



University of Technology, Sydney

**IN-DEPTH OPTIMIZATION OF
STOCK MARKET DATA MINING
TECHNOLOGIES**

A Thesis Submitted for the Degree of
Doctor of Philosophy

By

Li LIN

Supervisor: Prof. Chengqi Zhang

Co-Supervisor: Dr. Shichao Zhang

Faculty of Information Technology
University of Technology, Sydney

AUSTRALIA

29 July 2006

Certificate

Author: **Li Lin**

Title: **In-Depth Optimization of Stock Market Data Mining Technologies**

Degree: Philosopher Doctor

Date: 29 July 2006

I certify that this thesis has not already been submitted for any degree and is not being submitted as part of candidature for any other degree.

I also certify that the thesis has been written by me and any help that I have received in preparing this thesis and all sources used have been acknowledged in this thesis.

Signature of Student

Production Note:
Signature removed prior to publication.

Acknowledgements

I would like to extend my immense gratitude to my supervisor Professor Chengqi Zhang for his great selfless support and stimulating discussions, and for his prompting and careful reading of my draft which has been invaluable in the completion of this dissertation. He has provided an important mix of inspiration, sympathy and unfailing good humors, and he always gives me timely feedback for my research ideas, papers and dissertation drafts. I am lucky and happy to have been able to work with him during my PhD studies.

Also, I would like to thank the Faculty of Information Technology at University of Technology, Sydney (UTS) and CMCRC, SIRCA, AC3, SMARTS for providing me an excellent environment and the scholarship for my studies and research work. Especially for the CMCRC CEO Professor Mike Aitkin in the University of New South Wales, Australia, my co-supervisor Dr. Shichao Zhang and our team leader Dr. Longbing Cao in the University of Technology, Sydney, Australia.

Moreover, I appreciate all my other group members and my friends for plentiful help and happiness they have given to me, namely Ms. Li Liu, Mr. Jiaqi Wang, Mr. Wanli Chen, Mr. Jiarui Ni, Dr. Xiaowei Yan, Dr. Qingfeng Chen, Dr. Chunsheng Li. Mr. Zhenxing Qin, Mr. Yanchang Zhao, Ms. Dan Luo, A/Prof. Jie Lu, Dr. Guangquan Zhang, A/Prof. Maolin Huang, A/Prof. Sean He, A/Prof. Yuzhi Yan, A/Prof. Fenxi Zhang, Dr. Cynethia Nelson, Ms. Suneetha Uppu and more not listed here are also remembered in my heart.

Last but not the least, I am very grateful to my parents, my brothers, my wife Yinghui, my daughters Su and Si, who have been patient and supportive during my lives and studies in Australia. Their confidence and understanding are also a great encouragement to me in my academic endeavor.

I'd like to express my sincere thanks to those who once offered some help and concern to my study and life. My heart is filled with gratitude that words cannot express.

Contents

Certificate	i
Acknowledgements	ii
Contents	iv
Abstract	1
Chapter 1. Introduction	4
1.1 Problems	4
1.2 Solutions	8
1.3 Related Work	11
1.4 Significance	15
1.5 Structure of the Thesis	17
Chapter 2. Context	18
2.1 Data	18
2.2 Technical Trading Rules	20
2.2.1 Simple Moving Average	21
2.2.2 Filtered Moving Average	22
2.2.3 Enhanced Moving Average	23
2.2.4 Channel Break-Outs	25
2.2.5 Filter Rules	26
2.2.6 Support and Resistance	28
2.2.7 Other Trading Rules	29
2.3 Evaluation Criteria	29
2.3.1 Profit	30
2.3.2 Return	30
2.3.3 Sharpe Ratio	31
2.4 In-Sample and Out-of Sample Data Sets	32
2.5 Summary	33

Chapter 3. Domain Knowledge Integrated Applications	35
3.1 Problems	35
3.2 Domain-Driven Model	37
3.2.1 Related Concepts	37
3.2.2 Model	37
3.3 Human-Knowledge Interface	40
3.3.1 Domain Knowledge Database	40
3.3.2 Human-Machine Interface and Domain Knowledge In- tegration	41
3.4 Summary	44
Chapter 4. In-Depth Pattern Discovery and Related Applica- tions	45
4.1 Fundamental Concepts	45
4.2 DDID-PD Process Model	48
4.2.1 Constraint-Based Context	49
4.3 Mining In-Depth Patterns	50
4.3.1 Mining In-Depth Patterns	50
4.3.2 Human-Machine-Cooperated Interactive Knowledge Dis- covery	51
4.3.3 Loop-Closed Iterative Refinement	52
4.4 Mining Stock-Rule Pair in Real Stock Markets	53
4.4.1 Mining In-Depth Trading Rules (Sub-Domain)	53
4.4.2 Mining In-Depth Stock-Rule Pairs	57
4.5 Summary	58
Chapter 5. Optimized Algorithms	60
5.1 Standard Genetic Algorithm	60
5.1.1 Background	60
5.1.2 Solution	60
5.1.3 Conclusion	62
5.2 Robust Genetic Algorithms	64
5.2.1 Background	64
5.2.2 Robust Genetic Algorithms	64
5.2.3 Comparison	65
5.3 Fuzzy Set Methods	69
5.3.1 Fuzzy set	70
5.3.2 Output Literal Results	71
5.3.3 Evaluation and Conclusions	75
5.4 Multiple Criteria	76
5.4.1 Background	76

5.4.2	Solution	77
5.4.3	Conclusion	79
5.5	Summary	80
Chapter 6. Applications		82
6.1	Optimal Parameter Combination	82
6.2	Optimized Sub-Domain	83
6.3	Stock-Rules Pairs	86
6.3.1	Distribution of In-sample Set and Out-of-sample Set	87
6.3.2	Support and Confidence	88
6.4	Relationship between Investment and the Number of Pairs	90
6.4.1	Profit, Return, Top Percentage Pairs and Investment	90
6.4.2	Investment and the Number of Pairs	94
6.4.3	Stock-Rule Pair Return and Market Index Return	97
6.4.4	Maximal Return Point	101
6.4.5	Determination of Investment	103
6.5	Summary	106
Chapter 7. Evaluations		108
7.1	Financial Profitability	108
7.1.1	Economic Profitability	108
7.1.2	Sharpe Ratio	110
7.1.3	Predictability	112
7.2	Computational Performance	113
7.2.1	Execution Time	113
7.2.2	Memory	114
Chapter 8. Conclusions and Future Work		115
8.1	Conclusions	115
8.2	Future Work	117
Bibliography		119

Abstract

Stock trading is a number of stocks to be exchanged from one trader to another trader. It consists of a trader selling a number of stocks at a price and a volume, and another trader buying the same stocks at the same price and the same volume. Most traders want to buy a stock at a low price and to sell the stock at a high price in order to make a profit. However, it is difficult to know whether the current trading (buy/sell) price is low or high. Some researchers have presented technical trading rules which are mathematical formulas with many parameters to solve this problem, such as moving average rules, filter rules, support and resistance, channel break-out rules, and so on. All these rules are based on historical data to generate the best parameters and use the same parameters in future trading to make a profit. When the parameters of a trading rule are set properly, the trading rule can help the traders to make a profit (buy/sell at a low/high price). Experiments have shown that technical trading rules are profitable.

However, there are still some disadvantages and limitations to the technical trading rules in real stock market trading. First, the technical trading rules do not integrate domain knowledge (expert experiences and domain constraints, etc). For example, some trading rules pattern maybe only generate three signals during one year trading to get the most profit. However, the pattern is unreasonable and it is unprofitable in future trading, because the pattern is only a mathematical maximum, but it is impracticable in stock trading. Second, the output of a parameter for a trading rule is only one single value. Sometimes, it may be a noise so the trading rule is inapplicable in future trading. Third, present algorithms to calculate parameters of trading rules are inefficient. Most trades are performed through internet such that they can buy and sell stocks in online and a trade is completed in a second. Real markets are dynamic such that trading rules have to be updated all the time depending on changing situations (new data come

in, new parameters will be recomputed). Current enumerate algorithms waste too much time to get new parameters. However, a one-second short delay in real stock trading will lose the best trading chance. Fourth, when we evaluate the performance of a stock, we need not only to consider its performance (profit and return), but also to compare it to other stocks performance. At present, trading rules do not compare to the other stocks performance when they are selected to generate a signal, so the selected stocks or rules may be not the best ones. Fifth, in stock markets, there are many stocks and many trading rules. The problem is how to match and rank stocks and rules to combine a profitable and applicable pair. However, trading rules do not solve this problem. Lastly, trading rule techniques do not consider the sizes of investments. However, in real market trading, different investments will result in a different performance of a pair.

We propose in-depth data mining methodologies based on technical trading rules to overcome these disadvantages and limitations mentioned above. In this thesis, we present the solutions to combat the existing six problems.

To address the first problem, we designed a domain knowledge database to store domain knowledge (expert experience and domain constraints). During the computing procedure, we integrated domain knowledge and constraints. We observed the output more reasonable as we considered domain knowledge.

To address the second problem, we optimized a sub-domain output instead of a single value, in the sub-domain all combinations of parameter can get a near-best result. Moreover, in the sub-domain, some experienced traders can also set or micro-tune parameters by themselves and a better performance is guaranteed.

To address the third problem, we adopted genetic algorithms and robust genetic algorithms to improve the efficiency. Genetic algorithms and robust genetic algorithms can get a near-optimal result in an endurable execution time, and the result is also near to the best one.

To address the fourth problem, we applied fuzzy sets and multiple fitness functions to evaluate stocks. Because many factors influence the performance of a stock, it is necessary to create a multiple fitness function for genetic algorithms and robust genetic algorithms.

To address the fifth problem, we built a stock-rule performance table to rank stock-rule pairs and find the best matching pairs. The stock-rule pair results showed that the ranked performance is better than that of randomly matched pairs.

Finally, to address the sixth problem, we drew a graph of the relationship between investments and number of stock-rule pairs to search maximal points, and to decide the number of pairs for different sizes of investments.

In summary, the purpose of this thesis is to identify optimal methodologies in stock market trading, to make more profit with less risk for investors. The experimental results showed that the methodologies are more profitable and predictable.

Chapter 1 Introduction

Stock trading is becoming more and more popular around the world, and the money flow in stock markets is also growing quickly [ASX]. This leads to stock trading methods and strategies becoming the new focus of many people, such as traders, researchers, investors, brokers and dealers. However, many new problems also appeared with the development of stock trading, such as the execution of finding a parameter of a trading rule is inefficiency; a trading rule does not consider different investments. Nevertheless, as most traders want to make profit or a higher return through stock trading, the problems that the traders face to are (1) how to select stocks and trading rules from the global stock markets, (2) how to select the best trading price and volume, (3) how to grasp the best trading time-points and (4) how to generate effective trading alert signals. These problems become the highlights of stock market data mining research because a trading rule is fundamental methods to make profit.

We introduce the problems and solutions in the following sections, respectively.

1.1 Problems

In stock markets or any other similar financial markets, one target is making profit through trading which consists of some traders sell a stock and others buy the same stock. Since profit is total earnings deducting expenses, the key issue to attain this target is to select the best stock and to generate profitable trading signals so that traders can get more earning with less expense. Sometimes, some traders only buy or sell by their feelings and they can also make a profit. For this situation, it is not suitable for a computer-based system and it is just like a “gambling”. We do not discuss it in this thesis.

Many experiments and a long time practice to prove that trading rules are predictable tools to calculate profitable trading signals. “Predictable” means it can recommend trading signals for future trading. Currently, many trading platforms still incorporate this method to help their traders to generate trading

signals [ASX] [FOREX]. So, our methodology is based on trading rules and builds the further application platforms. However, trading rules still have some disadvantages and limitations need to be overcome. So we focus on the following disadvantages and limitations one by one:

First, in stock trading systems, some experts have many useful knowledge and experiences. Moreover, there are also many domain knowledge and business constraints, so how to integrate domain knowledge (expert experiences, knowledge, business constraints and domain constraints) into a trading system becomes an important issue for improving efficiency and effectiveness. For example, one important function of trading rules is predictability, which is using historical data (order book data) to predict the parameters of trading rules for future trading. So we must divide the historical data into two parts: in-sample data set and out-of-sample data set. In the first data set, we train trading rules and get the optimal parameter combination, and keep the parameters in the out-of-sample set to verify and evaluate its performance. Trading rules do not consider the size of the in-sample and out-of-sample set. Trading rules can always give an “optimized” parameter combination. However, the problem is the parameter may be without predictability if the two set sizes are not suitable. For example, if in-sample data size is only one day and out-of-sample data is one day too. The result is impossible to be a real pattern since one day is too short to find a pattern and it may be disturbed by noisy signals. In contrast, if the sizes are set more than one year, such as, in-sample data is ten years, and out-of-sample data is two years, it does not make sense either. The reason is a pattern cannot keep for such long time. So, the problem is how to decide a suitable data size for the two data sets in order that the predicted result is applicable and profitable. The proper sizes of the two sets are fundamental conditions of trading rules predictability. For these kinds of expert knowledge and other domain constraints, we can use a domain knowledge database to store them and automatically updated.

Second, most current trading rules only output the best “one” single value for each parameter to get the best performance [F-TRADE]. Sometimes, the combination is not a real pattern (combination of parameter) instead of a noise. A very little change of the parameters may lead to a significant decrease of the performance. The reason is that the output is not a real “optimal” value and it may be a noise. So our target is to find the “real” best parameters in-sample set such that it can keep the best performance out-of-sample set as well. Our solution is to find an optimal sub-domain for every parameter. A sub-domain is a small range for a parameter, instead of a single value. For example, 18-20 days for the long run of a moving average rule. In some case, the sub-domain can be a single value, so a single value is also included the definition of sub-domain. For example, the long run of a moving average rule is 18-18 days. The reason is sometimes the sub-domain can avoid most noise and output the real best results.

Third, in online stock market, real time processing is very important, otherwise the best trading chance will be missed and the generated alert signals cannot be successfully traded due to the dynamic of stock markets. So, it is very important to get the best signals in an endurable time - generally less than one second. Moreover, the market is dynamic so trading rules need to be updated every time when new data comes in. However, if we use current enumerating algorithms, it will take a long execution time to get a combination of parameters. In real stock markets, the best trading chance will be missed even if there is a one-second delay, so alert signals are useless. Because the markets situation is dynamic, we need to compute new parameters when new data come in. For example, moving window strategies change window in every minute. We present genetic algorithms to improve efficiency.

Fourth, since there are hundreds of thousand of stocks all over the world, how to find the best one to make the highest return is not easy. This is out of current technical trading rules methodology, but many current trading systems try to overcome. If we know one stock can make one per cent monthly return. Is that a

good stock? We are not sure because we do not know other stocks monthly return. However, we can select one stock if we know the stock is “very good” no matter how much its monthly return is. So a literal output is better than a numerical output. It can help us to make a decision rapidly and exactly. Moreover, how to evaluate a stock is better or worse is also another problem. Sometimes, we need not only a price, but also the comprehension of volume, liquidity and investments. All of these (price, volume, liquidity and investments) are not considered by technical trading rules. For example, one stock is better when investment is one thousand dollars, but, it may be no good when investment is one million dollars, because its volume is not much enough for one million dollars.

Fifth, trading rules can only generate the best parameter combination for a selected stock. There are more than hundreds of thousands of stocks around the world (many internet-based trading systems can select all stocks around the world to trade). Which stock can help traders to get the highest return? Some currently trading system can help investors to select some best stocks, but, the same stocks combined with different trading rules, their results are different and may be on the contrast. So, only stock recommendation is not enough, we need recommend the stock and trading rule pair to investors and even parameter combinations. However, there are no solutions for stock-rule pairs until now.

Last, most traders want to make a higher return, but the problem is trading rules methodology does not consider investments. For instance, one stock is better when investment is one thousand dollars, but, it may be not good when investment is one million dollars. In fact, different traders invest different amount of money in stock market. Investments may be from one-thousand dollars to millions of billions of dollars, so, it is very important to find the relationship between investments and the number of stocks so as to make a higher return. If we select too few stocks to trade, there will be much money in hand without making profit. If we select too many stocks to trade, some money

may be used to buy/sell “bad” signals such that better signals are missed, which leads to profit or return is not the best.

1.2 Solutions

About the problems we mentioned in Section 1.1, we propose to present some ideas to solve the existing problems.

