vector and s_n as the nth possible scenario; hence $S = \{s_1, \ldots, s_n\}$. Now let $M = \{m_1, \ldots, m_p\}$; $m_j \in S$ be a subset of S that is used for fitness evaluation.

Let I_i be an individual in the population, and $f_{I_i}^{m_j}$ be the fitness of individual I_i when evaluated on scenario m_j. Then F_{I_i} is the "fitness vector" of I_i when evaluated on a subset of scenarios, given by $F_{I_i} = \{f_{I_i}^{m_1}, \ldots, f_{I_i}^{m_p}\}$.

We use standard deviation to calculate the volatility of the fitness (performance) of the individual across this range of scenarios:

$$\sigma_{I_i} = \sqrt{\frac{1}{p}\sum_{j=1}^{p}(f_{I_i}^{m_j} - \overline{F_{I_i}})^2} \quad (1)$$

where: $\overline{F_{I_i}}$ = mean of F_{I_i}, given by $\frac{1}{p}\sum_{j=1}^{p} f_{I_i}^{m_j}$

The "Volatility-Adjusted Fitness" (VaF) of an individual is now defined as the mean fitness divided by volatility:

$$\mathrm{Va}F_{I_i} = \frac{\overline{F_{I_i}}}{\sigma_{I_i}} \quad (2)$$

3.1.2 MELG

For MELG we use a variation of the three-dataset methodology as outlined in (Panait and Luke 2003; Gagné et al. 2006). In our version of this methodology, the *training set* is used to evaluate the fitnesses of individuals in $n-1$ generations; elitism ensures that the best-of-generation individuals survive to the last generation, and the individuals in the last (nth) generation are tested against a different *in-sample volatility set*; a best-of-run individual is selected and its quality is assessed using yet another different *out-of-sample validation set*.

Note that a possible drawback of this methodology is that, where data samples are limited, either the training set or the out-of-sample validation set must be smaller than it would be in a two-dataset methodology. However, in our variation of the methodology the *in-sample volatility set* is a vector of subsets of the initial *training set*. We also choose to set the fitness vector to be $F_{I_i} = \{f_{I_i}^{m_0}, f_{I_i}^{m_1}, \ldots, f_{I_i}^{m_p}\}$, where $f_{I_i}^{m_0}$ is the fitness of the individual previously calculated in the $n-1th$ generation—thus, the fitness vector contains information about fitness on the whole training set treated as a single unit, as well as fitness on each of the scenarios.

Our methodology permits a more direct comparison with SGP, since we know that both SGP and MELG have been given identical training data—what is different is the way in which those data are presented to the population.

3.1.3 MEVO

The MEVO algorithm differs from MELG in that it uses a two-dataset method: the *in-sample volatility set* used in MELG is used not only in the final generation, but in every generation. The second dataset is therefore the out-of-sample validation set V.

Thus, evolutionary selection is based on the volatility-adjusted fitness $\mathrm{Va}F_{I_i}$, which is calculated from $F_{I_i} = \{f_{I_n}^{m_1}, \ldots, f_{I_n}^{m_n}\}$ (see above). F_{I_i} can be thought of as an "intermediate" fitness vector for each individual, and $\mathrm{Va}F_{I_i}$ is the "real" fitness of an individual. Note that MEVO is *not* exposed to the entire training set ($f_{I_n}^{m_0}$); we only expose it to the extreme scenarios. This might be thought to put MEVO at a disadvantage because it does not have access to as much information as MELG or SGP, but in early trials we discovered that using the entire training set as another scenario led to poor results.

3.2 Committee voting approach

Our stock-picking problem is an optimiser, not a classifier—we evolve a factor model (an equation) that provides a real-number estimate of attractiveness of a stock, with no hard threshold to indicate whether to buy or sell. That equation is then used by an investment simulator to make dynamic buy/sell decisions based on an assessment of the optimum tactics given the relative attractiveness of all available stocks.

We are concerned with not only the performance or fitness of the GP-evolved solutions but also the performance volatility of the GP-evolved solutions across a range of environment dynamics. For example, where market prices are rising ("bull market"), where market prices are falling ("bear market"), and where market prices are fluctuating with large amplitude ("volatile market").

We therefore identify individuals with widely differing behaviour—one that performs well in a bull market, and one that performs well in a bear market, etc. We assume that these individuals are the result of entirely separate GP evolutions using different training data. These individuals are then used at a committee stage in a majority voting algorithm.

The committee is implemented as part of our investment simulator. The simulator is used both during GP evolution (it is called by the fitness function) and during validation, but the committee is only used during validation.

4 Hedge fund simulation

To test the two new algorithms, we simulate a long/short market-neutral hedge fund of Malaysian equities. We choose the Malaysian market because it (in common with other emerging markets) is particularly volatile. The GP system evolves a non-linear equation that uses market data to determine whether a single stock should be selected to buy, or to sell.

4.1 System overview

Our test system comprises a GP system coupled with an investment simulator. The coupling between the two is the fitness function—the investment simulator is called each time the GP system needs to determine the fitness of an individual, at which point the individual is used to control the simulation of an hedge fund of Malaysian stocks. The

simulator is applied to training data giving monthly prices and other factors. Monthly returns on investment are calculated, and at the end of each simulated year the Sharpe ratio (Sharpe 1994) is calculated.

4.1.1 Fitness

The fitness f for an individual is the Sharpe ratio (Sharpe 1994), given by the following Equation 3:

$$\text{Sharpe ratio} = \frac{\overline{x_i} - \text{RFR}_\text{i}}{\sigma_i} \quad (3)$$

In Equation 3, $\overline{x_i}$ is the average monthly return on investment (ROI) over the sub-period i, σ_i is the standard deviation of monthly ROIs over the sub-period i, and $\text{RF}R_i$ is the average monthly Risk Free Rate for sub-period i. We set $\text{RF}R_i$ to 0.003 for all i (equivalent to 4% per annum).

