4-2003

Cooperative Coevolution of Technical Trading Rules

Lee A. Becker
Worcester Polytechnic Institute, lab@cs.wpi.edu

Mukund Seshadri
Worcester Polytechnic Institute, mukund@cs.wpi.edu

Cooperative Coevolution of Technical Trading Rules

Lee A. Becker & Mukund Seshadri
Department of Computer Science
Worcester Polytechnic Institute
Worcester, MA 01609 USA
{lab | mukund}@cs.wpi.edu

Abstract

This paper describes how cooperative coevolution can be used for GP of technical trading rules. A number of different methods of choosing collaborators for fitness evaluation are investigated. Several of the methods outperformed, at a statistically significant level, a buy-and-hold strategy for the S&P500 on the testing period from 1990-2002, even taking into account transaction costs.

1 Introduction

Genetic Programming (GP) has been applied to a wide range of problems in finance[1]. There have been a number of attempts to use GP for acquiring technical trading rules, both for Foreign Exchange Trading [2][3] and for S&P500 market timing by Allen & Karljalainen [4] and Neely[5]. These studies were not able to establish that GP-evolved technical trading rules could outperform a buy-and-hold strategy if transaction costs were taken into consideration. Our research involves five changes from that of previous studies. First, a complexity penalizing factor was added to the fitness function. This made the evolved rules more comprehensible by reducing the expression tree size and at the same time served to avoid overfitting. Second, additional derived technical indicators were used; these included Rate of Price Change, Price Resistance Markers, and Trend Line Indicators. These derived indicators also increased comprehensibility and they allowed us to incorporate domain at the cost of introducing bias in the search. The details and effects of these first two changes are discussed elsewhere[6]. Third, the fitness function used in evolution used the number of yearly periods in the training sample in which the rule outperformed a buy-and-hold strategy, and not the return obtained for using the rule for the entire in-sample period. This favored rules that worked well more of the time under a variety of conditions rather than ran up big returns in one portion of the period. The fourth change is the focus of this paper. This is the change from a single rule to two rules one a buy rule and a sell rule. This changed the nature of the GP from evolution to coevolution.

The remainder of the paper is organized as follows. Previous work on coevolution in genetic algorithms is discussed. The details of the application of GP coevolution to our domain are described. The results of our experiments are then presented and discussed.

2 Coevolution and its use in Evolutionary Computing

According to geneticists Futuyuma & Slatkin[7] following Jansen[8], "a rigorous definition of coevolution requires that a trait in one species has evolved in response to a trait of another species, which trait was itself evolved in response to the first species." In Evolutionary Computing considerable work on coevolution has been from a competitive point of view[9],[10]. This no doubt stems, in part, from the instances of competitive coevolution found in nature, for example, between parasites and their hosts or predators and their prey, or multiple competing predators. There is also a long tradition of competitive learning in AI, dating back to the work of Samuel's on checkers[11]. Finally, there are a number of tasks to which AI has been applied that are inherently competitive, for example, games and auctions.

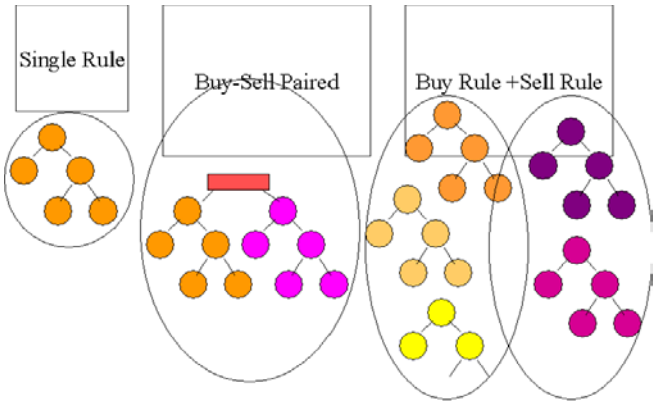
In coevolution, the fitness of individuals in one species cannot be evaluated independently of the individuals in the other species. As the species evolve, the fitness landscape of each is changing[12]; this has been referred to as "coupled fitness landscapes"[13]. A well-known issue associated with the fitness evaluation of multiagent systems is credit assignment. When the system performs well or poorly which of the individual agents gets the credit or blame. For competitive coevolution systems with only two agents, this is not a problem since one can use a complement. However, when multiple agents are cooperating to achieve a solution, this is a more serious issue. Holland's[14] work on classifier systems used the 'bucket brigade' algorithm for assigning credit.

