Usage Patterns of Trading Rules in Stock Market Trading Strategies Optimized with Evolutionary Methods

Krzysztof Michalak[1], Patryk Filipiak[2], and Piotr Lipinski[2]

[1] Institute of Business Informatics,
Wroclaw University of Economics, Wroclaw, Poland
`krzysztof.michalak@ue.wroc.pl`
[2] Institute of Computer Science,
University of Wroclaw, Wroclaw, Poland
`{patryk.filipiak,lipinski}@ii.uni.wroc.pl`

Abstract. This paper proposes an approach to analysis of usage patterns of trading rules in stock market trading strategies. Analyzed strategies generate trading decisions based on signals produced by trading rules. Weighted sets of trading rules are used with parameters optimized using evolutionary algorithms. A novel approach to trading rule pattern discovery, inspired by association rule mining methods, is proposed. In the experiments, patterns consisting of up to 5 trading rules were discovered which appear in no less than 50% of trading experts optimized by evolutonary algorithm.

Keywords: stock market, trading rules, evolutionary computing

1 Introduction

Evolutionary computing is often applied to various economic and financial problems [1, 7, 16] such as optimization of investment portfolios [3, 13, 15, 18] and supporting financial decision making [9, 12, 22]. Training of trading experts based on sets of trading rules is an important task with practical applications in stock market trading [6, 10, 11].

In this paper we consider optimization of trading experts based on a predefined set of trading rules. An individual rule may be parameterized by several variables such as time period for which underlying indicators are calculated and various coefficients. Most often "buy" and "sell" signals are generated in response to certain indicators crossing each other or crossing a selected threshold. In the latter case, thresholds used by the rule are also considered as parameters affecting the behaviour of this particular rule. Each expert is based on the same set of rules, but with differing weights from range $[-1, 1]$, so, in principle, it may skip certain rules (weight $= 0$) or even negate their behaviour (weight < 1). Final decision returned by an expert at any time instant is determined by calculating a weighted sum of outputs of all trading rules and by applying buy and sell decision thresholds assigned to that particular expert. Expert parameters form a vector with numeric coordinates each of which is interpreted accordingly as: rule weight, rule parameter (time period, coefficient, rule decision threshold) or expert decision threshold.

The approach used in this paper consists of application of two separate steps: evolutionary optimization that adjusts parameters of trading experts in order to achieve good performance and frequent set analysis algorithm which identifies patterns of trading rules that occur together in successful experts. Identification of frequent patterns in trading rules is intended as a further step in relation to work described in [14]. The first step, i.e. expert parameters optimization was performed using evolutionary algorithm described later in the paper which ran for a predefined number of generations. The performance of the trading expert was optimized separately for each trading day (with the assumption, that only intra-day trading is performed) and a separate optimization procedure was performed for each stock. From the optimized population a constant fraction of best experts was taken as input data for the process of discovering patterns of usage of trading rules.

2 Evolutionary Optimization

Trading experts generate decisions based on trading rules. The rules require various parameters, the signals generated by the rules are weighted and compared to decision thresholds. Therefore, the behaviour of a trading expert can be parameterized by a vector of constant length with coordinates corresponding to the above mentioned parameters.

Vector parameterizing a trading expert can be used as a chromosome of a specimen in an evolutionary algorithm. Such a chromosome can be stored in the following form:

$$weight_1, par_{1,1}, \ldots, par_{1,k_1}, \ldots, weight_n, par_{n,1}, \ldots, par_{n,k_n}, \Theta_{buy}, \Theta_{sell}$$

where:
$weight_i$ - i-th rule weight,
$par_{i,j}$ - j-th parameter of the i-th rule,
k_i - number of parameters for the i-th rule,
Θ_{buy}, Θ_{sell} - decision thresholds.

In order to optimize the performance of trading experts an evolutionary algorithm Algorithm 1 was employed.

In the presented algorithm the following operations are used:

SelectMatingPool - selection of a mating pool for crossover and mutation consisting of N_{mate} specimens. This selection is performed using a series of binary tournaments in which the better one of the two randomly selected specimens is included in the mating pool.