Note that we have chosen *not* to use a multiple-objective approach to fitness evaluation. At an early stage we experimented with using two objectives (high ROI and low volatility) but the system performed poorly. In financial investment, ROI and volatility are very closely linked (they are not properly independent objectives); the result was that the non-dominated set was very small and this adversely affected evolution, causing the system to converge on a local optimum with poorer performance than the solution found using the Sharpe ratio as a single objective.

4.2 The investment simulator

We simulate a market-neutral long/short hedge fund of Malaysian equities. The fund focuses on a basket of 33 Malaysian stocks, which it can buy ("go long") or sell (even if it doesn't own any—"go short"). Since all 33 stocks are quite well correlated, the market-neutral strategy simply entails buying the profitable stocks and selling (short if necessary) those stocks that are performing poorly.[3]

The training data are monthly prices (and other technical and fundamental data) over a period of 71 months. All trading occurs at the beginning of each month and the resulting stock mix is held for the duration of the month. At the beginning of each month, the simulator uses the individual provided by the GP system as a stock selection model that quantitatively measures the attractiveness of each stock; this model is a non-linear combination of technical and fundamental factors to predict the return expectation for each stock over a 4-week forward horizon.

For each month, we apply the stock selection model to the current month data—this is a table per stock with 19 factors (see Table 1) and 7,680 data points. A return prediction is assigned to each stock.

The stocks are grouped into four market sectors and within each sector all stocks are ranked according to the expected return. The portfolio simulator then makes the following fund management decisions:

- The long/short portfolio is both dollar neutral and sector neutral. Thus, at all times, 24 stocks are maintained in the portfolio with 12 long positions and 12 short positions equally distributed across all the sectors. According to the ranking, the top three stocks in each sector become the top fractile and the bottom three become the bottom fractile. The top fractile of each sector and the bottom fractile of each sector are chosen to hold long positions and short positions, respectively, in the portfolio.
- Sectors are equally weighted and each stock is given equal weight in the portfolio. Thus, each position accounts for approximately 4% of total portfolio value.
- Contract for differences (CFDs) are used instead of conventional shares to trade on stocks. We assume 20% notional trading requirement (margin), 0.25% trading commission, and 5% financing rate.

At the end of each month, all of the positions held in the portfolio are closed and the profit or loss of the portfolio during the month is calculated. At the beginning of the next monthly trading cycle, the simulator updates the expected return based on the new "current" data and a new desired long/short portfolio is formed.

5 Experiment

5.1 Multiple scenarios approach

Our primary research question for the multiple scenarios approach is, "are the best-of-run individuals from the two new systems more robust than the best-of-run individual from SGP when exposed to a volatile and previously unseen environment?"

Our experiment compares the performance of all three systems: SGP, MELG and MEVO. The basic GP parameter settings for the three systems are identical, as given in Table 2.

5.1.1 Data

All three systems use an Investment Simulator that has an investment universe of 33 Malaysian stocks. The training data for all three systems comprise time-series financial

[3] A *contrarian* strategy might do the opposite—sell the high stocks and buy the low stocks, on the expectation that mean-reversion will occur and the high stocks will fall while the low stocks will rise.

Table 1 Description of factors

1.	Closing stock price on 1st day of a month
2.	Closing stock price on last day of a month
3.	12-month MACD: moving average convergence and divergence
4.	Capitalisation = (number of shares) × (stock price (c))
5.	ROE = (net income) ÷ (shareholders' equity)
6.	ROE (this year) − ROE (prev. year)
7.	((total debt) ÷ (common equity)) × 100
8.	(sum of last 12-months of cash dividends) ÷ (stock price (c))
9.	(last 6 months' trailing earnings per share - prev. last 6 trailing earning per share) ÷ (absolute prev. last 6 months trailing earning per share)
10.	As above (replace 6 with 12)
11.	As above (replace 12 with 36)
12.	The rate of change in the reported last 12-month earnings per share over the 3 year time interval terminating on the date of the last interim period for which earnings were announced
13.	(last 12-month trailing earnings per share) ÷ (closing market price)
14.	(historical book value per share) ÷ (closing monthly market price)
15.	(cash earnings per share) ÷ (closing market price)
16.	One month dollar price change
17.	One year dollar price change
18.	(current year's net sales or revenue − previous year's net sales or revenue) ÷ (previous year's net sales or revenue)
19.	(last 12-month trailing earnings per share − last 12-month dividend per share) ÷ (last year's book value per share)

Table 2 GP parameter settings

Population size (N)	1,000
Method of generation	Ramped half and half
Function set	$\{+, -, *, /, \text{Exp}\}$
Terminal set	18 Firm-specific factors
Selection scheme	Fitness proportionate selection
Criterion of fitness	Monthly Sharpe ratio
Trees generated by elitism	10 (1%)
Trees generated by crossover	950 (95%)
Trees generated by mutation	40 (4%)
Termination criterion	100-generation evolution
Max. depth initial generation	6

data for the 33 stocks taken from the period 31st January 1999 to 31st December 2004.

SGP and MELG use a training dataset of financial time-series data taken from the period 31st January 1999 to 31st December 2004 (71 months).

For MELG (last generation only) and MEVO, the following three scenarios were chosen:

1. Bull market: 31/05/2003 to 31/12/2004 (19 months);
2. Bear market: 31/01/2000 to 31/05/2001 (16 months);
3. Volatile market: 31/01/1999 to 31/03/2000 (14 months).

Figure 1 shows the overall market index for Malaysian stocks, and a non-weighted portfolio index of the 33 investment stocks, for the overall period under study. It also indicates the three scenario periods (bull, bear and volatile) and the validation period. The market and portfolio indices both show considerable volatility—the portfolio index (constructed from the stocks in which our simulator invests) is slightly more volatile than the overall market index, and so beneficial effects displayed by our GP system cannot be due solely to "cherry-picking" the least volatile stocks.

