Multiobjective Algorithms for Financial Trading

Multiobjective Out-trades Single-Objective

Dome Lohpetch
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, UK
dl73@hw.ac.uk

David Corne
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, UK
d.w.corne@hw.ac.uk

***Abstract*— Genetic programming (GP) is increasingly investigated in finance and economics. One area of study is its use to discover effective rules for technical trading in the context of a portfolio of equities (or an *index*). Early work in this area used GP to find rules that were profitable, but were nevertheless outperformed by the simple "buy and hold" (B&H) strategy. Attempts since then tend to report similar findings, except for a handful of cases where GP methods have been found to outperform B&H. Recent work has clarified that robust outperformance of B&H depends on, mainly, the adoption of a relatively infrequent trading strategy (e.g. monthly), as well as a range of factors that amount to sound engineering of the GP grammar and the validation strategy. Here we add a comprehensive study of multiobjective approaches to this investigation, and find that multiobjective strategies provide even more robustness in outperforming B&H, even in the context of more frequent (e.g. weekly) trading decisions.**

Keywords-financial trading; genetic programming; multiobjective algorithms

I. INTRODUCTION

There are many opportunities in the areas of financial markets for machine learning and optimization methods [1], and applications of Genetic Programming (GP) [2—4], as well as other areas of evolutionary computation, are now common in this area [5—12] . GP is particularly popular in this field, with many studies reporting a variety of different GP applications in finance [5—12]. Our focus is the area known commonly as 'financial trading', but more strictly termed as *technical analysis* [13—16]. Technical analysis is the name given to the general area of study in which one attempts to forecast the future direction of equity prices, guided either largely or entirely by the patterns that are revealed (or perceived) from historical equity price data. Technical analysis therefore relies deeply on the notion that discernible or discoverable patterns and trends exist in equity price fluctuations over time, and that these can be exploited to predict price movements in the near future.

Simple, commonly used, and reasonably successful approaches to technical analysis use measures such as moving averages (the mean price for a given stock or index over a given recent period), relative strength indicators (a function of the ratio of recent upward movements to recent downward movements), and several others. A standard approach to engineering GP applications in this area is for GP trees to express trading rules, which combine technical indicator 'primitives' with other mathematical operations. Such a rule constitutes a 'signal', which may be interpreted, for example, as a recommendation to buy if the signal is above a threshold. The seminal attempts to use GP in this way were by Chen and Yeh [5] and by Allen and Karjalainen [7]. These initial studies, and many others that followed, tend to report that GP is able to find rules that are indeed profitable on unseen future data. But, rarely in such work is it also shown that such rules provide greater returns than a standard "buy-and-hold" (B&H) trading approach. The B&H strategy is: for a given trading period, buy the stock at the beginning of the period, and sell it at the end – hence, always a good strategy in an upwardly moving market, and far simpler than using technical indicators. Similarly, where rules are evaluated in a downward moving market, it is not generally clear if the rules developed by GP can outperform the simple strategy of investing in Treasury bills or the equivalent (i.e. a safe investment with a small but guaranteed return).

Nevertheless, there is a small amount of research in this area that has reported finding rules that outperform B&H [8,17,18]. Particularly notable early work was that by Becker and Seshadri [19—21], who modified Allen and Karjalainen's approach [7]. One of Becker & Seshadri's alterations was to adopt monthly trading rather than (as in Allen & Karjalainen) daily trading; others involved modifying the fitness function to reward consistency in performance, as well as simplicity of rules (and hence avoid overfitting). This work was clarified and replicated by Lohpetch & Corne in [24], who also established that adopting a training/validation/test approach also led to significant improvements in robustness. Later, in [25], Lohpetch & Corne evaluated performance in the context of each of monthly, weekly, and daily trading, and found a clear trend towards reduced robustness in terms of profitability (and outperformance of B&H) as trading interventions became more frequent. This is intuitively understandable, of course, in terms of the relative volatility of the market as seen at shorter timescales – it is easier, for example, for a short window of data to lead to misleading rules than a longer window of data, which smooths out the variations. This was confirmed recently by Gorse [26] on the same data source (a variety of periods of S&P Index data between 1970 and 2010), using a reinforcement learning method, rather than GP, to underpin the trading decisions. Then again, this finding could also be a commentary on the machine learning techniques involved, indicating that robust weekly and daily price patterns are particularly challenging to reliably detect, which allows for the possibility that improved engineering of the GP approach (say) might still lead to a profitable weekly or daily trader.

In this paper we aim to find out if improved performance at more frequent trading interventions can be achieved by taking a multiobjective approach. Specifically we engineer our GP method to be guided by multiobjective characterisation of a trading rule's fitness, and contribute the following beyond what has so far been done in this area:

- Comparison of many different multiobjective approaches to trading with each other, with earlier single-objective approaches, and with B&H.
- Comparison in the context of many different data splits (i.e. across a wide range of market environments)

The remainder is set out as follows. Section II provides a review of related work, recapping a little of the history reported in this section, but considering what has been done so far with multiobjective algorithms. Section III then sets out the basic GP approach and describes a number of ways of casting this as a multiobjective problem. In section IV we describe the experimental setup (data periods, training/validation/test splits, algorithms compared) and then present summaries of the results. We conclude in section V.