Potter & De Jong[15] describe an architecture for cooperative coevolution, in which individuals in one species are evaluated by using collaborators from each of the other coevolving species. There exist a number of logical possibilities for choosing collaborators:

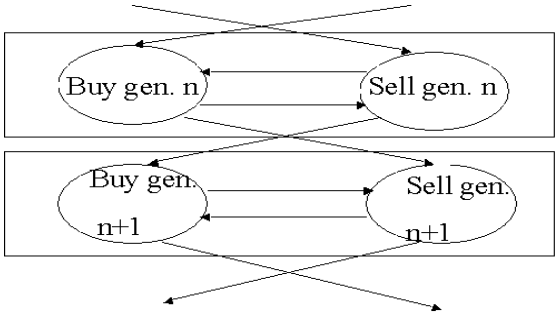
- All invididuals of each of the other species[16], but this is time-consuming and can lead to a combinatorial explosion if there are many species.
- The best individuals of each of the other species in the last generation[17].
- Randomly drawn individual(s) of each of the other species in the same generation[18][19].
- With a fixed partner (or set of partners) of the other species from the same generation[9].

3 Cooperative GP Coevolution of Technical Trading Rules

As depicted in the figure below, we started first with a single rule as our solution, we then moved to a pair consisting of a buy and a sell rule as our solution, and finally to two the best buy and the best sell rule as a solution. In the case of the pair and the last case, the buy rules and the sell rules were isolated species and could only mate, i.e. crossover, with rules of the same species. In the case of the pair, this differed from the cooperative coevolution architecture of Potter & DeJong[15] in that all members of one species did not have their fitness evaluated with same collaborators of the other species, rather each buy rule was always evaluated with the same sell rule as its collaborator. This is similar to the bipartite scheme used by Hillis[9] for competitive coevolution.



In addition to testing the benefit of having fixed collaborators, we were interested in comparing other methods for choosing collaborators. As mentioned above and depicted in the figure below the collaborators used to establish fitness for one species can come from individuals of other species of the same generation or from the previous generation or potentially with a mixture of both. In our case, we had needed only a single collaborator. We experimented using 5 randomly chosen individuals of the other species from the same generation and the 5 best individuals of the other species from the preceding generation.



In changing from using a single rule to two rules the buy vs. sell decision changes from that of the first table below to the second. The blank sells in the second table indicate the truth-value of the rule is not considered. Taking those truth-values into consideration would alter the decision behavior.

1-Rule Case	Rule	Action
In Market	T	*Do Nothing*
In Market	F	**Sell**
Out of market	T	**Buy**
Out of market	F	*Do Nothing*

2-Rule Case	Buy Rule	Sell Rule	Action
In Market		T	**Sell**
In Market		F	*Do Nothing*
Out of market	T		**Buy**
Out of market	F		*Do Nothing*

For each technical trading rule, non-leaf nodes in the expression tree allowed the logical operators (AND, OR, NOT) and the arithmetic comparison operators (>, <). The leaf nodes include the following types of technical indicators: Prices (Opening, Closing, High, Low of the current month), Moving Averages (2,3,6,10 month), Rate of Change (3,12 month), Price Resistance Markers (two previous 3-month moving average minima, two previous 3-month moving average maxima), and Trend Line Indicators (a lower resistance line extending the line connecting the 2 previous minima and a upper resistance line based on the 2 previous maxima).

We used a standard GP algorithm[20]. The software for this work used the GAlib genetic algorithm package, written by Matthew Wall at the Massachusetts Institute of Technology[21]. We used a population of 500 trees and ran for 150 generations. It was a steady state GA with half the population being replaced each

generation in the case of the single rule and the fixed partner and elitism in the case of the other two conditions. Given the difference in types of the nodes (boolean-valued vs. real-valued) there were constraints on the nodes that could be exchanged during crossover.