5.1.2 Out-of-sample validation

All three systems are validated on a previously unseen "out of sample" dataset, comprising time-series financial data for the 33 stocks taken from the period 31st July 1997 to 31st December 1998. During this period, the Malaysia stock market suffered great volatility including both the highest and lowest monthly returns in the entire period under study. From May 1998 to October 1998, the stock index lost more than 42%. Then from November 1998 onwards, there was a remarkable performance from the market index, rising 23.3% in November.

We have deliberately chosen this period as a real test of robustness of individuals in a dynamic and hostile environment. One expects episodes of extreme volatility in world stock markets, and in emerging markets in particular. A successful hedge fund stock selection model must be

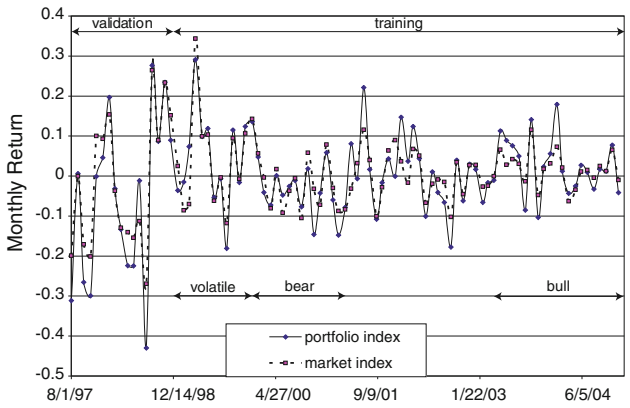

Fig. 1 Market and portfolio indices (fractional monthly returns, 31st July 1997 to 31st December 2004), scenarios and validation period

robust—be able to perform in both (extreme) up and down markets.

For the out-of-sample validation, we performed 25 complete training runs (each run being seeded with a different random number) of each of the three systems (SGP, MELG, and MEVO), and the best-of-run individual was selected from the final generation of each run.

The selected individuals were then validated on the previously unseen data; the results of the 25 runs are discussed in the following section.

5.2 Committee voting approach

Our primary research question for the committee voting approach is: *"does a voting system provide more robust results than the best-of-run individual from SGP when exposed to a volatile and previously unseen environment?"*

Our experiment compares the performance of an SGP individual with the Voting system comprising three best-of-run individuals derived from three GP evolutions with different training datasets. The basic GP parameter settings for the GP systems are identical, as given in Table 2.

5.2.1 Data

All systems use an Investment Simulator that has an investment universe of 33 Malaysian stocks. The training data for all systems comprises time-series financial data for the 33 stocks taken from the period 31st January 1999 to 31st December 2004.

SGP uses a training dataset of financial time-series data taken from the period 31st January 1999 to 31st December 2004 (71 months).

For the three special-case evolutions, the following three market contexts were chosen:

1. Bull market: 31/05/2003 to 31/12/2004 (19 months);
2. Bear market: 31/01/2000 to 31/05/2001 (16 months);
3. Volatile market: 31/01/1999 to 31/03/2000 (14 months).

5.2.2 The committee

During validation, the Voting investment simulator is augmented with a committee structure containing a team of three individuals.

In investment portfolio optimisation we trade monthly and aim to pick those stocks that will perform well *regardless* of whether the market in the following month will be bull, bear or volatile. Thus, we do not follow the otherwise obvious strategy of detecting the current market conditions and using an individual that has been trained only on that one market condition. Rather, the voting team comprises the best-of-run individual chosen from each of the final populations of three GP systems each of which has been trained on only one market condition—i.e. the three systems have undergone separate training with predefined, distinctively different training datasets representing the three market environments "bull", "bear" and "volatile".

Our expectation is that the behavioural correlation between team members is low. Each team member generates a predicted return for the next 30 days, for each stock in the portfolio, and an explicit mechanism is used to combine the members' solutions.

There are two possible combining mechanisms, either *Averaging* or *Voting* as shown in Fig. 2.

Averaging: The first mechanism averages the team members output. This results in a mean predicted return for the next 30 days for each stock. The stocks are then ranked in order of this mean predicted return; the top half is selected for buying and the bottom half is selected for selling.

Voting: With the second mechanism each team member uses its predicted returns to generate its own ranking of all the stocks; this is then converted into a buy decision for those stocks in the top half of the ranking and a sell decision for those stocks in the bottom half.

After the buy/sell recommendations have been calculated for all team members and for all stocks, a majority voting method is applied to each stock and a final buy or sell decision is derived for that stock. With majority voting, if a stock has more buy recommendations than sell recommendations, it will be bought: otherwise it is sold.

In our experiment we will consistently use the Voting mechanism.

