Journal of SOFTWARE – Practice and Experience, Wiley
To appear (accepted for publication, April 1998)

EDDIE Beats the Bookies

Edward P.K. TSANG
Department of Computer Science, University of Essex, Wivenhoe Park, Colchester, United Kingdom
edward@essex.ac.uk

James M. BUTLER
Standards and Policies Group, Reuters Ltd, 85 Fleet Street, London, United Kingdom
james.butler@reuters.com

Jin LI
Department of Computer Science, University of Essex, Wivenhoe Park, Colchester, United Kingdom
jli@essex.ac.uk

SUMMARY

Investment involves the maximisation of return on one's investment whilst minimising risk. Good forecasting, which often requires expert knowledge, can help to reduce risk. In this paper, we propose a genetic programming based system EDDIE, (which stands for Evolutionary Dynamic Data Investment Evaluator), as a forecasting tool. Genetic programming is inspired by evolution theory, and has been demonstrated to be successful in other areas. EDDIE interacts with the users and generates decision trees, which can also be seen as rule sets. We argue that EDDIE is suitable for forecasting because apart from utilising the power of genetic programming to efficiently search the space of decision trees, it allows expert knowledge to be channelled into forecasting and it generates rules which can easily be understood and verified. EDDIE has been applied to horse racing and achieved outstanding results. When experimented on 180 handicap races (real data) in the UK, it out-performed other common strategies used in horse race betting by great margins. The idea was then extended to financial forecasting. When tested on historical S&P-500 data EDDIE achieved a respectable annual rate of return over a three and a half year period. While luck may play a part in the success of EDDIE, our experimental results do indicate that EDDIE is a tool which deserves more research.

KEY WORDS: finance; forecasting; genetic programming; horse racing; investment

* Contact author:
Dr Edward Tsang, Department of Computer Science, University of Essex, Colchester CO4 3SQ, UK; tel: +44 1206 872774; fax: +44 1206 872788

INTRODUCTION

Financial investment is concerned with risking money in order to gain more money. The objective is to minimise risk and maximise return on investment (ROI). Information can often help us to reduce risk. For example, to predict the future price of a share, useful information include; its past and present prices, price-earning ratios, price-to-book values, dividends, market indices, who said what in the public domain, etc. The aim of our research is to build tools to help investors to make best use of the information available to them. Such tools should improve the productivity of the users by allowing them to examine more sets of rules in less time. To some users, such tools could help them to do what they did not have the knowledge to do without such tools.

Many factors could directly or indirectly affect the future price of an investment. Such factors are often inter-related, which adds to the difficulty of analysis. The combinatorial explosion problem prevents one from examining combinations of all the factors and their possible ways of interaction. We have developed EDDIE (Evolutionary Dynamic Data Investment Evaluator), a system using evolutionary computation [3] [20] [21] [22] [26] to help investors to evaluate investment opportunities. Our first implementation EDDIE-1 was applied to the horse racing domain, whose similarity with financial forecasting has been well documented [4] [17]. EDDIE-1 allows a potential investor to make hypotheses about the factors which are relevant to the result of a horse race. It then tests those hypotheses using historical data and evolves, by way of natural selection, a decision tree which aims to provide a good ROI. A decision tree generated by EDDIE-1 can be seen as a set of IF-THEN rules which classifies horses to winners or losers. The idea was then extended to financial forecasting. This paper reports the promising results so far.

GENETIC PROGRAMMING AS A TOOL FOR FORECASTING

In this section, we introduce the basic ideas in genetic programming, and explain how it can be used for building forecasting tools.

Genetic Algorithms (*GAs*) [10] [15] and Genetic Programming (*GP*) [21] are both inspired by Darwin's evolution theory. In typical GAs, a candidate solution is represented by a string. In our GP, a candidate solution is represented by a *genetic decision tree* (GDT). The basic elements of GDTs are *rules* and *forecast values*, which correspond to the *functions* and *terminals* in GP. Figure 1 shows an example of a simple GDT. A useful GDT in the real world is almost certainly a lot more sophisticated than this. In GP terms, the questions in the example GDT are *functions*, and the proposed actions are *terminals*, which may also be forecast values. In this example, the GDT is binary; in general, this need not be the case.

