Diverse Committees Vote for Dependable Profits

Wei Yan
Department of Computer Science
University College London
Gower Street, London, WC1E 6BT
w.yan@cs.ucl.ac.uk

Christopher D. Clack
Financial Computing and Quantitative Finance
Department of Computer Science
University College London
Gower Street, London, WC1E 6BT
c.clack@cs.ucl.ac.uk

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 an approach that uses a voting committee of GP individuals with differing phenotypic behaviour.

Categories and Subject Descriptors

I.2.M [**Artificial Intelligence**]: Miscellaneous

General Terms

Algorithms, Experimentation

Keywords

Genetic Programming, Robustness, Committee, Diversity, Finance, Dynamic Environment

1. INTRODUCTION

The hedge fund Long Term Capital Management made huge profits from mathematical models of bond and options markets — then lost $553,000,000 in just one day because the assumptions underlying their models were not robust to massive fluctuations in the markets [24].

The financial markets are highly dynamic, unpredictable and unforgiving. If 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 [34, 35]); 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 [4, 37]) and the insensitivity to a noisy fitness function [9, 11, 25]. 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 et al [1] model a dynamic environment as a sequence of fitness functions, each defined by changes to the previous. The model uses occasional macro-mutation for radical genotoype 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!

Our approach is substantially different to the prior work in this area, and 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.

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 [18, 38].

Robustness is a very broad theme and it is impossible to capture all its aspects by means of a single definition. Robustness is an ubiquitously observed property of biological systems. It is considered to be a fundamental feature of complex evolvable systems [18].

The definition of robustness in evolutionary systems varies from author to author, but in broad terms, it can 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

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

[34, 35] 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.

2. Robust to external changes (phenotypic robustness)

   (a) Robustness as the generalisation ability of the programs evolved using GP [2, 20, 21, 27, 29]. 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" [22]. 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 [17, 28]. Practical optimisation problems often require the evaluation of solutions through experimentation, stochastic simulation, sampling, or even interaction with the user. Thus, most practical problems involve noise. Jordanne et al. [17] 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 [15] 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 [26, 3]. 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 [10].

2.2 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) [19] 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 [36] [30].

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 [8] of the constituent classifiers. There is no guarantee of improvement with the MCS approach, but [6] 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 [13, 12, 14]

Zhang and Joung [39] 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" [7]. Dietterich [7] provides an informative review of these systems; see also [5, 23, 16].

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 [32, 33] 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 [40].

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. Firstly, 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 [16, 32].

2. Secondly, 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 [39].

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 [7, 5].

3. DESCRIPTION OF THE ALGORITHM

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

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. It can then be applied to each of the many stocks in a portfolio, to assist investment decisions.

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 [31] is calculated.

Fitness

The fitness f for an individual is the Sharpe ratio S, given by Equation 1.

$$S = \frac{\overline{x_i} - RFR_i}{\sigma_i} \qquad (1)$$

In Equation 1, S is the Sharpe Ratio, $\overline{x}_i$ 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 RFR_i is the average monthly Risk Free Rate for sub-period i. We set RFR_i to $0.00\dot{3}$ 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

The simulated Hedge 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 the stocks in this basket 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.[1]

The training data is monthly prices (and other technical and fundamental data) over a period of 71 months. Since we have only monthly data, 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 about 20 factors and 7,680 data points. A return prediction is assigned to each stock.

The stocks are grouped into 4 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 3 stocks in each sector become the top fractile and the bottom 3 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.

- CFDs (Contract for Differences) 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

[1] A market-neutral *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.

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.

4.3 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 pre-defined distinctively different training data sets 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 Figure 1.

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.

5. EXPERIMENT

Our primary research question 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 data sets. The basic GP parameter settings for the GP systems are identical, as given in Table 1.

Table 1: GP Parameter Settings

Population size (N)	1000
Method of generation	Ramped half and haif
Function set	{+, -, *, /, Exp}
Terminal set	18 firm-specific factors
Selection scheme	Fitness proportionate selection
Criterion of fitness	Monthly Sharpe ratio
Number of trees generated by elitism	10 (1%)
Number of trees generated by crossover	950 (95%)
Number of trees generated by mutation	40 (4%))
Termination criterion	100-generation evolution
Maximum depth of initial generation	6

5.1 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 data set 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: 31st May 2003 to 31st December 2004 (19 months);
2. Bear market: 31st January 2000 to 31st May 2001 (16 months);
3. Volatile market: 31st January 1999 to 31st March 2000 (14 months).