II. A Brief Account of Related Work

Previous studies using Evolutionary Algorithms (EA), including GP, for discovering technical trading rules has almost exclusively used a single objective approach; they can be classified into two main groups: the profit maximizing approach, and the risk adjustment approach. The profit maximizing approach focuses only on return, without concerning the associated risk [7,11,12,19—21], whereas the risk adjustment approach includes a risk-adjustment in the fitness evaluation process [6,8,10,17,18,23]. In brief, as we have indicated, the first significant attempt to discover trading rules was by Allen and Karjarleinen [7], which we will refer to hereafter as AK. In this study, GP was used with a single objective, which was to maximize excess return over B&H, to acquire technical trading rules for daily trading on the S&P 500 index from 1928 to 1995. Although Allen and Karjarleinen's study did not show consistent excess return compared with simple B&H, it provided inspiration for applying EAs in financial markets. Allen and Karjarleinen's fitness function did not consider any risk adjustment, so Neely [18] modified Allen and Karjarleinen's work by including four risk-adjustment methods, such as the Sharpe ratio and the X^* measure in his single objective fitness function. However, this modification still was not enough to outperform B&H. After the previous two studies, Becker and Seshadri [19—21] succeeded in finding GP-evolved technical trading rules, which could outperform B&H. Becker and Seshadri made plenty of changes to the method used by Allen and Karjarleinen, including the use of monthly data instead of daily data, reducing the GP operator set, and increasing the number of derived technical indicators among the GP operands.

As we have indicated, in two recent studies [24,25], Lohpetch & Corne have focused on indicating how the key details of the Becker and Seshadri may be replicated, and have carried out additional investigations which point towards guidelines for generating results that robustly outperform B&H; however these studies again used only a single objective based on excess return. In other work that has attempted to look specifically at outperforming B&H, Potvin et al [12], for example, showed that GP trading rules can be generally beneficial in falling or stable markets – but this is not particularly impressive, since B&H is naturally poor in such markets. In another line of work, risk metrics such as the Sharpe ratio [22] have been included in rules (or in their evaluation). The Sharpe ratio modifies the return of a rule by considering its variation in return over the period. So a rule that achieves a very high return, but with much variation over the period, may be evaluated as the same quality as a rule that has a low return, but is very consistent over the period (e.g. producing a similar positive return every month for a year). Such metrics typically reduce the fitness of rules that promote trading in volatile conditions, and therefore lead to rules more likely to be applied by investors. For instance, building on Fyfe et al. [6] (whose results were not superior to B&H), Marney et al. [8,17] made headway by including risk metrics, as did Cheng & Khai [10], however these attempts did not actually produce usable rules that compared well in comparison to B&H.

Considering such measures of risk, it is worth pointing out that the improvements made by Becker& Seshadri over the Allen and Karjarleinen approach include one which effectively transforms the fitness function into a measure of risk – this is the performance consistency measure, that we discuss later. It was confirmed in [24,25] that this was an essential element in outperforming B&H on the data in question. Overall, the incorporation of risk measures is clearly a promising thread of work in the automated trading context.

Next we turn to multiobjective algorithms, in which we can see again that risk measures are starting to be explored. With the rising use of EAs for multi-objective optimization (MOEAs), there have recently been several applications in the area of finance. Most of these have concentrated on investment portfolio optimization problems [29—36], which are the most popular applications of MOEAs to solve problems in this area [30] – this is essentially a problem of finding an ideal collection of companies/equities to constitute a trading portfolio. However there are two recent studies which used an MOEA to generate technical trading rules. The first of these [37] used an MOEA to generate technical trading rules with two objectives: to maximize total return and to minimize semi-variance as a risk factor, and the authors concluded that the higher return in the training data tended to correspond to larger volatility in the returns generated on the test data. In the second of these studies [27], the authors used Multi-Objective Particle Swarm Optimization method to yield technical trading rules with two objectives: percent profit and Sharpe ratio [28] in 3 contiguous training and testing periods. Both the best and average performances of this study were able to outperform all 5 technical indicators in the training periods, and they also beat the B&H strategy in two periods out of three. On test data, the performance of the best points among 30 Pareto fronts beat all indicators in both objective functions, but failed to beat the B&H strategy.

In the few studies so far using MOEA for technical trading rules, the results therefore seem inconclusive, however it certainly seems worth more investigation. In this paper, we therefore investigate a number of alternative ways of characterizing technical trading as a multi-objective task, engineering in the context of the GP method in [24,25] that has its roots in [7,19—21].

III. THE GP APPROACH AND MULTI-OBJECTIVE CHARACTERIZATIONS OF THE PROBLEM

Our basic approach uses standard GP, with a function set comprising arithmetic, Boolean and relational operators, while the terminal set comprises a collection of basic financial technical indicators, along with real and Boolean constants, and real-valued variables (such as *equity price*). An example of a rule specified by a GP tree (in fact a rule found in [19]) is in Figure 1. The abbreviations of technical indicators in Figure 1 can be found in section III-A.