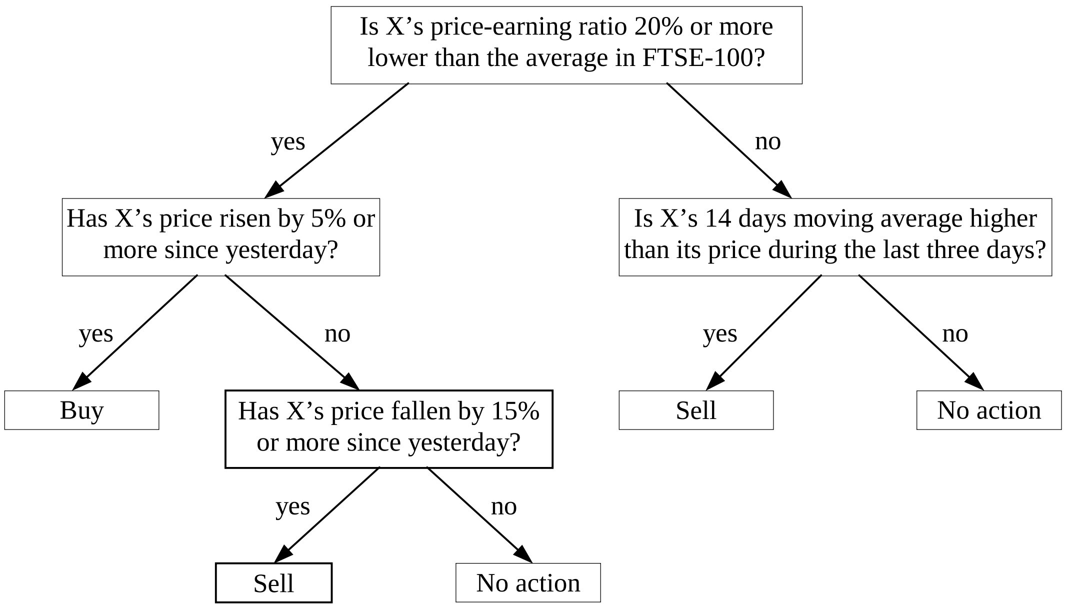

Figure 1. A (simplistic) GDT concerning the actions to take with Share X

A GDT can be seen as a set of rules. For example, one of the rules expressed in the GDT in figure 1 is:

IF X's price-earning ratio is 20% or more lower than the average in FTSE-100

AND X's price has risen by 5% or more since yesterday,

THEN Buy X.

GP is attractive for financial applications because it manipulates GDTs (as opposed to strings in GAs). This allows one to handle rule sets of virtually any size[1]. Besides, rules are easy to understand and evaluate by human users, which makes them more attractive than neural networks, most of which are black boxes [14] [34].

[1] In GAs, strings are normally of uniform size, with exceptions such as Messy GAs [16].

For a GP to work, one must be able to evaluate each GDT, and assign to it a *fitness* value, which reflects the quality of the GDT. Our GP maintains a set of GDTs called a *population* and works in iterations. In each iteration, GDTs are picked from the population weighted randomly using *fitness-proportionate reproduction*, which means that the fitter a GDT is, the greater chance it has of being picked. The set of all GDTs thus picked form a *mating pool* from which pairs of GDTs, which are referred to as *parents*, are picked. A branch in each parent is picked at random. The parents then exchange the subtrees under those branches, as shown in figure 2. This operation is called *crossover*. In figure 2, three nodes are cut off from Parent 1 at level 2. The subtree under the right hand branch of the root is picked from Parent 2, which is shaded for illustration purpose. Offspring 1 is generated by replacing the three nodes in Parent 1 by the part cut from Parent 2. Offspring 2 is generated similarly. Offspring are *mutated* occasionally, which is done by replacing random elements of the GDT by random (or heuristically determined) values. The possibly mutated offspring will then replace the old GDTs to form the new population. There are many variations in the way that the population is updated by new offspring, the way that the initial population is generated, the way that parents are picked, the way that crossover and mutation is done, etc. (e.g. see [3] [21] [22] [31]). These will not be elaborated here.