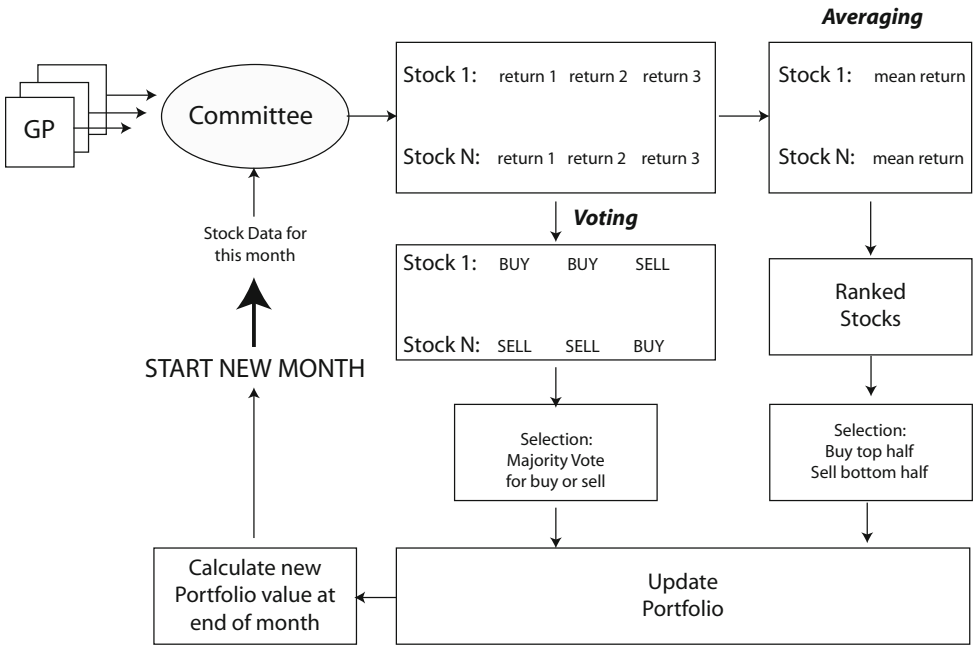

Fig. 2 The committee in action: either averaging or voting

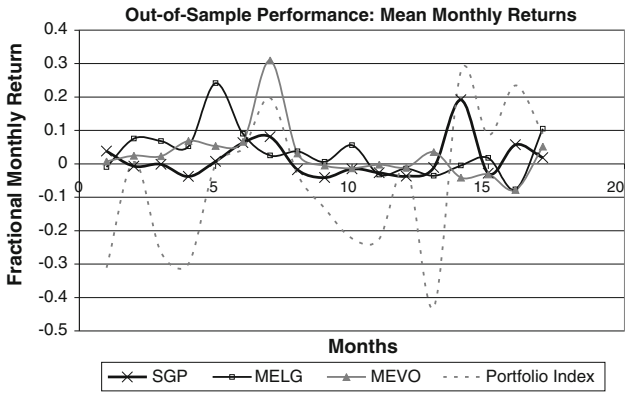

Fig. 3 Mean monthly fractional returns

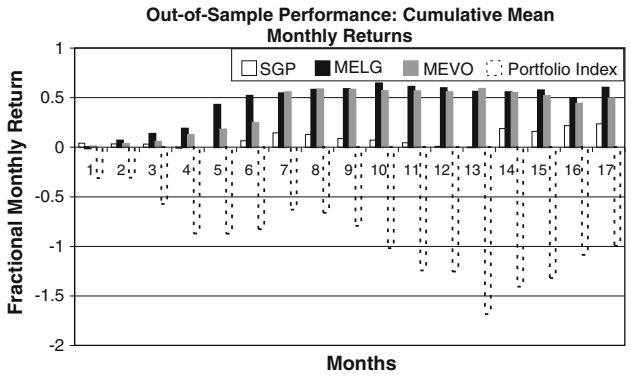

Fig. 4 Cumulative monthly fractional returns

6 Discussion of results: Multiple scenario approach

Figure 3 shows the mean monthly returns (over 25 runs) on the validation data for all three systems (SGP, MELG and MEVO), together with the portfolio index returns. The portfolio index (constructed from the stocks in which our simulator invests) shows considerable volatility—it is more volatile than the overall market index seen in Fig. 1, and so beneficial effects displayed by our GP system cannot be due solely to "cherry-picking" the least volatile stocks. SGP is not very volatile, but neither does it make much profit. Both MELG and MEVO appear to be adept at avoiding losses, yet still able to make good gains in positive months. Figure 4 shows vividly the difference between the large cumulative losses of the portfolio index compared with the cumulative gains of MELG and MEVO.

Figure 5 gives another view of the mean monthly returns by plotting the frequency distributions of returns in the validation period. The portfolio index (dashed) is very volatile, whereas all three GP systems are much less volatile (though with significant positive fat tails).

6.1 Robustness

So what does this tell us about "robustness", and how do we measure it? Simplistically, we might take robustness to be synonymous with "low variance"—i.e. the performance of the individual does not alter much, despite the extreme volatility of the market environment. However, in practice we have a much more exacting requirement: it is not helpful to an investor to know that an individual robustly (i.e. with low variance) makes a loss regardless of the market! A much more helpful measure is to know that the

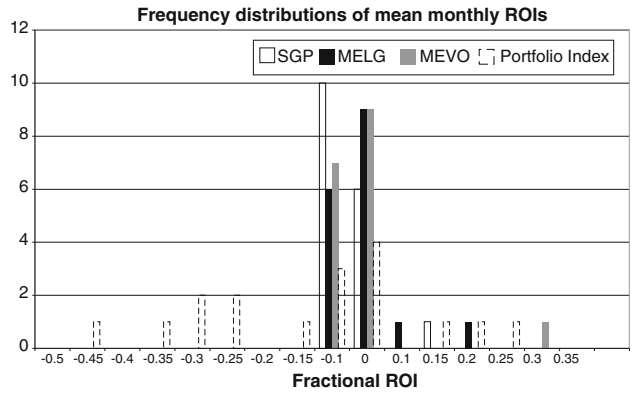

Fig. 5 Frequency distributions of mean monthly fractional returns

individual combines two qualities of (i) high ROI and (ii) low variance in the face of extreme volatility.

In Sect. 1 we stated our three measures of robustness:

1. when exposed to an out-of-sample volatile validation dataset, a more robust solution will have a lower standard deviation of returns, while the returns themselves do not decrease; or
2. when exposed to an out-of-sample volatile validation dataset, a more robust solution will have higher returns while the standard deviation does not increase; or
3. when exposed to an out-of-sample volatile validation dataset, the mean return per unit of risk of a more robust solution will not significantly reduce from that measured during training.

