Applied Soft Computing 55 (2017) 127–140

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

An intelligent hybrid trading system for discovering trading rules for the futures market using rough sets and genetic algorithms

Youngmin Kim [a], Wonbin Ahn [b], Kyong Joo Oh [b], David Enke [c,*]

[a] Laboratory for Investment and Financial Engineering, Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, 205 Engineering Management, 600 W. 14th Street, Rolla, MO, 65409-0370, USA

[b] Department of Information and Industrial Engineering, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul, 03722, Korea

[c] Laboratory for Investment and Financial Engineering, Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, 221 Engineering Management, 600 W. 14th Street, Rolla, MO, 65409-0370, USA

ARTICLE INFO

Article history:
Received 19 August 2016
Received in revised form 9 January 2017
Accepted 1 February 2017
Available online 9 February 2017

Keywords:
Intelligent hybrid trading system
Discovering trading rules
Rough sets
Genetic algorithms
Futures market

ABSTRACT

Discovering intelligent technical trading rules from nonlinear and complex stock market data, and then developing decision support trading systems, is an important challenge. The objective of this study is to develop an intelligent hybrid trading system for discovering technical trading rules using rough set analysis and a genetic algorithm (GA). In order to obtain better trading decisions, a novel rule discovery mechanism using a GA approach is proposed for solving optimization problems (i.e., data discretization and reducts) of rough set analysis when discovering technical trading rules for the futures market. Experiments are designed to test the proposed model against comparable approaches (i.e., random, correlation, and GA approaches). In addition, these comprehensive experiments cover most of the current trading system topics, including the use of a sliding window method (with or without validation dataset), the number of trading rules, and the size of training period. To evaluate an intelligent hybrid trading system, experiments were carried out on the historical data of the Korea Composite Stock Price Index 200 (KOSPI 200) futures market. In particular, trading performance is analyzed according to the number of sets of decision rules and the size of the training period for discovering trading rules for the testing period. The results show that the proposed model significantly outperforms the benchmark model in terms of the average return and as a risk-adjusted measure.

1. Introduction

Discovering potentially useful knowledge from nonlinear and complex stock market data, and then developing systems to support human decision-making, is one of the most important challenges in the financial market domain [3,5,12,42]. Due to the high complexity of the stock market, it is difficult to discover trading rules that are present in the stock market data. Therefore, the process of discovering trading rules is an important aspect of data mining since it can help investors obtain a set of symbolic trading rules that explain the relationship among input variables (i.e., technical indicators), while providing trading rules that are more transparent and can be better understood by investors [13].

Numerous research studies have been carried out to discover optimal trading rules based on technical indicators in the stock market using individual data-driven approaches, such as artificial intelligence, machine learning, pattern recognition, and data mining techniques, or by combining these computational techniques [7,13,29,44]. Recently, researchers have focused on the development of intelligent hybrid trading systems that utilize the advantages of each individual technique, thereby providing real potential synergy [4,26,43]. However, trading rules generated from some techniques are difficult for non-professional investors to easily interpret when they are mined from black box models, such as the Artificial Neural Networks (ANN) or Support Vector Machines (SVM) [46]. As such, these models are often not suitable for guiding actual investment behavior given that they rarely provide any insight into the nature of interactions between technical indicators and market direction [21].

Rough set theory has been successfully proven to be a useful technique for discovering potential knowledge from uncertainty, incompleteness, and imprecision data, such as stock market data [11,16,24,34]. Trading systems based on technical indicators and rough set analysis can mitigate the limitation of black box models

* Corresponding author.
E-mail addresses: kimym@mst.edu (Y. Kim), fsign@yonsei.ac.kr (W. Ahn), johanoh@yonsei.ac.kr (K.J. Oh), enke@mst.edu (D. Enke).

http://dx.doi.org/10.1016/j.asoc.2017.02.006
1568-4946/

since the generated 'If-then' decision rules can provide clear and explicit trading rules of knowledge that is applicable to investors. There are three advantages to developing an intelligent trading system using rough set analysis [41]. First, rough set analysis can handle the original data regardless of any external information or without any additional data assumptions. Second, rough set analysis can discover hidden knowledge or patterns in datasets and represent knowledge with decision rules of natural language. Third, the results (i.e., decision rules) are easy to understand for investors. Due to these advantages, rough set analysis has received significant attention for financial data analysis, including stock market analysis, as well as for identifying buying and selling signals for market timing, in addition to their use for generating trading rules [19].

Although rough set analysis can extract decision rules effectively from given datasets, two processes of data discretization and finding reducts (i.e., subsets of attributes) are required in order to generate decision rules based on technical indicators (i.e., continuous values). However, both processes are known to be an NP-hard problem [28,35,47] and are also related to the dimensionality reduction problem [14,17]. To transform continuous values into discrete values in accordance with certain cut points for discretization, an optimal discretization or cut points selection problem is considered [18,39]. Finding the reduct, which is a minimal subset of attributes, and then classifying a maximum number of objects, is also regarded as an optimization problem [15,28]. Nguyen and Skowron [30] proved that the problem of optimal discretization (or quantization) of real values attributes, including the problem of finding a minimal reduct, are NP-hard. They tried to solve these problems by applying Boolean reasoning and heuristic algorithms. Nevertheless, it is difficult to find a special procedure for their model that is appropriate for high-dimensional data. As a result, they suggested that adaptive heuristic methods might be needed to solve these problems. Evolutionary algorithms can be applied in this study since these problems can be defined as a combinatorial optimization problem with a binary search space [27].

To resolve two problems simultaneously, in this study a genetic algorithm (GA) is applied to search for both the cut points for discretization and the reducts in order to discover the optimal or sub-optimal trading rules. To be precise, although the GA is probably not finding an optimal solution when dealing with NP-hard problems, it can find a near-optimal solution [17,18]. Thus, the optimal (or the best) trading rule in this study represents the trading rule having the highest fitness value, while sub-optimal trading rules are rules having the next highest value to survive in the GA process. The GA is an effective and robust method for searching typical combinatorial optimization problem solutions in a variety of applications, and is particularly applicable for solving multi-parameter optimization problems [1,31]. Furthermore, Hu et al. [13] surveyed numerous related papers that focus on the application of evolutionary computation (EC) techniques for rule discovery within an intelligent trading system. Through the surveyed papers, the authors showed that genetic algorithms, when considered among other EC techniques (i.e., evolutionary algorithm, swarm intelligence, and hybrid EC techniques), are widely accepted for searching and discovering technical trading rules within an intelligent trading system.

The objective of this paper is to incorporate a GA into two processes (i.e., data discretization and reducts) of rough set analysis for discovering the optimal or sub-optimal trading rules in order to obtain better trading decisions. For our study, we develop a discovery trading rule mechanism that extends and advances the procedure of *RCTS* (the rule change trading system) developed by Kim and Enke [19]. Notice that *RCTS* generates numerous trading rules by combining the number of intervals for discretization and the cardinality of reducts. However, these trading rules are not evolved and are simply generated based on a decision table to deal with the classification problem, regardless of the trading performance. On the other hand, the discovery trading rule mechanism in this study extends the RTCS such that trading rules are evolved and discovered in the training period by optimizing trading performance measures. Furthermore, this study considers some important factors, such as the sliding window method (with or without a validation dataset), the fitness function, and the size of training data. Finally, this study examines whether the number of trading rules is important in the futures market since a trading systems using a single (optimal) rule, or too many trading rules, often makes them unsuitable for practical trading (see Section 2).

This study of an intelligent hybrid trading system for discovering trading rules has three distinct advantages. First, this study develops an intelligent hybrid trading system model using rough set analysis and a GA. Few studies have been conducted on a hybrid intelligent trading system that combines rough set analysis and a GA for discovering trading rules. Second, this study uses a GA for two processes of rough set analysis to explore reducts, while simultaneously searching the optimal or near-optimal cut points for data discretization for discovering the optimal or sub-optimal trading rules. This reduction and transformation of the data may shorten the running time, while also allowing the system to obtain more generalized results. Third, and more importantly, this study suggests a rule discovery mechanism that can discover trading rules automatically, with the discovered rules being quantitative and understandable for investors in an uncertain market. This moves the system from being a black-box model, to something more transparent, allowing for greater investor confidence of the decision-making and subsequent results. To evaluate an intelligent hybrid trading system, experiments were carried out on the historical data of the Korea Composite Stock Price Index 200 (KOSPI 200) futures market.

The remainder of the paper is organized as follows. Sections 2 and 3 briefly review the related research of developing a trading system for discovering trading rules and the basic concepts of rough set theory and a GA. Section 4 describes the construction procedure for the intelligent hybrid trading system for discovering trading rules. Section 5 presents an empirical study that was performed to evaluate the performance of the proposed model. Finally, concluding remarks and areas for future work are discussed in Section 6.

2. Related work

Trading rules based on technical indicators have been used widely in the stock, index, and currency markets to support human decision-making, in particular, for trading systems that forecast the future price trend or generate trading signals (i.e., buying and selling signals). Studies have shown positive empirical evidence regarding the profitability of technical trading rules [27]. In addition, Allen and Karjalainen [1] were the first to develop a technical trading rule discovery model using a GA, combining the arithmetic operators with technical indicators for the Standard & Poor's (S&P) 500 index. Thereafter, many researchers have attempted to develop trading system models based on a data-driven approach to discover technical trading rules in the financial markets. In particular, trading rule discovery models based on evolutionary learning, which aim to search for and apply optimal technical trading rules in a large search space, have been developed [13]. Dempster and Jones [8] developed a trading system for finding the optimal technical trading rules using genetic programming (GP) for the foreign exchange (FX) market. A stock trading model based on genetic network programming (GNP) has been proposed by Chen et al. [6]. Esfahanipour and Mousavi [9] implemented a GP for automatically generating risk-adjusted technical trading rules on individual stocks listed on the Tehran stock exchange (TSE). However, it is difficult for optimal trading rules to apply to various kinds of market situations since identifying the best individual trading rules only determines the timing of buying and selling stocks for the testing simulations of one specific market and timeframe [33,38,42].