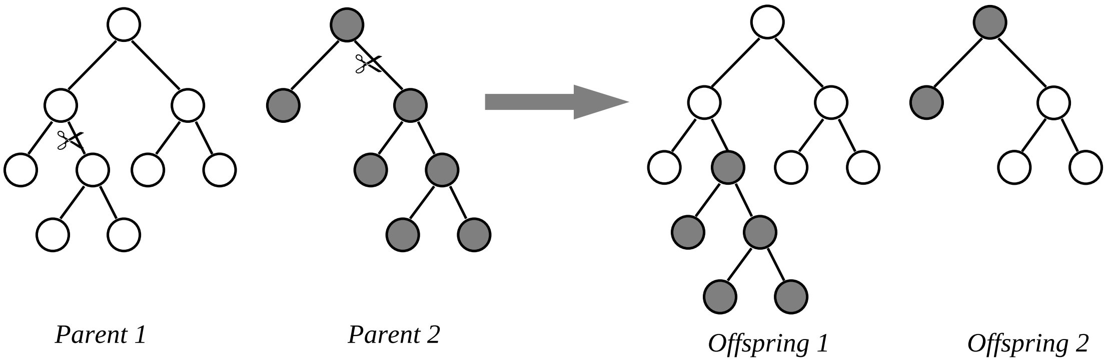

*Figure 2. Crossover in genetic programming (*cut off points marked by ✂)

For a GP to succeed, subtrees (or *genetic material* as they are sometimes referred to) which contribute to fitter candidate solutions (GDTs) get more chance of surviving in the population; this resembles survival of the fittest in nature. Besides, a good population (or "gene pool") must also have sufficient diversification to allow new candidate solutions, or "species", to be generated. So, a GP must reward the fittest candidate solutions but give weak ones reasonable chances to survive. Many successful applications of GP in various domains have been reported, e.g. see [3] [20] [22] [26].

Figure 3 shows the way that the proposed GP-based EDDIE system could be used as as forecasting tool. It should be emphasised that EDDIE does not replace the role of experts. It serves to improve the productivity of the users who may have various levels of expertise in the domain.

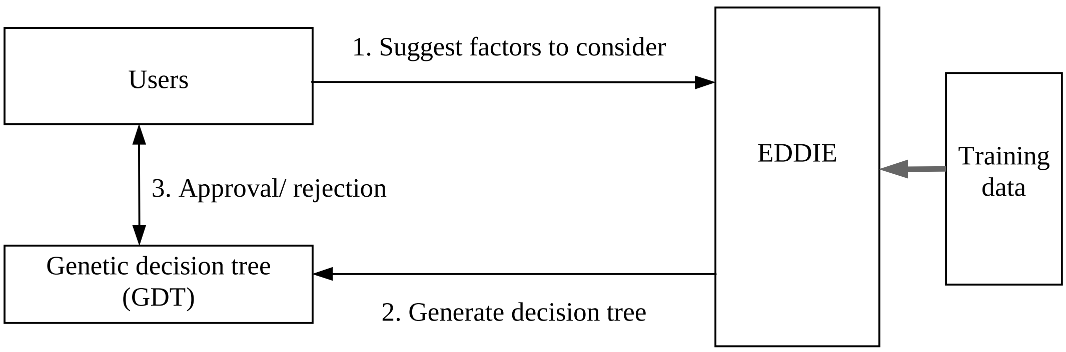

Figure 3. The role of EDDIE as a forecasting tool

Given a large database of facts about an investment opportunity, the users are responsible for suggesting a set of factors which they consider relevant. This is a point where expert knowledge can be channelled into the computer program. By using GP, what EDDIE does is to help the user to efficiently explore the space of models based on the factors suggested. With the help of historical data (often referred to as *training data*), the evolutionary mechanism will hopefully take into consideration the interactions between the factors and produce a GDT, which the human users (using their expertise) may approve or reject. This saves the users from building the rules, by looking at every possible interaction between the factors, themselves. GP contains an element of randomness, so EDDIE may produce alternative GDTs should it be asked to do so. Alternatively, the users may ask EDDIE to build alternative GDTs based on a modified set of factors. This generate and approve/reject cycle may continue until the users are satisfied with the GDTs generated or run out of time.