To address the first problem, we design a domain knowledge database and a human-machine interface to integrate domain knowledge (expert experience and domain constraints) in our system. For domain knowledge, we divided it into two kinds: the first kind of knowledge comes from human (expert or traders). For this kind of knowledge, traders can use the human-machine interface to input their knowledge to the system, even the knowledge is a vague one. For example, the short run of moving average may be between 10 to 20 days. Another kind of knowledge is computed by computer, such as system feedback or output. For example, the system concludes that for most stocks, while the in-sample set and out-of-sample set are about one month, trading rules can get a high return. We can set “one month” for both in-sample and out-of-sample set as a default size. The first kind of knowledge comes from human, and second kind of knowledge comes from computer. Both are stored in the domain knowledge database. [Chapter 3]

To address the second problem, we optimize a sub-domain instead of a single value. If we use an algorithm to get the best single value, it has a larger possibility to be a noise than we get a sub-domain, because noise is difficult to be found or filtered when we only keep one single value. The noise cannot give correct predicting signals out-of-sample. It is necessary to get the real best signals matching a real pattern in order to make a profit. Our solution to filter a noise is looking for the best sub-domain rather than a single value. So, the target of our algorithms is looking for a sub-domain in which the performances of all parameter combinations are better than that in other domains. The advantages of sub-domain methodology include: it is more robust to filter noise in case there is;

and traders can micro-tune a parameter in a sub-domain to make their own decisions. [Chapter 4]

To address the third problem, we implement genetic algorithms and robust genetic algorithms to improve efficiency. In real market trading system, in-sample data set and out-of-sample data set change along with the time elapses. For example, at 11:00, the in-sample data window is 9:00-10:00, the out-of-sample data window is 10:00-11:00. At 11:01, the in-sample data window is 9:01-10:01, and the out-of-sample data window is 10:01-11:01. However, current enumerate algorithms cannot get updated parameters in a reasonable short time. For example, using enumerate algorithms to optimize moving average, the number of parameter combination is more than 100000000, and for every combination the system needs to compute the best profit, return and/or Sharpe ratio, so it may be cost 60 minutes to get the best result when it considers one year intraday order book data. In contrast, if we use standard genetic algorithms (SGA), and set population number about 4000, 2-3 generations, we only need to computer 10000 times and our result is almost near-optimized (more than 90 per cent of the best one). The execution time of SGA is about 0.01 per cent of enumerate algorithms [Section 5.1]. It is about 0.3 second. However, sometimes, standard genetic algorithms cannot get a reasonable result, but, it is only a mathematical best value. For example, if we get 3 buy/sell signals in one year, we can get the best Sharpe ratio 28.251 [Figures 4.2 and 4.3]. The model is not a reasonable one, so it has not profitability out-of-sample. So, we need integrate domain knowledge or constraints into genetic algorithms to filter unreasonable results. The new algorithms with domain knowledge and constraint are called robust genetic algorithms (RGA) [Section 5.2]. Most experimental results show that RGA can give a more reasonable result in both in-sample and out-of-sample set.

To address the fourth problem to evaluate stocks, we present fuzzy set to transfer a numeric result into a literal result, so it is easy for traders to make their decision

[Section 5.3]. For example, if one stock can make “one per cent monthly return”. From numeric value it looks good, but, if other stocks can make more than two per cent monthly return, we know that “one per cent monthly return” is still not as good as we imagine. If one stock evaluation result is “Medium” instead of “one per cent monthly returns”, traders can make their decision very easily. Of course, the precondition is the output result is exact and believable for both numeric one and literal one. Another sub-problem to evaluate a stock is it needs multiple criteria because only one factor is not enough to evaluate a stock. For example, price, volume and investments need to be considered together. One stock may be good if investment is one thousand dollars, but, it may be not good when investment is one million dollars. In contrast, other stocks may be different. It may be no-good when investment is one thousand dollars, but, it may be good when investment is one million dollars. To rank all stocks, we need consider multiple criteria [Section 5.4].

To address the fifth problem to select the best stock-rule pairs, we build a stock-rule performance table. From the table, we can rank stock-rule pairs and classify them by “very good”, “good”, “medium”, “bad”, and so on. Moreover, when we build the performance table, we consider price, volume and investments, and draw the graphs for different investments [Section 6.4]. The performance includes profit, return and Sharpe ratio.

Finally, to address the sixth problem, we draw the graph of the relationship between investments and the number of stock-rule pairs. Our approach is based on the stock-rule performance table. We generate all trading signals. For different investments, we select different top percentage pairs and their signals. As a real market trading system, we sort all signals by time, and consider the money in hand to buy when we have enough available money and volume, and sell stocks when there is a sell signal at a possible volume (volume in hand or volume can be traded according to the order book). For all different combinations (investments and top percentage pairs), we can get return graphs

for different investments, for different top percentage pairs, and get maximal points for the number of pairs and investments [Section 6.5].

1.3 Related works

In 12th century France the *courratiers de change* were concerned with managing and regulating the debts of agricultural communities on behalf of the banks. As these men also traded in debts, they could be called the first brokers. Some stories suggest that the origins of the term "bourse" come from the Latin *bursa* meaning *a bag*. Bruges, the sign of a purse (or perhaps three purses), hung on the front of the house where merchants met. However, it is more likely that in the late 13th century commodity traders in Bruges gathered inside the house of a man called Van der Bourse, and in 1309 they institutionalized this until now informal meeting and became the "Bruges Bourse". The idea spread quickly around Flanders and neighboring counties and "Bourses" soon opened in Ghent and Amsterdam. In the middle of the 13th century Venetian bankers began to trade in government securities. In 1351 the Venetian Government outlawed spreading rumors intended to lower the price of government funds. There were people in Pisa, Verona, Genoa and Florence who also began trading in government securities during the 14th century. This was only possible because these were independent city states not ruled by a duke but a council of influential citizens. The Dutch later started joint stock companies, which let shareholders invest in business ventures and get a share of their profits - or losses. In 1602, the Dutch East India Company issued the first shares on the Amsterdam Stock Exchange. It was the first company to issue stocks and bonds. The New York Stock Exchange, which is in operation since March 1792, is the second largest stock exchanges in the world. It sports the tag line "The world puts its stocks in us" [GOOGLE] [NYSE]. More and more people were interesting in stock traded, but they found it difficult to analyze the daily jumble of up-a-quarter and down-an-eighth or whether stocks generally were rising, falling or staying even. So, Charles Dow, Edward Jones and Charles Bergstresser devised their stock average

(Dow-Jones indexes) to make sense of this confusion. They began in 1884 with eleven stocks, nine railroads and two industrials. It was the precursor to the Dow Jones Industrial Average, which is launched in 1896 [Dowjones] [Robert 1932] [Stephen et al 1998] [William et al 1998].

Dow Jones average indexes are widely spread around the world and many traders want to make a profitable trade. The best way is buying at a low price and selling at a high price. Since the computer aided systems are widely used in the stock market trading and the order book data are available, the new technology based on the computer is introduced, that is technical trading rules, and many researchers have founded a number of technical trading rules, such as moving average, filter rules, channel break-out. [Acar et al 1997] [Allen et al 1999] [Oslen 2004]

A number of researchers and research projects have already taken advantage of trading rules [Aarts et al 2005] [Brooks et al 2005] and optimization techniques [Mihael 2002] in the field of data mining. However, most of the projects concentrated only on trading rules. We focus on mining in-depth pattern (using trading rule as a tool to mine patterns instead of finding a new trading rule) based on trading rules.

Sullivan [Sullivan et al 1999] summarized trading rules and their performance against profit. They also made the experiments that trading rules can help traders to get profitable signals. Allen [Allen et al 1999] performed the GA to build new trading rules, and Neely [Neely et al 1996] proved some evidence that trading rules had predictive ability.

Olsen [Olsen 2004] tested whether moving average trading rule profits declined over the period from 1971 to 2000. Some previous studies have reported mixed results regarding the success of technical trading rules in currency markets. However, his optimized rules are for successive 5-year in-sample periods from 1971 to 1995 and tested over subsequent 5-year out-of-sample periods. Results show that risk-adjusted trading rule profits have declined over time-from an

average of over 3% in the late 1970s and early 1980s to about zero in the 1990s. Thus, market inefficiencies reported in previous studies may have been only temporary inefficiencies. Leigh [Leigh et al 2002] implemented a recognizer for two variations of the “bull flag” technical charting heuristic and used that recognizer to discover trading rule on the New York Security Exchange [NYSE]. Composite index out-of-sample results indicate that these rules are effective. Their work is a little different from ours since our work is based on trading rules instead of seeking trading rules. Prinzie [Prinzie et al 2005] has done the research about the data mining models with practical constraints or thresholds. It can improve model performance as the model is optimized for the given implementation environment, if the implementation constraints/thresholds are known in advance. Prinzie illustrated the relevance of this constrained optimization of data mining models on a direct-marketing case only. Lam [Lam et al 2000] presented a method to find trading signals that is similar to seeking a new trading rule.

F-TRADE [F-TRADE] is a technical stock-trading platform based on multi-agents. It is developed by University of Technology, Sydney (UTS) Data mining group under the support of CMCRC [CMCRC] and UTS. F-TRADE focuses on trading rules and more enhanced versions than other systems. The advantages of F-TRADE include that it can integrate new trading rules developed by users when the users want to do some tests or experiments. In addition, the platform can also integrate a user’s new data into the system to find trading signals by trading rules offered by the system. However, it also has the following limitations: First, users (traders or researchers) can set parameters, but, the parameters cannot be generated automatically. For example, F-TRADE does not tell users the best sub-domain for short run of moving average unless users already know that. Second, the output of the system is the best one value. In many cases, it is a really good result. However, sometimes, it may be a noise. Third, the system computes all possible parameter combination by enumerate algorithm, so efficiency need be improved. Fourth, the system only compute the

one stock-rule pair offered by users, so it cannot give a stock-rule pair list or some new recommendations. Fifth, the user must select stocks and rules by themselves. The other trading systems recommend the stocks to the users, but, one good stock combined with a “wrong” rule, the result may be not good any more. Sixth, the system does not consider investments since it is based on technical trading rules. All our algorithms are developed for the F-TRADE necessity and will be integrated into F-TRADE.

Many previous works have been done about trading rules method. The advantage of these works is that trading rules can give a profitable result. However, there are some limitations in the previous works. (1) The current works did not consider the integration of domain knowledge (expert experience and domain constraints) which is necessary in real market trading. (2) The best trading rule is only one single group of combination of parameters, such that it is easy to be influenced by a noise. (3) The current works did not consider computing efficiency and effectiveness. However, in real market trading, trading rules need be updated all the time, when new data come in. (4) Some stocks have a good performance for making profit, but, when situations changed (such as investments changed), the profit is not good any more, because the volume is not suitable for new investments. The reason is that price is not the unique criterion, other features (volume or liquidity) are also very important to evaluate a stock. (5) For only one rule and one stock, trading rules method can help traders to find the profitable trading signals, but, the current research did not pay attention to which stock and rule “pair” is the best for making profit. (6) The current research did not consider the relationship between the various investments and the number of trading stocks.

In the above section, we have introduced the problems of stock markets. We introduce the significance in the following section.

1.4 Significance

In stock markets, the target of most investors is to make higher profit at less risk. One ideal solution is “choosing the best stocks and the best trading rules to generate the best trading signals”. So, the most important problems are the selection of stocks and trading rules and the generation of trading alert signals (signals in short), efficiency and effectiveness and considering different investments. These problems become the focus of data-mining task in stock market. For example, to speed up an algorithms efficiency; given a rule, what are the best targeted stocks; or, given a stocks, what are the best rules; given stocks and rules, to find the best training (in-sample) and testing (out-of-sample) windows sizes; given timeframe, stocks and rules, to look for the best stock-rule pairs for making more profit; given a stock-rule pairs, to decide the best investment strategies, say investment amount; or given investment amount, to decide the best stock-rule pairs, say the number of stock-rule pairs; to combine mathematical algorithms with financial domain knowledge to remove noisy signals; to represent and to integrate domain knowledge into the system; and so on. If any one of the above is wrong, the target cannot be archived.

Our research is in-depth data mining strategy based on trading rules methods. The advantages of this strategy is predictable and profitable [Robert 1999]. Moreover, it overcomes the limitations of existing trading rules, such as, trading rules discarded investments and inefficiency.

The significances of our research are listed in the following paragraphs.

Firstly, we considered domain knowledge, business constraints and expert experience, such that the results with domain knowledge are more reasonable and applicable.

Secondly, we implemented robust genetic algorithms, which improved efficiency, such that the parameters of a trading rule can be archived in an endurable time. In real time trading, only one-second delay will miss a trading chance.

Thirdly, we evaluated stocks by multiple criteria, because many features and factors have different impacts on profit or return. Such as, trading rules only considered the price changing, but, trading rules ignored volume. In practice, when investment is a big amount, the volume is also very important. It maybe leads to a different return.

Fourthly, different stocks matched with different trading rules. The different pairs have different performance. To find the best pairs for making profit is beyond trading rules applications. However, it is very important in stock trading, so we sorted them through building a stock-rule performance table. Our experimental result shows that the sorted pairs can help to make a profit.

Lastly, we considered investments when we evaluate stock-rule pairs. Currently, all trading rules do not consider investments. However, different investments influence returns. Such as, when investment is one thousand dollars, a monthly return may be two per cent, but, when investment is one million dollars, the monthly return may be not two per cent even if all other conditions are the same.

Figure 1.1 shows the relationship of our work among stock market trading workflows. It makes stock trading become more intelligent and profitable.

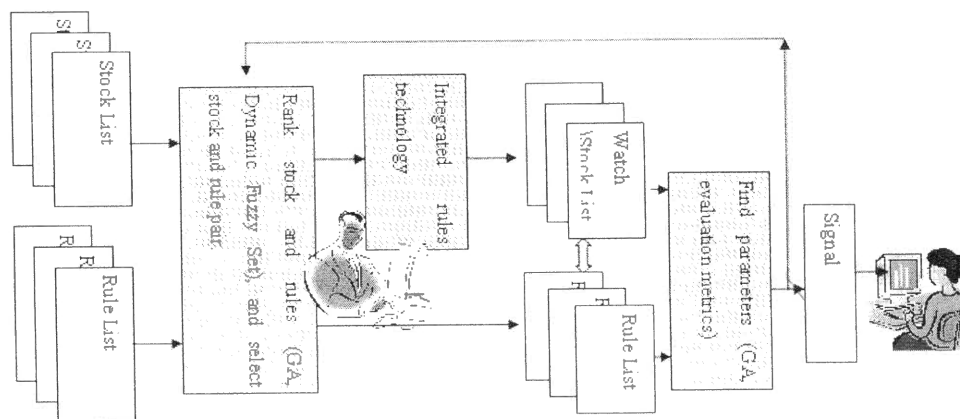


Figure 1.1 The structure of system trading process in stock markets and the functions of the system in this dissertation (shadow parts).

1.5 Structure of the Thesis

Organization of this dissertation is as follows. In chapter 2, we introduce background and related work on technical trading rules.

Chapters 3 to 6 form the core of this dissertation. Chapter 3 demonstrates domain driven concepts and related applications. Chapter 4 describes in-depth data mining method and an instance. Chapter 5 presents optimal algorithms including genetic algorithms, robust genetic algorithms, fuzzy set literal output and multiple criteria evaluation metrics. Chapter 6 collects some applications about these problems and solutions which are genetic algorithms, robust genetic algorithms, in-depth sub-domain, ranking stock-rule pairs and the relationship between investments and the number of stock-rule pairs. Chapter 7 evaluates the results and Chapter 8 summarizes the conclusions and previews future work.

In this dissertation, all experiments were implemented in the same computer conditions: Laptop DELL Latitude D600, CPU IBM (M) 1.3G Hz, 512M RAM, 30G Hard disk, Windows 2000 English version, Borland C++ Builder 6.0. [Lin et al 2004a] [LINLIONLINE].

Chapter 2 Context

To conduct our research, firstly, we need to specify an appropriate universe of trading rules from which the current trading systems may have been applied. In stock markets, when brokers or dealers want to buy or sell a share, some of them will depend on a technical trading rule. Robert Edwards and John Magee [Edwards et al 1992] defined technical trading rules as “the science of recording the actual history of trading (price changes, volume of transactions, etc.) in a certain stock or in “the averages” and then deducing from that pictured history the probable future trend.”