Crossover - a single-point crossover operator.

Algorithm 1 Evolutionary algorithm used for optimizing trading experts.

N_{pop} - population size
N_{mate} - mating pool size
P_{mut} - probability of mutating a single parameter

P_1 = InitPopulation()
Evaluate(P_1)
for $i = 1 \to N_{Gen}$ **do**
  P'_i = SelectMatingPool(P_i, N_{mate})
  P'_i = Crossover(P'_i)
  P'_i = Mutate(P'_i, P_{mut})
  Evaluate(P'_i)
  $P_i = P_i \cup P'_i$
  P_{i+1} = Reduce(P_i, N_{pop})
end for

Mutate - a mutation procedure which changes any given vector coordinate with P_{mut} probability.

Evaluate - specimen evaluation based on simulated performance of the trading expert. For a given specimen trading decisions are generated by an expert based on parameters vector equal to the genotype of the specimen. The evaluation of the specimen is the return which would be achieved if the suggestions of the expert were used to buy and sell stocks in the intra-day trading model. Additionally, a specimen is required to perform at least one transaction during the day (specimens that do not generate any "buy" signals are therefore given an evaluation of 0). A commision of 0.4% is applied to each transaction (each buy and each sell).

Reduce - reduction of $P_i \cup P'_i$ back to N_{pop} specimens with N_{elite} best specimens guaranteed to be copied to the next population and the rest filled using fitness proportionate selection.

Parameters of the evolutionary algorithm in the experiments were set as presented in Table 1.

3 Analysis of Trading Rule Sets

In this paper we aimed at finding sets of trading rules that appear together in trading experts optimized using evoluationary algorithm. The data set for pattern discovery was a top 20% of specimens present in the last population in evolutionary algorithm (after N_{gen} generations). These 20% of specimens were selected separately for each stock and each trading day. To increase the size of the data set (which is only 20 specimens for $N_{pop} = 100$) specimens for each stock obtained on all trading days were used together in a single set for pattern discovery.

Table 1. Parameters of the evolutionary algorithm

Desctiption	Symbol	Value
Number of generations	N_{gen}	30
Size of the population	N_{pop}	100
Size of the mating pool	N_{mate}	50
Number of the best specimens copied to the next population	N_{elite}	20
Probability of mutating each parameter	P_{mut}	0.1

The general approach is similar to frequent set discovery techniques used in association rule mining [4, 5, 17]. The algorithm that generates sets of trading rules that are frequently found in good trading experts is presented in Algorithm 2. It builds frequent rule sets incrementally, starting from sets containg one element each (which are simply equivalent to individual rules). Building larger sets is based on the observation, that if a set is frequent then all of its subsets must also be frequent. Based on this property it is possible to build larger frequent sets by extending the already found smaller frequent sets. The support needs to be calculated for the newly constructed sets, but there is a gain in that the number of candidate sets is lower than when all possible sets of given size would be considered. This algorithm is parameterized by the number *minSupport* which defines what percentage of specimens a rule set must appear in to be considered frequent. In the experiments this number was set to *minSupport* = 50%. The rule is considered to be used in a particular expert if the absolute value of its weight is not less than $minWeight$ parameter. In the experiments the minimum weight was set to $minWeight = 0.7$.

Algorithm 2 Frequent rule set discovery algorithm.

$i = 1$
C_1 = family of sets, each containing one rule
S_1 = CheckSupport(C_1)
while $|S_i| > 1$ **do**
  C_{i+1} = Combine(S_i)
  S_{i+1} = CheckSupport(C_{i+1})
  $i = i + 1$
end while

The procedures used in this algorithm are as follows:

CheckSupport - processes a family C_i of candidate sets of rules. For each set of rules $R \in C_i$ iterates over the entire data set V (consisting of N specimens). If in a given specimen each of the rules in the candidate set R has a weight with absolute value at least $minWeight$ the count m is increased. The procedure returns those candidate sets from C_i, which have $m \geq minSupport * N$. The CheckSupport procedure is detailed in Algorithm 3.