If we are doing weekly trading (for example) then a rule such as that in Figure 1 is to be interpreted as follows. The rule essentially makes an assertion: "The 3-week moving average (MA-3) is less than the lower trend line (LRTL) and the 2-week moving average (MA-2) is less than the 10-week moving average (MA-10) and the lower trend line (LRTL) is greater than the second previous 3-week moving average maxima (MX-2)." In the context of a particular trading intervention, we evaluate this assertion and find that it is either true or false. This then corresponds to a trading action as follows. If we are currently out of the market, and the assertion is true, then *buy*; if we are currently in the market and the assertion is false, then *sell*. The overall procedure assumes a fixed amount is to be invested (e.g. $1,000) whenever the decision is *buy*.

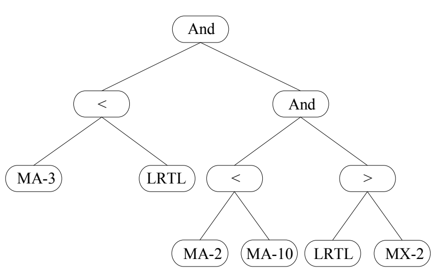

Figure 1. An example trading rule in the GP population

The next few subsections explain the approach in further detail, albeit briefly. The data we use (as in [7,19—21,24,25,26] is the Standard and Poors 500 (S&P 500) index – a fixed set of 500 stocks which aggregate to daily price indicators (opening, closing, high, low). When considering weekly and monthly trading, the opening price (for example) for a week or a month is the opening price on the first day of that week or month.

A. GP Function and Terminal Sets

The function set is: AND, OR and NOT, together with the relational operators ">" and "<". We use strongly typed GP, to automatically enforce appropriate inputs for each operator. The terminal set is as follows, where 'unit', depending on the experiment, is either day, week or month

- Opening, closing, high, low prices for the current unit (OPRICE, CPRICE, HRPICE, LPRICE)
- Volume for current unit (VOL) and volume for previous unit (VOL-LAG);
- 2,3,6 and 10-unit moving averages (MA-2, MA-3, MA-6 and MA-10);
- Rate of change indicator: 3-unit and 12-unit (ROC-3 and ROC-12);
- Price Resistance indicators: the two previous 3-unit moving average minima (MN-1 and MN-2), and the two previous 3-unit moving average maxima (MX-1 and MX-2);
- Trend Line Indicators: a lower resistance line (LRTL) based on the slope of the two previous minima; an upper resistance line (URTL) based on the slope of the two previous maxima.

Please note that 'unit' can be: day for daily data, week for weekly data and month for monthly data.

The l-unit moving average at time m is the mean of the closing prices of the l units from m back to $m-(l-1)$. The l-unit rate of change indicator measured at m is:

$$100(p(m)-p(m-(l-1)))/p(m-(l-1))$$

where $p(x)$ indicates the closing price for time x. Previous maxima MX1 and MX2 are obtained by considering the 3-unit moving averages at each point in the previous 12 units. Of the two highest values, the one closest in time to the current is MX1, and the other is MX2. The two previous minima are similarly defined. Finally, to identify trend line indicators, the two previous maxima are used to define a line in the obvious way, and the extrapolated value of that line from the current time becomes the upper trend line indicator; the lower trend line indicator is defined similarly by using the two previous minima.

B. *Main Details in Calculating the Fitness Objectives*

In the single-objective formulations previously used, the fitness function has three elements. The first is 'excess return', which is the excess of the return over that of B&H. strategy. This is:

$$E = r - r_{bh} \qquad \text{(eq. 1)}$$

where r is the return on an investment of $1,000, and r_{bh} is the corresponding return from B&H. To calculate r we use [7,19—21]:

$$r = \sum_{t=1}^{T} r_t I_b(t) + \sum_{t=1}^{T} r_f(t) I_s(t) + n \ln\left(\frac{1-c}{1+c}\right) \qquad \text{(eq. 2)}$$

where:

$$r_t = \log P_t - \log P_{t-1} \qquad \text{(eq. 3)}$$

which indicates the continuously compounded return, where P_t is the price at t. $I_b(t)$ is 1 if the rule indicates *buy* at time t, 0 otherwise. Similarly defined is the sell signal, $I_s(t)$. The first component of r hence calculates return on investment over the times when the investor is in the market. The second component, $r_f(t)$ indicates the risk-free return, which is taken for any particular day t from US Treasury bill data (available from http://research.stlouisfed.org/fred/data/irates/tb3ms). Hence, the second component represents time out of market, assuming that the investor's funds are earning a standard risk-free interest. Finally, the third component corrects for transaction costs, where c denotes the one-way transaction cost. The cost of a single transaction is assumed to be 0.5% – e.g. $5 for a transaction of volume $1,000. The number of transactions during the period is n. This component estimates the compounded loss from the cost of transactions.

In the single-objective approach, the second main part of the fitness function, r_{bh}, is calculated as:

$$r_{bh} = \sum r_t + \ln\left(\frac{1-c}{1+c}\right) \qquad \text{(eq. 4)}$$

where r_t is as indicated above, calculating the return of buying at the first day and selling at the last day of the period.