In the first step, we should introduce some notations for the thesis being completed.

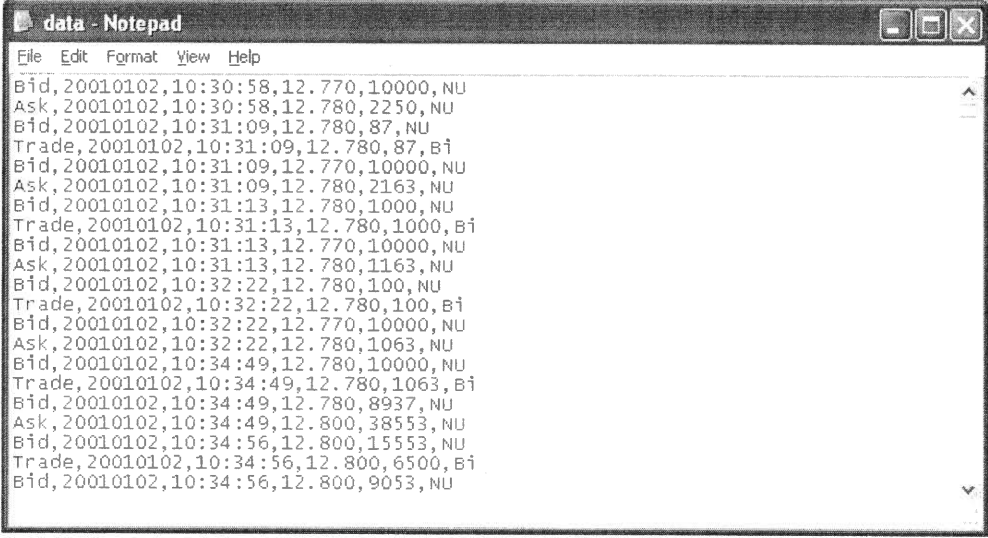
2.1 Data

Definition 2.1 Data. Values (numerical or literal) collected through record keeping or by polling, observing, or measuring, typically organized for analysis or decision making. More simply, data is facts, transactions and figures.

In this dissertation, the data is stock market exchange intraday order book information. The order record includes: order type (Bid/Ask/Trade), date (YYYYMMDD), time (HH:MM:SS), price, volume, and two character flag. See Figure 2.1.

In this dissertation, the stocks (shares) are randomly selected from Australian Stock Exchange (ASX). For commercial confidential reason, we only give the meaningless codes for these selected stocks: A01, A02, A03, A04, A05, B06,B07, B08, C09, C10, C11, F12, F13, G14, I15, J16, M17, M18, M19, M20, O21, P22, Q23, Q24, S25, S26, S27, T28, T29, W30, W31, W32 and W33, totally 33 shares numbered from 1 to 33 respectively. Time period is five-year intraday order book on-market data, from 199810101 (1 January 1998) to 20021231 (31 December

2002). The reason for not selecting other stocks is they may not have the everyday intraday data during the whole five years or randomly missed [SIRCA]. In different experiments, the time period may be different.



```
data - Notepad
File Edit Format View Help
Bid,20010102,10:30:58,12.770,10000,NU
Ask,20010102,10:30:58,12.780,2250,NU
Bid,20010102,10:31:09,12.780,87,NU
Trade,20010102,10:31:09,12.780,87,Bi
Bid,20010102,10:31:09,12.770,10000,NU
Ask,20010102,10:31:09,12.780,2163,NU
Bid,20010102,10:31:13,12.780,1000,NU
Trade,20010102,10:31:13,12.780,1000,Bi
Bid,20010102,10:31:13,12.770,10000,NU
Ask,20010102,10:31:13,12.780,1163,NU
Bid,20010102,10:32:22,12.780,100,NU
Trade,20010102,10:32:22,12.780,100,Bi
Bid,20010102,10:32:22,12.770,10000,NU
Ask,20010102,10:32:22,12.780,1063,NU
Bid,20010102,10:34:49,12.780,10000,NU
Trade,20010102,10:34:49,12.780,1063,Bi
Bid,20010102,10:34:49,12.780,8937,NU
Ask,20010102,10:34:49,12.800,38553,NU
Bid,20010102,10:34:56,12.800,15553,NU
Trade,20010102,10:34:56,12.800,6500,Bi
Bid,20010102,10:34:56,12.800,9053,NU
```

Figure 2.1 The example of intra-day order book data format (stock code “A01”).

Definition 2.2 Data Mining is more or less a collection of different techniques and tools for various types of data. [Jaeger et al, 1996]. These techniques and tools can be used as information extraction activities whose goal is to discover hidden facts contained in databases. Using a combination of machine learning, statistical analysis, modeling techniques and database technology, data mining finds patterns and subtle relationships in data and infers rules that allow the prediction of future results. Typical applications include market segmentation, customer profiling, fraud detection, evaluation of retail promotions, and credit risk analysis.

Definition 2.3 Data Mining Method. Procedures and algorithms designed to analyze the data in databases. It is using computer technology to find some rules or suggestions from a great deal of data, and these rules and suggestions exist in some special data domains.

Definition 2.4 Training Data. A data set used to estimate or to train a model.

Definition 2.5 Testing Data. A data set independent of the training data set, used to fine-tune the estimates of the model parameters (i.e., weights).

Definition 2.6 Bootstrapping. Training data sets are created by re-sampling with replacement from the original training set, so data records may occur more than once. In other words, this method treats a sample as if it were the entire population. Usually, final estimates are obtained by taking the average of the estimates from each of the bootstrap test sets.

Definition 2.7 Order Book. Order book is a file which records all the data of the stock market since the first date of trading [ASX]. It includes all orders (ask orders, bid orders, modified orders, deleted orders). For each order, it includes order id, share code, broker id, date, time, price, volume, bid/ask flag, house and correlative flags, etc.

Definition 2.8 Market Return. The return of the market portfolio.

Definition 2.9 Market Index (Index). A charted index intended to gauge price changes in the overall market.

In this dissertation, we specially refer to the order book of a stock market, it keeps: share name, trading date and time, trading price, volume, value, buyer house, seller house, the best bid/ask price at the trading time, etc, during the past time period. All this data is offered by AC3 [AC3], CMCRC [CMCRC] and SIRCA [SIRCA].

2.2 Technical Trading Rules

Definition 2.10 Technical Trading Rules. [Achelis 1995] [Robert 1999] Technical trading rules are used by financial market traders to assist them in determining their investment or speculative decisions. These rules can be based on either technical or fundamental analysis. This thesis considers only rules based on technical indicator. A technical indicator is a mathematical formula that transforms historical data on price and/or volume into a single number. These

indicators can be combined with price, volume or each other to form trading rules.

Definition 2.11 Technical Analysis. Technical Analysis is the study of price changes, rates of change, averages, volume, and open interest of futures markets and trends. [Investionary]

So, sometimes, we also say the technical trading rules are based on technical analysis.

2.2.1 Simple Moving Average (SMA)

Moving averages (MA, see Figure 2.2) are used to identify trends in prices. A Moving average is simply an average of current and past prices over a specified period of time. An MA of length θ is calculated as

$$MA_t(\theta) = \frac{1}{\theta} \sum_{i=0}^{\theta-1} P_{t-i} \quad (2.1)$$

Where

$$\forall \theta \in \{1,2,3,\dots\}.$$

By smoothing the short-term fluctuation or noise in the price series, the MA is able to capture the underlying trend in the price series over a time period. An MA can be used to formulate a simple trend-following rule also referred to as a momentum strategy.

A simple MA rule can be constructed by comparing the price to the near past average price. We can use a binary function to formulize it as:

$$S(\Theta)_t = P_t - MA_t(\theta) \begin{cases} > 0, & 1 \\ \leq 0, & 0 \end{cases} \quad (2.2)$$

The function $S(\Theta)$, will return a zero or one corresponding to a buy or sell signal respectively.

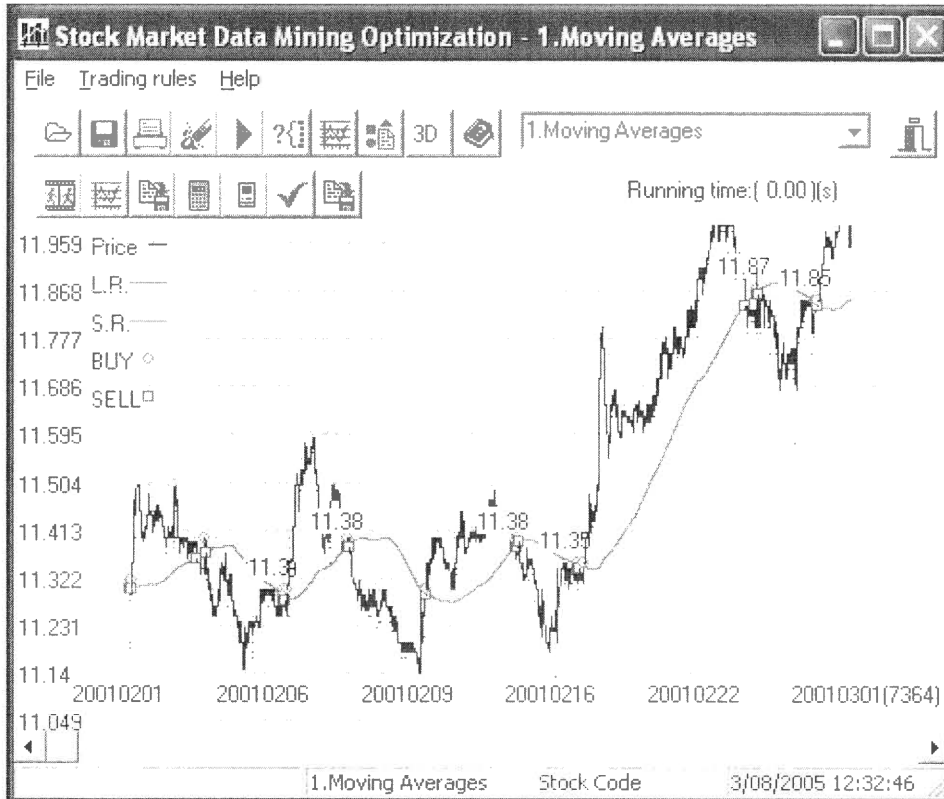


Figure 2.2 Simple moving average rules.

In Figure 2.2, an alert signal is generated, when the price is crossover the average price.

2.2.2 Filtered Moving Average (FMA)

For removing some noisy signals, sometimes, we need to check whether the signal is a noise or not. So, we can remove the noise through adding a filter.

Filtered Moving Average (FMA, see Figure 2.3) is adding one parameter more Fix Band (d), the equation of FMV is:

$$S(\Theta)_t = P_t - (1 + d_{t-1})MA_t(\theta) \begin{cases} > 0, & 1 \\ \leq 0, & 0 \end{cases} \quad (2.3)$$

That means, when the price is higher/lower d per cent than the average price during the past θ day, it generates a Buy/Sell alert signal.

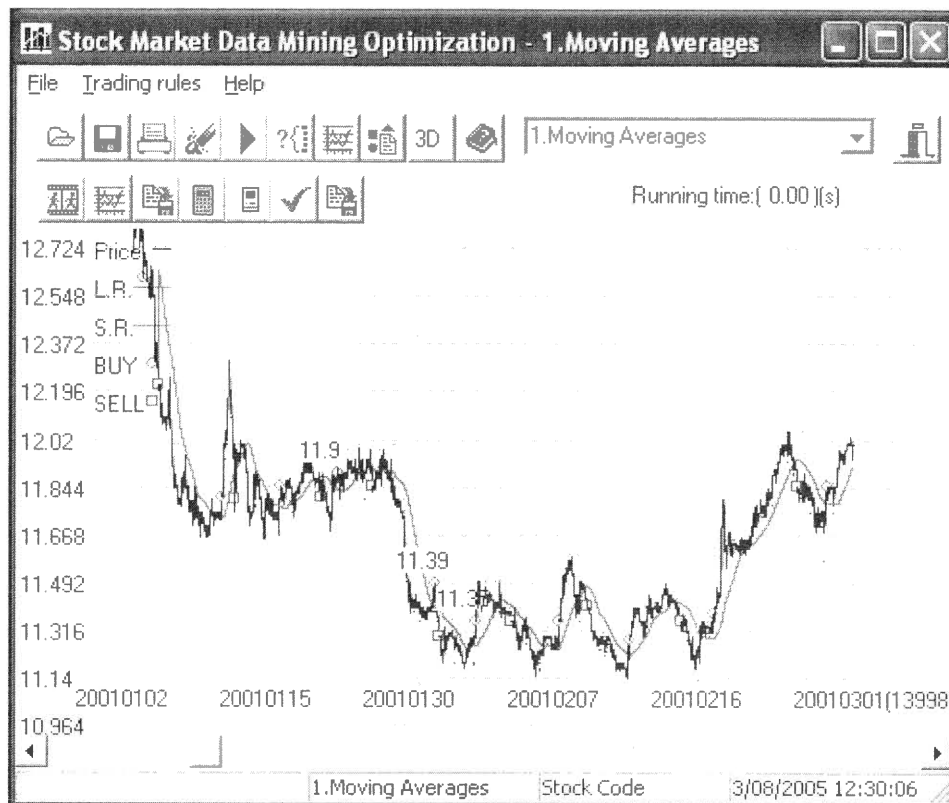


Figure 2.3 Filtered moving average rules.

In Figure 2.3, an alert signal is generated when the price is higher/lower a certain percentage of the average price. Otherwise, the crossover is ignored.

2.2.3 Enhanced Moving Average (EMA)

Enhanced Moving Average [Robert 1999] cross-over rules are one of the most popular and common trading rules discussed in the technical analysis literature. The standard moving average rule, which utilizes the price line and the moving average of price, generates signals. In an uptrend, long commitments are retained as long as the price trend remains above the moving average. Thus, when the price trend reaches a top, and turns downward, the downside penetration of the moving average is regarded as a sell signal. Similarly, in a downtrend, short positions are held as long as the price trend remains below the moving average. Thus, when the price trend reaches a bottom, and turns upward, the upside

penetration of the moving average is regarded as a buy signal. There are numerous variations and modifications of this rule.

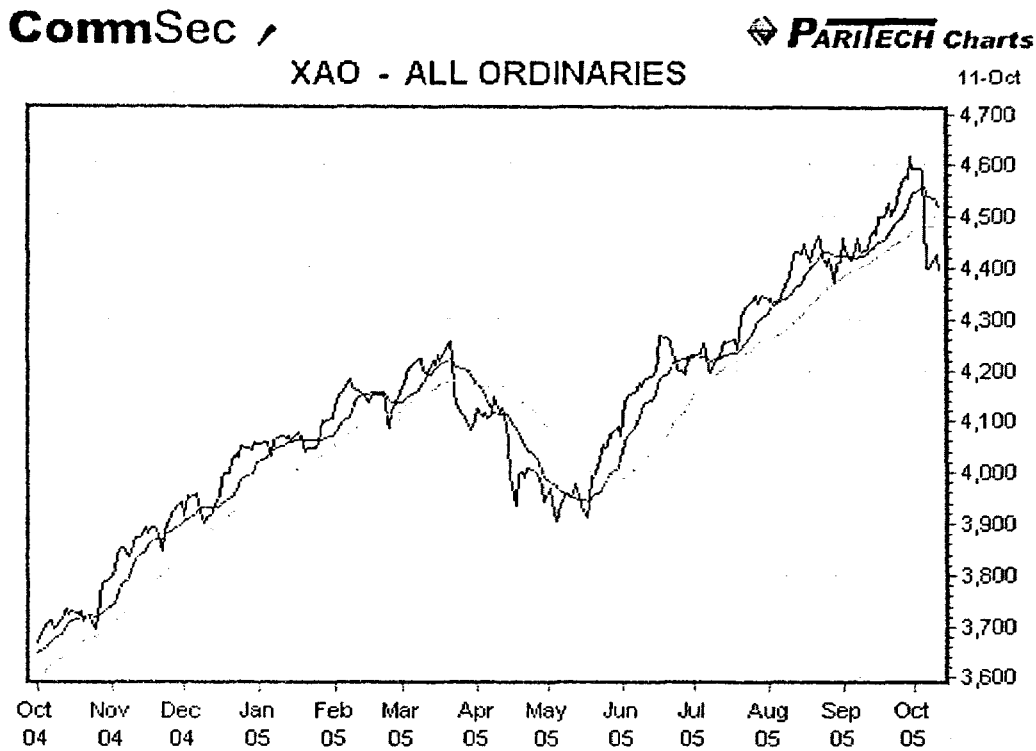


Figure 2.4 Enhanced Moving Average in ASX for all stocks (Oct 2004 to Oct 2005 intra-day data).