Algorithm 3 CheckSupport - a procedure for selecting rule sets with support $\geq minSupport$.

C_i - a family of candidate sets
S_i - a family of frequent sets
V - data set of the best specimens generated by evolutionary algorithm
N - the size of the data set V $(N = |V|)$

$S_i = \emptyset$
for $R \in C_i$ **do**
  $m = 0$
  for $v \in V$ **do**
    if $\forall k \in R \cdot |v[k]| \geq minWeight$ **then**
      $m = m + 1$
    end if
  end for
  if $m \geq minSupport * N$ **then**
    $S_i = S_i \cup \{R\}$
  end if
end for

Combine - given a family of frequent sets S_i of size i generates a family of candidate sets C_{i+1} of size $i + 1$. This family includes all sets that have the property, that each subset of size i of each candidate set is in S_i (i.e. is a frequent set). The Combine procedure is detailed in Algorithm 4.

Algorithm 4 Combine - a procedure generating a family C_{i+1} of candidate sets of size $i + 1$ based on a family S_i of frequent sets of size i.

C_{i+1} - a family of candidate sets of size $i + 1$
S_i - a family of frequent sets of size i

$C_{i+1} = \emptyset$
for $T_j \in S_i$ **do**
  for $T_k \in S_i$ **do**
    if $|T_j \backslash T_k| = 1$ **and** $|T_k \backslash T_j| = 1$ **then**
      $C_{i+1} = C_{i+1} \cup \{T_J \cup T_k\}$
    end if
  end for
end for

---

4 Experiments

The experiments were performed on high-frequency data from the New-York Stock Exchange. 50 stocks were analyzed: AAPL, AMZN, BIDU, BIIB, C, CAT, CF, CHK, CMI, CRM, CSCO, CVX, DIA, EMC, EOG, F, FAS, FFIV, GE, GLD, GLW, GOOG, GS, HPQ, IBM, INTC, IVV, JPM, MDY, MO, MS, MSFT, NFLX, NKE, NVDA, ORCL, PFE, PX, SPG, SPY, T, TLT, UNP, V, VALE, VZ, WFC, WYNN, XLF and YHOO. Minute quotations were used from trading days in the period from 2012.01.01 to 2012.06.30. This period contains 125 trading days.

A set of 78 trading rules was used. The rules were based on indicators such as Moving Averages [21], Chaikin Oscillator [19], Overbought/Oversold [20], RSI [2], Williams %R Oscillator [8], etc. Due to space limitations it is not possible to describe all the rules in detail. Some of the indicators were based on moving averages. These indicators were used in conjunction with both the simple moving average (SMA) and the exponential moving average (EMA). Thus, the total number of rules active in the system was 96.

The rules were further parameterized. For example if moving average was used, the period of the moving average was modified by the evolutionary algorithm. Periods were adjusted in the range $[5, 30]$. Other parameters and decision thresholds were adjusted within ranges depending of the meaning of the parameter.

The results of experiments are summarized in Table 2. This table shows the maximum size of frequent rule set obtained for each stock and the number of frequent rule sets of the maximum size generated in the experiments. The maximum size of frequent rule sets varies from 2 to 5 depending on what stock the rules are related to. For most of the stocks frequent rule sets were found containing at least 3 rules. In the experiments no frequent rule sets containing 6 or more rules were found. While such sets were generated as candidate sets, none of them has reached the required support of at least 50% (however, some of the 6-elements sets have reached the support of about 47.5%).

The number of n-element frequent rule sets found in the experiments (for n=2,3,4 and 5) is shown in Figures 1-4.

Frequent rule sets of larger size seem to be more interesting because they may indicate more complex dependencies among the rules. The maximum frequent rule set size obtained in the experiments was 5. In Table 3 examples of frequent rule sets of size 5 are presented.

Abbreaviations used in Table 3 have the following meaning.