Recent research has looked to overcome this limitation by generating a large number of trading rules according various market situations. Mabu et al. [25] extracted a large number of buying and selling rules using GNP with rule accumulation, and then determined whether to buy or sell stocks based on the extracted trading rules through a unique classification mechanism. Wang et al. [42] proposed a complex stock trading strategy with a reward/penalty mechanism using particle swarm optimization (PSO). They used moving average (MA) and trading range break-out (TRB), combining various parameters of two technical indicators to generate numerous trading rules, and then updated all the trading rule weights based on the reward/penalty mechanism. Zhang et al. [46] proposed an evolutionary trend reversion model for stock trading rule discovery called eTrendRev, which can adapt to varying stock market situations given that the proposed system consists of classification rule mining, evolutionary learning, and a reinforcement learning technique. Kim and Enke [19] developed the rule change trading system (*RCTS*) using rough set analysis for the futures market. They generated possible trading rules combined from both data discretization and reducts methods, and changed the trading rules according to the previous trading results. Simultaneously, a GA was employed to determine the thresholds of market timing for buying and selling signals in the KOSPI 200 index futures market. However, although the above studies present trading systems with a number of trading rules in order to resolve the limitation of trading models with market specific optimal trading rules, they can still negatively affect the buying or selling signals, as well as take a long time for even one trial. Therefore, to avoid these limitations, the appropriate number of trading rules should be considered before applying them to the current market being considered.

One of the limitations to consider for each of the aforementioned studies is the different sample size of the training or learning dataset necessary in order to build the trading system models. The sample period for training is an important factor when using the data-driven approach since it has a large influence on the experiment results [13]. In this empirical study we will also consider if the number of trading rules and the size of the training dataset for discovering trading rules have an impact on the proposed model.

3. Background

3.1. The fundamental concepts of rough set theory

Rough set theory has been successfully applied to gain knowledge in various industry domains. It provides a mathematical approach to deal with vagueness and uncertain data based on three concepts, including the indiscernibility relation, set approximation, and reducts. In this section we briefly review the fundamental concepts of rough set theory [34,36,37].

The starting point of rough set analysis is to construct a decision table $S = \langle U, C, D \rangle$ consisting of columns (i.e., attributes) and rows (i.e., objects), where U is called the universe, and C and D are disjoint sets of condition and decision attributes, respectively [31]. Any subset B of C can classify a binary relation on universe U, if $a(x) = a(y)$ (i.e., x, $y \in U$), and is called an *B-indiscernibility* relation $IND(B)$, where $a(x)$ and $a(y)$ denote the value of attribute a (i.e., $\forall a \in B$) for object x and y, respectively. Thus, the indiscernibility relation is an equivalence relation that can be defined by a set of attributes.

The set approximation is based on the indiscernibility relation and defines three regions based on the equivalent classes. To approximate a set X ($X \subseteq U$) on the basis of the indiscernibility relation B, the B-lower ($\underline{B}X$) and B-upper ($\bar{B}X$) approximations can be defined by the set X. In other words, $\underline{B}X$ consists of all objects that belong with certainty to the set X and $\bar{B}X$ includes all objects that possibly belong to the set X. The difference between the upper and the lower approximation is called the boundary region of the set X (i.e., $BN_B(X) = \bar{B}X - \underline{B}X$). If the boundary region of the set X is empty (i.e., $BN_B(X) = \phi$), then the set X is the crisp set, whereas if the boundary region of X is non-empty ($BN_B(X) \neq \phi$), the set X is defined as a rough set with respect to B.

A reduct is the minimal subset of attributes C that provides the same discernibility as the set of attributes C. Thus, dispensable or unnecessary attributes can be eliminated without losing any information since attributes in a reduct contain all the information. The intersection of all reducts is called the "core", which is a set of the most essential attributes. After finding the reducts, decision rules are generated with values of the condition and decision attributes to describe approximations in logical terms. As a result, the decisions rules are expressed in the form "*If condition(s) then decision(s)*".

For this study, various technical indicators are used as conditional attributes and the directional movement is defined as the decision attribute for constructing a decision table with the time series. Therefore, rough set theory based on above concepts can be applied to the stock market domain for generating buying and selling signals (i.e., decision rules).

IF MACD [*,−0.18) and RSI [40.72, 58.79) and ROC [0.59,*) then Up

IF MACD [0.16, *) and RSI [*, 40.72) and ROC [0.59,*) then Down

For example, the aforementioned decision rules are generated by rough set analysis to present the extracted knowledge from the stock market data. Condition attributes consists of technical indicators, including the MACD (Moving Average Convergence Divergence), RSI (Relative Strength Index), and ROC (Rate of Change), each with their corresponding ranges. The value of the decision attribute is determined as either Up or Down for classifying the market direction. This simple decision rule structure also allows investors to easily understand and verify the result, thereby avoiding the transparency limitations of black box models by allowing the trading rules to identify or explain relationships among the technical indicators.

3.2. The basic principle genetic algorithms

During the last few decades, GAs have been applied to a wide range of problems, such as optimal resource allocation, transportation, and scheduling problems, among others. GAs belong to the class of evolutionary computation that is inspired by natural evolution using adaptive heuristic search methods [18,30]. GAs mimic the principles of natural selection and biological evolution to produce several solutions to a given problem. This gives GAs the ability of solving combinatorial optimization problems. Calculus based methods (e.g., hill climbing) typically have a difficult time dealing with these types of problems. Although initially randomized, GAs allow one to explore the search space for better solutions based on historical information.

The general process of a GA begins with a population of randomly created individual solutions. The individuals (solutions) in the population are then evaluated/scored by a user-defined fitness function. Two individuals (i.e., parents) are selected based on the fitness measure. These individuals reproduce to generate one or more offspring, after which new offspring are generated by applying a recombination operator, such as crossover and mutation. The aforementioned process is repeated until a best/suitable solution is found, or a certain number of generations is reached. The basic pseudo-code of a GA is as follows:

1. Generate an initial random population of individuals
2. generations = 0
3. Evaluate the fitness of the individuals
4. **while** termination condition is not met **do**
5. Select two individuals (two parents) for reproduction
6. Generate new individuals (the offspring) using crossover and mutation operators
7. Evaluate the fitness of the new individuals
8. Replace the worst individuals of the population by the best individuals
9. generations = generations + 1
10. **end while**
11. **return** the optimal population

In this study, a GA is simultaneously employed to optimize two processes of rough set analysis to explore optimal reducts and cut points for data discretization, and for discovering trading rules. In general, rough set theory is good for knowledge representation while GAs are good for solving combinatorial optimization problems. Therefore, the advantage of an intelligent hybrid trading system model using rough set analysis and a GA, as proposed and developed here, is that the discovered trading rules generated by resolving optimization problems are simple, yet explicit.

4. An intelligent hybrid trading system for discovering trading rules

The framework of an intelligent hybrid trading system for discovering trading rules is given in Fig. 1. In the first phase, a decision table consisting of conditional attributes (i.e., technical indicators) and a decision attribute (i.e., Up (+1) or Down (-1)) are generated for rough set analysis. The second phase presents the rule discovery mechanism that evolves decision rules extracted from rough set analysis for discovering optimal and sub-optimal trading rules. In the final phase, the trading strategy is established to generate buying and selling signals.

4.1. Phase 1: data transformation for generating the decision table

In phase 1, rough set analysis starts by generating a decision table S = <U, C, D> that is composed of a finite set of m objects $U = \{x_1, x_2, \ldots, x_m\}$, a finite set of n conditional attributes $C = \{c_1, c_2, \ldots, c_n\}$, and a decision attribute $D = \{d\}$. The real-time transaction data of basic quantities (i.e., open, high, low, close price, and trading volume) during the time interval $(t-1, t]$ transforms into each formula of n technical indicators to generate the values of conditional attributes. The values of the decision attribute is determined by the change of the closing price, which is labeled either Up (1) or Down (-1). Up (1) means that the next closing price at time $t+1$ is greater than or equal to the closing price at time t, whereas Down (-1) means that the next closing price at time $t+1$ is less than the closing price at time t. For the analysis, several popular technical indicators (i.e., $n = 37$) were chosen by experts and researchers, such as the Moving Average Convergence Divergence (MACD), the Relative Strength Index (RSI), the Stochastic Oscillator (SO), and the Williams%R, among others. Appendix A provides detailed calculations for each technical indicator used as conditional attributes in a decision table.

4.2. Phase 2: a rule discovery mechanism

In phase 2, the rule discovery mechanism consists of two steps that are developed for (1) extracting decision rules using rough set analysis and to (2) evolve the extracted decision rules through the GA for discovering optimal decision rules. The two steps repeat until the stopping condition of the GA is reached.