We examine several of these. For example, more than one moving average (MA) can be used to generate trading signals. Buy and sell signals can be generated by crossovers of a slow moving average (long term) by a fast moving average (short term), where a slow MA is calculated over a greater number m of days than the fast MA. The moving average for a particular day is calculated as the arithmetic average of prices over the previous n days, including the current day. Thus, a fast moving average has a smaller value of n than a slow moving average. There are two types of “filters” we impose on the moving average rules. The filters are said to assist in filtering out false trading signals (*i.e.*, those signals that would result in losses). The fixed percentage band filter requires that the buy or sell

signal exceed the moving average by a fixed multiplicative amount b . We record the MA (n, m, b)

When fast moving average is higher than slow moving average, then generating a “BUY” alert; else, when fast moving average is lower than slow moving average, then generating a “SELL” alert (see Figure 2.4 [ASX] and Figure 2.5).



Figure 2.5 Enhanced moving average in ASX stock market data mining (year 1998-2002 inter-day data).

2.2.4 Channel Break-Outs

A channel (sometimes referred to as a trading domain) can be said to occur when the high over the previous n days is within x per cent of the low over the previous n days, not including the current price (See Figure 2.6). Channels have their origin in the “line” of Dow Theory which was set forth by Charles Dow around the beginning of the last century. The rules we develop for testing the channel break-out are to buy when the closing price exceeds the channel, and to sell when

the price moves below the channel. Long and short positions are held for a fixed number of days. Additionally, a fixed percentage band can be applied to the channel as a filter.

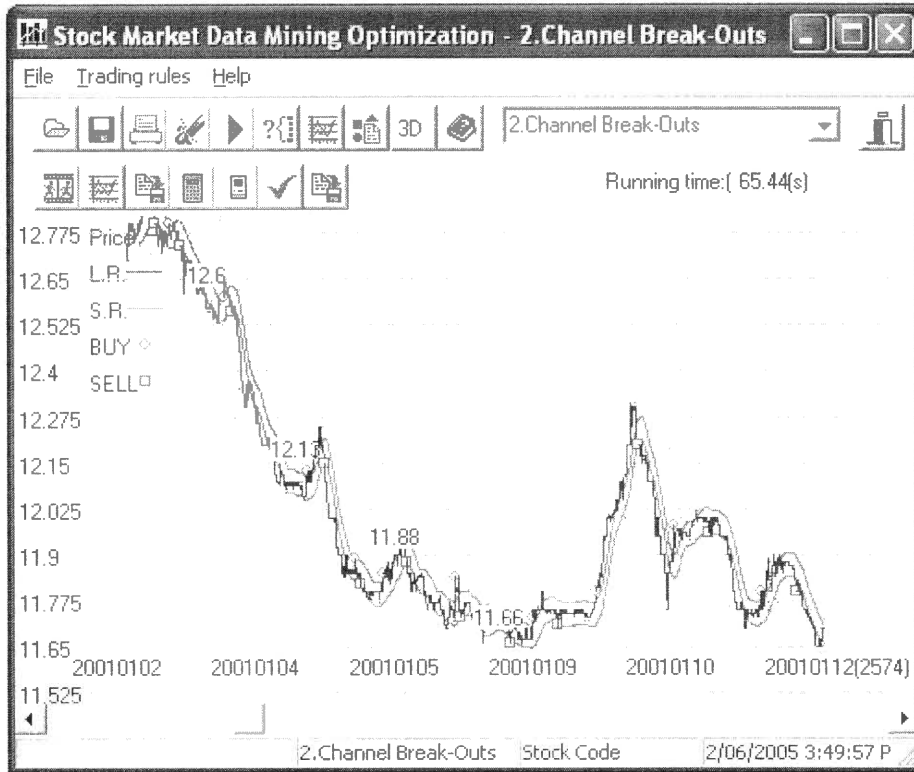


Figure 2.6 Channel break-out rules.

2.2.5 Filter Rules

The standard Filter rules (see Figure 2.7) were defined as:

An x per cent filter is defined as follows: If the daily closing price of a particular security moves up at least x per cent, buy and hold the security until its price moves down at least x per cent from a subsequent high, at which time simultaneously sell and go short. The short position is maintained until the daily closing price rises at least x per cent above a subsequent low at which time one covers and buys. Moves less than x per cent in either direction are ignored. [Ball 1978]

The first item of consideration is how to define subsequent lows and highs. We do this in two ways. As the above excerpt suggests, a subsequent high is the highest closing price achieved while holding a particular long position. Likewise, a subsequent low is the lowest closing price achieved while holding a particular short position. Alternatively, a low (high) can be defined as the most recent closing price that is less (greater) than the e previous closing prices. Next, we expand the universe of filter rules by allowing a neutral position to be imposed. This is accomplished by liquidating a long position when the price decreases y per cent from the previous high, and covering a short position when the price increases y per cent from the previous low. Following BLL [Brock et al, 1992], we also consider holding a given long or short position for a pre-specified number of days effectively ignoring all other signals generated during that time. [Richard 1988]

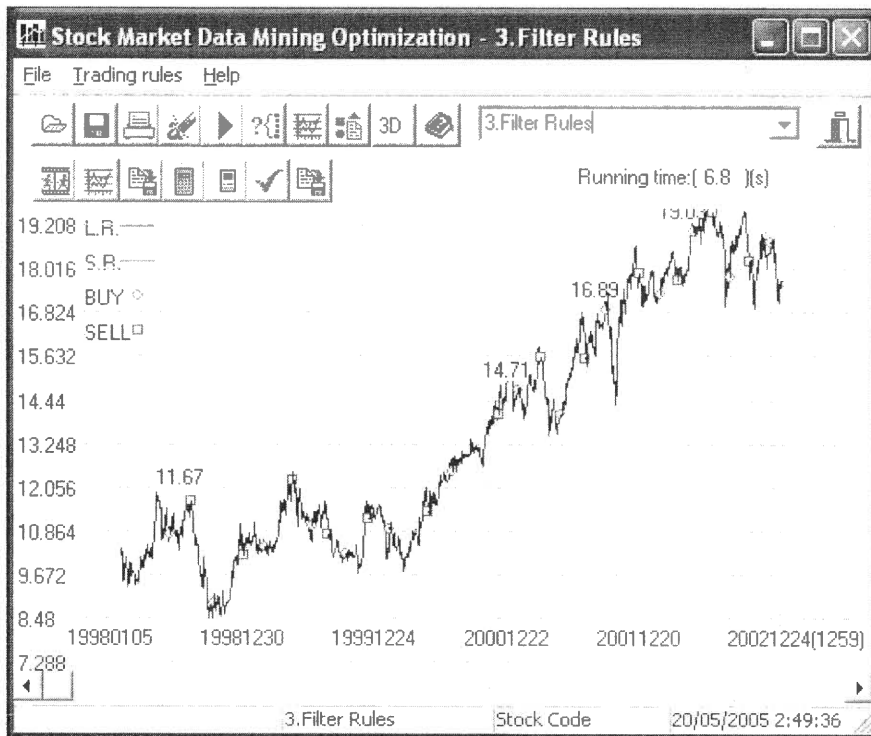


Figure 2.7 Filter rules in ASX.

2.2.6 Support and Resistance

The notion of support and resistance is discussed as early as in Wyckoff, 1910 [Ryan et al 1999] and tested in BLL [William et al 1992] under the title of “trading domain break”.

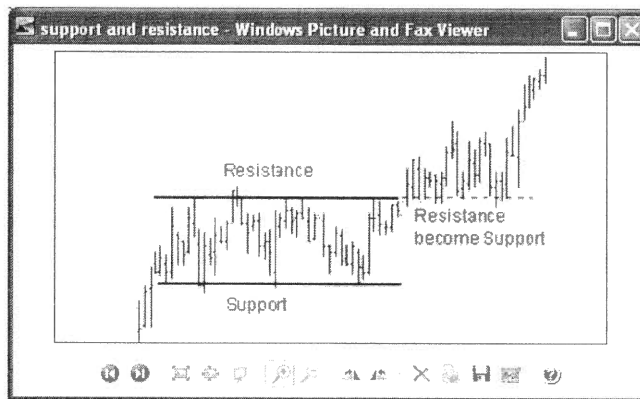


Figure 2.8 Definition of support and resistance.

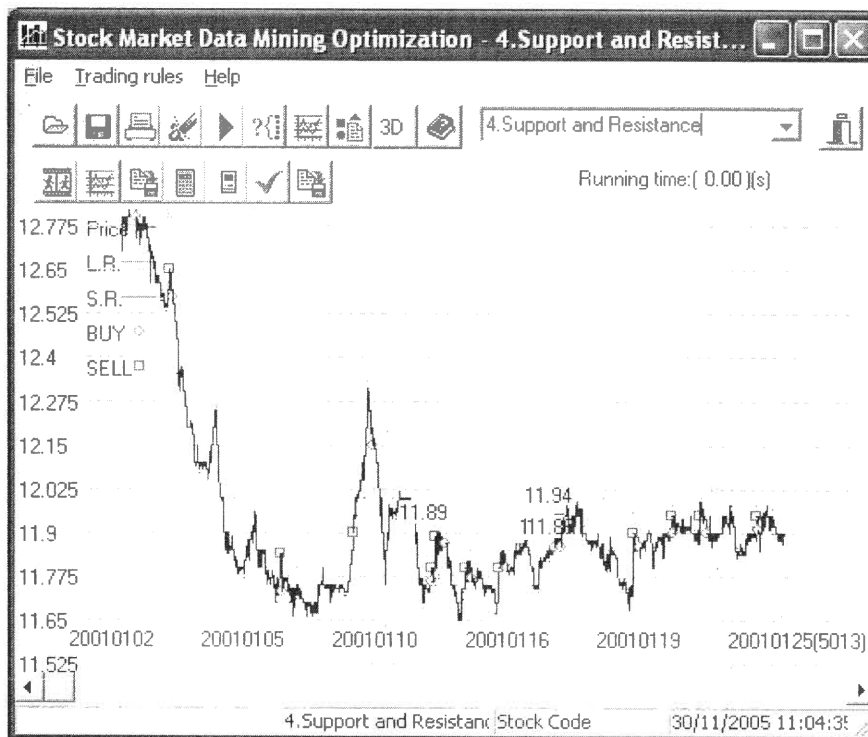


Figure 2.9 Support and resistance rules.

A simple trading rule based on the notion of support and resistance (S&R, see Figures 2.8 and 2.9) is to buy when the closing price exceeds the maximum price over the previous n days, and sell when the closing price is less than the minimum price over the previous n days. Rather than base the rules on the maximum (minimum) over a prespecified domain of days, the S&R trading rules can also be based on an alternate definition of local extrema. That is, define a minimum (maximum) to be the most recent closing price that is less (greater) than the e previous closing prices. As with the moving average rules, a fixed percentage band filter, and time delay filter. Also, positions can be held for a pre-specified number of days.

2.2.7 Other Trading Rules

In this dissertation, we have done all experiments on the above six rules. The data is based on the Australian Stock Exchange (ASX) order book data during 01/01/1998 (1 January 1998) to 31/12/2002 (31 December 2002).

There are some other trading rules, such as abnormal return, on-balance volume averages and benchmark. These rules will be inserted in F-TRADE in the future work.

In this dissertation, we presented the trading rules include: Enhanced Moving Average, Filter rules, Channel-break-out, Simple Moving Average, Filtered Moving Average and Support & Resistance [see Chapter 2] numbered from "Rule 1" to "Rule 6" respectively.

2.3 Evaluation Criteria

Most traders and researchers in stock market want to make a profit and a high return at low risk, so the evaluation criteria should include profit, return, risk and Sharpe ratio. Sometimes, a return includes a profit, so in this dissertation, we only consider one of them because the result is almost the same of the two criteria.

2.3.1 Profit

Definition 2.12 Profit. [Investopedia] The same as net-income: total earnings deducting expenses. In other words, profit is the money a business makes after accounting for all the expenses. Profit is the goal of every company.

However, when we mentioned the profit, we ignored the sizes of investments, so it did not reflect the real reward. Most finance experts use return as the criterion to compare the performance for different algorithms and stocks, so we also used the return as the evaluation metrics through this dissertation except mentioned by other metrics (sometimes, Sharpe ratio is used for considering both the return and the risk).

2.3.2 Return

Sometimes, we should consider the profit to the sizes of investments. So the return is important and more applicable than profit.

Definition 2.13 Return (One-step Return). [Thomas et al 1998]

$$R = \frac{y(t) - y(t-1)}{y(t-1)} \quad (2.4)$$

$y(t)$ is the trading price at time t .

A common variant is the log-return (R).

$$R = \log \frac{|y(t)|}{|y(t-1)|} \quad (2.5)$$

The log-returns are often used in academic research while the former version is most common in the trading community. If the natural logarithm is used, the two measures are very similar for small changes, since $\ln\left(\frac{a}{b}\right) \sim \frac{a}{b} - 1 = \frac{a-b}{b}$.

Since the last century, there have been about one hundred trading rules, such as: moving average, channel break-out, filter rules, support & resistance, on-balance volume averages, etc [Ryan et al 1999][Investopedia].

Definition 2.14 Index Return.

$$IR = \frac{I(t) - I(t-1)}{I(t-1)} \quad (2.6)$$

$I(t)$ is the market index at time t . IR reflects the whole market performance and return. Generally, $t-1$ is the first day of a month or a year, and t is the last day of a month or a year.

2.3.3 Sharpe ratio

In stock market trading, most investors want to get more profit and return but take less risk. Yet, the profit and risk often contradict each other. For example, if a trader wants to get a higher return, he will take more risk. The evaluation criteria also include Sharpe ratio (SR), which considered both return and risk. Sharpe ratio is derived by William F. Sharpe. [Investopedia]

Definition 2.15 Risk. The chance that an investment's actual return will be different than expected. [Pratt 1964]

This includes the possibility of losing some or all of the original investment. It is usually measured by using the historical returns or average returns for a specific investment.

Definition 2.16 Sharpe ratio. It is calculated by subtracting the risk free rate from the rate of return for a portfolio and dividing the result by the standard deviation of the portfolio returns.

$$SR = (R_p - R_f) / \sigma_p \quad (2.7)$$

where R_p is Expected portfolio return, R_f is Risk free rate and σ_p is portfolio standard deviation. The Sharpe ratio tells us whether the returns of a portfolio are because of smart investment decision or a result of excessive risk. When SR is higher, it means one can get more return with less risk. So through this dissertation, when we say a stock is better we mean its SR is higher than others unless otherwise mentioned.

2.4 In-Sample and Out-of Sample Data Set

The technical trading rules are not only used to analyze the historical data, but mostly also used for predicting purpose in the future, so we should consider its predictive ability for the future trading. One reasonable method is dividing the data set into two subsets: in-sample set and out-of-sample set. We train the data mining algorithm to get parameters in in-sample set, and keep the parameters in out-of-sample set to test, evaluate and verify it. Now, we give their definitions.

Parameters
Stock Market Data Mining Optimization
 (Version 2.0)
 University of Technology, Sydney Capital Market CRC

Inputs:
 Stock Code:
 In-Sample Start:
 In-Sample End:
 Out-Sample End:
 Days/Year:
 Investment(\$):
 Volume:
 Optimization target:
 Generations of GA:
 Return Definition:
 Transaction Cost: (\$) (%)
 Support(training set): %
 Support(testing set): %
 Risk Free Return: %

Parameters	Min	Max	Step
Fix Band X	0	0.316	0.001
Short Run	0	41	1
Long Run	0	89	1
Hold Day	0	30	1
Filter	0.00	1	0.001
Pre Day	30	100	1
Fix Band Y	0.00	1	0.001
Delay Day	0	30	1

In-Sample Results	Out-of-Sample Results
Total signals(B/S): 321(161/160)	Total signals(B/S): 372(186/186)
Average Return/Trade: 0.083%	Average Return/Trade: 0.021%
Win/Lose numbers: 104/216	Win/Lose numbers: 106/265
Profit(AU\$): 116.73	Profit(AU\$): 35.3
Confidence: 100 %	Confidence: 100 %
Standard deviation : 0.008	Standard deviation : 0.003
Sharpe Ratio: 0.107	Sharpe Ratio: 0.076

Figure 2.10 The parameters are determined by in-sample (training) set, and the output evaluation is come from out-of-sample (testing) set.