The excess return E, calculated as described, was originally the objective function in [7], but improvements in [19,21] arose from two adjustments. One of these is an adjustment to fitness according to the size of the tree. Given a actual fitness value f, the adjusted fitness value modified by the complexity-penalizing factor can be computed as following equation:

$$\textit{adjusted fitness value} = \frac{5f}{\max(5,\ \textit{actual-depth})} \qquad \text{(eq. 5)}$$

where *actual-depth* is the depth of the tree being evaluated, and the constant 5 is a 'desired' depth.

The other aspect of the fitness function which led to more consistent results was what we call *Performance Consistency* (PC). E is calculated for each successive period of K units covering the entire test period. The value returned is simply the number of these periods for which E was greater than both the corresponding B&H return and the risk-free return.

Putting all these together, the most successful single objective approach so far calculates f, the PC-based fitness (i.e. a number from 0 to X, where there were X periods covering the test data), and adjusts it for tree complexity, returning $5f/\max(5, \textit{actual-depth})$ as the fitness of a rule.

C. *Single and Multi-Objective Approaches used in this study*

The experiments reported in this paper refer to three separate configurations of single-objective approach. That is, in a single-objective experiment, the objective used is one of the following three:

- CMR – Market Return (essentially equivalent to excess return)
- CPC_LK12 – Performance Consistency with 12-unit periods.
- CPC_LK24 – Performance Consistency with 24-unit periods.

In each case, the initial "C" indicates that fitness is modified by the complexity-penalizing factor as indicated in eq. 5. In CPC_LK12, for example, and an experiment involving monthly trading which the unseen test data cover a 60-month period, fitness (before the complexity modification) is either 0, 1, 2, 3, 4 or 5, according to in how many of the separate 12-month periods the rule was able to outperform both B&H and risk-free return.

In the multi-objective approaches tested, each used a combination of either two or three objectives, chosen from the above as well as the additional list of objectives below:

- MR – Market Return
- 2MR – 2 Separated Market Return (Divide the period into 2 sub-periods and use MR as fitness value for each one) – where used, obviously this counts as two objectives
- PC_LK12 – Performance Consistency with 12-unit periods.
- PC_LK24 – Performance Consistency with 24-unit periods.
- CXP – Complexity Penalizing Factor – standalone measure of the tree complexity – simply the depth of the tree.
- 2CMR – 2MR weighted by complexity penalizing factor

In total we test seven distinct multiobjective approaches, as follows:

- MR-CXP – 2 objectives, MR and CXP
- 2MR-CXP – 3 objectives: MR for two sub-periods, and CXP
- MR-PC_LK12-CXP – 3 objectives, as indicated.
- MR-PC_LK24-CXP – 3 objectives, as indicated.

- 2CMR – 2 objectives, as 2MR, but both objectives complexity penalized
- CMR-CPC_LK12 – 2 objectives, as indicated.
- CMR-CPC_LK24 – 2 objectives, as indicated.

D. Some details of the Multi-Objective Approach

The overall multi-objective algorithmic approach we use in this study was classic and straightforward, essentially similar in all details to NSGA II as described in [38]. An archive was maintained, with fixed size 10, containing the approximation to the Pareto front so far, and using crowding distance as the arbiter when new individuals were generated that would otherwise overflow the archive. However the key element we need to describe is how the returned Pareto front was exploited to make trading decisions on unseen test data. This was done as follows.

Experiments were done in the context of three continuous trading periods, called training, validation, and evaluation. In all single-objective experiments reported here, GP evolves a rule, guided by performance in the training period, but at the same time recording (but not using) the performance of each evaluated rule on the validation period. The rule that performed best on the validation period is then evaluated over the evaluation period, and this is the result returned and summarised in our results section (how this is handled for multiobjective algorithms is discussed next). This regime of training/validation/evaluation (which we call regime 2) was found clearly more robust in previous work [24,25] than a straightforward training/evaluation split in which we use the rule that was best on training data.

In the context of multiobjective approaches, all used the same configuration as in single-objective experiments, except that instead of evaluating a single rule, we always use a set of rules, and this set is the current content of the non-dominated archive. We test two approaches: in the majority-voting approach (MJV) the set of rules is used simply to make a majority decision: each individual rule either signals 'buy' or 'sell' in the current environment. The signal from the set is the majority decision; in the case of a tie, we simply use the decision made by the rule (breaking other ties randomly) that was flagged as achieving the best return (MR) on the validation data. For comparison, we also test a basic 'Normal Trading' approach, in which the result of a set of rules is taken to be the average result of the rules treated individually. This corresponds, in the trading context, to giving each rule an equal share of the money to invest – given transaction costs that are strictly in proportion to the amount traded, this arguably yields valid results, though (as we expected and as we will see) is not a particularly effective approach.

E. Operators, Initialization and other GP parameters

We used the standard GP mutation operators 'grow', 'shrink', 'switch' and 'cycle' [3], and standard subtree-swap crossover [2]. No constraint was placed on tree size (other than selection pressure from one or more of the objectives in some configurations), however the population was initialized by growing trees to a maximum depth of 5. The population size was always 500 (with archive maximum size 10 in the multi-objective approaches, though archives tended to be much smaller). As discussed, the multiobjective method was NSGA-II; In single-objective run, in each generation, the current best was copied into the next generation, and the rest were the produced by crossover of two parents (probability 0.7) or mutation of a single parent, with parents selected via rank-based selection. Each run on training data continued for 50 generations.