EDDIE-1, A GENETIC PROGRAMMING TOOL FOR HORSE RACING

Our research in EDDIE was encouraged by the success of our early work in applying EDDIE-1, an implementation of EDDIE, to horse racing, which many argue is closely related to financial forecasting [4] [17]. (In fact, de la Maza [25] also applied his results in horse racing research to finance.) In EDDIE-1, our task was to predict the winners and losers in a horse race. It is commonly believed that past performance, trainer-jockey information, etc. of the participating horses can help in predicting the outcome of a race. All functions in EDDIE-1 take two arguments and all terminals take integer values which forecast the future performance of each horse in the coming race. Each function in a GDT consists of attributes and operators. Attributes are characteristics that a horse may have, such as the position in which it finished its last race or the weight the horse is set to carry in the coming race. Operators consist of various logical and relational operations such as *equals* (one attribute equals another). For example, an expert might believe that the amount of prize money that a horse races for in some way denotes its class. In other words the better the horse, the better the class of race it competes in (hence the higher the prize money it competes for). We then code for attributes that a horse may have, such as the amount of prize money the horse competed for in its last, 2nd last and 3rd last races. We then code relevant operators, in this case relationals; <, > and =. This allows EDDIE-1 to generate functions such as,

Winner's Prize Money in Last Race < Winner's Prize Money in This Race?

This function asks if the prize money in the horse's last race is less than the prize money available in the horse's current race. As a source of information the work of a horse racing expert was used so as to propose some hypotheses as to what made one horse better than another [28]. All GDTs in the EDDIE-1 system are binary decision trees in that each function in the decision tree returns a Boolean value and then branches into one of two possible ways depending on whether the answer to the rule was true or false.

Each terminal in a GDT consists of an integer which represents a forecast for each horse in a given race. As questions in the GDT are asked, of each horse, a *walk* is performed on that GDT, and one progressively moves down the tree until a terminating forecast value is reached. The decision on that horse is made and that decision is a confidence rating on how well that horse is expected to perform in the coming race. The horse with the highest confidence value in any given race is deemed to be the most probable winner.

Once all attributes and associated operators (that code for the expert's hypotheses) have been provided EDDIE-1 is ready to generate a population of GDTs. EDDIE-1 randomly selects attributes and operators so as to generate random functions and, from these functions, generates GDTs such as the following,

(W1 < W0 (P1 = 1 34 20) (P2 < 3 28 30))

Here we see a GDT that asks if the value in attribute W1 is less than the value held in attribute W0. If this is true then the rule (P1 = 1 34 20) is fired otherwise the rule (P2 < 3 28 30) is fired. If for example W1 was not less than W0 then EDDIE-1 would try to determine if P2 was less than 3. If P2 was less than 3 then the horse being evaluated would be given a rating of 28, otherwise it would be given a rating of 30 and the evaluation would terminate.

Each GDT in the population is tested against a training database and scored depending on how well it makes the correct predictions. Those GDTs with the highest scores are said to have a high fitness with regards to solving the problem of choosing the winner of a horse race. Mates are chosen, as a function of fitness, and are paired off into a mating pool and crossover ensues. Crossover swaps a single, randomly selected, sub-tree between two parents and thus generates two new GDTs or offspring. The offspring then replace the parents in the population. The different structure of the two offspring, although probably different to their parents, may be better fitted to the task of selecting the winner of a horse race. This is the way of natural selection. In the case of EDDIE-1, we want to generate GDTs which are better suited to selecting the winner of a horse race. When the new population has been created the process then starts again with the new population and a further round of crossovers take place and so on. After a number of generations, the best GDT discovered so far is used for selecting the winner of a horse race. If the best GDT is not producing a high enough ROI then the expert's hypotheses are deemed to be wrong. The expert then allows for new rules by adding and replacing attributes and operators as the expert's knowledge deems necessary. This process continues until a good ROI is obtained.

PERFORMANCE OF EDDIE-1 ON HORSE RACING

A database of 180 real UK handicap races in 1993, for horses rated from 0 to 80 by the official handicapper, was used[2]. This type of race was selected because it is generally recognised as being difficult to predict the outcome [6]. It is therefore an excellent way of testing EDDIE-1. The 180 races were split into two databases of 150 training races and 30 testing races. The training races were used to generate GDTs to find winners and the testing races were used to test the ROI produced by GDTs generated by EDDIE-1. We used a population size of 1,000, crossover rate 90%, reproduction rate 10% (straight copying to the new population) and no mutation. EDDIE-1 was asked to run for 50 generations.