We used S&P500 data from January 1954 through December 2002. With the exception of the Prices, all the technical indicators are derived. The latter were pre-processed, and the need to assure the two previous minima and maxima required data from 1954-1959. We trained on data from 1960-1990 and tested on data from 1991-2002. For testing, we assumed transaction costs of .5% for each buy or sell. For months when we were out of the market, we credited the interest rate on 3-month T-bills.

4 Results and Discussion

The table below compares the performance of the best technical trading rule or pair of rules generated by each of ten runs of 150 generations. The last row gives the means.. The columns hold the IN-sample and OUT-of-sample returns for the following four experimental conditions: SINGLE rule, PAIRED rule, i.e. always evaluated with the same collaborator, best buy rule and best sell rule as evaluated with the BEST five individuals of the PREVious generation, and evaluated with five RANDOMly chosen individuals.

SINGLE		PAIRED		BEST PREV		RANDOM	
IN	OUT	IN	OUT	IN	OUT	IN	OUT
7705	2064	10816	2807	13820	3256	13832	2742
7143	2654	14957	3476	14377	3036	12757	2438
8126	3983	22074	3434	18898	2399	17746	3376
7541	2166	18467	3475	14819	2837	16117	3374
6916	2953	19861	2856	12893	3222	14747	2837
7644	3437	12374	2823	14525	2807	16249	3059
7131	3184	11530	3258	21525	2567	12911	2807
7143	2654	12667	2911	12893	2807	6233	3876
7968	2982	10646	3541	14809	2971	15243	2807
7110	3691	11818	2837	18795	2527	12077	2842
7443	**2977**	**14521**	**3142**	**15735**	**2843**	**13791**	**3016**

In considering the performance of the trees, one should note that with a buy and hold strategy if one invested $1000 in the S&P500 at the beginning of 1960, it would have grown to $5457 at the end of 1990. This was the in-sample period on which the GP was trained. For the out-of-sample period, $1000 would have grown to $2638. In terms of the mean the paired rules performed best. Both the paired rules and the best buy and best sell rules evaluated with random collaborators outperformed the buy-and-hold at 99% statistical significance. Neither the single rule nor the best buy and best sell evaluated with the best individuals of the previous generation outperformed the buy-and-hold at 95% significance. Given the large variances the difference in means between any of the experimental conditions was not statistically significant. The best performance was in the case of the PAIRED rules, which were evaluated with a fixed collaborator. It appears that in the tradeoff between diversity of collaborators and the compatibility of a fixed (set of) collaborator(s), compatibility may be more important.

Looking at the IN-sample results, all of the pairs of rules (PAIRED, BEST PREV, RANDOM) performed much better than the single rule. This is apparently due to the increased expressive power, which is partially a function of the number of nodes in the two expression trees. The greater expressive power allows the pairs of rules to fit the training data better. The large difference in performance did not carry over to the OUT-of-sample, which suggests they may have overfit the training data. Therefore we increased the complexity penalizing factor mentioned above, which results in reduced tree size for the rules in the pairs. The results of the PAIRED rules can be compared with those of the PAIRED MODIFIED rules with the increased complexity penalizing factor in the fitness function. As expected they fit the in-sample data less well, or didn't overfit it as much, but generalized better to the out-of-sample data. The combined size of two trees evolved in the PAIRED MODIFIED is approximately the same as that of the single rule, yet its performance is superior. This may be attributable to the specialization gained by separating the buy and sell rules.

SINGLE		PAIRED		PAIRED MODIF.	
IN	OUT	IN	OUT	IN	OUT
7705	2064	10816	2807	11712	3102
7143	2654	14957	3476	8394	3750
8126	3983	22074	3434	7950	3514
7541	2166	18467	3475	8260	3390
6916	2953	19861	2856	12258	3750
7644	3437	12374	2823	8039	3437
7131	3184	11530	3258	8992	3772
7143	2654	12667	2911	8394	3834
7968	2982	10646	3541	11031	3329
7110	3691	11818	2837	9746	2728
7443	**2976.8**	**14521.00**	**3141.80**	**9477.60**	**3460.60**

In conclusion, the results seem to indicate that there is value in the specialization gained by separating the buy and sell rules and cooperatively coevolving the rules.