F. Data splits

In this context, a data split indicates a particular period of time, itself divided into three contiguous periods of training, validation and test data. In much of the earlier work in this thread of research, only one or a small number of data splits were used, which does not lead to results that are particularly robust in terms of the conclusions that can be made. In [24,25] we used 4 different data splits for each of daily, weekly and monthly trading. In this paper, we use eight different data splits for each of daily, weekly and monthly data (i.e. 24 different data splits altogether). In this way, for each trading frequency, we hope to underpin a good level of robustness in any claims about trends that emerge from the results. The data splits used need considerable space to specify, so we point the reader to their full description within supplementary material at http://doiop.com/data_splits .

IV. Results

Every combination of algorithm/data-split/regime including, where applicable, MJV or normal trading was repeated 20 times independently, and the results are summarised here. A complete collection of results is available from the authors, and also in part at http://doiop.com/data_splits. Here we summarise by providing graphs that indicate (a) the relative performance of the multiobjective strategies compared with single-objective strategies, and (b) the relative performance of the multiobjective strategies in comparison with B&H. In each case, the display of results is oriented to show the picture across all data splits relevant to the experiment.

Figure 2 shows the performance of all multi-objective methods in comparison (implicitly) with single-objective configurations for monthly trading. For example, 2CMR (MJV) achieves 75% on this plot. This means that in 75% (i.e. 15) of 20 independent tests, where each test includes an experiment of both this method (2CMR (MJV)) and all three single-objective approaches on each of the 8 monthly data splits, 2CMR (MJV) outperformed *all* of the single-objective methods.

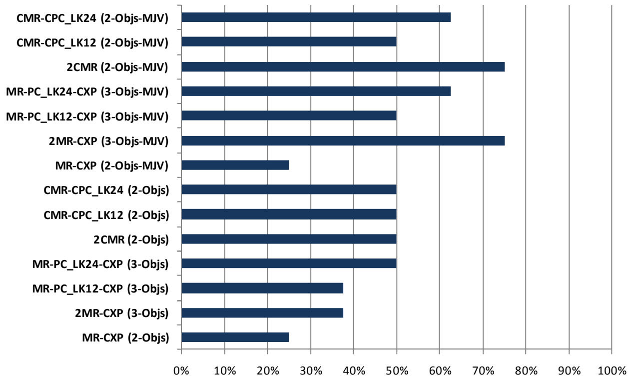

Figure 2. Performance of multi-objective methods in comparison with single-objective configurations for monthly trading splits. E.g. 2CMR (MJV) achieves 75%, meaning that: averaged over all 8 monthly data splits, 2CMR (MJV), in 75% of the 20 independent tests, outperformed *all* of the single-objective methods.

The relative performance of multiobjective vs single objective does not seem highly convincing from the figure at first sight, since there is no majority that extend beyond the 50% mark. However, recall that each single multiobjective approach is compared against *all* single-objective methods for each bar. Also, it is quite clear that the majority-voting (MJV) techniques, the top 7 methods in this plot (and all plots) consistently perform strongly against single-objective methods, except for MR-CXP.

Turning to performance in relation to Buy-and-Hold, Figure 3 shows, for each multiobjective approach, how often (averaged over the 8 monthly splits) the return from trading over the evaluation period outperformed buy-and-hold. This is a fairly convincing result for multiobjective approaches in monthly trading.

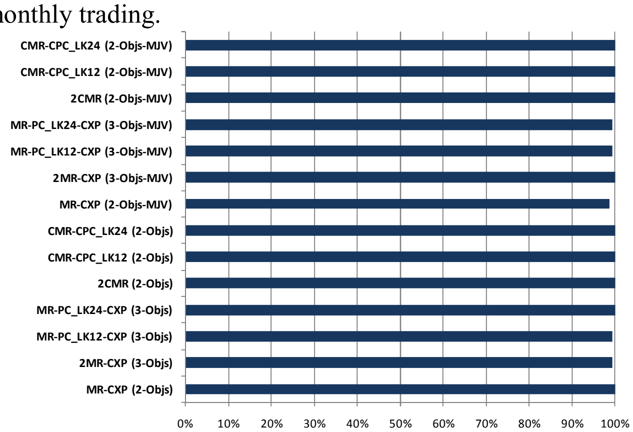

Figure 3. Performance of multi-objective methods in comparison with the buy-and-hold strategy for monthly trading splits. E.g. most algorithms achieve 100%, meaning that: averaged over all 8 monthly data splits, the method out-performed B&H in all of the 20 independent tests.

Turning now to more frequent trading interventions, Figures 4 and 5 show the results (corresponding respectively to Figures 2 and 3) over the weekly trading splits.