Figure 2 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.

5.2 Out-of-Sample Validation

The two systems are validated on a previously unseen "out of sample" data set, 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

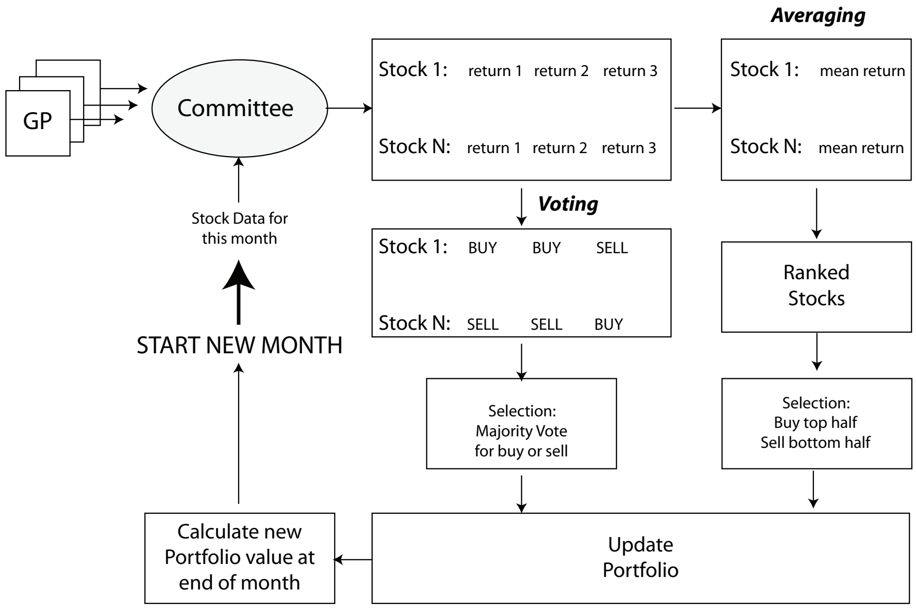
Averaging
GP
Committee
Stock 1: return 1 return 2 return 3
Stock N: return 1 return 2 return 3
Stock 1: mean return
Stock N: mean return
Voting
Stock 1: BUY BUY SELL
Stock N: SELL SELL BUY
Stock Data for this month
START NEW MONTH
Ranked Stocks
Selection: Majority Vote for buy or sell
Selection: Buy top half Sell bottom half
Calculate new Portfolio value at end of month
Update Portfolio

Figure 1: The Committee in action: either Averaging or Voting.

stock markets, and in emerging markets in particular. A successful hedge fund stock selection model must be robust — be able to perform in both (extreme) up and down markets.

For out-of-sample validation, the best-of-run individual was selected from the final generation of each of the GP evolutions. For the validation, 25 runs were recorded and the results are discussed in the following section.

6. DISCUSSION OF RESULTS

Figure 3 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.

Whilst observation of the frequency distribution of monthly returns provides some useful information, it does not tell us much about the "robustness" of the new Voting system compared with that of SGP.

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 hardly 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 individual combines two qualities of (i) high return on investment (ROI) and (ii) low variance in the face of extreme volatility. These two measures are given in Figure 4 and Figure 5 respectively. 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

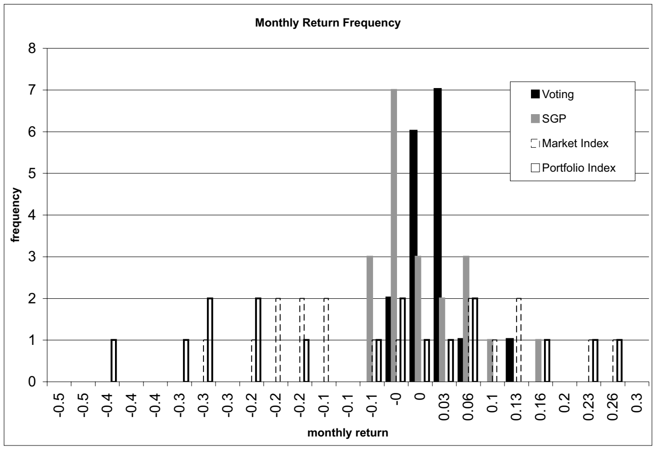

Figure 3: Frequency distributions of mean monthly fractional returns.