The performance of our three systems, using robustness measures 1 and 2 above, are illustrated in Fig. 6, which shows standard deviation plotted against returns. Specifically, the figure plots returns in excess of the risk free rate, and we have added data for the portfolio index and for a popular non-genetic technical strategy (using Moving Average Convergence Divergence (MACD) to select

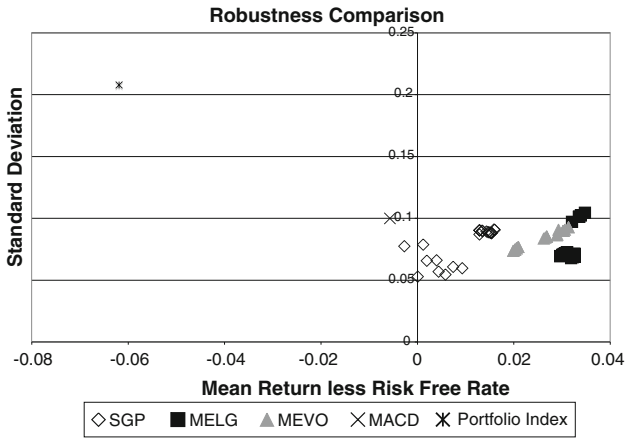

Fig. 6 Robustness comparison

stocks). The portfolio index is shown to be not at all satisfactory, with both low returns and high standard deviation; the MACD approach performs much better than the portfolio index, but not as well as any of the three GP systems. In terms of robustness:

1. The three GP systems and MACD all have similar standard deviations (a ranked *T*-test indicates no significant difference in the GP distributions) and so by this measure no one system is more robust than another.
2. By contrast, the three GP systems and MACD differ in their returns while their standard deviations do not differ, so by measure 2 they are not equally robust. In order (from least to most robust) we have MACD, then SGP, the MEVO and finally (the best) MELG. We further quantify this in the following:

Fund managers use a very similar approach to our robustness measures 1 and 2—they use the Sharpe ratio (Sharpe 1994) (see Sect. 4.1) which determines the ROI (in excess of the risk free rate) per unit of risk (given by the standard deviation). Since the standard deviations in this case are the same, a Sharpe ratio comparison also provides a quantitative comparison of returns and thus of our robustness measure 2. Therefore, we have calculated the Sharpe ratios (across 25 runs) for each of the three GP systems.

Comparison of the Sharpe ratio distributions shows that all three systems achieve substantially better results than the portfolio index (as expected from Fig. 6) and a ranked *T*-test comparison of the Sharpe ratios indicates a statistically significant difference among all three systems. The *p*-values (the probabilities that two compared distributions are from the same population) are presented in Table 3. The means of the Sharpe ratio distributions are 0.125 (SGP), 0.305 (MEVO), and 0.421 (MELG)—MELG is substantially the most robust system of the three.

Our third measure of robustness determines how much the mean return per unit of risk reduces when moving from the training set to the validation set. This is shown in Fig. 7. The percentage reductions in Sharpe ratio (and associated *p*-values from a ranked *T*-test) were 65% for SGP (1.4×10^{-11}) 25% for MEVO (3.1×10^{-8}) and just over 2% for MELG (0.92), indicating a substantial robustness advantage for MELG.

Table 3 Summary of ranked *T*-test (*p*-values)

Compare SGP with MELG	4.01×10^{-16}
Compare SGP with MEVO	4.13×10^{-16}
Compare MEVO with MELG	4.62×10^{-9}

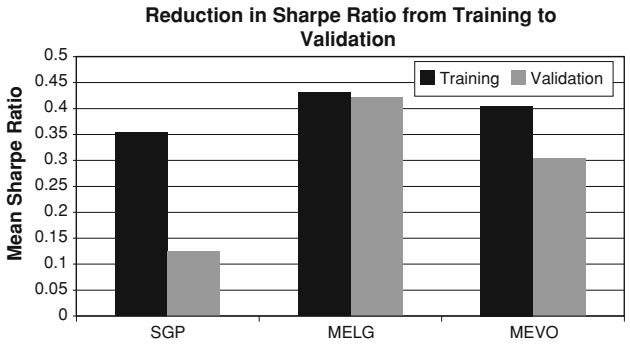

Fig. 7 Robustness measure: drop in Sharpe ratio

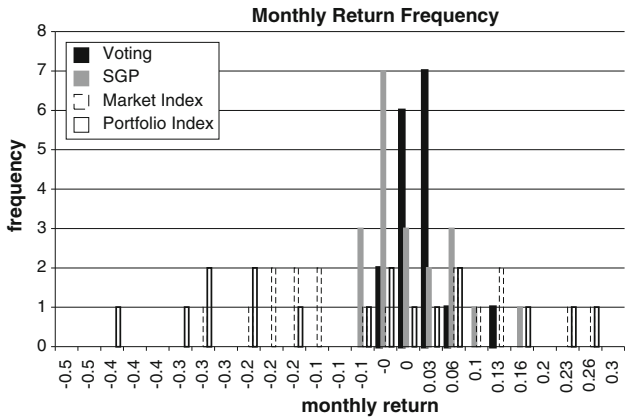

Fig. 8 Frequency distributions of mean monthly fractional returns

7 Discussion of results: Voting committee approach

Figure 8 plots the frequency distributions of returns in the validation period. The market and portfolio indices (dashed lines) are both very volatile; SGP makes a fairly consistent slight loss balanced by some gains in a positive short fat tail. The Voting system makes a fairly consistent slight gain but with a short fat positive tail.

7.1 Robustness

In this experiment, as in the previous experiment, we observe robustness measures 1, 2 and 3 (see Sect. 1).

In each case there were 25 training runs, each run being seeded with a different random number: for SGP, the reported mean ROI and standard deviation are the results of applying the best individual from the final generation to the validation data; for the Voting system, a voting pool of three individuals was selected from final generations of each of the 25 runs—the voting pools were then applied to the validation data and the mean ROI and standard deviation calculated.