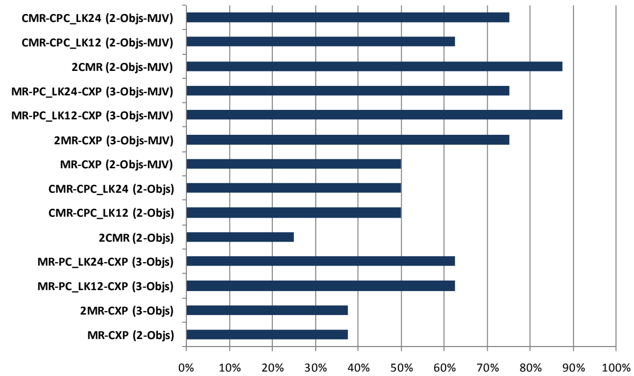

Figure 4. Performance of multi-objective methods in comparison with single-objective configurations for weekly trading splits.

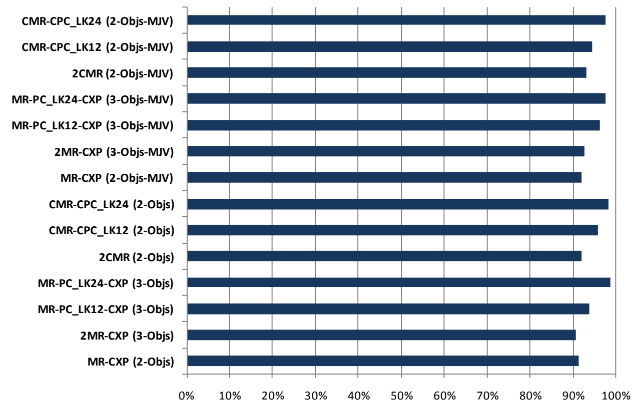

Figure 5. Performance of multi-objective methods in comparison with the buy-and-hold strategy for weekly trading splits.

The superiority of multiobjective methods over single objective methods is much clearer in this context. It seems clear that multiobjective methods can somehow spread the risk (associated with more volatility) over multiple rules in a way that boosts performance; this notion is reflected in the observation that 'normal trading' does better in this context than it did in the monthly trading context.

Finally, we see the corresponding results for daily trading regimes in Figures 6 and 7. These figures clearly tell a different story. With potential for daily trading intervention, Figure 7 reveals that the multiobjective methods that we have tested can outperform B&H at best roughly half the time. However it should still be recognised that this in itself is not particularly poor performance – all of the 'B&H' results compared against in this paper have the advantage, entirely unfair in reality, of being able to choose between either the B&H return or the risk-free return, whichever would have generated most over the trading period. What is certainly clear, however, from Figure 6 is that the multiobjective methods we have studied are outperformed by the single objective methods in the daily trading context.

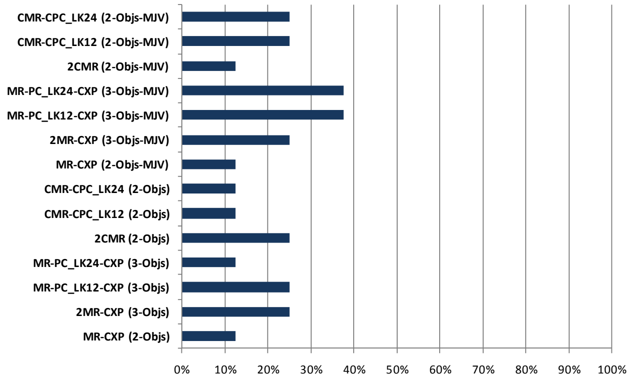

Figure 6. Performance of multi-objective methods in comparison with single-objective configurations for daily trading splits.

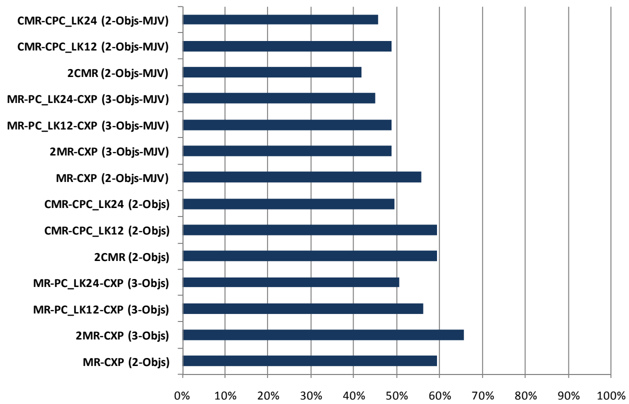

Figure 7. Performance of multi-objective methods in comparison with the buy-and-hold strategy for daily trading splits.

When we consider these results in terms of the relative performance of the seven different ways of characterising performance with multiple objectives, there seem to be only two fairly clear observations that emerge. These are best revealed by appeal to figures 3, 5 and 7, in which performance is compared directly with B&H, rather than confounded by the variations in performance of the single-objective methods. First, as already noted, the majority voting approach tends to lead to better performance for monthly and weekly trading, although in daily trading the 'normal trading' approach, simply spreading the investment decision over each rule, seems clearly to have an advantage, and this seems independent of other aspects of the multiobjective configuration. We expect to achieve a better understanding of this in future work, which will look at variations on the majority voting approach, which give individual rules different weights according to measures of the risk and return associated with them. However, turning to comparison of the seven objective combinations, one clear observation is that MR-CXP, in monthly and weekly trading, was always the least effective approach. It is notable that MR-CXP is the only one of the multiobjective configurations that did not include an objective that related to risk. When PC_LK12 or PC_LK24 (or their complexity modified variants) were involved, the corresponding objectives essentially gave a measure of the variance in performance of the rule over the trading period, which is precisely what is attempted in a risk-based metric such as the Sharpe ratio. Similarly, when 2MR or 2CMR were involved, these broke down the (perhaps modified) excess return measure into two separate periods, again revealing information about variation in performance and hence providing a simple estimate of the rule's (or ruleset's) risk.