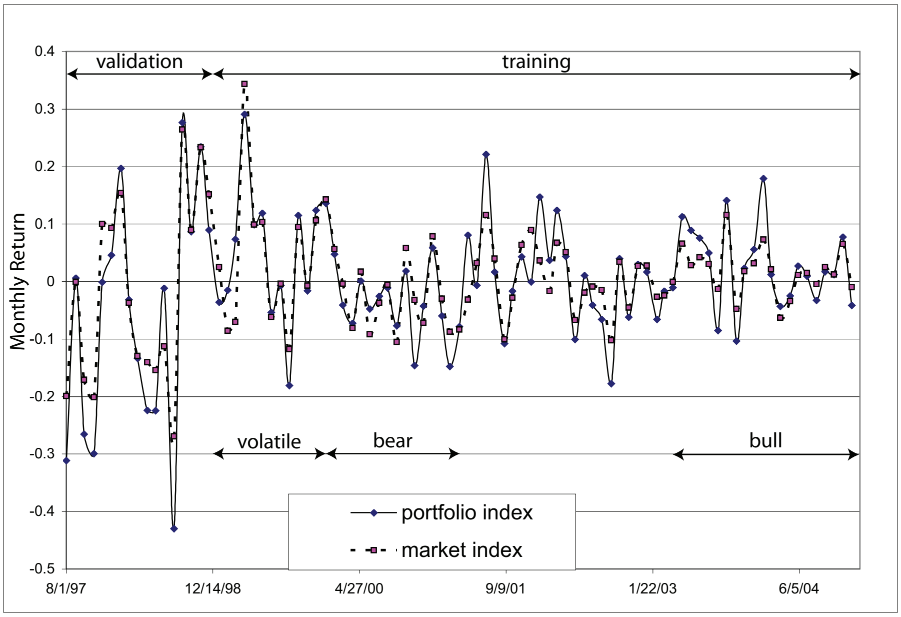

Figure 2: Market and portfolio indices (fractional monthly returns, 31st July 1997 to 31st December 2004), scenarios and validation period.

3 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 Voting system appears to achieve both a higher mean ROI and a lower standard deviation than SGP; to determine whether this difference is significant we apply a ranked T-test, giving p-values of 3.8×10^-8 for the ROI and 4×10^-4 for the standard deviation (see Table 2).

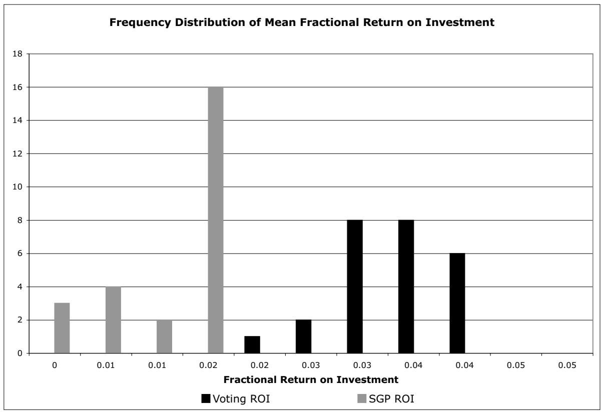

Figure 4: Distributions of mean returns on investment. Voting average = 0.035703, σ = 0.005425, **SGP average** = 0.013821, σ = 0.005856.

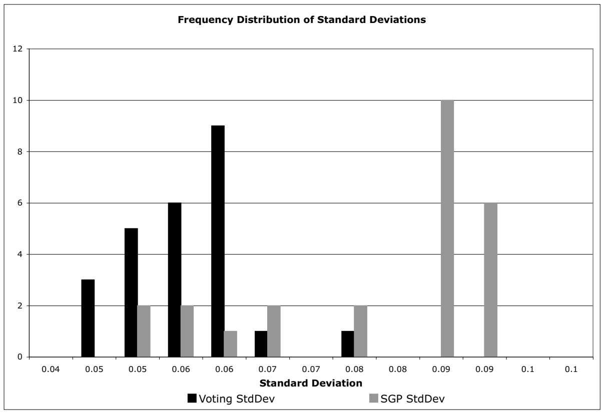

Figure 5: Distributions of standard deviation. Voting average = 0.05838, σ = 0.006705, **SGP average** = 0.005856, σ = 0.013663.