Five runs were made to evolve GDTs for selecting the winner in each horse race. The results of the five runs are shown in figure 4. The run with the best strike rate (*i.e.* run 2, which produced the greatest percentage of winners from bets made) was chosen as the system to run on the test database.

Run	Number of Bets	Number of Winning Bets	Strike Rate
1	127	44	34.6%
2	123	46	37.4%
3	119	37	31.1%
4	119	43	36.1%
5	125	40	32.0%

Figure 4. Results of teaching phase

After we obtained a decision tree we applied it to a test database of thirty races, of which the decision tree had no prior knowledge. For a test of the methodology, EDDIE-1 was pitted against three other systems; Favourites[3], Handicapper[4] and Chance[5], so as to see how it compared against them in maximising ROI. For comparison, an imaginary bet of £1 was placed on each horse that was uniquely selected, (*i.e.* EDDIE-

[2] 1993 data was used because this was the most recent data available to the authors at the time when these tests were done. Since then, like de la Maza [14], our reserach has moved on to financial applications.
[3] Betting on the favourite in the market is a common system but often yields wins with low odds and hence a low return on investment.
[4] The official handicapper is a good judge of a horses ability. His task is to burden each horse with a certain amount of weight so as to contend with horses of varying ability. The horse with the highest handicap is therefore thought to be the best horse in the race and is bet upon accordingly.
[5] Chance is a system that merely bets on any randomly chosen horse in the race. A method used by too many horse race bettors!

1 did not select two or more horses as the possible winner of the race; when the GDT gives two or more horses the same rating, the race was not bet on) and the percentage ROI using the actual odds was calculated for each system. The results are shown in figure 5. The result shows that EDDIE-1 beats all the other strategies and scored an ROI of 88.2%.

System	Number of Bets	Number of Winning Bets	Strike Rate	ROI
EDDIE-1	17	4	23.5%	88.2%
Favourites	21	6	28.6%	44.9%
Handicapper	28	3	10.7%	-63.4%
Chance	30	0	00.0%	-100.0%

Figure 5. Results of testing EDDIE-1 on 30 handicap races

EDDIE-1 produced a ROI better than systems commonly used in horse racing today. The Favourites system, as expected, chose horses that had short prices and so gave poor value for money. The Handicapper system showed that a human expert can make mistakes. Maybe if the official handicapper had used EDDIE-1 he/she may have made better decisions as to the abilities of the horses involved. The Chance system showed that in a dynamic investment market, like horse racing, one can never leave anything to chance, and that expert knowledge is always required to make a system work.

Tests on past results should always be interpreted with care [11] [18]. EDDIE-1 was developed and tested EDDIE-1 on only 180 races (input of the racing data was laborious; this prevented us from testing EDDIE-1 on more data). Therefore, we cannot rule out the possibility that luck may have played a part in the above results. However, the high return of EDDIE-1 and its superiority over the other strategies in these tests (based on real life data), plus the attractive properties of GP described earlier, suggest that EDDIE deserves more research.

EDDIE IN FINANCIAL FORECASTING

The promising results in EDDIE-1 encouraged us to extend it to financial forecasting. We took the S&P-500 data from 2 April 1963 to 2 July 1970 (1,800 trading days) as training data to generate GDTs, and tested them on data from 6 July 1970 to 25 January 1974 (900 trading days). We implemented EDDIE-3, an adaptation of EDDIE for the S&P-500 data. We used a population size of 1,200, crossover rate of 90%, reproduction rate of 10% and a mutation rate of 1%. The termination condition was 10 hours on a Pentium II (300 MHz) or 40 generations, whichever reached first. We used GP to generate rules for predicting whether the following goal is achievable at any given day:

Goal *G*: the index will rise by 4% within 63 trading days (3 months).

We used {If-then-else, And, Or, Not, <, >} as functions. In other words, unlike EDDIE-1, GDTs in EDDIE-3 are no longer binary trees. The crossover operator was modified to look after the types of the branches. Terminals were indicators, numbers or *conclusion*. Indicators were derived from rules in the finance literature, such as [2] [7] [12] [36]. Examples of indicators are:

Filter_63: Today's price − the minimum price of the previous 63 trading days;

TRB_50: Today's price − the maximum price of the previous 50 trading days

TRB_50 is derived from the *Trading Range Breakout* rule [7]. Conclusions could be either *Positive* (meaning that G is predicted to be achievable) or *Negative*.