EMV - Ease of Movement Value

LinDir - Linear Regression Direction

NormPrice - Price normalized by Dow Jones Index

OvO - Overbought / Oversold

RoC - Rate of Change

RSI - Relative Strength Index

TBR - Top / Bottom Reversal

V&P - Volume and Price

Table 2. Maximum sizes of frequent rule sets and the number of sets of the maximum size

Ticker	Max set size	Num. sets	Ticker	Max set size	Num. sets	Ticker	Max set size	Num. sets
AAPL	4	471	FFIV	3	235	NVDA	3	2907
AMZN	3	1	GE	3	14	ORCL	3	4237
BIDU	3	141	GLD	3	2057	PFE	3	1622
BIIB	3	1466	GLW	5	7	PX	2	1267
C	3	249	GOOG	4	2	SPG	3	86
CAT	3	191	GS	3	518	SPY	3	1456
CF	3	724	HPQ	3	1310	T	5	29
CHK	3	288	IBM	3	567	TLT	3	277
CMI	4	4	INTC	3	197	UNP	3	1
CRM	3	105	IVV	3	15	V	3	277
CSCO	3	227	JPM	3	920	VALE	3	1190
CVX	3	74	MDY	3	4227	VZ	3	569
DIA	2	2177	MO	3	3	WFC	2	899
EMC	3	2748	MS	3	1777	WYNN	5	14
EOG	3	9	MSFT	3	118	XLF	3	196
F	3	192	NFLX	5	1	YHOO	4	5
FAS	4	5	NKE	2	1725			

Table 3. Examples of frequent rule sets of size 5 obtained in the experiments

Ticker	Rules		
GLW	$EMV_{(3)}$+TBR, LinDir,	$EMV_{(3)}$+Trend, $EMV_{(3)}$+NormPrice	$EMV_{(3)}$+WiO
NFLX	$EMV_{(2)}$+RoC, $EMV_{(2)}$+NormPrice,	K-Stochastic, RSI	NormPrice+WiO
T	NormPrice+MA, NormPrice+$V\&P_{(1)}$,	NormPrice+two MAs, $V\&P_{(2)}$	NormPrice+TBR,
WYNN	$EMV_{(1)}$+OvO, RoC,	$EMV_{(2)}$+$V\&P_{(2)}$, WiO	NormPrice+two MAs,

WiO - Williams Oscillator

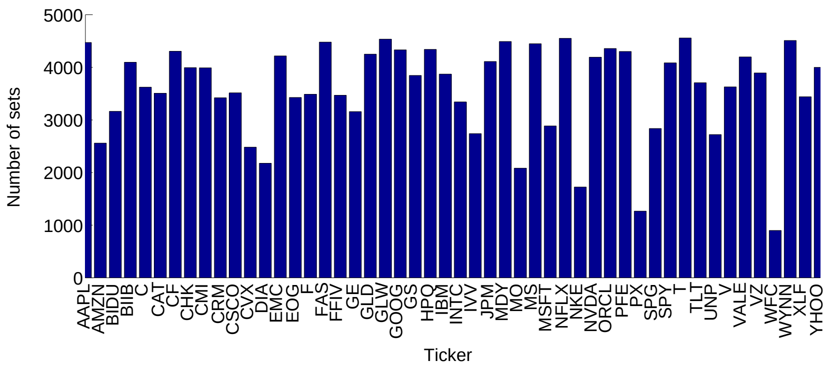

Fig. 1. Number of 2-element frequent rule sets for stocks used in the experiments.