Alternatively, we could use the Sharpe Ratio (see Section 4.1) as a measure of robustness. Figure 6 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 return on investment!), but the Voting system is substantially superior. A ranked T-test result is displayed in Table 2 and indicates a convincing difference between the two systems.

7. 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 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).

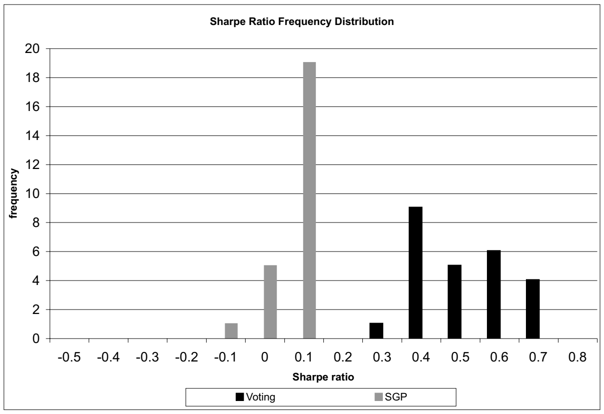

Figure 6: Frequency distributions of Sharpe ratios. Voting average = 0.567292236, σ = 0.111250913, **SGP average** = 0.12483279, σ = 0.062651544.

Table 2: 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}

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

We used the Sharpe ratio [31] as a practical definition of robustness — i.e. a more robust solution would have a better balance of high return on investment and low standard deviation of returns. Sharpe ratios were calculated for 25 validation runs of the investment simulator using (i) the best-of-run individual from a standard GP system (SGP), and (ii) a committee of three best-of-run individuals from three GP systems utilising different sets of training data.

Statistical analysis indicated overwhelmingly that the Voting system provides a substantial improvement in robustness when compared to SGP.

Further work in this area 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.

8. ACKNOWLEDGMENTS

The authors thank Dr Gerard Vila 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.

9. REFERENCES

[1] V. S. Aragon and S. C. Esquivel. An evolutionary algorithm to track changes of optimum value locations in dynamic environments. *Journal of Computer Science and Technology*, 4(3):127–134, 2004.

[2] T. F. Bersano-Begey and J. M. Daida. A discussion on generality and robustness and a framework for fitness set construction in genetic programming to promote robustness. In J. R. Koza, editor, *Late Breaking Papers at the 1997 Genetic Programming Conference*, pages 11–18, Stanford University, CA, USA, 13–16July 1997. Stanford Bookstore.

[3] C. P. Bowers. Formation of modules in a computational model of embryogeny. In *The 2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 537–542, 2005.

[4] J. Branke. Creating robust solutions by means of evolutionary algorithms. In *PPSN V: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, pages 119–128, London, UK, 1998. Springer-Verlag.

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

[6] B. Buxton, W. B. Langdon, and S. J. Barrett. Data fusion by intelligent classifier combination. *Measurement and Control*, 34(8):229–234, 2001.

[7] T. G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, volume 1857 of *LNCS*, pages 1–15. Springer, 2000.

[8] J. P. Egan. *Signal Detection Theory and ROC Analysis*. Academic Press, 1975.

[9] J. M. Fitzpatrick and J. J. Grefenstette. Genetic algorithms in noisy environments. *Machine Learning*, 3:101–120, 1988.

[10] A. Gierer, S. Berking, H. Bode, C. N. David, K. Flick, G. Hansmann, H. Schaller, and E. Trenkner. Regeneration of hydra from reaggregated cells. *Nature New Biology*, 239:98–101, 1972.

[11] U. Hammel and T. Bäck. Evolution strategies on noisy functions: How to improve convergence properties. In Y. Davidor, H.-P. Schwefel, and R. Männer, editors, *Proceedings of theInternational Conference on Evolutionary Computation, theThirdConference on Parallel Problem Solving from Nature(PPSN III)*, volume 866, pages 159–168, Jerusalem, 9–14 1994. Springer.

[12] M. Herbster. 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*, volume 2111 of *LNCS*, pages 444–460. Springer, 2001.

[13] M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2):151–178, 1998.

[14] M. Herbster and M. K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.

[15] J. Herrmann. A genetic algorithm for minmax optimisation problems. volume 2, pages 1099–1103, 1999.

[16] K. Imamura, T. Soule, R. Heckendorn, and J. Foster. Behavioral diversity and a probabilistically optimal GP ensemble. *Genetic Programming and Evolvable Machines*, 4:235–253, 2003.