We call a trading day a *positive position* if G holds. The whole training and test period contained roughly 50% of positive positions. We tested the accuracy of our predictions by comparing it against random decisions (which assumed that G can be achieved half of the times). If the *efficient market hypothesis* [24] holds, our rules should perform no better than random decisions. Figure 6 shows the results of 10 runs by random decisions and 10 runs by EDDIE-3. Rules generated by EDDIE-3 achieved an average accuracy of 53.59%, which is consistently better than random decisions, which achieved an accuracy of 49.47%. All the rules generated by EDDIE-3 achieved better accuracy than random runs, with one exception (random run 5 achieved an accuracy of 52.22%, which was better than 50.89%, accuracy achieved by the rule generated in run 7 by EDDIE-3).

We would also like to know the quality of the predictions. For example, if a prediction is wrong, how wrong is it? We tested the annual rate of return (ARR) by the rules generated by EDDIE-3 using the following trading behaviour:

Hypothetical trading behaviour: we assume that whenever a positive position is predicted, one unit of money was invested in a portfolio reflecting the S&P-500 index. If *the S&P index does rise by 4% within 63 days, then we sell the portfolio at a profit of 4%. If not, we sell the portfolio on the 63rd day, regardless of the price.*

We ignored transaction costs and any difference between buying and selling prices. Rules generated by EDDIE-3 were tested against random decisions in the test data. Results show that all rules generated by EDDIE-3 out-performed all random runs. EDDIE-3 achieved an ARR of 42.71% in average, compared with 38.03% by random runs. (Note that the returns are higher than 16% because the holding period by both EDDIE and random rules were often less than 63 days.)

	Random runs		**EDDIE-3**	
Runs	Accuracy	ARR	Accuracy	ARR
1	48.78%	39.21%	56.67%	42.10%
2	49.89%	37.39%	55.22%	42.57%
3	48.44%	36.30%	53.44%	45.42%
4	50.67%	38.06%	54.11%	42.14%
5	51.22%	39.85%	54.89%	42.76%
6	49.78%	37.93%	52.22%	42.16%
7	48.44%	38.40%	50.89%	41.73%
8	49.89%	37.80%	51.44%	42.58%
9	48.89%	39.34%	55.33%	43.50%
10	48.67%	36.07%	51.67%	42.13%
Mean	**49.47%**	**38.03%**	**53.59%**	**42.71%**

Figure 6. Results of testing EDDIE-3 in S&P-500

Following is one of the simplest GDTs produced by EDDIE-3 (from run 2 shown in figure 6):

```
(IF  (Filter_63  <  6.639448) THEN
        (IF  (TRB_50  >  –0.631248)
                THEN Positive
                ELSE (IF  (Filter_63 < 5.241992) THEN Positive ELSE Negative ));
        ELSE (IF  (NOT (Filter_63  >  11.944961))
                THEN Positive
                ELSE (IF  (NOT (TRB_50  >   –0.509885)) THEN Positive ELSE Negative)))
```

A number of issues are worth pointing out. Firstly, although the number of runs is relatively small, the results are significant because the amount of data tested was large and the results were consistent. Empirical results should always be interpreted with great care [11] [18], but the results at least suggested that EDDIE is a financial tool that is worth further investigation.

Secondly, the training and test data were chosen arbitrarily. It was later discovered that the testing period was actually more bullish than the training period. The fact that GDTs generated from a less bullish market performed well in the test data demonstrated the robustness of EDDIE-3.

Thirdly, it should be pointed out that our calculation of ARR assumes that funds are available whenever a positive position is predicted, and such funds have no cost when idle. While these assumptions are acceptable for evaluating the quality of the predictions, investors should not expect returns as high as 38.03% to 42.71% in reality. Exactly what ARR an investor may get by using EDDIE depends on many things, including the amount of capital available and cost of the capital.