It therefore seems clear that risk-oriented measures are valuable in this context for monhly or weekly trading, and this value is exploited well by the multi-objective approaches that incorporated such measures. The reverse seems to be the case, however, in the more volatile context of daily trading. We speculate that this may be because large variations in the returns of a particular rule (or ruleset) are far more frequent in the daily context; so, metrics that estimate that variation, especially over smaller timescales, are likely to be more affected by noise. However it is worth pointing out that our experimental setup may mitigate against finding robust rules for daily trading, simply because the technical indicator primitives we use are forced, in that case, to consider only relatively brief timescales (see section III.A). For example, rules can refer at most to a 10-day moving average.

V. Conclusion

In this paper we have investigated the use of several configurations of multiobjective GP algorithms for developing trading rules for trading on the stock market, specifically in association with the S&P 500 portfolio of companies. This continues a thread of research that dates back to Allen and Karjeleinen's work [7] which showed that GP could produce profitable rules, and has since been supplemented by many efforts, increasingly aimed at understanding how to configure GP to provide robust and convincing performance (in terms of outperforming buy-and-hold and risk-free return strategies). Building on earlier work that indicated robust and convincing performance could be achieved in the context of monthly trading, but less so for weekly trading (and rarely for daily trading), we show that multiobjective approaches, especially when using a majority voting approach, are able to generally outperform single-objective approaches, as well as robustly outperform B&H and risk-free return. This is the case for monthly trading, and even more clearly so for weekly trading, in which improving on the performance shown for single-objective approaches in [24], we see good evidence that multiobjective GP for evolving trading rules seems clearly viable in the context of weekly trading. These results are to some extent underpinned in their robustness by the fact that they emerge after considering 8 different data splits that cover a range of different trading environments.

In comparing each of seven different characterizations of the problem into multiple objectives, we find that the worst performance (again, considering only the monthly and weekly cases) came from the only configuration that involved no objective related to measuring risk – i.e. variation in performance over the trading period. Similarly, we find that, in the monthly and weekly scenarios, the approach in which trading was done via majority vote over a set of non-dominated rules was more effective than the approach in which performance was effectively averaged over the rules in the set. Again, these observations were reversed in the case of daily trading, which we speculate is due to the increased volatility in that scenario, confounding any clear measures of risk. However it could also be that our approach to evolving rules for daily trading was compromised by forcing the rules to reason over only short-timescale moving average and similar indicators.

REFERENCES

[1] Brabazon, A. & O'Neill, M. (2005). Biologically Inspired Algorithms for Financial Modelling (Natural Computing Series), New York: Springer.

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

[3] Angeline, P. J. (1996). Genetic Programming's Continued Evolution. *Advances in Genetic Programming*, Vol. 2, editor, P. J. Angeline and K. Kinnear, Cambridge, MA: MIT Press, pp. 89-110.

[4] Banzhaf, W., Nordin, P., Keller, R. E. & Francone, F. D. (1998). *Genetic Programming - An Introduction*: On the Automatic Evolution of Computer Programs and Its Applications, San Francisco: Morgan Kaufmann

[5] Chen, S. H. & Yeh, C. (1996). Toward a Computable Approach to the Efficient Market Hypothesis: An Application of Genetic Programming, *J. of Econ Dynamics & Control*, 21: 1043-1063.

[6] Fyfe, C., Marney, J. P. & Tarbert, H. (1999). Technical Trading versus Market Efficiency: A Genetic Programming Approach, *Applied Financial Economics*, 9: 183-191.

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

[8] Marney, J. P., Fyfe, C., Tarbert, H. & Miller, D. (2001). Risk Adjusted Returns to Technical Trading Rules: A Genetic Programming Approach, *Computing in Economics and Finance*, Soc. for Computational Economics, Yale University, USA, June 2001.

[9] Chen, S. H. (2002). Genetic Algorithms and Genetic Programming in Computational Finance, Boston, MA: Kluwer.

[10] Cheng, S. L. & Khai, Y. L. (2002). GP-Based Optimisation of Technical Trading Indicators and Profitability in FX Market, *Proc. 9th ICONIP* , Vol. 3, pp. 1159-1163.

[11] Farnsworth, G. V., Kelly, J. A., Othling, A. & Pryor, R. (2004). Successful Technical Trading Agents Using Genetic Programming, SANDIA Report SAND2004-4774, Sandia National Laboratories.

[12] Potvin, J. Y., Soriano, P. & Vallée, M. (2004) Generating Trading Rules on the Stock Markets with Genetic Programming, *Computers and Operations Research*, Vol. 31(7) (June 2004): 1033 - 1047

[13] Pring, M. J. (1980). *Technical Analysis Explained*, New York: McGraw-Hill.