Definition 2.17 In-Sample Set (see **Definition 2.4 Training Data**). A data used to estimate or train a model. It is used to find the trading rules.

Definition 2.18 Out-of-Sample Set (see **Definition 2.5 Testing Data**). A data set independent of the in-sample data set, used to fine-tune or evaluate the estimates of the model parameters.

In this dissertation, the in-sample data set is set one-month period order book data, and out-of-sample data set is just consecutive one-month order book data except mentioned differently.

The strategy is that we use the in-sample data to train and find the trading rules, and, to evaluate the performance in the out-of-sample data set, so that the result can be used in the future trading prediction. (See Figure 2.10).

The parameters in Figure 2.10 are derived from in-sample set and kept to get an output directly from out-of-sample set. The results of in-sample set and out-of-sample set can be clarified in Figure 2.10. The Sharpe ratio of in-sample set and out-of-sample set are 0.107 and 0.076 respectively.

Definition 2.19 In-Depth Mining [Cao et al 2005] refers to a further mining either on existing patterns/rules or in selected/refined data sets.

Definition 2.20 Sub-Domain is a subset of a domain. Here, after we remove some part(s) of domain, only keep a subset of domain, in which the output result is optimized. For example, we can get a more profit or lower risk in this subset.

2.5 Summary

In this chapter, we explained and introduced the related notations, definitions, concepts, data, environments, conditions, technical trading rule and backgrounds of the project including both computer science and finance so as to make this thesis completed.

Firstly, we gave a brief introduction about the order book data, all the experiments in this dissertation are based on the order book data.

Secondly, we mentioned the technical trading rules. These rules are presented by previous domain experts and proved profitable and predictable [Ryan et al 1999] [Cao et al 2005]. Our research is based on the trading rules and more applications are further developed, for example, domain driven data mining technology and applications, robust genetic algorithms and the relationship between the sizes of investments and the number of stock-rule pairs, all of these will be introduced in the rest chapters.

Lastly, we described the criteria and related concepts, such as: Sharpe ratio, market index return, in-sample set and out-of-sample set, etc. These will be used to evaluate the performance of the system and compared to other systems.

Chapter 3 Domain-Knowledge Integrated Applications

Data mining algorithms must fit application domain so that it can improve performance. However, for data mining applications, on the one hand, developers put their complex business logic in the domain model, which makes the data in the database pretty simple, and creates a model in which to "get" to the business value we have to go through the domain model. On the other hand, business executions want / crave / require "ad-hoc" reporting, with the degree of "ad-hoc-ness" varying from simple reports to data warehouses. However this is impossible to use one domain model supports all of the complex logic. [Eric Evans, 2004]

3.1 Problems

Traditional data mining is a data-driven trial-and-error process where data mining algorithm extracts patterns from data according to some models [Mihael 2002]. It targets fully automated mining process, algorithms and tools. A data mining system is expected to be an automated tool without human involvement and the capability to adapt to external environment constraints.

Unfortunately, data mining in the real world is highly constraint-based [Ng et al 1998]. Real-world patterns interesting to business are often hidden in a large quantity of data with complex data structures and source distribution (data constraints). The real-world business process, problems and requirements are often tightly embedded in domain-specific information and expertise (domain constraints). Nonetheless, most of mined patterns would not be interesting or actionable to business even though the patterns are sensible to research, or there exists interestingness conflicts between academia and business (interesting constraints). Furthermore, the rules automatically discovered from domain-

specific data often do not make sense to real business process or regulations, or the rules must be integrated with other business rules so that the rules can be deployed into real life (rule constraint).

To solve the above mentioned constraints in the real world, it is essential to bring n new supports to existing data mining methodologies. Some real experience and lessons learned in artificial intelligence and pattern recognition have taught us the significance of the involvement of domain knowledge and even domain experts in solving complex real world problems. Similarly, in order to effectively mine and deploy interesting patterns from the aforementioned constraint-based context, the involvement of domain knowledge and experts and the consideration of constraint are essential for knowledge discovery in complex business data and analyzing domain-specific problems. Combining these two aspects, we feel it is crucial to develop a new data mining methodology for advising the process of real world data analysis and evaluation and refinement of mining results in a more effective way. This leads to the domain-driven in-depth pattern discovery (DDID-PD) framework.

The key ideas of the DDID-PD framework include (1) dealing with constraint-based context, (2) mining in-depth patterns, (3) supporting human-machine cooperative knowledge discovery, and (4) viewing data mining as a loop-closed iterative refinement process. Handling constraint-based context can improve the quality and effectiveness of data mining by extracting and transforming the domain-specific datasets in terms of guides taken from domain experts and their knowledge.

In-depth pattern mining can discover more interesting and actionable patterns from domain-specific perspective. In this framework, data mining and domain experts complement each other on an in-depth granularity via an interactive interface. The involvement of domain-specific data mining techniques and reduction of the complexity of the knowledge producing process can be implemented in real stock markets. A system following the DDID-PD framework

can embed effective supports for domain knowledge and experts' feedbacks, and refine the lifecycle of data mining in an iterative manner. Therefore, DDID-PD can benefit the real-world knowledge discovery of more interesting and actionable patterns from specific domains compared with current data-driven data mining methodology.

3.2 Domain Driven Model

3.2.1 Related Concepts

The domain-driven is a new concept still in discussion, so we give some definitions.

Definition 3.1 Human-Machine Cooperation. [Cao et al 2005] The in-depth pattern discovery is conducted under the cooperation of business analysts and data analysts.

Definition 3.2 Domain-Driven Pattern Discovery. [Cao et al 2005] It is not only a data-driven trial-and-error process, but rather a highly domain-dependent. It gets involved in domain expertise and constraints in a human-machine cooperation context.

Domain experts should object to terms or structures that are awkward or inadequate to convey domain understanding; developers should watch for ambiguity or inconsistency that will trip up design.

3.2.2 Model

Using a proven set of basic building blocks along with consistent language brings some sanity to the development effort. This leaves the challenge of actually *finding* an incisive model, one that captures subtle concerns of the domain experts and can drive a practical design. A model that sloughs off the superficial and captures the essential is a *deep model*. This should make the software more in tune with the way the domain experts think and more responsive to the user's needs. [also see Twocrows]

model elements gives developers a steady platform from which to apply the modeling approaches.

Figure 3.1 shows the DDID-PD model in the system, and Figure 3.2 shows the layered architecture. Figure 3.3 shows the process of data mining process.

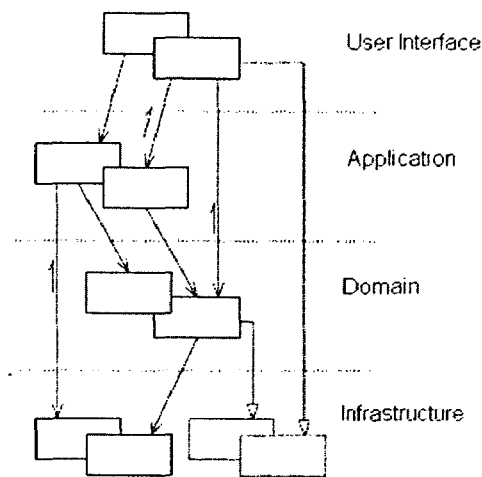


Figure 3.2 Layered model.

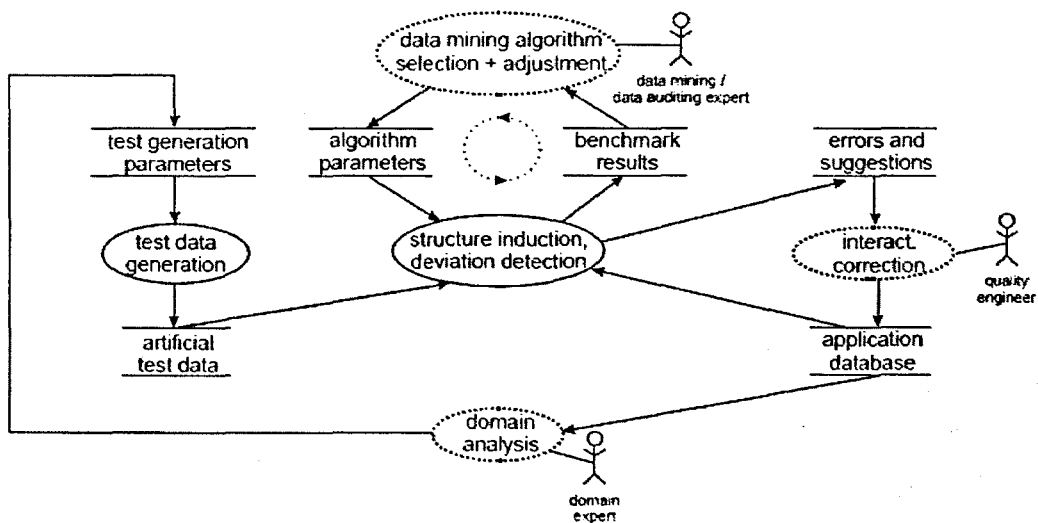


Figure 3.3 Domain-driven data mining process.

In designing a large system, there are so many contributing components, all complicated and all absolutely necessary to success, that the essence of the domain model, the real business asset, can be obscured and neglected.

3.3 Human-Machine Interface

3.3.1 Domain Knowledge Database

For any data mining systems, domain knowledge becomes more and more important in improving efficiency and effectiveness, because it can reduce unnecessary computation and search. In stock markets, expert experiments are also very important to finding optimal parameters, stocks, rules, etc. Because the experts in different fields have different kind of knowledge, the best method to integrate domain knowledge into a system is human-machine interface.

For a system to satisfy all different users, the human-machine interface is one simple and actionable method for overcoming the problems in a single system, it permits different users to select different parameters and interface to solve their own problems.

Table 3.1 The structure of domain knowledge database and some instances.

Parameter name	Default	Lower bound of	Upper bound of
Short run	10	1	50
Long run	40	20	100
Signals per year	-	20	150
Fix Band X	0.1	0.0	1.0

The structure of domain knowledge database (see Table 3.1) is concentrated on parameters. Most of the data in this database is a relational formula. The data come from expert experiences, domain constraints and feedback of system output. A parameter can be of trading rule or not, such as the size of in-sample set, the size of out-of-sample set, and the sizes of investments.

Sometimes, for a parameter, we do not know the default value, lower bound or upper bound value, or some of them, we can ignore it or set it as the smallest/biggest value for lower/upper bound. For example, Long run of a moving average, default value is 50 (we have not any experience for it), lower bound is 1 (we have any experience for it), and upper bound is 50 (expert experience).

After we embedded the domain knowledge data base into the system, it improves not only efficiency but also effectiveness. Moreover, it can remove useless domain.

Figure 3.4 shows domain knowledge operations.

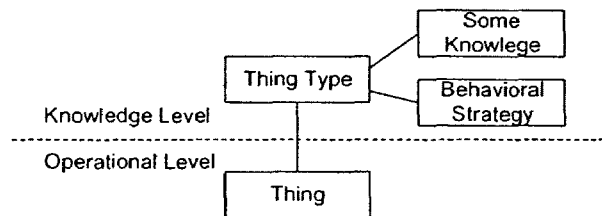


Figure 3.4 Knowledge level and operational level.

3.3.2 Human-Machine Interface and Domain Knowledge Integration

In stock market data mining, domain knowledge is various, trading rule name, parameters for each trading rule, stock names, time duration for every stock, the size of in-sample set and out-of-sample set, the number of generation of GA. If all of these we compute by enumerate algorithm, the execution time and memory may be a terrible disaster. However, if experts have already known some of them, or even a sub-domain of them, such as the in-sample data size is 9 months from Jan 1 to Oct 1, 2002 and out-of-sample data size is 3 months from Oct 1 to Dec 31 for year 2002, stock name is W30. This result is computed by our system and can be added into domain knowledge data base to improve efficiency. It helps users to improve performances.



Figure 3.5 The result for the optimized in-sample and out-of-sample data set. (2002 ASX order book, Moving average)

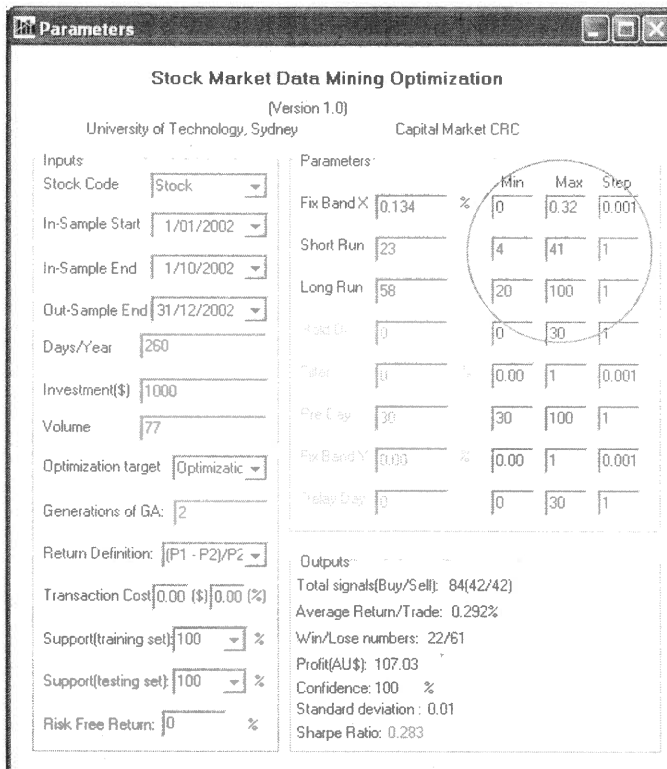


Figure 3.6 The user-interface which can be integrated with domain knowledge (ASX index).

In Figure 3.6, the in-sample and out-of-sample ranges come from domain knowledge data base. (See Figure 3.5.)

Figure 3.6 shows the human-machine interface, it makes the system more convenient and efficient, such as, setting smaller range and more reasonable parameters. Figure 3.7 shows the output of the system.

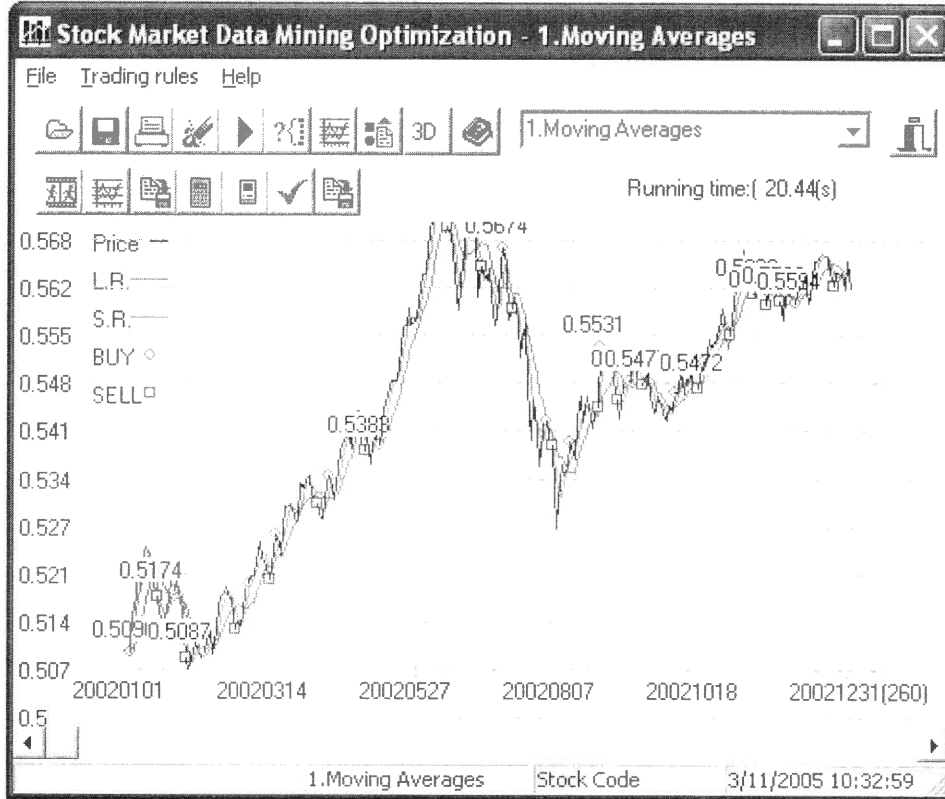


Figure 3.7 The result for the optimized in-sample and out-of-sample data set. (Moving average)

Further, domain knowledge includes both expert experience and feedback of the system output. Domain knowledge is accumulated automatically into the knowledge database, such as, the better stock-rule pairs, length of in-sample data set, length of out-of-sample data set, fix-band size, delay-day, hold-day, long run and short run, etc. An effective database improves the performance by setting a default value for each parameter.