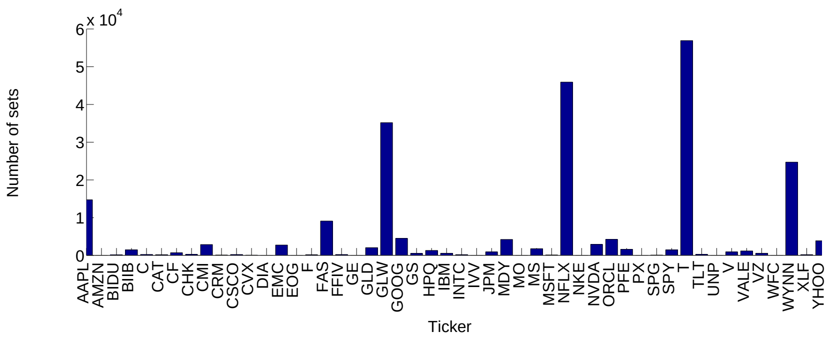

Fig. 2. Number of 3-element frequent rule sets for stocks used in the experiments.

5 Conclusion

In this paper a task of finding repetitive patterns in trading experts was approached. Trading experts were constructed from trading rules. Parameters and weights of the trading rules were optimized using evolutionary algorithm in order to improve performance of the experts. Based on the optimized experts a discovery of usage patterns of the trading rules was performed.

The novel approach to this topic proposed in this paper is inspired by frequent set discovery methods used in assiociation rule mining. These methods

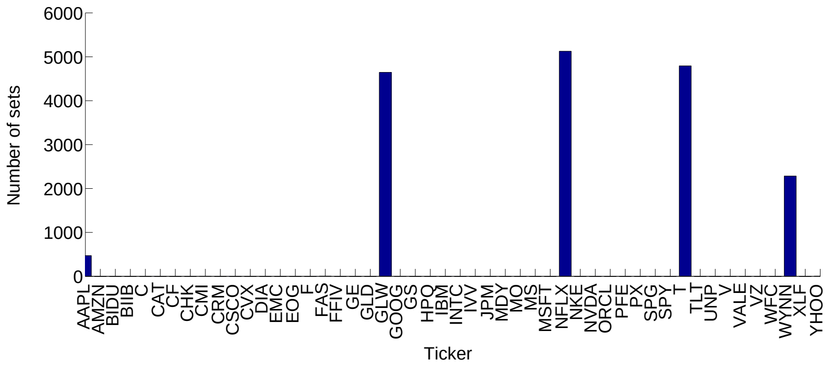

Fig. 3. Number of 4-element frequent rule sets for stocks used in the experiments.

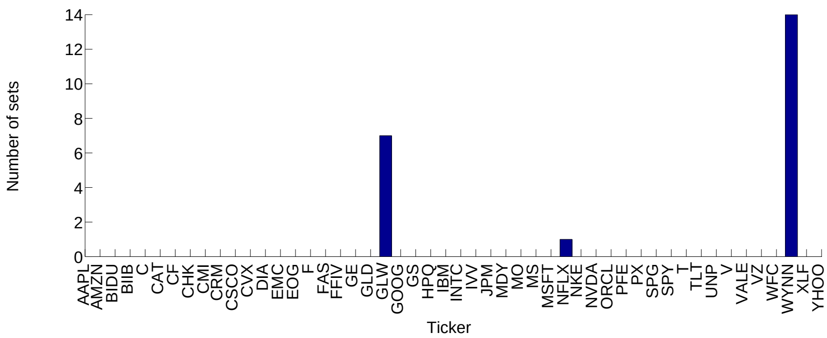

Fig. 4. Number of 5-element frequent rule sets for stocks used in the experiments.

were adapted to finding patterns emerging in a population of trading experts optimized by an evolutionary algorithm to achieve good performance in stock market trading. The results of experiments have shown that patterns of up to 5 trading rules emerge in the population. Each of the patterns is supported in at least 50% of the optimized experts.

The results of the experiments suggest, that repetitive patterns are present in trading rule sets. In the context of evolutionary optimization of trading rules such patterns could be used to improve solutions in dynamically changing environment of stock market trading, especially for shortening generation of good populations for evolutionary algorithms.