RELATED WORK

There have been previous attempts at developing horse racing experts. SEAGUL [25] uses a version of genetic algorithm known as a genetic classifier [15] which classifies horses as winners or losers and gives a reasonable ROI. The problem with SEAGUL is that it only classifies horses as winners or losers. If, however, the horse classified as the winner does not arrive at the race track then all of the other horses, having been classified as losers, will not provide a betting opportunity. HOBBES [27] is an expert system which uses production rules to determine the winner of a horse race. However, HOBBES requires an expert to have complete knowledge of the complexities of horse racing.

Fan *et. al.* pointed out the usefulness of using ordinal data in financial forecasting [13]. Genetic programming is useful for classification in general. EDDIE-1 has been shown to classify horses into winners and losers. EDDIE-1 is superior to the above horse racing investment systems in that it can develop its own rules from hypotheses put forward by the expert. We argue that an expert, when assisted by EDDIE-1, can produce results better than an expert working on their own.

Genetic algorithms have been used in financial markets with a fair amount of success [1] [5] [9] [30] [35] [37] [39]. In Reuters, Butler used EDDIE for data mining and decision support in the financial forecasting [8]. Neely et. al. applied genetic programming to foreign exchange forecasting and reported some success [29].

EDDIE-1 and EDDIE-3 generated decision trees. In artificial intelligence, CLS [19] and ID3 (and its descendent C4.5) [32] [33] also generate decision trees. One weakness of CLS and ID3 is that when some instances with identical attributes have different outcomes in the training data, the branch which corresponds to those attributes is deemed inconclusive, even if only one instance displays a different outcome. EDDIE will be able to reflect the outcomes of the majority.

FUTURE WORK

Our goal is to develop useful tools for financial forecasting. Successful results presented in this paper encourage us to further this work. We are also exploring the possibility of incorporating in our future work constraint satisfaction techniques, which have been demonstrated to be useful in genetic algorithms [23] [38] [40].

Acknowledgement

This project is partly supported by the Research Promotion Fund (DDP540), University of Essex. Jin Li is supported by the Overseas Research Scholarship and the University of Essex Scholarship.

References

1. Agapie, Ad. & Agapie, Al., *Forecasting the economic cycles based on an extensions of the Holt-Winters model, a genetic algorithms approach*, Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr), New York City, March 1997, 96-99

2. Alexander, S.S., *Price movement in speculative markets: Trend or random walks, No. 2,* in Cootner, P. (ed.), the random character of stock market prices, MIT Press, Cambridge, MA, 1964, 338-372
3. Angeline, P. & Kinnear, K.E.Jr. (ed.), *Advances in genetic programming II*, MIT Press, 1996
4. Asch, P., Malkiel, B.G. & Quandt, R.E. *Market Efficiency in Racetrack Betting*. Journal of Business, Vol. 57, No. 2, 1984, 165-175
5. Bauer, R.J. *Genetic Algorithms and Investment Strategies*. Wiley, 1994
6. Braddock, P. *Braddock's complete guide to: Horse Race Selection and Betting*. Longman. 1983
7. Brock, W., Lakonishok, J. & LeBaron, B., *Simple technical trading rules and the stochastic properties of stock returns*, Journal of Finance, 47, 1992, 1731-1764
8. Butler, J.M., *Eddie beats the market, data mining and decision support through genetic programming*, Developments, Reuters Limited, Vol1, July 1997
9. Chen, S-H. & Yeh, C-H., *Speculative trades and financial regulations: simulations based on genetic programming*, Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr), New York City, March 1997, 123-129
10. Davis, L. (ed.) *Handbook of genetic algorithms*, Van Nostrand Einhold, 1991
11. Economist, *The mathematics of markets*, A survey of the frontiers of finance, 9th October, 1993, 1-24
12. Fama, E.F. & Blume, M.E., *Filter rules and stock-market trading*, Journal of Business 39(1), 1966, 226-241
13. Fan, D.K., Lau, K-N. & Leung, P-L. *Combining ordinal forecasting with an application in a financial market*. Journal of Forecasting, Vol. 15, No.1, Wiley, January 1996, 37-48
14. Goonatilake, S. & Treleaven, P. (ed.), *Intelligent systems for finance and business*, Wiley, New York, 1995
15. Goldberg, D.E. *Genetic Algorithms in Search*, Optimization and Machine Learning. Addison-Wesley. 1989.
16. Goldberg, D.E., Deb, K., Kargupta, H. & Harik, G. *Rapid, accurate optimization of difficult problems using fast messy genetic algorithmss*, IlliGAL Report No.93004, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, February 1993
17. Hausch, D.B. & Ziemba, W.T. *Transactions Costs, Extent of Inefficiencies, Entries and Multiple Wagers in a Racetrack Betting Model*. Management Science, Vol. 31, No. 4, 1985, 381-394
18. Hooker, J.N., Testing heuristics: we have it all wrong, Journal of Heuristics, Vol.1, No.1, 1995, 33-42
19. Hunt, E., Marin, J. & Stone, P. *Experiments in induction*, Academic Press, New York, 1966
20. Kinnear, K.E. (ed.), *Advances in genetic programming*, MIT Press, 1994
21. Koza, J.R. *Genetic Programming: on the programming of computers by means of natural selection*. MIT Press, 1992
22. Koza, J., Goldberg, D., Fogel, D. & Riolo, R. (ed.), *Procedings, First Annual Confrence on Genetic programming*, MIT Press, 1996
23. Lau, T.L. & Tsang, E.P.K., Solving the processor configuration problem with a mutation-based genetic algorithm, International Journal on Artificial Intelligence Tools (IJAIT),