This technique may be applicable to other domains that are traditionally thought of as having a single undifferentiated solution. For such domains at least, one should consider using fixed collaborators from the other species for fitness evaluation.

References

[1] Chen, S.-H. 2002. *Genetic Algorithms and Genetic Programming in Computational Finance.* Boston, MA: Kluwer.

[2] Neely, C., Weller, P., & Dittmar, R. 1997. Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach. *Journal of Financial and Quantitative Analysis* 32:405-26.

[3] Thomas, J. & Sycara, K. 1999. The Importance of Simplicity and Validation in Genetic Programming for Data Mining in Financial Data. *Proceedings of the joint AAAI-1999 and GECCO-1999 Workshop on Data Mining with Evolutionary Algorithms*, July, 1999.

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

[5] Neely, C.J. 1999. Risk-Adjusted, Ex Ante, Optimal, Technical Trading Rules in Equity Markets. Working Paper 199-015D. Federal Reserve Bank of St. Louis. To appear in *International Review of Economics and Finance.*

[6]Becker, L.A. & Seshadri, M. 2003. Comprehensibility & Overfitting Avoidance in Genetic Programming for Technical Trading Rules Worcester Polytechnic Institute, Computer Science Technical Report WPI-CS-TR-03-09.

[7] Futuyma, D.J. & Slatkin, M. 1983. *Coevolution.* Sunderland, MA: Sainauer.

[8] Jansen, D.H. 1980. When is it coevolution? *Evolution* 34:611-612.

[9] Hillis, D.W. 1990. Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D* 42:228-234.

[10] Angeline, P.J. & Pollock, J.B. 1993. Competitive Environments Evolve Better Solutions for Complex Tasks. In *Proceedings of the Fifth International Conference on Genetic Algorithms Mining*, 264-270. San Mateo, CA: Morgan Kaufmann.

[11] Samuel, A. 1959. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development* 3:210-229.

[12] DeJong, K. & Spears, W. 1993. On The State of Evolutionary Computation. In *Proceedings of the Fifth International Conference on Genetic Algorithms Mining*, 618-623. San Mateo, CA: Morgan Kaufmann.

[13] Kauffman, S.A. & Johnsen. 1991. Coevolution to the edge of chaos: Coupled fitness, landscapes, poised states, and co-evolutionary avalanches. In C.G. Langton, C. Taylor, J.D. Farmer, & S. Rasmussen (Eds.), *Artificial Life II, SFI Studies in the Sciences of Complexity,* 10: 325-369. Addison-Wesley.

[14] Holland, J.H. 1986. Escaping brittleness: The possibilities of general purpose learning algorithms applied to parallel rule-based systems. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, (Eds.), *Machine Learning,* Volume 2: 593-623.Los Altos: Morgan Kaufman.

[15] Potter, M.A. & DeJong, K. 2000. Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. *Evolutionary Computation* 8(1):1-29.

[16] Axelrod, R. 1987. The evolution of strategies in the iterated prisoner's dilemma. In L.D. Davis, (Ed.), *Genetic Algorithms and Simulated Annealing*, 32-41. New York, Morgan Kaufmann.

[17] Cliff, D. & G. F. Miller, G.F. 1995. Tracking the Red Queen: Measurements of adaptive progress in co-evolutionary simulations". In Moran, F., Moreno, A., Merelo, J.J. & Cachon, P. (Eds.) *Advances in Artificial Life: Proceedings of the Third European Conference on Artificial Life (ECAL95). Lecture Notes in Artificial Intelligence 929*: 200-218. Springer Verlag.

[18] Price, T.C. 1997. Using co-evolutionary programming to simulate strategic behavior in markets. *Journal of Evolutionary Economics* 7:219-254.

[19] Phelps, S., Parsons, S., McBurney, P., & Sklar, E. 2002. Co-evolution of Auction Mechanisms and Trading Strategies: Towards a Novel Approach to Microeconomic Design. In *Proceedings of ECOMAS-2002 Workshop on Evolutionary Computation in Multi-Agent Systems, at Genetic and Evolutionary Computation Conference (GECCO-2002).*

[20] Koza, J. R. 1992. *Genetic Programming: On the Programming of Computers by means of Natural Selection*. Cambridge, MA: MIT Press.

[21] http: //lancet.mit.edu/ga/