3.4 Summary

If the design, or some central part of the design, does not map to the domain model, that model is of little value, and the result of the system is worse, such as inefficient or ineffective. At the same time, complex mappings between models and design functions are difficult to understand and, in practice, impossible to maintain as the design changes.

In stock market data mining, domain knowledge is most important as well, especially for different type of investors: scholars, researchers, dealers, brokers and market managers, who have different targets and different knowledge. It is difficult and impossible for a general computer system to deal with all the problems investors face to, so a domain-driven system is essential and necessary.

In this chapter, a Domain-Driven Model architecture and process are introduced and some experiments are given. With domain expert knowledge, the performance of the system can be improved and verified. Moreover, the domain database can be accumulated from expert experiences and system feedback automatically.

Chapter 4 In-Depth Pattern Discovery and Related Applications

4.1 Fundamental Concepts

Businesses which rely on queries, reports and OLAP (On-Line Analytical Processing) systems often consider these activities to be data mining; but, at best, the queries, reports and OLAP systems are only the first step. The businesses run into trouble when they try to generalize from the information they've uncovered and use the information as a guide to future behavior. A description is not the same as a prediction.

Data mining uses a variety of data analysis tools to discover patterns and relationships in data that can be used to make reasonably accurate predictions [CRISP-DM]. It is a process, not a particular technique or algorithm. I want to emphasize that the goal of data mining is prediction, generalizing a pattern to other data. Exploring and describing the database is merely the starting point.

The traditional approach falls short on several counts when it comes to making useful predictions. First, the analyst may fail to select the most appropriate attributes (columns in the database). It may be easy to decide that *annual purchases* are a more significant number than *customer ID*; but when you're dealing with 5 million cases, each of which has 200 attributes, it is extremely difficult to identify everything that is important.

As database structure grows increasingly complex (e.g., 50 million cases each with 2,000 attributes) [Cao et al 2005], it becomes virtually impossible for any individual to know the data well enough to say with confidence which variables affect behavior. The difficulty is exacerbated by the fact that the best predictors may not be individual attributes, but rather a combination of attributes.

Because data mining is essentially an iterative process, quantitative results go through a reality check and are revised as needed until a meaningful predictive model evolves. The knowledge of the domain expert guides the analysis of the data and the manipulation of variables.

Data mining also addresses another failing of the descriptive approach. Even after a pattern is unearthed through a series of queries, the analyst can't be sure whether that pattern holds true for anything other than the collection of data used to find it. The analyst may try to identify potential buyers of a certain product after building a profile of customers who have already bought that product, but will this profile apply to people who are not yet customers?

For example, analysis may show that 75 per cent of purchasers for a certain retail product are male. Therefore, the retailer decides to target men as the likeliest potential buyers in the future. However, if the store's overall customer distribution is 75 per cent male and 25 per cent female, there's not much new information in the fact that 75 per cent of this particular product's buyers are male. Data mining might reveal that education and age are better predictors of buying behavior than gender. Perhaps this product will be especially popular with a particular demographic segment of women, implying a very different promotional strategy than initially planned.

Data mining methodology, on the other hand, tries to verify that the patterns you find can be used for prediction (i.e., that the patterns are applicable beyond the original database). It does this using a variety of techniques, such as dividing the database and developing a predictive model on one portion that is then tested on the other portion. Data mining can assess both the mathematical accuracy and the potential costs and revenues of a particular predictive pattern. (If it costs \$100 each to reach the ideal buyer for your \$25 product, you might want to modify your marketing plan.)

Clearly, there is more to data mining than just summarizing and querying the database, but running algorithms should only require 10 to 20 per cent of a

project's time and resources. The bulk of the effort needs to be spent on data preparation, which includes building the data mining database, exploring the data and transforming the data for mining. As predictive models are generated, the models need to be evaluated to ensure that the models are meaningful. The ultimate results can be very rewarding.

Existing association rule mining algorithms are specifically designed to find strong patterns that have high predictive accuracy or correlation. Many useful patterns, for example, out-expectation patterns with low supports, are certainly pruned in these mining algorithms. This chapter introduces our research developing novel theories, techniques and methodologies for discovering hidden interactions within data, such as class-bridge rules and out-expectation patterns. These patterns are essentially different from traditional association rules, but are much more useful than traditional ones to applications such as cross-sales, trend prediction, detecting behavior changes, and recognizing rare but significant events. This delivers a paradigm shift from existing data mining techniques. [Zhang et al 2005]

In the DDID-PD framework, a collection of concepts are proposed in terms of applicable requirements from the real world. These concepts bring either new ideas or deep things into the existing data mining framework, and enhanced the efficiency and effectiveness of real-world data mining.

Definition 4.1 Generic Pattern. [Cao et al 2005] Referring to patterns automatically discovered by data mining models while taking little consideration of business requirements and interestingness.

Definition 4.2 In-Depth Pattern. [Cao et al 2005] Referring to patterns which are highly interesting and actionable in business decision-making. These patterns are created through refining model or tuning parameters to optimize generic patterns; these patterns may also be directly discovered from data set with sufficient consideration of business requirement and constraints.

In-depth patterns are not only interesting to data miners, but also to business decision-makers. In the afore-mentioned trading strategies, more actionable trading strategies can be found via model refinement or parameter tuning.

4.2 DDID-PD Process Model

The components of the DDID-PD framework are shown in Figure 4.1. [Cao et al 2005] The lifecycle of DDID-PD is as follows (the sequence is not rigid, some phase may be bypassed or moved back and forth in a real problem).

P1 . Problem understanding and definition;

P2 . Data understanding;

P3 . Data processing;

P4 . Modeling;

P5 . Results evaluation;

P6 . Based on feedbacks and progress of the phases from *P2* to *P5*, it is quite possible that each phase may be iteratively reviewed starting from *P1* via the interaction with domain experts in a back-and-forth manner for the refinement of mining results;

P7 . Results post-processing; or

P6' : In-depth modeling on the mined results where applicable then going to *P7*;

P8 . Going back and reviewed phases from *P2* on may be required;

P9 . Deployment;

P10 . Knowledge and report delivery.

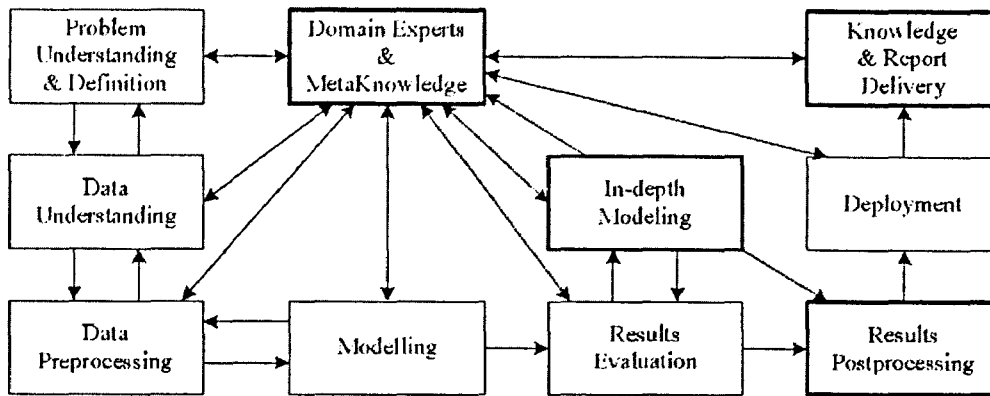


Figure 4.1 Domain-Driven In-Depth pattern discovery process models.

The DDID-PD process highlights four highly correlated ideas that are critical for the success of a data mining process in the real world. The four ideas are (1) constrained-based context, multiple types of constraints widely exist in the domain problem and its analysis objectives; (2) in-depth pattern mining, another round of modeling on the first-round results may be necessary for mining patterns really interesting and applicable to business; (3) human-cooperated interactive knowledge discovery, the involvement of domain experts and their knowledge and the interaction between experts and mining system in the whole process are important for effective execution of the mining; (4) a loop-closed iterative refinement is the outcome of iterative refinement.

The following sections outline them one by one. [Cao et al 2005]

4.2.1 Constraint-Based Context

In human society, everyone is constrained by either personal situations or social regulations. Similarly, advanced knowledge discovery and smart decision-making need consider real-world aspects such as environmental reality, expectations and constraints in the real-world process. More specifically, the following four kinds of constraints play important roles in building an effective and efficient data mining system. The constraints consist of domain-specific, functional and environmental constraints, and form a constraint-based data

mining context [Ng et al 1998]. The constraints are: data constraints, domain constraints, rule constraints and interesting constraints. We give the explanation in the following.

Data constraints: this is related to data quantity, data structures, data semantic complexity, data distribution, etc.

Domain constraints: it involves domain type, domain knowledge, human capability and role, business process and workflow, characteristics, qualitative and quantitative hypothesis and conditions, etc.

Rule constraints: it includes rule representation, rule explanation, rule interestingness to analytical goals, rule deployment in the integration with real-world business process and environment, etc.

Interesting constraints: this is driven whether by academic or industrial process and workflow, problem requirements and analytical goals, etc.

All the above constraints may vary from domain to domain. The constraints may get involved in the whole process or only specific local process.

4.3 Mining In-Depth Patterns

4.3.1. Mining In-Depth Patterns

Existing data mining algorithms, such as association rule mining or decision-tree, often generate a large number of patterns, but most of them either are redundant or do not reflect the true interestingness. This has hindered the deployment and adoption of data mining in the real applications. Taking trading rules in finance as an example, a trading rule, e.g. Moving average rule, usually implies millions of individual rules. However, most of them are not applicable for a specific business environment. Therefore, it is essential to further refine these rules so that more interesting and actionable rules can be discovered and recommended for more smart and effective decision-making. To overcome this disadvantage, in deploying data mining into the real world, we need to discover more interesting

and actionable rules based on a domain-specific problem and its business requirement. This leads to in-depth mining.

In data mining applications, the involvement of domain knowledge and constraint are often necessary for conducting in-depth mining. More importantly, some appropriate in-depth mining techniques should be developed on the demand of a domain-specific problem.

4.3.2 Human-Machine-Cooperated Interactive Knowledge Discovery

Real-world data mining should be a human-machine-cooperated knowledge discovery process rather than an autonomous system. Domain experts consist of the center or an essential constraint of the data mining process via dynamic expert-model intersection. In fact, the experts and their knowledge play significant roles in the whole data mining process such as business and data understanding, features selection, hypotheses proposal, model selection and learning, and evaluation and refinement of algorithms and resulting outcomes. For instance, domain experts can narrow down the selection of features and models, and create high quality hypothesis and efficient constraints based on their domain knowledge, which effectively accelerates the mining process.

Instead of producing patterns or knowledge directly from data, the domain-driven data mining methodology allows domain experts and/or their knowledge to be front or center of the mining process, and interact with data integration, feature selection, interpretation of algorithms and resulting outcomes. For instance, domain experts can incorporate their knowledge into data and feature selection and constraint on business data and problems. This point may also be called as human-centered [Mihael 2002], human-involved, supervised or guided data mining.

From above, domain-driven in-depth data mining supports in-depth analysis with the assistance of domain knowledge. Furthermore, the mining is actually the interaction between domain-expert and mining system. To support the dynamic

interaction, user-friendly human-machine interfaces are necessary. The interface from domain experts can be online and instantly embedded into the mining system and knowledge base on requirement, and refine tune the quality of final mined rules. This actually makes a data mining process and tool as highly interactive and dynamic rather than as fully automated as previously imagined. For this commitment, the knowledge base including expert systems, AI, PR and cognitive science needs to be involved. A good option is to build intelligent agents-based data mining platform to support user modeling, user interface, and so on. This is also called interactive mining.

4.3.3 Loop-Closed Iterative Refinement

The data mining process and its system are closed rather than open, since it encloses iterative refinement and feedback of hypotheses, features, models, evaluation and explanations in a human-involved context. The real-world mining process is iterative because the evaluation and refinement of features, models and outcomes cannot be completed once but rather is based on iterative feedbacks and interaction during the whole process.

The data mining process and its system are closed with iterative refinement and feedback of hypotheses, features, models, evaluation and explanations in the human-involved or –centered context. It iteratively evaluates and tunes features and models based on feedbacks from and the involvement of domain experts and their knowledge, and the interaction with the domain problem.

Specific data mining process needs to be designed for a particular problem. In the process, we may consider how to involve domain expert knowledge, feedbacks, fine-tuning work, evaluation and modification in an iterative and incremental manner.

To support the loop-closed iterative refinement, some appropriate human-computer interaction interface should be designed.

4.4 Mining Stock-Rule Pairs in Real Stock Markets

4.4.1 Mining In-Depth Trading Rules (Sub-Domain)

In stock market, since a long time ago, financial researchers have developed many trading rules to support traders' decision-making [Frenkel et al 2004]. These rules actually indicate possible patterns hidden in stock markets [Leigh et al 2002]. For example, the trading strategy Moving average (MA) actually indicates a correlated pattern between two features namely short-run moving average and long-run moving average. The pattern MA is defined in Chapter 2.

The in-depth pattern mining on existing trading rules aims to mine more actionable rules which can better serve traders' objectives. See Chapter 6.2. (We call the actionable rules "in-depth rules". Different combination of parameters is a different "model" (also called "rule"), but some of them are not actionable in future trading. Our purpose is to look for an actionable and profitable rule)

This pattern actually consists of a large number of rules (we call them generic rules) from finance perspective, such as a group value of a parameter is actionable and profitable. However, traders do not know which rule is actionable for assisting in their specific trading decision.

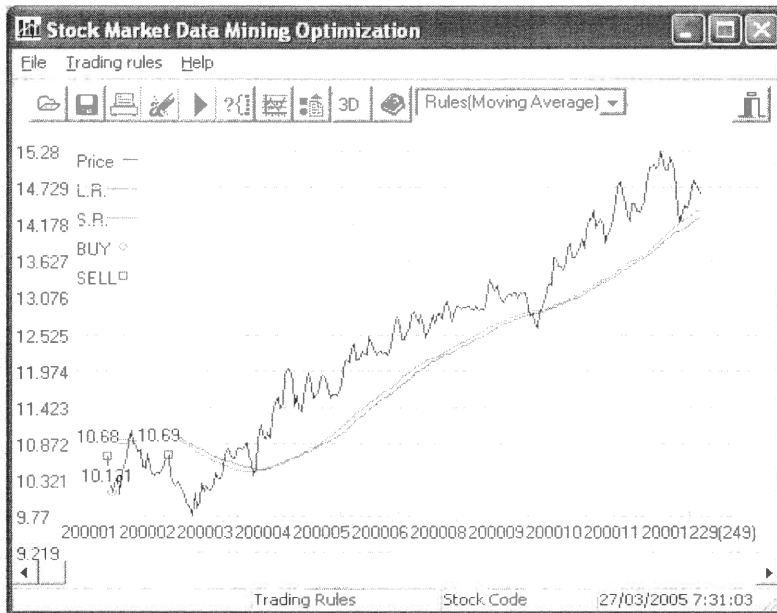


Figure 4.2 The signals of one single best value for moving average and stock W33 intraday data from ASX.