The performance of the two systems, using robustness measures 1 and 2 are illustrated in Fig. 9.

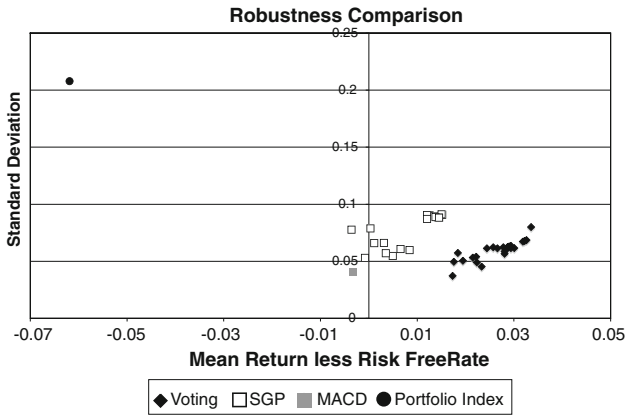

Fig. 9 Robustness comparison

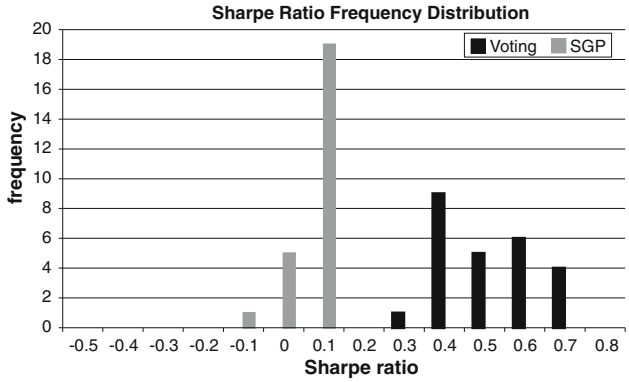

Fig. 10 Frequency distributions of Sharpe ratios. Voting average = 0.567292236, $\sigma = 0.111250913$, SGP average = 0.12483279, $\sigma = 0.062651544$

Table 4 Ranked *T*-test results (*p*-value)

SGP vs voting (mean ROI)	3.8×10^{-8}
SGP vs voting (standard deviation)	4.0×10^{-4}
SGP vs voting (Sharpe ratio)	4.32×10^{-16}

- The Voting system and SGP system have similar standard deviation, and so by this measure no one system is more robust than another.
- By contrast, the Voting system is superior in terms of return. Figure 10 gives the frequency distribution for the Sharpe ratio for both SGP and the Voting system. As above, in each case there were 25 training runs, each run being seeded with a different random number. Both systems beat the portfolio index Sharpe ratio of -0.297 (a negative ROI!), but the Voting system is substantially superior. A ranked *T*-test result is displayed in Table 4 and indicates a convincing difference between the two systems. Therefore, the Voting system is more robust than SGP in terms of measure 2.

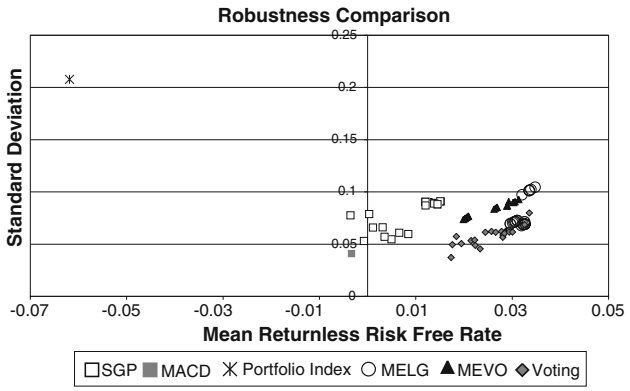

Fig. 11 Robustness comparison of all three systems

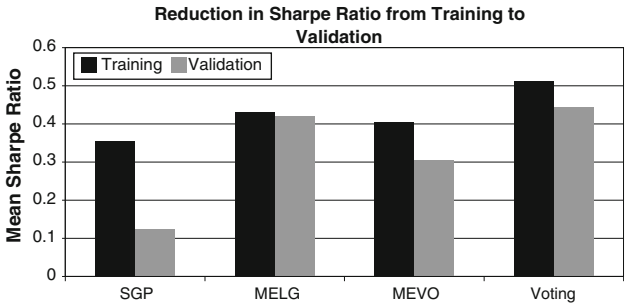

Fig. 12 Drop in mean Sharpe ratios for all three systems

The percentage reductions in Sharpe ratio (and associated p-values from a ranked T-test) were 65% for SGP (1.4×10^{-11}), 13% for the Voting system (5.6×10^{-4}). The performance of the voting system in the different environments is more resilient to changes, thus more robust.

7.2 Multiple scenario approach versus voting approach

We compare the robustness of the two approaches (MELG versus MEVO versus voting system) in Figs. 11 and 12. Figure 11 compares the robustness of the three systems in terms of performance in relation to associated risk (robustness measures 1, 2). Figure 12 compares the robustness of the three systems in relation to change of the performance in different environments (robustness measure 3). In Fig. 11, MEVO is slightly outperformed by MELG and the voting system as, given the same level of risk, it does not yield higher returns than the other two systems. MELG more consistently gives higher returns than the voting system, at the cost of higher risk. The same pattern emerges in Fig. 12; again, MELG has the best performance, beating MEVO, and SGP has the worst performance.

8 Summary and conclusion

In a volatile and unforgiving financial environment, it is possible to obtain a substantial improvement in the robustness of hedge fund stock selection through (1) the use of carefully selected scenarios of extreme market behaviour during GP training; and (2) the use of a voting committee comprising an odd number of GP individuals trained on a variety of different training sets (and therefore with differing phenotypic behaviour).