4.2.1. Extract decision rules using rough set analysis

As rough set analysis can only handle discrete attributes instead of continuous (or numerical) attributes, one of the data preprocessing methods, data discretization, is preceded to create

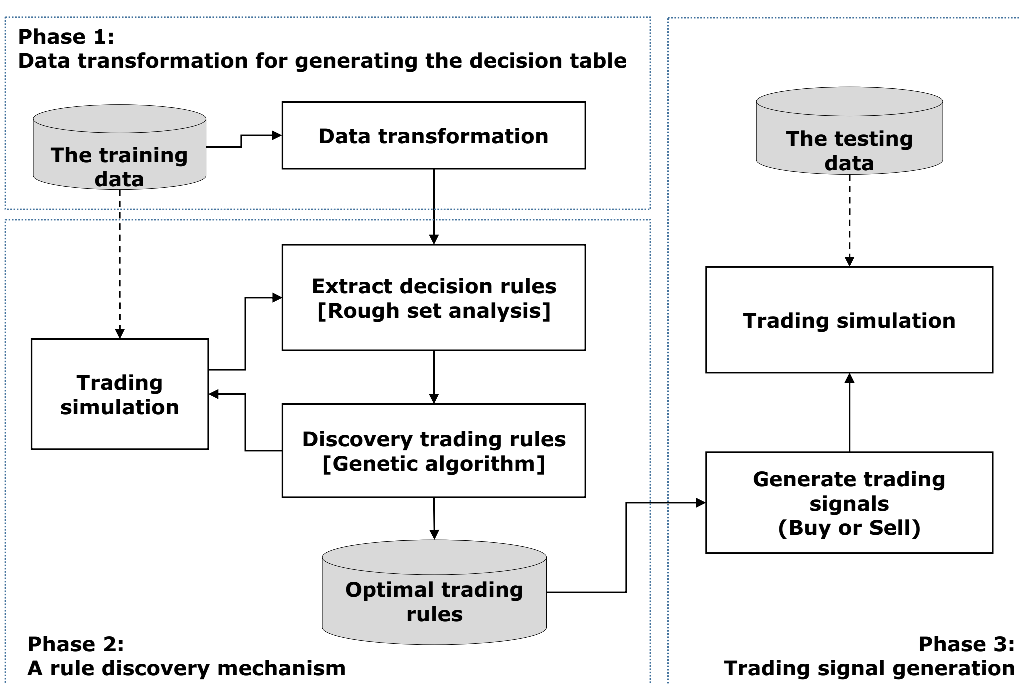

Fig. 1. The framework of an intelligent hybrid trading system for discovering trading rules.

the subset of attributes (i.e., reducts). Generally, the discretization of a continuous attribute ($c \in C$) involves converting it into a finite number of intervals with a set of candidate cut points $P^c = \{P_1^c, P_2^c, \ldots, P_{k-1}^c, P_k^c\}$, where P_1^c and P_k^c are the minimum and the maximum values of c, respectively, and $P_1^c < P_2^c < P_3^c < \ldots < P_{k-1}^c < P_k^c$. Note here that a cut point is a real number within the range of continuous values. One interval is less than or equal to the cut point, while the other interval is higher than the cut point since the cut point divides the range into two intervals [20]. Thus, a conditional attribute can be defined as $c = [P_1^c, P_2^c] \cup (P_2^c, P_3^c] \cup \ldots \cup (P_{k-1}^c, P_k^c]$. In addition, the pair $(c, I_{(i,k)}^c)$ defines the conditional attribute ($c \in C$), while $I_{(i,k)}^c$ denotes i^{th} interval ($i = 1, 2, \ldots, k$) k intervals of the corresponding conditional attribute. The number of cut points needs $k-1$ for the upper limit, while the lower limit is 1 except for the minimum (P_1^c) and the maximum (P_k^c) cut points. For example, the equal width method is one data discretization method where a continuous attribute is divided in intervals of equal width. The relative strength index (RSI), which ranges from 0 to 100, is one of the most popular technical momentum indicators. As an example, if the RSI is discretized using the equal width method (in this case, $k = 4$), it can be denoted as $RSI = [0, 25] \cup (25, 50] \cup (50, 75] \cup (75, 100]$. A pair $(RSI, I_{(2,4)}^{RSI})$ refers to an RSI with the 2nd (25, 50] among 4 intervals.

After the discretization process, the original decision table $S = <U, C, D>$ is replaced with a new decision table $S^{dis} = <U, C^{dis}, D>$. Rough set analysis then explores subset of attributes with the discretized data for extracting decision rules. Generally, some attributes may be redundant in the decision table (or information system), and thus can be removed without losing essential information. Therefore, the creation of reducts is an essential process in rough set analysis since reducts refer to a minimal subset of attributes. For this study, the rule discovery mechanism, which enables the GA to randomly search reducts, is used to explore all the possible reducts among combinations of technical indicators (refer to the next step).

Based on generated reducts, decision rules are expressed in the form of '*IF condition(s), THEN decision*', which combines the predecessor with the successor of the decision rule. As a result, a *decision rule*(*dr*) can be expressed as:

$$\begin{aligned}
&\textit{IF value of technical indicator } (c_1) \textit{ belongs to } \left(c_1, I_{(i,k)}^{c_1}\right)\\
&\textit{And value of technical indicator } (c_2) \textit{ belongs to } \left(c_2, I_{(i,k)}^{c_2}\right)\\
&\textit{And} \ldots\ldots\\
&\textit{And value of technical indicator } (c_n) \textit{ belongs to } \left(c_n, I_{(i,k)}^{c_n}\right)\\
&\textit{Then Up} (+1) \textit{ or Down } (-1).
\end{aligned}$$

Thus, a set of finite decision rules can be expressed as $DR = \{dr_1, dr_2, \ldots, dr_l\}$. Each decision rules is evaluated for quality by measurements such as support, accuracy, and coverage [45]. The support of a decision rule is the number of objects that corresponds with the predecessor of the rule. The accuracy is calculated by dividing the support of the successor of the rule by the support of the predecessor of the rule. The coverage is calculated by dividing the support of the predecessor of the rule by the total number of objects. In this study, insufficient decision rules are filtered out by criteria, such as support (>20% of the total number of objects in each training dataset), and coverage and accuracy greater than 20% and 50%, respectively. This means that a particular decision rule should account for at least 20% of each trading dataset; decision rules were rarely generated when the support and coverage were greater than 30% in this experiment. The reason for the accuracy greater than 50% is because it is dealing with a problem of classifying the market direction into two categories (i.e., either up or down).

4.2.2. Discovery of optimal decision rules using the GA

For this step, a GA is employed to improve the decision rules extracted from the previous step. If too many or too few cut points for discretization or cardinality of reducts are considered when generating decision rules based on rough set analysis, the rules are difficult to apply to the market since they are often either too complicated or too simplistic [19]. Thus, we consider some constrains to resolve these limitations when implementing the GA, which needs three sets of parameters for a chromosome (or an individual) to represent decision rules. The first set is a reduct (i.e., 3 = cardinality of reduct) among 37 technical indicators. The second and third sets represent the number of cut points (i.e., 1, 2, and 3) and the candidate cut points (i.e., divide an attribute into 20 equal intervals) for discretization of the corresponding reduct elements. The representation of the chromosome structure used in the GA is shown in Fig. 2. As a result, the GA searches the optimal or sub-optimal reducts at the same time as the number of cut points and candidate cut points for discretization in the rule discovery mechanism.

To explore the optimal parameters for discovering optimal and sub-optimal rules, it is necessary to define a suitable objective function to evaluate the chromosome fitness. In this study, the Sharpe Ratio is used in the training period as the fitness, similar to previous studies [2,10,41]. The Sharpe ratio is a risk-adjusted measure of return [40]. The objective function denotes SR as follows:

$$\textit{Maximize } SR = \frac{E(R)}{\sigma}$$

where $E(R)$ is the average return rate of the trading results, and σ is the standard deviation of the return rate. Additionally, the number of trades (i.e., $\geq$30) is a constraint for statistical sampling.

The encoded chromosomes are evolved to maximize the fitness function (i.e., Sharpe ratio). In order to ensure the propagation of the elite chromosome, this process implements the genetic operations, including crossover and mutation, with elitist selection. In this study, this elitism selects the chromosomes with the best fitness values to be the off-spring of the next generation while the remaining chromosomes execute the generic operations. In addition, to enhance the quality of the decision rules in this process, decision rules with lower support and coverage are filtered out.

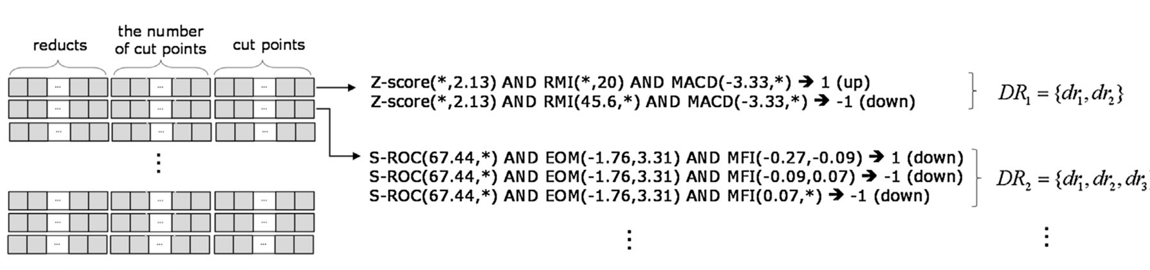

Fig. 2. Chromosome structure of the GA.