[17] E. Jordaan, A. Kordon, L. Chiang, and G. Smits. Robust inferential sensors based on ensemble of predictors generated by genetic programming. In X. Yao, E. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. Rowe, P. T. A. Kabán, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *LNCS*, pages 522–531, Birmingham, UK, 18-22Sept. 2004. Springer-Verlag.

[18] H. Kitamo. *Foundations of systems biology.* MIT Press, ISBN 0-262-11266-3, 2001.

[19] J. Kittler and F. Roli. Multiple classifier systems. In *Proceedings of 2nd International Workshop, MCS2001.* Springer Verlag, 2001.

[20] I. Kuscu. Promoting generalization of learned behaviours in genetic programming. In A. E. Eiben, T. Back, M. Schoenauer, and H.-P. Schwefel, editors, *Fifth International Conference on Parallel Problem Solving from Nature*, volume 1498 of *LNCS*, pages 491–500, Amsterdam, 27-30Sept. 1998. Springer-Verlag.

[21] I. Kuscu. Generalisation and domain specific functions in genetic programming. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, volume 2, pages 1393–1400, La Jolla Marriott Hotel La Jolla, California, USA, 6-9July 2000. IEEE Press.

[22] I. Kushchu. Genetic programming and evolutionary generalization. *IEEE Transactions on Evolutionary Computation*, 6(5):431–442, October 2002.

[23] Y. Liu, X. Yao, and T. Higuchi. Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, 4(4):380, 2000.

[24] R. Lowenstein. *When Genius Failed*. Fourth Estate, 2002.

[25] B. L. Miller and D. E. Goldberg. Genetic algorithms, selection scheme, and the varying effect of noise. *Evolutionary Computation*, 4(2):113–131, 1996.

[26] J. F. Miller. Evolving a self-repairing, self-regulating, french flag organism. pages 129–139, 2004.

[27] F. W. Moore and O. N. Gacia. A new methodology for reducing brittleness in genetic programming. In *Proceedings of the National Aerospace and Electronics 1997 Conferences, NAECON-97*, 1997.

[28] V. Nissen and J. Propach. On the robustness of population-based versus point-based optimisation in the presence of noise. *IEEE Transactions on Evolutionary Computation*, 2(3), 1998.

[29] L. Panait and S. Luke. Methods for evolving robust programs. In *Genetic and Evolutionary Computation — GECCO 2003*, volume 2724 of *LNCS*, pages 1740–1751. Springer, 2003.

[30] R. Ranawana and V. Palade. Multi-classifier systems: Review and a roadmap for developers. *International Journal of Hybrid Intelligent Systems*, 3(1):35–61, 2006.

[31] W. F. Sharpe. The sharpe ratio. *J. Portfolio Management*, 21:49–58, 1994.

[32] T. Soule. Voting teams: A cooperative approach to non-typical problems using genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, pages 916–922. Morgan Kaufmann, 1999.

[33] T. Soule. Heterogeneity and specialization in evolving teams. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 778–785. Morgan Kaufmann, 2000.

[34] T. Soule. Operator choice and the evolution of robust solutions. In R. L. Riolo and B. Worzel, editors, *Genetic Programming Theory and Practise*, chapter 16, pages 257–270. Kluwer, 2003.

[35] T. Soule, R. B. Heckendorn, and J. Shen. Solution stability in evolutionary computation. In N. Cicekli, editor, *ISCIS XVII Seventeenth International Symposium On Computer and Information Sciences*, pages 237–241, University of Central Florida, Orlando, Florida, Oct.28-30 2002. CRC Press.

[36] C. D. Stefano, A. D. Cioppa, and A. Marcelli. 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*, pages 671–675, Washington, DC, USA, 2003. IEEE Computer Society.

[37] S. Tsutsui and A. Ghosh. Genetic algorithms with a robust solution searching scheme. *IEEE Transactions on Evolutionary Computation*, 1(3):201–208, 1997.

[38] A. Wagner. *Robustness and Evolvability in Living Systems.* Princeton University Press, 2005.

[39] B.-T. Zhang and J.-G. Joung. Building optimal committees of genetic programs. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature – PPSN VI*, pages 231–240, Berlin, 2000. Springer.

[40] M. Zhu and H. Chipman. Darwinian evolution in parallel universes: A parallel genetic algorithm for variable selection. *Technometrics*, 48(4):491–502, 2006.