(http://www.wspc.com.sg/journals/journals.html), World Scientific, Vol.6, No.4, December 1997, 567-585

24. Malkiel, B., *Efficient market hypothesis*, in Newman, P. , Milgate, M. & Eatwell, J. (eds.), New palgrave dictionary of money and finance, Macmillan, London 1992
25. de la Maza, M. *A SEAGUL Visits the Race Track*. Proceedings of the 3rd International Conference on Genetic Algorithms. Morgan Kaufman, 1989, pp. 208-212
26. McDonnell, J.R., Reynolds, R.G., & Fogel, D.B. (ed.) *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, MIT Press, 1995
27. McNatton, S.W. *HOBBES: A Predicting Expert System for Thoroughbred Horse Racing*. (Research Thesis). University of Kentucky (USA), 1994
28. Mordin, N. *Betting For a Living*. Aesculus Press, 1993
29. Neely, C., Weller, P. & Ditmar, R., Is technical analysis in the foreign exchange market profitable? a genetic programming approach, in Dunis, C. & Rustem, B. (ed.), Proceedings, Forecasting Financial Markets: Advances for Exchange Rates, Interest Rates and Asset Management, London, May 1997
30. Oussaidene, M., Chopard, B., Pictet, O. & Tomassini, M., Parallel genetic programming: an application to trading models evolution, in Koza, J., Goldberg, D., Fogel, D. & Riolo, R. (ed.), *Procedings, First Annual Confrence on Genetic programming*, MIT Press, 1996, p.357
31. Perry, J.E. *The effect of population enrichment in genetic programming*, Proc., IEEE International Conference on Neural Networks, 1994, 456-461
32. Quinlan, J.R., *Induction of decision trees*, Machine Learning, Vol.1, 1986, 81-106
33. Quinlan, J.R., *C4.5: programs for machine learning*, Morgan Kaufmann, San Mateo, 1993
34. Reeves, C.R. (ed.), *Modern heuristic techniques for combinatorial problems*, Blackwell Scientific Publishing, 1993
35. Rolf, S., Sprave, J. & Urfer, W., *Model identification and parameter estimation of ARMA models by means of evolutionary algorithms*, Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr), New York City, March 1997, 237-243
36. Sweeney, R.J., *Some new filter rule test: Methods and results*, Journal of Financial and Quantitative Analysis, 23, 1988, 285-300
37. Trigueros, J., *A nonparametric approach to pricing and hedging derivative securities via genetic regression*, Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr), New York City, March 1997, 1-7
38. Tsang, E.P.K. *Foundations of constraint satisfaction*, Academic Press, London, 1993
39. Vacca, L., *Managing options risk with genetic algorithms*, Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr), New York City, March 1997, 29-35
40. Warwick, T. & Tsang, E.P.K. *Tackling car sequencing problems using a generic genetic algorithm*, Evolutionary Computation, Vol.3, No.3, 1995, 267-298