Thus, a set of decision rules $DR = \{dr_1, dr_2, \ldots, dr_l\}$ mined from rough set analysis is replaced with the set of optimal decision rules (DR^*) through GA optimization. The elite chromosomes in the last generation are applied as trading rules in the testing period. When using the GA for discovering the optimal sets of decision rules, the GA parameters are significant since they affect the scope of the search space and the time complexity during the evolutionary process [31]. Nevertheless, there is not a general rule for setting the GA parameters [18,32]. Therefore, the appropriate population size and elitism rate are considered by the number of sets of trading rules in this study (i.e., maximum is 200). Others parameters are determined experimentally through trial-and-error for good performance. Through preliminary experiments, the population size (1000), elitism rate (0.2), crossover rate (0.5), and mutation rate (0.06) are predetermined. The stopping criteria is determined as 1000 generations in this study.

4.3. Phase 3: trading signal generation

In phase 3, the sets of decision rules generated from elite chromosomes in the last generation needs to be converted to generate signals for trading the futures contract. In order to get some insight into how the decision rules are converted into the buying, selling, and holding signals, we calculate the sum of the decision values (Up (=1) or Down (=-1)) of the decision rules. According to the number of sets of decision rules (J), a trading signal TS at time t is calculated as follows:

$$TS_t = \sum_{i=1}^{J} DR^*_{i,t}$$

where J is the number of the elite chromosomes in the last generation (i.e., 1st$\sim J^{\text{th}}$ of decision rules) during the training data, and trading signal TS_t is the sum of the decision value of rules at time t. To generate a trading signal, the threshold level of determining the buying and selling signals is set to zero (i.e., $TS_t = 0$), such that if TS_t is positive (i.e., $TS_t > 0$) at t and the position of the futures contract was none at time $t-1$, the trading signal recommends the long position (i.e., buy signal). On the other hand, if TS_t is negative (i.e., $TS_t < 0$) at time t and the position of the futures contract was none at time $t-1$, the trading signal recommends the short position (i.e., sell signal). If TS_t is zero (i.e., $TS_t = 0$) at time t and the position of the futures contract was none at time $t-1$, no action signal is generated, whereas if TS_t is zero (i.e., $TS_t = 0$) at time t and the position of the futures contract was the long (short) position at time $t-1$, the long (short) position is the holding signal. The trading strategy is described as follows:

IF *TS_t is more than 0*
If the position is none ($t-1$) then ***Long*** *position*
Else If the position is ***Short*** *position ($t-1$) then* ***Exit*** *position*
Else ***Holding*** *position*
Else IF *TS_t is less than 0*
If the position is none ($t-1$) then ***Short*** *position*
Else If the position is **Long** *position ($t-1$) then* ***Exit*** *position*
Else ***Holding*** *position*
ELSE No *position (i.e., $TS_t = 0$)*

5. Experimental results and analysis

5.1. Datasets

The dataset used for the experiments in this study involves the 30-min open, high, low, close, and volume of the Korea Stock Price Index 200 (KOSPI 200) futures contract. The dataset was obtained from the CHECKExpert terminal of the Korea Securities Computing Corporation (KOSCOM). The training and testing periods included trending up, down, and flat market dynamics, which is appropriate for evaluating the trading performance.

5.2. Experimental design

The main aim of designing experiments is to investigate the effectiveness of the rule discovery mechanism that consists of functions for searching cut points for discretization and reducts, step by step. Therefore, we design three experimental trading models according to the finding reducts methods: (1) random approach, (2) correlation approach, and (3) GA approach. In order to explore the impact of the number of sets of decision rules on the performance of the trading system, we examine trading performance by using a different number of decision rules sets. In terms of trading performance, common evaluation measures, including the annualized return ratio (%) and the Sharpe ratio are used. The maximum drawdown ratio (%), which indicates the maximum cumulated profit percent drop from a price peak to bottom during the testing period, is also used. In addition, a 0.25% transaction cost is applied for calculating trading performance.

5.3. Experimental results

5.3.1. Random approach

In this experiment, reducts are created based on 10 out of 37 technical indicators which are randomly selected for each training period, employing the GA discretization method. We need to identify whether various technical indicators are helpful for discovering trading rules when creating reducts for developing the rule discovery mechanism. The experiments are compared to the different objective function (i.e., maximize Sharpe ratio vs profit) for evaluating trading rules. A sliding window method, with or without the validation dataset for validating trading rules, is also utilized. For the sliding window method with the validation dataset, the data is divided into a training dataset (2 months), a validation dataset (1 month), and a testing dataset (1 month). The validation dataset is designed to prevent over-fitting the trading rules. That is, a set of extracted trading rules in the training dataset is evaluated using each trading performance (i.e., Sharpe ratio or profit) in the validation dataset before applying discovered trading rules into the testing dataset. A set of decision rules is then rearranged by the trading performance. On the other hand, the data for the sliding window method without a validation dataset is divided into a training dataset (3 months) and a testing dataset (1 month). The testing dataset is the same period from 01/02/2008 to 12/31/2014.

Table 1 shows the trading performance of the rule discovery mechanism with the random approach. The number of sets of decision rules are determined in descending order (i.e., $J \in \{1, 6, 15, 30, 60, 120\}$) of each fitness (i.e., Sharpe ratio or profit) among all combination reducts (i.e., the total number of reducts $\leq \frac{10\times9\times8}{3\times2\times1} = 120$) based on a random selection of 10 technical indicators for each training period. From the results of the experiment (see Table 1), it is apparent that the rule discovery mechanism using the fitness function of the Sharpe ratio maximization generally outperforms the rule discovery mechanism using the fitness function of the profit maximization. Furthermore, the rule discovery mechanism using the sliding window method without a validation set outperforms, on average, the sliding window method with a validation dataset. Both the sliding window without the validation dataset and the objective function (i.e. the Sharpe ratio) are determined next using experiments for correlation and the GA approaches. According to the number of sets of decision rules, proposed $J = 30, 60$ obtain an annualized return rate of 5.66% (the maximum drawdown: 28.99%) and 5.51% (the maximum drawdown: 22.86%), respectively.

Table 1
Trading performance of the random approach.

Model	Measure	Return rate (%) by the number of sets of decision rules (=J)					
		1	6	15	30	60	120
Maximize Sharpe ratio	Annualized return (%)	1.55	2.29	4.45	5.66	5.51	3.53
	Standard deviation	10.27	6.68	12.85	11.08	11.1	7.71
	Maximum drawdown (%)	−41.07	−32.64	−40.13	−28.99	−22.86	−28.29
	Sharpe ratio	0.15	0.34	0.35	0.51	0.5	0.46
Maximize profit	Annualized return (%)	3.19	3.43	−0.45	4.04	3.62	1.8
	Standard deviation	10.39	22.75	12.89	14.75	10.67	13.22
	Maximum drawdown (%)	−29.97	−74.31	−40.78	−41.9	−38.02	−50.53
	Sharpe ratio	0.31	0.15	−0.04	0.27	0.34	0.14
Maximize Sharpe ratio and validation dataset	Annualized return (%)	3.87	3.09	3.74	4.34	3.44	2.41
	Standard deviation	17.02	14.77	10.65	14.63	17.45	15.32
	Maximum drawdown (%)	−28.76	−41.24	−29.76	−35.83	−21.79	−36.87
	Sharpe ratio	0.23	0.21	0.35	0.3	0.2	0.16
Maximize Profit and validation dataset	Annualized return (%)	1.26	−4.85	1.85	−2.46	2.41	0.11
	Standard deviation	14.98	11.69	13.17	13.1	13.48	18.64
	Maximum drawdown (%)	−42.22	−74.66	−40.43	−43.8	−45.74	−41.28
	Sharpe ratio	0.08	−0.42	0.14	−0.19	0.18	0.01

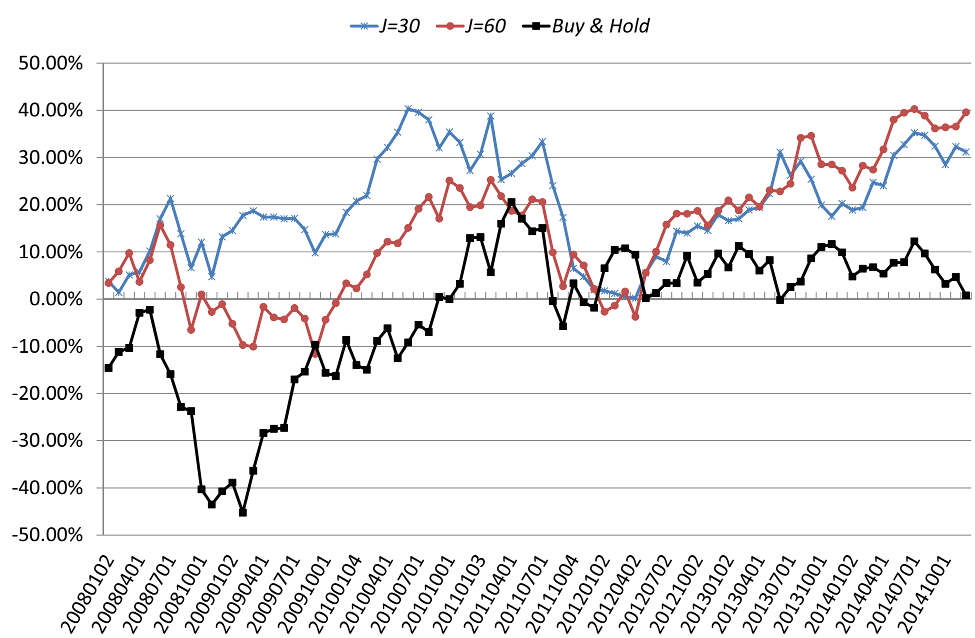

Fig. 3. Cumulative return for the rule discovery mechanism with the random approach.