Our system used a GP system to evolve a non-linear factor model for stock-picking, coupled with an Investment Simulator that modeled a long-short, market-neutral, sector-neutral hedge fund trading CFDs in the highly volatile Malaysian stock market. Historical stock data (both technical and fundamental) were used from the period 1997–2004.

We introduced three practical measures of robustness: the first two compared volatility against returns on investment, and the third compared the Sharpe ratio during training with the Sharpe ratio during validation (see Sect. 1).

For the Multiple Scenario approach, experiments were run on three GP systems (MELG, MEVO and a "standard" GP system—SGP) with 25 runs of each, and comparisons were made with both a portfolio index and a non-genetic simple technical analysis for stock picking.

Although robustness measure 1 showed no significant difference between the three GP systems, statistical analysis of measures 2 and 3 indicated overwhelmingly that MELG provides the most robust individual, with SGP being the least robust. All three GP systems were shown to have better performance than the non-genetic technical analysis, and this in turn performed very much better than the portfolio index.

For the Committee voting approach, experiments were run on two GP systems: (i) SGP, and (ii) a committee of three best-of-run individuals from three GP systems utilising different sets of training data. Statistical analysis indicated that the Voting system provides a remarkable improvement in terms of robustness measure 2 when compared to SGP.

When we compare the three best systems—MELG, MEVO and the voting system—MELG and the voting system show a slight edge over MEVO in terms of all three robustness measures. In Fig. 12 the voting system appears to be slightly better than MELG; however, there is no statistically significant difference between MELG and the voting system when we compare the distributions of their Sharpe ratios (we obtain a ranked T-test p-value of 0.268589).

Further work in this area includes extending the experiment to a larger universe of stocks; combining the

scenarios mechanism with other robustness-enhancing techniques; and investigating better ways to present the extreme scenarios to the population. For example, we are trying to gain a better understanding of why using the complete training set as a separate scenario for MEVO did not give good results. Also it includes combining the voting mechanism with other robustness-enhancing techniques, experimenting with different sizes of committee and different ways to obtain good individuals with widely differing phenotypic behaviour, and attempting to gain a better understanding of the mechanisms of robustness.

Acknowledgments The authors thank Dr Gerard Vila and Prospect Wealth Management for suggestions and discussions, SIAM Capital for financial support, and Reuters for access to financial data. We also thank the anonymous referees for their helpful comments.

References

Aragón VS, Esquivel SC (2004) An evolutionary algorithm to track changes of optimum value locations in dynamic environments. J Comput Sci Technol 4(3):127–134

Bersano-Begey TF, Daida JM (1997) A discussion on generality and robustness and a framework for fitness set construction in genetic programming to promote robustness. In: Koza JR (ed) Late breaking papers at the 1997 genetic programming conference. Stanford Bookstore, Stanford University, California, pp 11–18

Bowers CP (2005) Formation of modules in a computational model of embryogeny. In: The 2005 IEEE Congress on evolutionary computation, vol 1, pp 537–542

Branke J (1998) Creating robust solutions by means of evolutionary algorithms. In: PPSN V: Proceedings of the 5th international conference on parallel problem solving from nature. Springer-Verlag, London, pp 119–128

Brown G, Wyatt J, Harris R, Yao X (2005) Diversity creation methods: a survey and categorisation. Inf Fusion 6(1):5–20

Buxton B, Langdon WB, Barrett SJ (2001) Data fusion by intelligent classifier combination. Meas Control 34(8):229–234

Dietterich TG (2000) Ensemble methods in machine learning. In: Proceedings of the first international workshop on multiple classifier systems. LNCS, vol 1857. Springer, New York, pp 1–15

Egan JP (1975) Signal detection theory and ROC analysis. Academic Press, New York

Fitzpatrick JM, Grefenstette JJ (1988) Genetic algorithms in noisy environments. Mach Learn 3:101–120

Gagné C, Schoenauer M, Parizeau M, Tomassini M (2006) Genetic programming, validation sets, and parsimony pressure. In: Collet P, Tomassini M, Ebner M, Gustafson S, Ekárt A (eds) Proceedings of the 9th European conference on genetic programming. Lecture notes in computer science, vol 3905. Springer, Budapest, pp 109–120. http://link.springer.de/link/service/series/0558/papers/3905/39050109.pdf

Gierer A, Berking S, Bode H, David CN, Flick K, Hansmann G, Schaller H, Trenkner E (1972) Regeneration of hydra from reaggregated cells. Nat New Biol 239:98–101

Hammel U, Bäck T (1994) Evolution strategies on noisy functions: How to improve convergence properties. In: Davidor Y, Schwefel HP, Männer R (eds) Proceedings of the international conference on evolutionary computation, the third conference on parallel problem solving from nature (PPSN III), vol 866. Springer, Jerusalem, pp 159–168. http://citeseer.ist.psu.edu/hammel94evolution.html

Haynes T, Wainwright R, Sen S, Schoenefeld D (1995) Strongly typed genetic programming in evolving cooperation strategies. In: Eshelman L (ed) Genetic algorithms: Proceedings of the sixth international conference (ICGA95). Morgan Kaufmann, Pittsburgh, pp 271–278. http://www.mcs.utulsa.edu/~rogerw/papers/Haynes-icga95.pdf

Herbster M (2001) Learning additive models online with fast evaluating kernels. In: Proceedings of the 14th annual conference on computational learning theory and and 5th European conference on computational learning theory. LNCS, vol 2111. Springer, New York, pp 444–460

Herbster M, Warmuth MK (1998) Tracking the best expert. Mach Learn 32(2):151–178

Herbster M, Warmuth MK (2001) Tracking the best linear predictor. J Mach Learn Res 1:281–309

Herrmann J (1999) A genetic algorithm for minimax optimization problems. In: Proceedings of the Congress on evolutionary computation, vol 2, pp 1099–1103

Huynen M, Stadler P, Fontana W (1996) Smoothness within ruggedness: the role of neutrality in adaptation