References

1. Bauer, R.: Genetic Algorithms and Investment Strategies. Wiley, Chichester (1994)
2. Bauer, R. J., Dahlquist, J. R.: Technical Markets Indicators: Analysis & Performance pp. 129 John Wiley & Sons (1998)

3. Best, M. J.: Portfolio Optimization. Chapman&Hall/CRC (2010)
4. Borgelt, Ch.: Frequent item set mining. In: Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, Volume 2, Issue 6 pp. 437-456, John Wiley & Sons (2012)
5. Chopra, D., Vishwakarma, D.: Efficient Frequent Item set Discovery Technique in Uncertain Data. International Journal of Engineering and Advanced Technology (IJEAT), Volume 1, Issue 6 (2012)
6. Dempsey, I., O'Neill, M., Brabazon, A.: Adaptive Trading with Grammatical Evolution. In: Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006), pp. 2587–2592, IEEE, Los Alamitos (2006)
7. Dempster, M., Jones, C.: A Real-Time Adaptive Trading System using Genetic Programming. Quantitative Finance 1, pp. 397–413, (2001)
8. Kirkpatrick, Ch, D., Dahlquist, J. R, Technical Analysis: The Complete Resource for Financial Market Technicians. pp. 440–441 FT Press (2010)
9. Li, J., Taiwo, S.: Enhancing Financial Decision Making Using Multi-Objective Financial Genetic Programming. In: Proceedings of IEEE Congress on Evolutionary Computation, 2006, pp. 2171–2178, (2006)
10. Lipinski, P.: Dependency Mining in Large Sets of Stock Market Trading Rules. In: Enhanced Methods in Computer Security, Biometric and Intelligent Systems, ed. J. Pejas, A. Piegat, pp. 329–336, Kluwer Academic Publishers (2005)
11. Lipinski, P.: Discovering Stock Market Trading Rules using Multi-Layer Perceptrons. In: Proceedings of 9th International Work Conference on Artificial Neural Networks, IWANN 2007, LNCS, vol. 4507, pp. 1114–1121, Springer (2007)
12. Lipinski, P.: ECGA vs. BOA in Discoverying Stock Market Trading Experts. In: Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2007, pp. 531–538, ACM (2007)
13. Lipinski, P.: Evolutionary Strategies for Building Risk-Optimal Portfolios. In: Natural Computing in Computational Finance, SCI, vol. 100, pp. 53–65, Springer (2008)
14. Lipinski, P.: Frequent Knowledge Patterns in Evolutionary Decision Support Systems for Financial Time Series Analysis. In: Natural Computing in Computational Finance, SCI, vol. 293, pp. 131–145, Springer (2010)
15. Michalak, K., Filipiak, P., Lipinski, P.: Evolutionary Approach to Multiobjective Optimization of Portfolios That Reflect the Behaviour of Investment Funds. In: Artificial Intelligence: Methodology, Systems, and Applications - 15th International Conference, AIMSA 2012, Varna, Bulgaria, LNCS, vol. 7557, pp. 202-211, Springer (2012)
16. Michalak, K., Lipinski, P.: Prediction of high increases in stock prices using neural networks. Neural Network World, 15, vol. 4, pp. 359-366, Springer (2005)
17. Park, B. J.: Efficient Tree-based Discovery of Frequent Itemsets. International Journal of Multimedia and Ubiquitous Engineering, 7(2) (2012)
18. Radziukyniene, I, Zilinskas, A.: Evolutionary Methods for Multi-Objective Portfolio Optimization. In: Proceedings of the World Congress on Engineering 2008, pp. 1155–1159, Newswood Limited (2008)
19. Ranganatham M.: Investment Analysis and Portfolio Management. pp. 387–388 Pearson Education (2004)
20. Schwager, J. D.: Technical Analysis. pp. 174–178 John Wiley & Sons (1995)
21. Srivastava, U. K., Shenoy, G. V., Sharma, S. C.: Quantitative Techniques For Managerial Decisions. pp. 392–394 New Age International (1989)
22. Tsang, E., Li, J., Markose, S., Er, H., Salhi, A., Iori, G.: EDDIE in Financial Decision Making. Journal of Management and Economics 4(4), (2000)