Fig. 3 describes an illustration of the accumulated return rate of the rule discovery mechanism using the Sharpe ratio maximization and sliding window method without the validation dataset according to the number of sets of decision rules (i.e., J = 30, 60). When comparing to the buy-and-hold strategy, this random approach generates a higher accumulated return rate and Sharpe ratio. However, although this approach is better than the buy-and-hold strategy during given testing period, it is difficult to apply as a practical trading system; this approach cannot protect the downside risk from the financial crisis in 2011. This result implies that various technical indicators might help to improve the trading performance by defending the downside risk since reducts are generated by only 10 technical indicators, randomly selected for each training period in this experiment.

5.3.2. *Correlation approach*

In this experiment, two experiments are compared according to the function of determining cut points for discretization. Since we need to identify whether the discretization method employing the GA (DMGA) is useful, the rule discovery mechanism without the function of searching reducts is used for comparing the other discretization method (i.e., equal frequency binning method, EFB) as applied by previous studies [19,22,23,45]. Reducts (i.e., subset technical indicators) in rough set analysis are generated equally, but the result of the discretization is different. We used the correlation analysis between the value of each technical indicator and the closing price of the KOSPI 200 futures for creating equal reducts. As a result, 10 technical indicators with high correlation for each training period are chosen, but discretization of each indicator is different. In this experiment the size of the training period is three months, with a testing period of one month for evaluating trading performance (i.e., the testing dataset: 01/02/2008–12/31/2014). A moving window approach is used again for training and testing. The number of sets of decision rules are determined in descending order of fitness (i.e., $J \in \{1, 6, 15, 30, 60, 120\}$) among all combination reducts (i.e., the total number of reducts $\leq \frac{10\times9\times8}{3\times2\times1} = 120$).

Tables 2 and 3 summarize trading performance of the rule discovery mechanism with the DMGA and EFB, respectively, according to the number of sets of decision rules. When comparing the rule discovery mechanism with the EFB, the rule discovery mechanism with DMGA generates a higher annualized return rate and Sharpe ratio, while also offering less maximum drawdown for all experiments regardless of the number of sets of decision rules, apart from the best set of decision rules (i.e., $J = 1$). The rule discovery mechanism with the DMGA generated a Sharpe ratio from 0.30 to 0.44. These ratios are much higher than those achieved using other models. In particular, the rule discovery mechanism with DMGA ($J = 15$, 30) led to more profit (the annualized return rate: 6.37% and 6.87%) and less risk (the maximum drawdown: 20.32% and 23.84%) than other trading performance. The results indicate that it is worthwhile to conduct the rule discovery mechanism with DMGA, which makes the intelligent hybrid trading system more adaptive to discovering trading rules.

Figs. 4 and 5 present an illustration of the accumulated return rate of the trading systems using the rule discovery mechanism with DMGA and EFB, respectively. As shown, two models (with the number of sets of decision rules, $J = 15$, 30, 60) outperform the buy-and-hold strategy during the testing period. The results indicate that the rule discovery mechanism is efficient for discovering trading rules, and that the rule discovery mechanism with DMGA is more stable than the rule discovery mechanism with EFB. However, although the trading system using the rule discovery mechanism with EFB is better than the buy-and-hold strategy during the given testing period, it is difficult to apply as a trading system in practice since the maximum drawdown is higher than the buy-and-hold strategy. Interestingly, trading systems using the best set of decision rules ($J = 1$) and multiple rules ($J = 120$) do not achieve satisfactory levels during the testing period. The results imply the number of decision rules may have an impact on the trading performance.

5.3.3. GA approach

In this approach, we consider two parameters to help identify the impact of the size of the training or learning period for discovering both trading rules and the number of sets of decision rules for trading futures contracts. As discussed earlier, the data size for training/learning is an important factor that has been observed to have a large influence on the experimental results since the appropriate periods for training can improve trading performance. Thus, the intelligent hybrid trading system for discovering trading rules is trained with a 3, 6, 9, and 12 month period for discovering trading rules, and then is tested on the following one month period for 84 months (see Table 4).

In this experiment, the trading performance obtained from the proposed trading system is carefully analyzed according to the combinations of size of training periods *TP* (i.e., $TP \in \{3, 6, 9, 12\}$) for the sliding window method and the number of sets of decision rules (i.e., $J \in \{1, 10, 25, 50, 100, 200\}$). The total number of experiments was 24. The rule discovery mechanism that consists of both functions for searching cut points for discretization and reducts is applied to discover trading rules in the experiments. Furthermore, the trading performance is evaluated against the results of the buy-and-hold trading strategy and the conventional approach, which generates trading rules using rough set analysis without the rule discovery mechanism (i.e., *Phase 2*) [22,23,45].

From Tables 5–8, the rule discovery mechanism with $TP = 6$ and $J = 50$ generates the best trading performance (the annualized return rate: 10.59%, standard deviation: 8.00, Sharpe ratio: 1.32), while the rule discovery mechanism with $TP = 12$ and $J = 100$ is the worst trading performance among the 24 experiments (the annualized return rate: −10.69%, standard deviation: 18.04, Sharpe ratio: −0.59). In particular, when we analyzed the overall trading performance according to the size of the training periods, the average profitability through six months periods (i.e., $TP = 6$) obtained better trading performance in all aspects (see Tables 5–8). The results indicate that the use of a six-month training period is more profitable on average than other training periods. It also implies that short training periods (i.e., $TP = 3$) do not provide a rule discovery mechanism with the information needed to discover decision rules. On the other hand, the longer training periods (i.e., $TP = 9$, 12) are quite bad for the rule discovery mechanism because the search space is too large to easily search and identify solutions.

In terms of the number of sets of decision rules, using the best decision rule (i.e., $J = 1$), or multiple decision rules (i.e., $J = 200$), is not optimal for trading systems regardless of the size of trading period. Multiple trading rules generate whipsaw signals, while trading system with the best rule or a single rule is not appropriate to apply to the testing period due to the over-fitting problem.

The cumulative returns of the rule discovery mechanism with $TP = 6$ and $J = 25$, 50 during the testing period are given in Fig. 6, along with the buy-and-hold trading strategy and the conventional approach. For the examined dataset, the annualized return rate of the buy-and-hold trading strategy is 0.11% (standard deviation: 22.55), the maximum drawdown ratio is 43.01%, and the Sharpe ratio is 0.005, whereas the annualized return rate of the conventional approach is 2.03% (standard deviation: 6.96), the maximum drawdown ratio is 21.5%, and the Sharpe ratio is 0.29. According to the number of sets of decision rules, our model earns remarkable total return rates at the end of the seven years (i.e., 72.31% and 74.11%, for $J = 25$, 50, respectively). Furthermore, the maximum drawdown of the proposed models with $TP = 6$ and $J = 25$, 50 is 24.60% and 15.81% during the testing period, respectively. For statistical analysis, the Wilcoxon signed rank test is conducted instead of the paired *t*-test due to the non-normal distributions of monthly trading returns. At a 10% statistical significance level, the difference between the buy-and-hold trading strategy and the proposed model is significant (*p*-value: 0.041, Z-value: −2.043). The difference between the conventional approach and the proposed model is also significant (*p*-value: 0.059, Z-value: −1.889). As a result, the proposed model with $TP = 6$ and $J = 50$ obtains the highest cumulative returns with the lowest drawdown percentage. The proposed model is also more prominent when the market condition is flat, which offers opportunity for traders. Moreover, the best trading model outperforms *RCTS* (the annualized return rate: 10.33%, standard deviation: 21.86, Sharpe ratio: 0.47, the maximum drawdown: 46.05%) when comparing to the previous study by Kim and Enke [19]. These results indicate that the proposed model is relatively more stable and provides reliable trading performance since both the Sharpe ratio and the maximum drawdown are improved by more than 50%. To test for a difference in the trading performance between the proposed model and *RCTS*, a statistical test is conducted with respect to the monthly trading returns according to the Wilcoxon rank-sum test (or the Mann-Whitney test), which is a nonparametric alternative to the two-sample *t*-test. The result is statistically significant at the 10% significance level (*p*-value: 0.083, Mann-Whitney U: 2157.50, Wilcoxon W: 4784.50).

Table 2
Trading performance of the rule discovery mechanism with EFB.

Year	Return rate (%) by the number of sets of decision rules (=J)					
	1	6	15	30	60	120
2008	8.99	14.86	30.63	20.88	25.20	−26.09
2009	15.61	18.00	15.04	10.59	19.04	22.32
2010	12.53	0.62	21.20	15.64	15.30	29.44
2011	−25.48	−36.98	−33.49	−33.83	−34.68	7.73
2012	−3.98	8.54	10.46	9.29	10.99	5.58
2013	12.25	0.36	−2.92	−1.60	5.99	−1.83
2014	−3.88	−0.51	−7.73	−7.03	−1.28	−3.90
Annualized return (%)	2.29	0.70	4.74	1.99	5.79	4.75
Standard deviation	13.49	16.83	19.84	17.10	18.35	16.89
Maximum drawdown (%)	53.85	54.11	46.18	53.19	53.87	36.57
Sharpe ratio	0.17	0.04	0.24	0.12	0.32	0.28

Table 3
Trading performance of the rule discovery mechanism with DMGA.