[14] Ruggiero, M. A. (1997). *Cybernetic Trading Strategies*, NY: Wiley.

[15] Murphy, J. J. (1999). *Technical Analysis of the Financial Markets*, NY: New York Inst. of Finance.

[16] Lo, A. W., Mamaysky, H. & Wang, J. (2000). Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation, *J. of Finance*, 55:1705-1770, 2000.

[17] Marney, J. P., Miller, D., Fyfe, C. & Tarbert, H. (2000). Technical Analysis versus Market Efficiency: A Genetic Programming Approach, *Computing in Economics and Finance*, Society for Computational Economics, Barcelona, Spain, July 2000 (paper #169).

[18] Neely, C. (2001). Risk-adjusted, ex ante, optimal technical trading rules in equity markets, Working Papers 99-015D, Revised August 2001, Federal Reserve Bank of St. Louis.

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

[20] Becker, L.A. & Seshadri, M. (2003). Cooperative Coevolution of Technical Trading Rules, Worcester Polytechnic Institute, Computer Science Technical Report WPI-CS-TR-03-15.

[21] Becker, L.A. & Seshadri, M. (2003). GP-evolved technical trading rules can outperform buy and hold, In *Proc. 6th Int'l Conf. on Comp. Intelligence and Natural Computing*, North Carolina USA, 2003.

[22] Sharpe, W. F. (1966). "Mutual Fund Performance". *Journal of Business* **39** (S1): 119–138. doi:10.1086/294846.

[23] Marney, J. P., Tarbert, H. & Fyfe, C. (2005). Risk Adjusted Returns from Technical Trading: A Genetic Programming Approach, *Applied Financial Economics*, 15: 1073-1077.

[24] D. Lohpetch, D. Corne (2009) Discovering Effective Technical Trading Rules with Genetic Programming: Towards Robustly Outperforming Buy-and-Hold, in World Congress on Nature and Biologically Inspired Computing (NABIC) 2009, pp. 431—437, IEEE Press.

[25] D. Lohpetch, D. Corne (2010) Outperforming Buy-and-Hold with Evolved Technical Trading Rules: Daily, Weekly and Monthly Trading, EvoApplications 2010, 10pp, Springer LNCS

[26] D. Gorse (2011) Application of stochastic recurrent reinforcement learning to index trading, in Proc. European Symposium on Artificial Neural Networks –ESANN 2011, to appear.

[27] Antonio C. Briza and Prospero C. Naval Jr. (2008). Design of stock trading system for historical market data using multiobjective particle swarm optimization of technical indicators. In *GECCO*, pp. 1871–1878.

[28] Burton G. MALKIEL. (2003). The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, 17(1):59–82.

[29] Chang, T. J., Meade, N., & Beasley, J. E. (2000). Heuristics for Cardinality Constrained Portfolio Optimization. *Computers and Operations Research*, 27 (13), 1271–1302.

[30] Carlos A. Coello. *Evolutionary Multi–Objective Optimization and its Use in Finance*, chapter Handbook of Research on Nature Inspired Computing for Economy and Management. Idea Group Publishing, 2006.

[31] Doerner, K., Gutjahr, W. J., Hartl, R. F., Strauss, C., & Stummer, C. (2004). Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection. *Annals of Operations Research*, 131 (1–4), 79–99.

[32] Ehrgott, M., Klamroth, K., & Schwehm, C. (2004). An MCDM approach to portfolio optimization. *European Journal of Operational Research*, 155 (3), 752–770.

[33] Lin, D., Wang, S., & Yan, H. (2001). *A multiobjective genetic algorithm for portfolio selection*. Working Paper, Institute of Systems Science, Academy of Mathematics and Systems Science Chinese Academy of Sciences, Beijing, China.

[34] Shoaf, J. S., & Foster, J. A. (1996). A Genetic Algorithm Solution to the Efficient Set Problem: A Technique for Portfolio Selection Based on the Markowitz Model. In *Proceedings of the decision sciences institute annual meeting* (pp. 571–573). Orlando, Florida.

[35] Streichert, F., Ulmer, H., & Zell, A. (2004). Comparing Discrete and Continuous Genotypes on the Constrained Portfolio Selection Problem. In K. D. et al. (Ed.), *Genetic and evolutionary computation–gecco 2004. proceedings of the genetic and evolutionary computation conference. part ii* (pp. 1239–1250). Seattle, Washington, USA: Springer-Verlag, Lecture Notes in Computer Science Vol. 3103.

[36] Vedarajan, G., Chan, L. C., & Goldberg, D. E. (1997). Investment Portfolio Optimization using Genetic Algorithms. In J. R. Koza (Ed.), *Late breaking papers at the genetic programming 1997 conference* (pp. 255–263). Stanford University, California: Stanford Bookstore.

[37] Swee C. Chiam, Kay Chen Tan, and A. Al Mamun. (2009). Investigating technical trading strategy via a multi-objective evolutionary platform. *Expert Systems with Applications*, 36(7):10408–10423.

[38] Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. *Proceedings of the Parallel Problem Solving from Nature VI (PPSN-VI)*, (pp. 849-858).