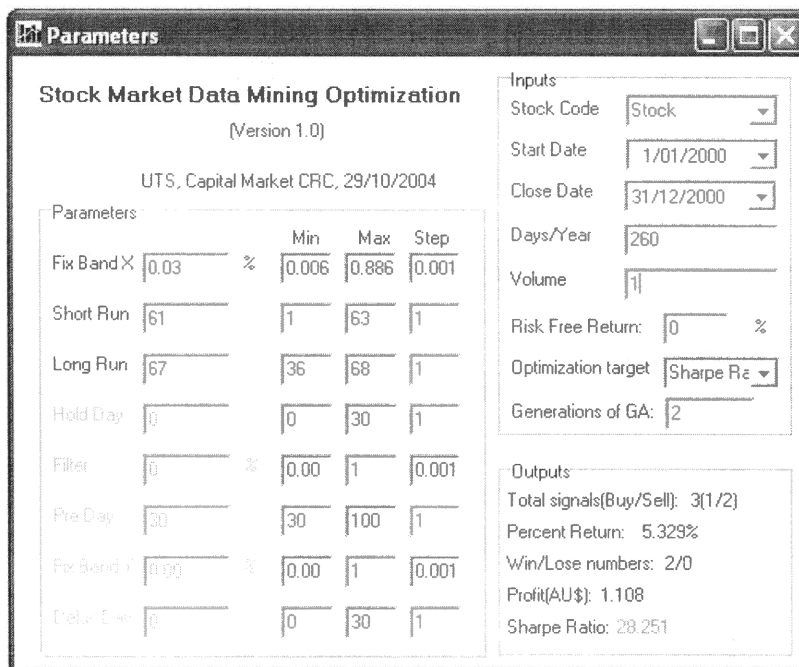


Figure 4.3 For generic MA rules, the best parameter value of moving average.

Moreover, from the above Figures 4.2 and 4.3, the result is the best mathematical one, but it is not able to predict the future trading in market. So we improve it by considering the domain knowledge, such as, Sharpe ratio range, the number of trades and return range, etc. See the improved result in Figures 4.4 and 4.5.

See Figure 4.2, Sharpe ratio 28.251 is the maximal value by mathematical formula.

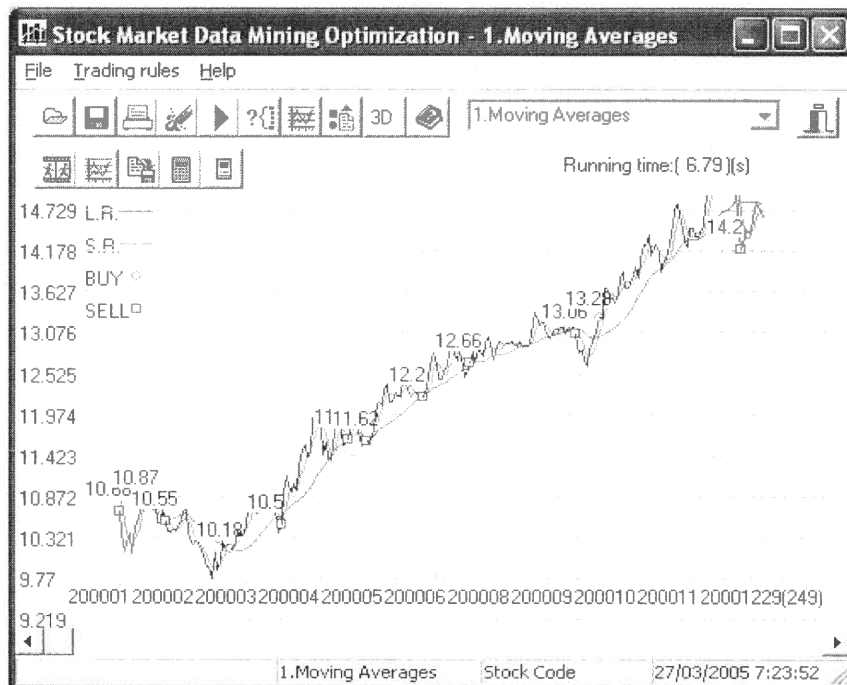


Figure 4.4 The best sub-domain of moving average trading signals.

However, it is a result of unreasonable and impracticable model (a combination of parameters is a model, also called a rule). For one-year trading, there are only three trading signals in January, but, no trading during the other 11 months. The reason is Sharpe ratio is a statistics value, sometimes, it can be a strange point. When some trading signals are equally distributed beside mean value, it may get a significant maximal value. In a real market, price changing trends cannot be regarded equally distributed around the mean value. Such as the example shown in Figure 4.2, the three signals are distributed equally. One-year model is decided

by one-month model, but the other 11 months data are out of pattern. Moreover, we can see the profit in Figure 4.2 is only A\$ 1.108, less than that of the model in Figure 4.3, whose profit is A\$ 3.44. When a trading model is incorrect, the result is pointless and it cannot generate actionable or profitable signals out-of-sample set.

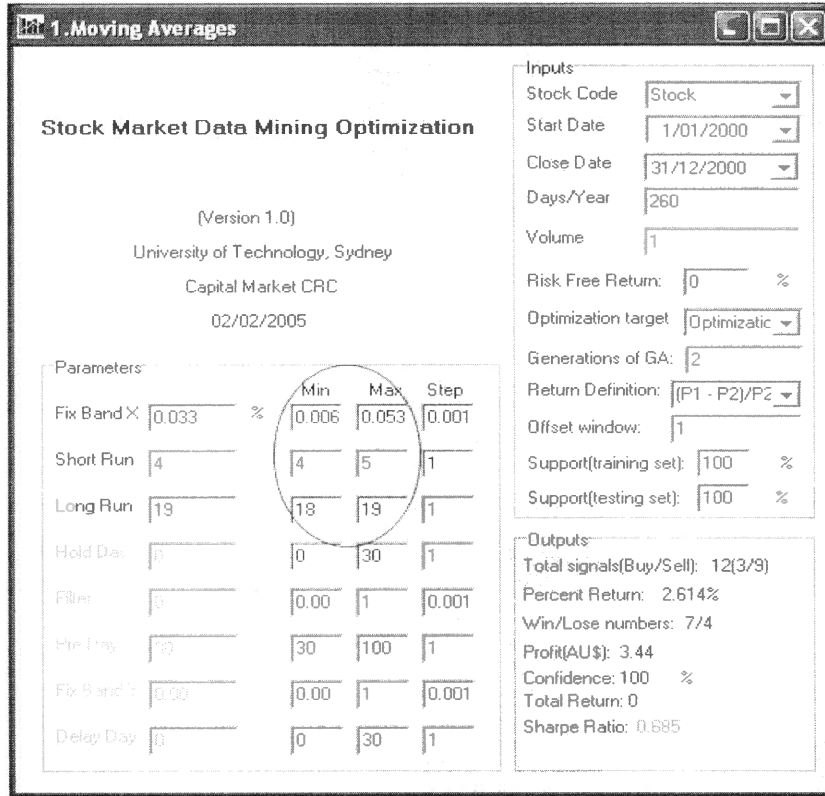


Figure 4.5 The best sub-domain of moving average.

In order to find the optimized rules from the generic rules, genetic algorithm (GA) [Lin et al 2004a] and robust genetic algorithm (RGA) [Lin et al 2005c] and a human-machine interaction interface are developed so that financial experts can supervise the construction of some features dynamically and interactively to narrow down the search space. Taking the MA as an example, an in-depth rule MA (4, 19, 0.033) is found in the in-sample data. The Sharpe ratio is greatly improved to positive scope compared with the generic results. This demonstrates that the in-depth mining with the involvement of domain knowledge can lead to

more interesting and actionable rules for trading support. (See Figures 4.4 and 4.5.)

4.4.2 Mining In-Depth Stock-Rule Pairs

It is assumed that some trading rules are suitable for a cluster of stocks, while others are more effective to guide the trading of other stocks in the market. This hypothesis actually indicated whether there are correlations between trading rules and stocks. If yes, and if we can find the correlation, then it would be very helpful for guiding the real trading.

Based on this hypothesis, we developed algorithm to search the in-depth correlations between trading rules and stocks in real stock data. The basic ideas of the Stock-Rule pair mining algorithms are as follows.

(1) Mining in-depth rules for individual stock;

For each security, a set of in-depth rules are discovered for each cluster of trading rules by the robust genetic algorithm (RGA). Furthermore, in-depth rules can be discovered from all classes of rules for all stocks respectively. As a result, a Stock-Rule set is found in which a trading rule is matched with one or multiple stocks.

(2) Mining the highly Stock-Rule pairs;

In the above step, multiple in-depth rules from different rule class may be found suitable for one stock. It is necessary to discover a highly correlated rule for a specific stock from the above resulting set. This leads to the most suitable rule for a stock, and forms a correlated Stock-Rule pair.

(3) Refining and evaluating the Stock-Rule pairs.

For finding the interesting and actionable Stock-Rule pairs, the assistance of domain experts and their suggestions are essential for the refinement and evaluation of pairs found in the above steps.

We have done the Stock-Rule pair correlation in ASX, the six rules (Simple moving average, Filter moving average, Enhanced moving average, Channel break-out, Filter rule, Support and resistance) and 26 stocks are tested in the system. The result is described in Chapter 6. From the result, we can get the relationship between the sizes of investments and return, the number of pairs and the sizes of investments.

4.5 Summary

In this chapter, some concepts of DDID-PD are defined. In the real world, correlated patterns interesting to business are often hidden in domain-specific data and constraint-based context. This often leads to the scenario as too many rules are mined while few of them are truly interesting to business when using usual correlation mining techniques. Therefore, in-depth pattern discovery should be conducted on the domain-specific constraint-based context. To this end, we have done some experiment in the DDID-PD framework to guide the real-world data mining. The main phases and components of the DDID-PD framework (shown in Figure 4.1) include almost all phases of the well-known industrial data mining methodology CRISP-DM. while there are three big differences from the CRISP-DM [CRISP-DM]: (1) Some new essential components highlighted by thick rims, such as result post-processing and in-depth modeling. (2) In the DDID-PD, the phases of CRISP-DM highlighted by shadow are enhanced by dynamic interaction with domain experts and the consideration of constraints and domain knowledge. (3) The life cycle of the DDID-PD is actually different from that of CRIPS-DM.

Deploying the DDID-PD framework, the stock-rule pair in real stock market has been analyzed. It has been found that there are correlations between stocks and trading rules based on the knowledge of financial experts. The experiment shows that the mined correlations ideas of DDID-PD framework are interesting and

actionable to real trading. The ideas of DDID-PD can assist in mining interesting real-world patterns in an effective and efficient manner.

Chapter 5 Optimized Algorithms

5.1 Standard Genetic Algorithm (SGA)

5.1.1 Background

To set a parameter value of technical trading rules has a profound impact on the profitability of these rules. In order to maximize the profit, the parameter values must be chosen optimally. In this optimization problem, it is important to be aware of two issues. First, there are a great number of possible parameter values. Second, the profit surface is characterized by multiple optima. Genetic algorithms are very efficient and effective approaches to this type of problem. [Robert 1999] [O'Reilly 2005] [Aytug et al 2000]

Efficiency refers to the computational speed of the optimization technique. Through a recombination procedure known as crossover, mutation and by maintaining a population of candidate solutions, the genetic algorithm is able to search quickly through the profitable areas of the solution space. Effectiveness refers to the global optimization properties of the algorithm. Unlike other search or optimization techniques based on gradient measures, a genetic algorithm avoids the possibility of being anchored at local optima due to its ability to introduce random stocks into the search process through mutations. Since a genetic algorithm is an appropriate global optimization method, it can be used to search for the optimal parameter values for trading rules. [Vose 1996]

5.1.2 Solution

Genetic algorithms are heuristic for function optimization, where the extreme of the function (i.e., minimal or maximal) cannot be established analytically. A population of potential solutions is refined iteratively by employing a strategy inspired by Darwinist evolution or natural selection. Genetic algorithms promote “survival of the fittest” [Vose 1999][Neely et al 1996] [Nix et al 1992]. To

improve efficiency and keeping a near-optimal value, we present standard genetic algorithms to solve this problem [Robert 1999] [Davis 1987]. Algorithm 5.1 is a standard genetic algorithm (we do not do any improvement on the genetic algorithm, so we call it standard genetic algorithms).

```

P ← InitializePopulation();
Generation ← 3                ; all the initial values can be set by users
Population ← 3000
Fix Band X ← 0.000 .. 0.100   // per cent
Long run ← 1 .. 100
Short run ← 1 .. 100         // Short run < Long run
Delay day ← 0 .. 30
Hold day ← 0 .. 30
While (not stop (P)) do
    Parents[1..2] ← SelectParents(P);
    Offspring[1] ← Crossover(Parents[1]);
    Offspring[2] ← Mutation(Parents[2]);
    P ← Selection(P, Parents[1..2], Offspring [1..2]);
Endwhile.

```

Algorithm 5.1 Standard genetic algorithm

In algorithm 5.1, different parameters have different domain, so we use function RANDOM () to generate a random value between 1 and 1000, and transform it into the domain of each parameter. For example, a long run of a moving average rule is between 1 to 100, and it is bigger than a short run; a Fix-band-X of a moving average rule is between 0 per cent to 10 per cent. Crossover rate is 25 per cent and mutation rate is 1 per cent.

5.1.3 Conclusion

In this sub-section, we show the comparison of standard genetic algorithm and enumerate algorithm. Through these comparisons, we can conclude the result that the genetic algorithm can get an output in real time.

```
Max = -Maxium;
For Fix Band X from 0.000 to 0.100 step 0.001
  For Long run from 1 to 100 step 1
    For Short run from 1 to 100 step 1           // Short run < Long run
      For Delay day from 0 to 30 step 1
        For Hold day from 0 to 30 step 1
          Max = (Max, Compute_Sharpe_Ratio() );
        End For (Hold day)
      End For (Delay day)
    End For (Short run)
  End For (Long run)
End (Fix Band X)
Output (Max);                               // The best result;
```

Algorithm 5.2 Enumerate Algorithm

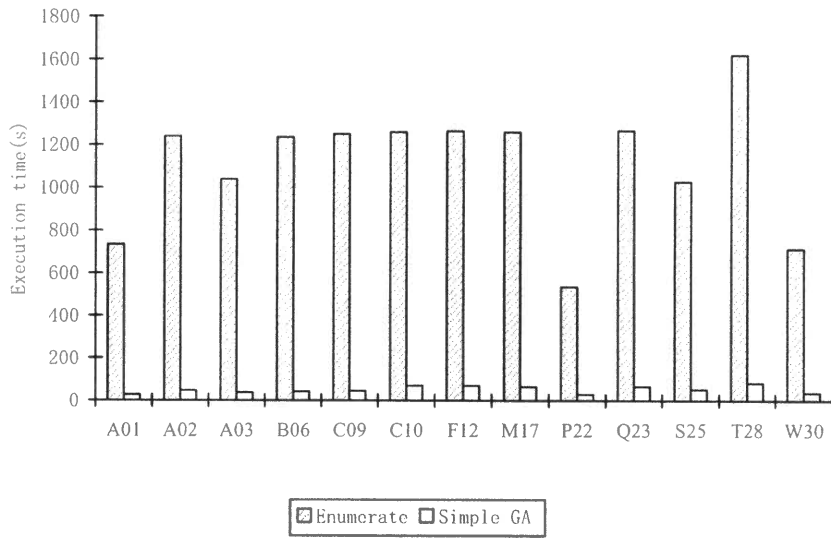


Figure 5.1 The execution time of enumerate algorithm and SGA.

In Figure 5.1, Filter rule, order book data and the time period is from 01 January 1998 to 20 February 2001, in-sample data, [Lin 2004a]

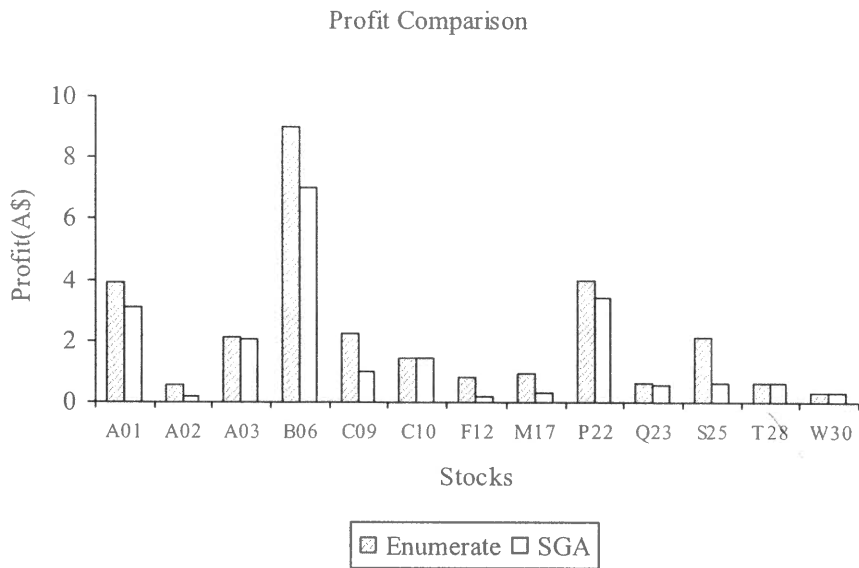


Figure 5.2 The profit comparison between enumerate algorithm and SGA.