Year	Return rate (%) by the number of sets of decision rules (=J)					
	1	6	15	30	60	120
2008	−10.33	2.83	8.33	−12.99	−21.04	16.94
2009	−13.85	−18.27	−16.89	−4.11	8.12	23.20
2010	7.52	29.12	29.99	38.07	44.12	1.83
2011	−2.11	4.73	6.84	10.46	−6.31	−25.69
2012	−4.88	−2.77	3.73	0.94	23.37	9.76
2013	35.08	19.83	19.96	16.40	−12.99	21.52
2014	−0.58	−1.73	−7.69	−0.68	8.07	−2.41
Annualized return (%)	1.55	4.82	6.32	6.87	6.19	6.45
Standard deviation	15.11	14.41	14.60	15.51	20.72	15.86
Maximum drawdown (%)	30.36	26.91	20.32	23.84	36.61	44.78
Sharpe ratio	0.10	0.33	0.43	0.44	0.30	0.41

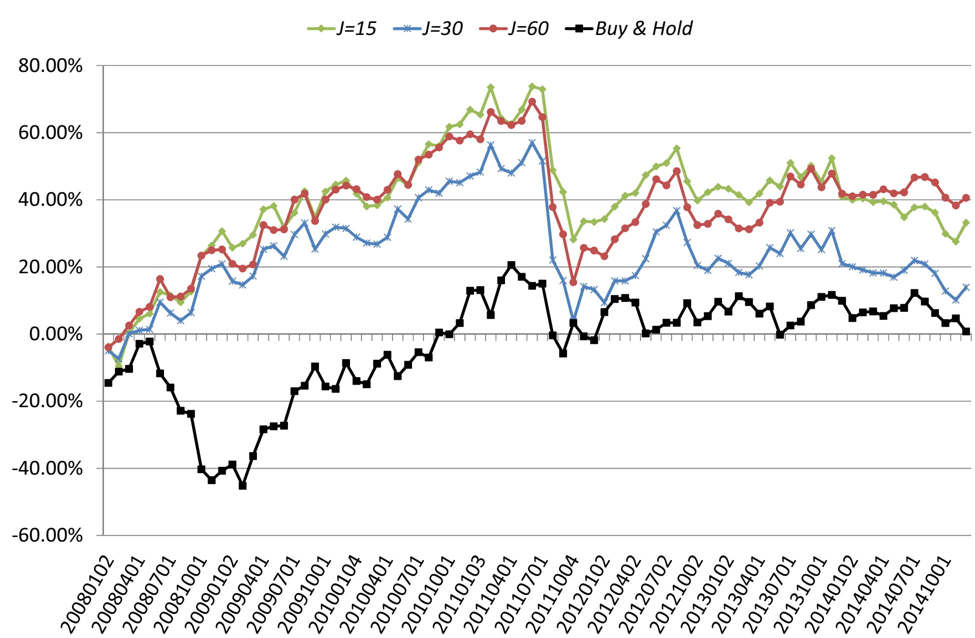

Fig. 4. Cumulative return for the rule discovery mechanism with EFB.

Table 4
Sliding window scheme and the experimental dataset.

Training dataset		Testing dataset		Total number of testing windows
Start date	Window size	Start/End date	Window size	
Oct. 1, 2007	3 months	Jan.1, 2008/ Dec. 31, 2014	1 month	84
Jul. 1, 2007	6 months			
Apr. 1, 2007	9 months			
Jan. 1, 2007	12 months			

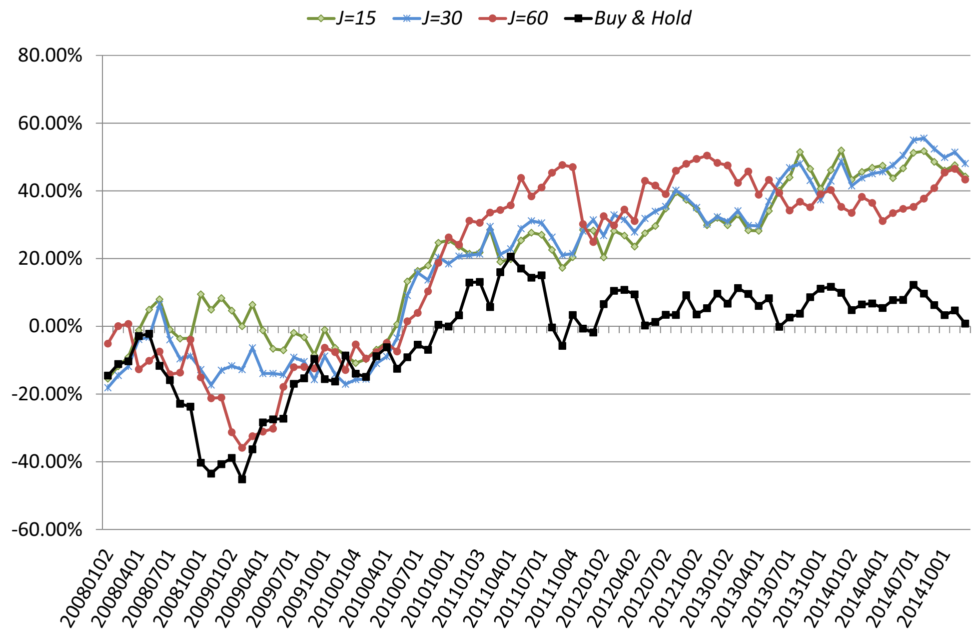

Fig. 5. Cumulative return for the rule discovery mechanism with DMGA.

Table 5
Trading performance of the rule discovery mechanism using a 3 month training period.

Year	Return rate (%) by the number of sets of decision rules (=J)					
	1	10	25	50	100	200
2008	−9.53	9.42	10.91	9.56	11.18	10.47
2009	22.62	13.16	15.53	18.30	4.07	1.53
2010	6.20	4.60	−9.37	−8.86	7.95	8.46
2011	−4.35	−5.37	−2.32	2.64	−5.24	−14.55
2012	2.34	13.68	25.01	12.93	−1.94	−0.58
2013	6.94	−10.37	−7.18	−8.41	−3.77	10.27
2014	12.31	10.99	11.52	12.55	13.51	8.95
Annualized return (%)	5.22	5.16	6.30	5.53	3.68	3.51
Standard deviation	9.82	8.79	11.86	9.95	6.95	8.41
Maximum drawdown (%)	30.57	30.12	28.20	16.83	23.54	27.50
Sharpe ratio	0.53	0.59	0.53	0.56	0.53	0.42

Table 6
Trading performance of the rule discovery mechanism using a 6 month training period.

Year	Return rate (%) by the number of sets of decision rules (=J)					
	1	10	25	50	100	200
2008	26.83	6.98	12.85	6.83	4.21	−2.88
2009	−21.64	9.69	−7.11	1.87	13.51	5.54
2010	−1.38	8.97	3.71	4.47	0.11	5.18
2011	−2.22	−17.38	9.03	6.69	18.90	8.18
2012	3.96	21.17	25.48	25.37	10.14	9.56
2013	−16.45	4.26	23.52	19.79	2.83	11.14
2014	23.18	12.10	4.84	9.07	−5.20	−4.88
Annualized return (%)	1.75	6.54	10.33	10.59	6.36	4.55
Standard deviation	16.89	10.95	10.62	8.00	7.67	5.70
Maximum drawdown (%)	47.80	34.85	24.60	15.81	24.19	28.10
Sharpe ratio	0.10	0.60	0.97	1.32	0.83	0.80

6. Concluding remarks

For this study, an intelligent hybrid trading system using rough set analysis and a GA was developed for discovering trading rules in the futures market. A novel rule discovery mechanism was developed for discovering optimal and sub-optimal trading rules by employed by the GA to evolve decision rules extracted by rough set analysis. Testing results illustrated the profitability of the trading system according to the number of sets of decision rules and the size of training period in the KOSPI 200 futures market. An intelligent hybrid trading system for discovering trading rules with a six-month training period and 50 sets of decision rules provided the highest annualized return rate compared to other experimental combinations, and also implies that combining the appropriate number of sets of decision rules and the size of training period can improve trading performance. The performance of the trading system indicates that the rule discovery mechanism is able to find profitable trading rules in the data. The results further demonstrate

Table 7
Trading performance of the rule discovery mechanism using a 9 month training period.

Year	Return rate (%) by the number of sets of decision rules (=J)					
	1	10	25	50	100	200
2008	−12.67	−2.15	−18.21	−27.46	23.19	5.01
2009	−5.94	0.26	−6.14	1.36	−11.69	−8.78
2010	−12.38	0.28	5.73	−2.02	7.26	14.85
2011	33.06	18.49	28.31	25.65	0.00	6.09
2012	1.09	−7.09	30.61	29.44	31.51	16.57
2013	21.17	21.32	9.76	23.07	14.81	13.21
2014	9.88	6.75	1.98	−1.83	1.13	−3.13
Annualized return (%)	4.89	5.41	7.44	6.89	9.46	6.26
Standard deviation	16.13	9.95	16.27	18.85	13.70	8.82
Maximum drawdown (%)	49.16	35.34	36.98	33.06	20.51	28.69
Sharpe ratio	0.30	0.54	0.46	0.37	0.69	0.71

Table 8
Trading performance of the rule discovery mechanism using a 12 month training period.