Imamura K, Soule T, Heckendorn R, Foster J (2003) Behavioral diversity and a probabilistically optimal GP ensemble. Genet Program Evolvable Mach 4:235–253

Ito T, Iba H, Kimura M (1996) Robustness of robot programs generated by genetic programming. In: Koza JR, Goldberg DE, Fogel DB, Riolo RL (eds) Genetic Programming 1996: Proceedings of the first annual conference. MIT Press, Stanford University, California, pp 321–326

Jordaan E, Kordon A, Chiang L, Smits G (2004) Robust inferential sensors based on ensemble of predictors generated by genetic programming. In: Yao X, Burke E, Lozano JA, Smith J, Merelo-Guervós JJ, Bullinaria JA, Rowe J, Kabán PTA, Schwefel HP (eds) Parallel problem solving from nature—PPSN VIII. LNCS, vol 3242. Springer-Verlag, Birmingham, pp 522–531. doi: 10.1007/b100601. http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=3242&spage=522

Kitamo H (2001) Foundations of systems biology. MIT Press, Cambridge. ISBN: 0-262-11266-3

Kittler J, Roli F (2001) Multiple classifier systems. In: Proceedings of 2nd international workshop, MCS2001. Springer-Verlag, New York

Kuscu I (1998) Promoting generalization of learned behaviours in genetic programming. In: Eiben AE, Back T, Schoenauer M, Schwefel HP (eds) Fifth international conference on parallel problem solving from nature. LNCS, vol 1498. Springer-Verlag, Amsterdam, pp 491–500

Kuscu I (2000) Generalisation and domain specific functions in genetic programming. In: Proceedings of the 2000 Congress on evolutionary computation CEC00. IEEE Press, La Jolla Marriott Hotel La Jolla, California, USA, vol 2, pp 1393–1400. doi: 10.1109/CEC.2000.870815. http://citeseer.ist.psu.edu/502977.html

Kushchu I (2002) Genetic programming and evolutionary generalization. IEEE Trans Evol Comput 6(5):431–442

Liu Y, Yao X, Higuchi T (2000) Evolutionary ensembles with negative correlation learning. IEEE Trans Evol Comput 4(4):380. http://citeseer.ist.psu.edu/article/liu00evolutionary.html

Lowenstein R (2002) When genius failed. Fourth Estate

Miller JF (2004) Evolving a self-repairing, self-regulating, french flag organism. In: GECCO 2004, pp 129–139

Miller BL, Goldberg DE (1996) Genetic algorithms, selection scheme, and the varying effect of noise. Evol Comput 4(2): 113–131

Moore FW, Garcia ON (1997) A new methodology for reducing brittleness in genetic programming. In: Proceedings of the National Aerospace and Electronics 1997 conferences, NAECON-97

Nissen V, Propach J (1998) On the robustness of population-based versus point-based optimisation in the presence of noise. IEEE Trans Evol Comput 2(3):107–119

Panait L, Luke S (2003) Methods for evolving robust programs. In: Genetic and evolutionary computation—GECCO 2003. LNCS, vol 2724. Springer, New York, pp 1740–1751

Ranawana R, Palade V (2006) Multi-classifier systems: review and a roadmap for developers. Int J Hybrid Intell Syst 3(1):35–61

Reynolds CW (1992) An evolved, vision-based behavioral model of coordinated group motion. In: Meyer JA, Wilson SW (eds) From animals to animats (Proceedings of simulation of adaptive behaviour). MIT Press, Cambridge

Rosca J (1996) Generality versus size in genetic programming. In: Koza JR, Goldberg DE, Fogel DB, Riolo RL (eds) Genetic Programming 1996: Proceedings of the first annual conference. MIT Press, Cambridge, pp 381–387

Sharpe WF (1994) The Sharpe ratio. J Portf Manag 21:49–58

Soule T (1999) Voting teams: a cooperative approach to non-typical problems using genetic programming. In: Proceedings of the genetic and evolutionary computation conference. Morgan Kaufmann, vol 1, pp 916–922. http://www.cs.uidaho.edu/tsoule/research/vote2.ps

Soule T (2000) Heterogeneity and specialization in evolving teams. In: Proceedings of the genetic and evolutionary computation conference (GECCO-2000). Morgan Kaufmann, pp. 778–785. http://www.cs.bham.ac.uk/~wbl/biblio/gecco2000/ESO56.ps

Soule T (2003) Operator choice and the evolution of robust solutions. In: Riolo RL, Worzel B (eds) Genetic programming theory and practise, chap 16. Kluwer, Dordrecht, pp 257–270

Soule T, Heckendorn RB, Shen J (2002) Solution stability in evolutionary computation. In: Cicekli N (ed) ISCIS XVII seventeenth international symposium on computer and information sciences. CRC Press, University of Central Florida, Orlando, pp 237–241

Stefano CD, Cioppa AD, Marcelli A (2003) Exploiting reliability for dynamic selection of classifiers by means of genetic algorithms. In: ICDAR '03: Proceedings of the seventh international conference on document analysis and recognition. IEEE Computer Society, Washington, pp 671–675

Tsutsui S, Ghosh A (1997) Genetic algorithms with a robust solution searching scheme. IEEE Trans Evol Comput 1(3):201–208

Wagner A (2005) Robustness and evolvability in living systems. Princeton University Press, Princeton

Zhang BT, Joung JG (2000) Building optimal committees of genetic programs. In: Schoenauer M, Deb K, Rudolph G, Yao X, Lutton E, Merelo JJ, Schwefel HP (eds) Parallel problem solving from nature—PPSN VI. Springer, Berlin, pp 231–240. http://citeseer.ist.psu.edu/zhang00building.html

Zhu M, Chipman H (2006) Darwinian evolution in parallel universes: a parallel genetic algorithm for variable selection. Technometrics 48(4):491–502