Year	Return rate (%) by the number of sets of decision rules (=J)					
	1	10	25	50	100	200
2008	−30.87	−43.94	−33.38	−32.73	−35.92	−31.27
2009	−11.25	−29.23	−30.55	−27.76	−34.19	−23.97
2010	8.71	7.29	20.13	1.62	6.73	−2.66
2011	−11.12	−17.81	19.13	−0.28	−10.82	−3.47
2012	−8.01	6.84	16.25	29.21	−1.30	0.85
2013	2.51	25.48	22.71	9.71	15.40	31.83
2014	12.40	−7.14	−14.36	−16.53	−14.74	−14.32
Annualized return (%)	−5.37	−8.36	−0.01	−5.25	−10.69	−6.14
Standard deviation	13.63	22.08	23.31	20.26	18.04	18.99
Maximum drawdown (%)	76.03	94.19	50.55	77.15	81.72	63.80
Sharpe ratio	−0.39	−0.38	0.00	−0.26	−0.59	−0.32

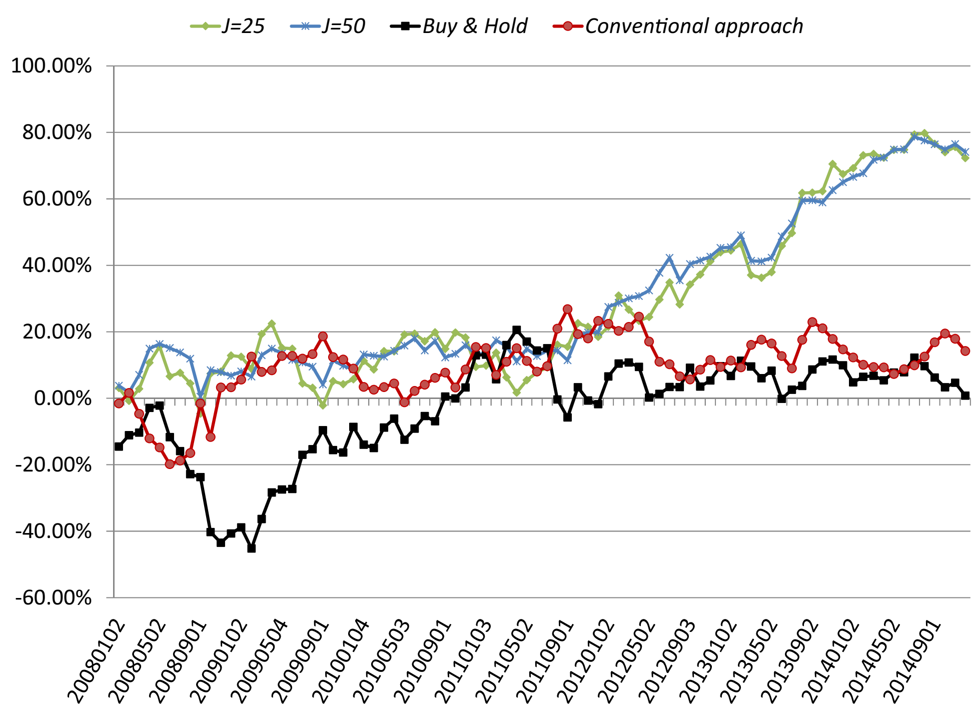

Fig. 6. Cumulative return for the rule discovery mechanism using a 6 month training period.

that rough set analysis can help to generate decision rules in the rule discovery mechanism, while the GA can help to improve the decision rules.

Our study provides two important contributions. First, a rule discovery mechanism for resolving the optimization problem of data discretization and reducts in rough set analysis by using the GA is presented. Although these processes may cause information loss, our intelligent trading system model performs better when applying the GA approach since it yields the highest return and lowest risk for the tested dataset. Consequently, we can summarize that our study extends discovery trading rule models by solving the optimization problem. Second, we examined the trading performance according to comparable approaches (i.e., random, correlation, and GA approach), as well as other factors, including the sliding window method with or without the validation dataset, the number of trading rules, and the size of the training period. This has resulted in a comprehensive experimental study that covers most of the trading system topics currently in the literature. Such a comprehensive comparison had not previously been performed.

However, financial data, such as stock market data, is known to be complex and high-dimensional data. Therefore, although the GA is a popular search algorithm, as the higher size and dimensionality of the data increase, the longer and more difficult it can be to find a solution given the increased search space. Thus, it is necessary to utilize data reduction techniques, such as object (i.e., instance) and attribute selection methods before searching for a solution.

For further study, complexity analysis may be necessary for applying the system to real-time market trading. It might be possible to extend the proposed model by generating decision rules according to the current market situation (i.e., up, down, and flat trend). Furthermore, it may be possible to develop an expert system by using the rule discovery mechanism as a knowledge discovery tool. Thus, a hybrid knowledge-based system consisting of decision support and an expert system can be employed to develop a new trading system.

Acknowledgements

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Science, ICT & Future Planning (NRF2014R1A2A1A11052670).

Appendix A. Technical indicators

Notes: Where H: High price, L: Low price, C: Close price, V: Volume, MA: Moving average, and EMA: Exponential moving average at time t.

Technical Indicators	Formula	Values
%D (Stochastics%D)	$\%D(n) = \sum_{i=0}^{n-1} \%K_{t-i}/n$	$n = 12$
%K (Stochastics%K)	$\%K(n) = \frac{C - L_{n,\min}}{H_{n,\max} - L_{n,\min}} \times 100$	$n = 12$
Band%b	$Band\%b(n) = \frac{C - L(n)}{U(n) - L(n)}$ $U(n) = \bar{C}(n) + \left[\alpha \times \sqrt{\frac{\sum_{i=0}^{n-1}\left(C_{t-i} - \bar{C}(n)\right)^2}{n}}\right]$ $L(n) = \bar{C}(n) - \left[\alpha \times \sqrt{\frac{\sum_{i=0}^{n-1}\left(C_{t-i} - \bar{C}(n)\right)^2}{n}}\right]$	$n = 20$ $\alpha = 2$
Band width	$Bandwidth(n) = \frac{U(n) - L(n)}{\bar{C}(n)}$	$n = 20$
CCI (Commodity Channel Index)	$CCI(n) = \frac{M - \bar{M}(n)}{d(n) \times 0.015}$ $M = \frac{H+L+C}{3}$ $d(n) = \frac{1}{n}\sum_{i=0}^{n-1} \lvert M_{t-i} - \bar{M}_t(n) \rvert$	$n = 9$
CMF (Chaikin Money Flow)	$CMF(n) = \sum_{i=0}^{n-1} MFV_{t-i} / \sum_{i=0}^{n-1} V_{t-i}$ $MFV = \frac{(C-L)-(H-C)}{H-L} \times V$	$n = 20$
CO (Chaikin's Oscillator)	$CO(m,n) = EMA(AD, m) - EMA(AD, n)$ $AD = AD_{t-1} + \left(\frac{(C-L)-(H-C)}{(H-L)}\right) \times V$	$m = 3$ $n = 10$
CV (Chaikin Volatility)	$CV(m,n) = (\frac{K}{K_{t-n}} - 1) \times 100, K = EMA(H - L, m)$	$m = 10$ $n = 10$
Disparity	$Disparity(n) = \frac{C}{MA(C,n)} \times 100$	$n = 20, 60$
DPO (Detrended Price Oscillator)	$DPO(n) = C - MA(C, (n/2) + 1)$	$n = 14$
EOM (Ease of Movement)	$EOM = \left(\frac{H+L}{2} - \frac{H_{t-1}+L_{t-1}}{2}\right) / \frac{V}{H-L}$	
FI (Force Index)	$FI(n) = (C - C_{t-1}) \times V$	$n = 2, 13$
MACD (Moving Average Convergence-Divergence)	$MACD(m,n) = EMA(C, m) - EMA(C, n)$	$m = 12$ $n = 26$
MAO (MA Oscillator)	$MAO(m,n) = MA(C, m) - MA(C, n)$	$m = 5$ $n = 20$
MFI (Money Flow Index)	$MFI = 100 - \frac{100}{(100+MR)}$ $TP = \frac{H+L+C}{3}, MF = TP \times V$ $MR = \frac{PositiveMF}{NegativeMF}$	
MI (Mass Index)	$MI(n) = \sum_{i=0}^{n-1} \frac{EMA_{t-i}(r,n)}{EMA^2_{t-i}(r,n)}, r = H - L$	$n = 9$
Momentum	$Momentum(n) = \frac{C}{C_{t-n}} \times 100$	$n = 10$

NCO (*Net Change Oscillator*)	$NCO(n) = C - C_{t-n}$	$n = 12$
PO (*Price Oscillator*)	$PO(m, n) = \frac{MA(C,m)-MA(C,n)}{MA(C,m)}$	$m = 5$ $n = 10$
Psychology	$Psy(n) = \frac{thenumberofuptrend}{n}$	$n = 10$
RMI (*Relative Momentum Index*)	$RMI(n) = \frac{\sum_{i=0}^{n-1} Up_{t-i}}{\sum_{i=0}^{n-1} Up_{t-i} + \sum_{i=0}^{n-1} Down_{t-i}} \times 100$	$n = 14$
ROC (*Rate of Change*)	$ROC(n) = (\frac{C}{C_{t-n}} - 1) \times 100$	$n = 12$
RSI (*Relative Strength Index*)	$RSI(n) = 100 - \frac{100}{1+RS(n)}$ $RS(n) = \sum_{i=0}^{n-1} Up_{t-i} / \sum_{i=0}^{n-1} Down_{t-i}$ where $Up_t(Down_t)$ is upward (downward) price change	$n = 14$
S-ROC (*Smoothed Rate of Change*)	$SROC(m, n) = \frac{EMA(C,n)}{EMA(C,m)} \times 100$	$m = 10 n = 20$
Slow%d (*Stochastics slow%d*)	$\%d(n) = \sum_{i=0}^{n-1} \%D_{t-i}/n$	$n = 12$
SO (*Stochastics Oscillator*)	$SO(n) = \sum_{i=0}^{n-1} \%K_{t-i} \times 100$	$n = 12$
Sonar	$Sonar(n) = EMA(C,\ n) - EMA_{t-n}(C,\ n)$	$n = 25$
Sonar Signal	$SonarSignal(m, n) = EMA(Sonar(n), m)$	$m = 9$ $n = 25$
TRIX	$TRIX(n) = \frac{EMA^3(C,n)-EMA^3_{t-1}(C,n)}{EMA^3_{t-1}(C,n)}$	$n = 12$
VMA (*Volume Moving Average*)	$VMA(n) = MA(V,\ n)$	$n = 20, 60$
VO (*Volume Oscillator*)	$VO(m, n) = \frac{\overline{V}(m)-\overline{V}(n)}{\overline{V}(n)} \times 100$	$m = 12$ $n = 26$
VROC (*Volume Rate of Change*)	$VROC(n) = (\frac{V}{V_{t-n}} - 1) \times 100$	$n = 14$
Williams%R	$Williams\%R(n) = \frac{H_{n,\max}-C}{H_{n,\max}-L_{n,\min}} \times 100$	$n = 14$
Z-score	$Zscore(n) = \frac{C-MA(n)}{SD(C,n)}$ where $SD(C, n)$ is the standard deviation of closing prices for n periods	$n = 20$

References

[1] F. Allen, R. Karjalainen, Using genetic algorithms to find technical trading rules, J. Financ. Econ. 51 (1999) 245–271.

[2] S.D. Bekiros, Fuzzy adaptive decision-making for boundedly rational traders in speculative stock markets, Eur. J. Oper. Res. 202 (1) (2010) 285–293.

[3] M.A. Boyacioglu, D. Avci, An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: the case of the Istanbul stock exchange, Expert Syst. Appl. 37 (2010) 7908–7912.

[4] T. Chavarnakul, D. Enke, A hybrid stock trading system for intelligent technical analysis-based equivolume charting, Neurocomputing 72 (2009) 3517–3528.

[5] Y.-S. Chen, C.-H. Cheng, C.-L. Chiu, S.-T. Huang, A study of ANFIS-based multi-factor time series models for forecasting stock index, Appl. Intell. (2016), http://dx.doi.org/10.1007/s10489-016-0760-8.

[6] Y. Chen, S. Mabu, K. Shimada, K. Hirasawa, A genetic network programming with learning approach for enhanced stock trading model, Expert Syst. Appl. 36 (2009) 12537–12546.

[7] W.C. Chiang, D. Enke, T. Wu, R. Wang, An adaptive stock index trading decision support system, Expert Syst. Appl. 59 (2016) 195–207.

[8] M.A.H. Dempster, C.M. Jones, A real-time adaptive trading system using genetic programming, Quant. Financ. 1 (2001) 397–413.

[9] A. Esfahanipour, S. Mousavi, A genetic programming model to generate risk-adjusted technical trading rules in stock markets, Expert Syst. Appl. 38 (2011) 8438–8445.

[10] Y. Hsu, A. Chen, J. Chang, An inter-market arbitrage trading system based on extended classifier systems, Expert Syst. Appl. 38 (4) (2011) 3784–3792.

[11] J. Hu, W. Pedrycz, G. Wang, K. Wang, Rough sets in distributed decision information systems, Knowl. Based Syst. 94 (2016) 13–22.

[12] Y. Hu, B. Feng, X. Zhang, E.W.T. Ngai, M. Ngai, Stock trading rule discovery with an evolutionary trend following model, Expert Syst. Appl. 42 (2015) 212–222.

[13] Y. Hu, K. Liu, X. Zhang, L. Su, E.W.T. Ngai, M. Liu, Application of evolutionary computation for rule discovery in stock algorithmic trading: a literature review, Appl. Soft Comput. 36 (2015) 534–551.

[14] A. Janusz, D. Ślęzak, Utilization of attribute clustering methods for scalable computation of reducts from high-dimensional data, Proc. Federated Conf. Comput. Sci. Inf. Syst. (2012) 295–302.

[15] X. Jia, L. Shang, B. Zhou, Y. Yao, Generalized attribute reduct in rough set theory, Knowl. Based Syst. 91 (2016) 204–218.

[16] Y. Jing, T. Li, C. Luo, S.-J. Horng, G. Wang, Z. Yu, An incremental approach for attribute reduction based on knowledge granularity, Knowl. Based Syst. 104 (2016) 24–38.

[17] K.-j. Kim, H. Ahn, Simultaneous optimization of artificial neural networks for financial forecasting, Appl. Intell. 36 (2012) 887–898.

[18] K. Kim, I. Han, Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, Expert Syst. Appl. 19 (2) (2000) 125–132.

[19] Y. Kim, D. Enke, Developing a rule change trading system for the futures market using rough set analysis, Expert Syst. Appl. 59 (2016) 165–173.

[20] S. Kotsiantis, D. Kanellopoulos, Discretization techniques: a recent survey, GESTS Int. Trans. Comput. Sci. Eng. 32 (1) (2006) 47–58.

[21] R.K. Lai, C. Fan, W. Huang, P. Chang, Evolving and clustering fuzzy decision tree for financial time series data forecasting, Expert Syst. Appl. 36 (2009) 3761–3773.

[22] S.J. Lee, J.J. Ahn, K.J. Oh, T.Y. Kim, Using rough set to support investment strategies of real-time trading in futures market, Appl. Intell. 32 (2010) 364–377.

[23] S.J. Lee, K.J. Oh, T.Y. Kim, How many reference patterns can improve profitability for real-time trading in futures market? Expert Syst. Appl. 39 (2012) 7458–7470.

[24] C. Luo, T. Li, Z. Yi, H. Fujita, Matrix approach to decision-theoretic rough sets for evolving data, Knowl. Based Syst. 99 (2016) 123–134.

[25] S. Mabu, K. Hirasawa, M. Obayashi, T. Kuremoto, Enhanced decision making mechanism of rule-based genetic network programming for creating stock trading signals, Expert Syst. Appl. 40 (2013) 6311–6320.

[26] N. Mehdiyev, D. Enke, Interest rate prediction: a neuro-hybrid approach with data preprocessing, Int. J. Gen. Syst. 43 (5) (2014) 535–550.

[27] L. Menkhoff, The use of technical analysis by fund managers: international evidence, J. Bank. Financ. 34 (11) (2010) 2573–2586.

[28] M.J. Moshkov, M. Piliszczuk, B. Zielosko, Partial covers, reducts and decision rules in rough sets –Theory and applications, Stud. Comput. Intell. 145 (2008), Springer.

[29] S. Mousavi, A. Esfahanipour, M.H.F. Zarandi, A novel approach to dynamic portfolio trading system using multitree generic programming, Knowl. Based Syst. 66 (2014) 68–81.
[30] H.S. Nguyen, A. Skowron, Quantization of real values attributes, rough set and Boolean reasoning approach, Wrightsville Beach, NC, in: Proc. 2nd Joint Annual Conf. Inf. Sci., 1995, 1995, pp. 34–37.
[31] K.J. Oh, T.Y. Kim, S.-H. Min, H.Y. Lee, Portfolio algorithm based on portfolio beta using generic algorithm, Expert Syst. Appl. 30 (2006) 527–534.
[32] M. Ozturk, I.H. Toroslu, G. Fidan, Heuristic based trading system on Forex data using technical indicator rules, Appl. Soft Comput. 43 (2016) 170–186.
[33] R. Pardo, The Evaluation and Optimization of Trading Strategies, Wiley, 2008.
[34] Z. Pawlak, Rough sets, Int. J. Comput. Inf. Sci. 11 (1982) 341–356.
[35] Z. Pawlak, Rough set approach to knowledge-based decision support, Eur. J. Oper. Res. 99 (1) (1997) 48–57.
[36] Z. Pawlak, Rough sets and intelligent data analysis, Inf. Sci. 147 (2002) 1–12.
[37] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, W. Ziarko, Rough set, Commun. ACM 38 (11) (1997) 88–95.
[38] M. Pring, Technical Analysis Explained: the Successful Investor's Guide to Spotting Investment Trends and Turning Points, McGraw-Hill, 1991.
[39] S. Ramírez-Gallego, S. García, J.M. Benítez, F. Herrera, Multivariate discretization based on evolutionary cut points selection for classification, IEEE Trans. Cyb. 46 (2016) 595–608.
[40] W.F. Sharpe, The sharpe ratio, J. Portfolio Manage. 21 (1994) 49–58.
[41] L. Shen, H.T. Loh, Applying rough sets to market timing decisions, Decis. Support Syst. 37 (2004) 583–597.
[42] F. Wang, P.L.H. Yu, D.W. Cheung, Combining technical trading rules using particle swarm optimization, Expert Syst. Appl. 41 (2014) 3016–3026.
[43] P.S. Wiles, D. Enke, A hybrid neuro-fuzzy model to forecast the soybean complex, in: Proceedings of the 2015 American Society of Engineering Management Conference, Indianapolis, IN, October, 2015.
[44] P.S. Wiles, D. Enke, Optimizing MACD parameters via genetic algorithms for soybean futures, Procedia Comput. Sci. 61 (2015) 85–91.
[45] J. Yao, J.P. Herbert, Financial time-series analysis with rough sets, Appl. Soft Comput. 9 (2009) 1000–1007.
[46] X. Zhang, Y. Hu, K. Xie, W. Zhang, L. Su, M. Liu, An evolutionary trend reversion model for stock trading rule discovery, Knowl. Based Syst. 79 (2015) 27–35.
[47] N. Zhong, J. Dong, Using rough sets with heuristics for feature selection, J. Intell. Inf. Syst. 16 (2001) 199–214.