In Figure 5.2, the rule is Enhanced Moving Average, 1 January 2001 to 31 January 2001, intra-day order book data, volume = 1 and without transaction cost. Figure 5.2 shows the SGA result is near to the enumerate result, but execution time of SGA is only 1 per cent of enumerate algorithm (Figure 5.1).

5.2 Robust Genetic Algorithm (RGA)

5.2.1 Background

For SGA, we only consider the mathematical maximal value, but pay no attention to noisy signals. Sometimes, we are not able to get applicable signals and parameters, because the parameters are noise (see Figures 5.3 and 5.4). In order to get the applicable and predictable parameters in future trading, we need consider getting the real parameters for a real market changing trend.

5.2.2 Robust Genetic Algorithms

We should consider SGA with domain knowledge in order to get more reasonable results both in Mathematics and Finance. We call the new algorithm the robust genetic algorithm (RGA) [Lin et al 2005c]. Domain knowledge comes from domain knowledge data base (see Chapter 3.3), for example:

- (1) The number of signal in one month should be at least more than 10;
- (2) The trading frequency;
- (3) The parameter ranges, such as, short day (short run) may be about 20 and long day (long run) maybe about 50;
- (4) The output range. For example, Sharpe ratio is between -2 to 2, and the maximal range is -4 to 4.

Domain knowledge data base can be accumulated by system feedback and domain experts. For example, the difference between long day and short day is nearly equal to the short day.

5.2.3 Comparison

The comparison results of SGA and RGA are discussed in the following section.

Figure 5.3 shows the interface and result. The stock name is W30. Trading rule is the Enhanced Moving Average (EMA) [Chapter 2]. In-sample data is one-month from 1 January 2001 to 31 January 2001 and out-of-sample data is one-month continuously after in-sample data without transaction cost. The following figures are under the same conditions.

Algorithm 5.3 is used to find an optimized sub-domain output.

```
P ← InitializePopulation();
Generation ← 3 // all the initial values can be set by users
Population ← 3000
Fix Band X ← 0.000 .. 0.100 // per cent
Long run ← 1 .. 100
Short run ← 1 .. 100 // Short run < Long run
Delay day ← 0 .. 30
Hold day ← 0 .. 30
While (not stop (P[top 5 percent])) do
  Parents[1..2] ← SelectParents(P);
  Offspring[1] ← Crossover(Parents[1]);
  Offspring[2] ← Mutation(Parents[2]);
  P[top 5 percent] ← Selection(P[200], Parents[1..2], Offspring [1..2]);
Endwhile.
```

Algorithm 5.3 Robust genetic algorithm to find an optimized sub-domain.

Algorithm 5.3 outputs top 5 percent value as a sub-domain.



Figure 5.3 The signals generated by SGA. There are only five signals during two months. It is not a reasonable model, although its Sharpe ratio is the best.

Figure 5.4 shows the parameters of SGA.

Parameters

Stock Market Data Mining Optimization
[Version 1.0]
University of Technology, Sydney Capital Market CRC

Inputs		Parameters				
Stock Code	Stock	Fix Band X	0.632 %	Min: 0.067	Max: 0.859	Step: 0.001
In-Sample Start	1/01/2001	Short Run	5	4	78	1
In-Sample End	31/01/2001	Long Run	23	23	100	1
Out-Sample End	28/02/2001	Hold Day	0	0	30	1
Days/Year	260	Filter	0 %	0.00	1	0.001
Investment(\$)	1000	Pre Day	30	30	100	1
Volume	77	Fix Band Y	0.00 %	0.00	1	0.001
Optimization target	Sharpe Ra	Delay Day	0	0	30	1
Generations of GA:	2	Outputs				
Return Definition:	$(P1 - P2)/P2$	Total signals(Buy/Sell)	5(3/2)			
Transaction Cost	0.00 (\$) 0.00 (%)	Average Return/Trade:	0.085%			
Support(training set)	100 %	Win/Lose numbers:	2/2			
Support(testing set)	100 %	Profit(AU\$):	1.54			
Risk Free Return:	0 %	Confidence:	100 %			
		Standard deviation:	0.001			
		Sharpe Ratio:	0.707			

Figure 5.4 The output result of SGA. The best Sharpe ratio is 0.707 and there are only five signals. The number of signal is not acceptable although it is really the “best” value depending on mathematic definition.

Figures 5.5 and 5.6 show the result of RGA. The pattern is more reasonable than that of SGA, because the number of signals of SGA is only 5 (3 buy signals and 2 sell signals in two months period totally 13998 transactions, but the RGA has 82 signals in the same period. It is more reasonable.)

Parameters

Stock Market Data Mining Optimization

(Version 1.0)
University of Technology, Sydney Capital Market CRC

Inputs

Stock Code:

In-Sample Start:

In-Sample End:

Out-Sample End:

Days/Year:

Investment(\$):

Volume:

Optimization target:

Generations of GA:

Return Definition:

Transaction Cost: (\$) (%)

Support(training set): %

Support(testing set): %

Risk Free Return: %

Parameters

	Min	Max	Step
Fix Band X: <input type="text" value="0.125"/> %	<input type="text" value="0.001"/>	<input type="text" value="0.304"/>	<input type="text" value="0.001"/>
Short Run: <input type="text" value="16"/>	<input type="text" value="4"/>	<input type="text" value="51"/>	<input type="text" value="1"/>
Long Run: <input type="text" value="31"/>	<input type="text" value="18"/>	<input type="text" value="98"/>	<input type="text" value="1"/>
Hold Day: <input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="30"/>	<input type="text" value="1"/>
Filter: <input type="text" value="0"/> %	<input type="text" value="0.00"/>	<input type="text" value="1"/>	<input type="text" value="0.001"/>
Pre Day: <input type="text" value="30"/>	<input type="text" value="30"/>	<input type="text" value="100"/>	<input type="text" value="1"/>
Fix Band Y: <input type="text" value="0.00"/> %	<input type="text" value="0.00"/>	<input type="text" value="1"/>	<input type="text" value="0.001"/>
Delay Day: <input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="30"/>	<input type="text" value="1"/>

Outputs

Total signals(Buy/Sell):

Average Return/Trade:

Win/Lose numbers: 23/58

Profit(AU\$): 106.24

Confidence: 100 %

Standard deviation : 0.01

Sharpe Ratio: 0.297

Figure 5.5 The robust parameters combination.

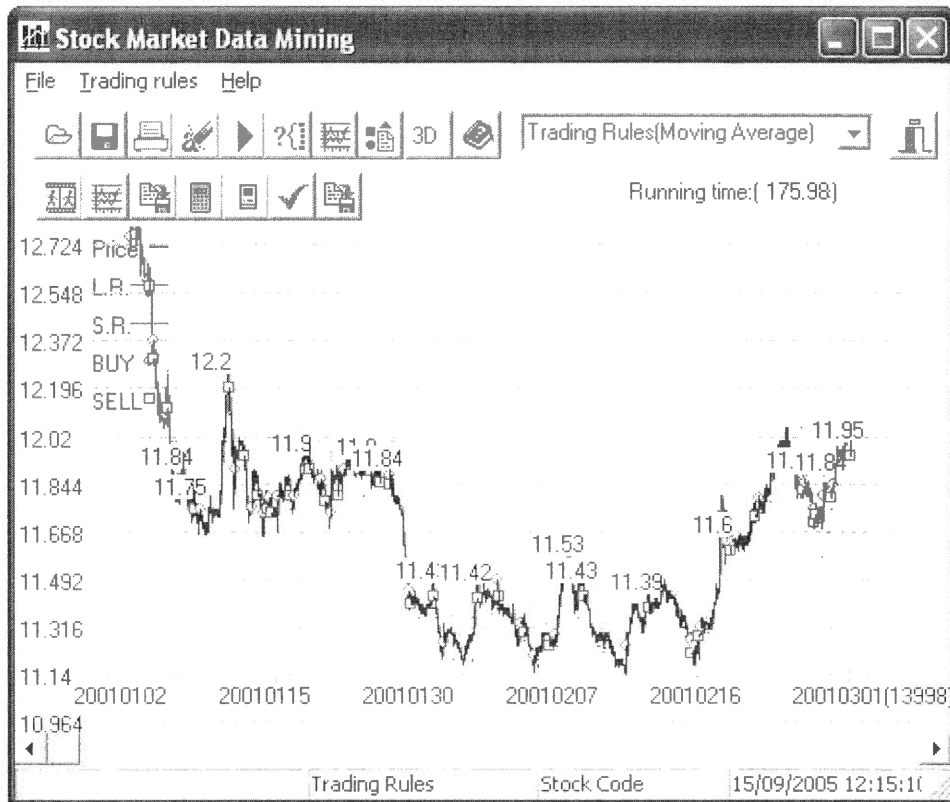


Figure 5.6 The robust alert signals which are more reasonable and applicable.

5.3 Fuzzy Set Methods

If the system tells the traders that the Sharpe ratio of a stock-rule pair is 0.123, what decision will traders make? Choose this pair or give it up. It is difficult for traders to make any decisions, because traders do not know whether the pair is good or not. However, if the system tells traders that a pair is “very good”, the traders can make a decision easily. So the traders want the system to evaluate a stock by “very good”, “better”, “good”, “normal”, “medium”, “bad”, “very bad”, and so on, instead of a numerical value of Sharpe ratio or return. [Lin et al 2004b]

For example, if we say the Sharpe ratio of “A01-MA” is 0.458, we cannot get the position of the stock-rule pair in the list. We cannot tell “A01-MA” is a good pair or bad. For another example, if we know two Sharpe ratios of pair “A01-MA”

and “A03-Filter rule” are 0.458 and 0.5 respectively. We cannot tell the performance of these two pairs. Are these two pairs both good? Or one is good, another is bad? Or both are bad? So the result is not clearly depending on the “numerical results” only.

To overcome the numerical problem, we try to use some “literal words” instead of “numerical values” to output the result of the pairs. For example “A03-Filter” rule is “good”. “A01-MA” is “medium”. So, it is easier to be accepted by traders and other users. Fuzzy set is the best tool to implement this function, so we adopt Fuzzy set method in this section to realize this function.

Firstly, we introduce some conception of Fuzzy set to make this thesis completed. [Allen et al 1993]

5.3.1 Fuzzy Set

Definition 5.1 (Fuzzy set, [Lotfi 1969]). Given an arbitrary set X , a fuzzy set (on X) is a function from X to the unit interval $I=[0,1]$,

$$\mu : X \rightarrow I \quad (5.1)$$

For convenience, we use the following notation to stand for a fuzzy set,

$$X = \{x_1, x_2, x_3, \dots, x_n\} \quad (5.2)$$

Then a fuzzy set μ on X has been denoted by the collection of pairs of the functional relation μ ,

$$\{(x_1, \mu(x_1)), (x_2, \mu(x_2)), (x_3, \mu(x_3)), \dots, (x_n, \mu(x_n))\} \quad (5.3)$$

Here, $\mu(x)$ is the membership degree of x . In our system, for each rule R , we build a fuzzy set for it, and the element of the set is the stocks (S), the membership degree is RS .

$$R = \{(s_1, SR(s_1)), (s_2, SR(s_2)), (s_3, SR(s_3)), \dots, (s_n, SR(s_n))\}$$

$SR(s)$ is the best Sharpe ratio of stock s under the rule R . Sometimes, $SR(s)$ is larger than 1 or less than 0, for this situation, we can use a linear transformation to transfer it into unit interval $[0, 1]$.

So, we can weigh all the stock-rule pairs with Definition 5.1.

Algorithm 5.2 The algorithm to classify a stock-rule pair (membership function).

Step 1: Select a stock S and a rule R . We divided the historical data into two sets: in-sample set and out-of-sample set. In this experiment, the in-sample set and out-of-sample set are one-year data and continued one-month order book trading data, respectively;

Step 2: On the in-sample set, we compute the best SR with RGA, and keep the parameter values for out-of-sample set computing usage;

Step 3: On the out-of-sample set, we compute SR with the parameters we learn from the in-sample set;

Step 4: SR is the membership degree of the stock S associated to the rule R .

Step 5: Insert S in the set R with the weight SR if it is higher than a threshold. (The threshold can be changed by user or other criterion.)

Through algorithm 5.2, if a stock-rule set has a higher weight (Sharpe ratio), it has a better performance (more profit and lower risk). On the contrary, the weight is lower.

5.3.2 Output Literal Results

After we get all stock-rule pairs membership weights, we can classify them into some subsets which include: "Very good", "Good", "Medium", "Bad", "Very Bad" and so on. The membership weight is consistent with the performance, because the membership function is defined by Algorithm 5.2.

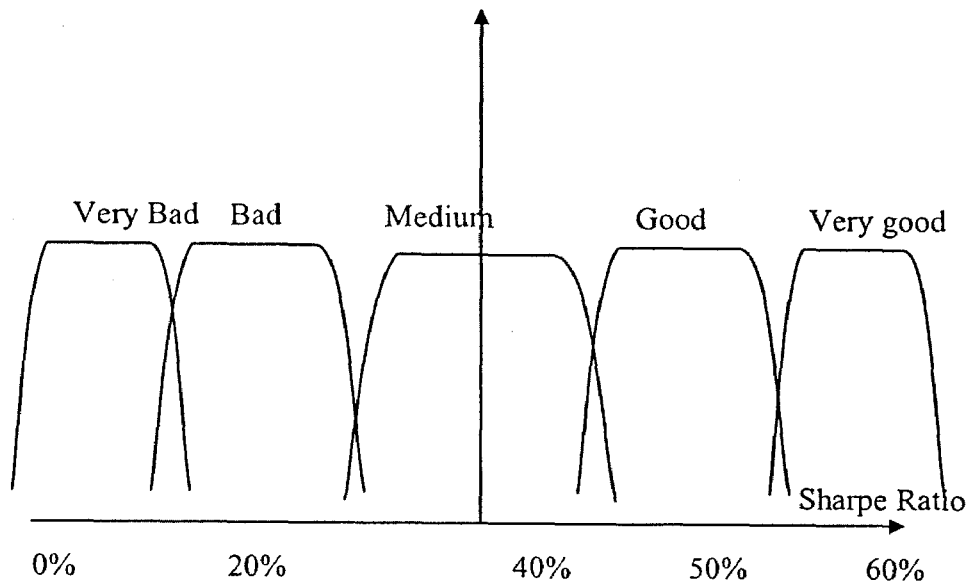


Figure 5.9 Fuzzy set definition.

From Table 5.1 and Figure 5.10, it is easy to know the different stock-rule pairs can get the different results. So there are not the best stocks or rules at all time, just the best stock-rule pairs for making more profit with less risk.

Table 5.1 The comparison of Sharpe ratio (SR) in out-of-sample sets of Moving Average, Filter rule and Channel Break-out and the literal output (classified by one rule respectively).

Stocks	Rules (Sharpe ratio)		
	Moving Average	Filter rule	Channel Break-out(Literal Result)
A01	0.458(Very bad)	0.323(Bad)	0.957(Medium)
A02	0.588(Bad)	0.347(Medium)	1.066(Good)
A03	0.411(Very bad)	0.5(Good)	0.817(Medium)
A04	0.516(Bad)	0.349(Medium)	1.104(Good)

A05	0.804(Good)	0.706(Good)	1.353(Good)
B06	0.529(Bad)	0.251(Very bad)	0.846(Medium)
B07	1.229(Very good)	0.395(Medium)	2.053(Very good)
C09	0.715(Medium)	0.322(Bad)	0.984(Medium)
C10	0.476(Very bad)	0.557(Good)	0.334(Very bad)
C11	0.475(Very bad)	0.319(Bad)	0.784(Bad)
F12	0.669(Medium)	0.446(Medium)	1.007(Good)
F13	0.91(Good)	0.296(Very bad)	1.103(Good)
G14	0.684(Medium)	0.395(Medium)	1.425(Good)
I15	1.223(Very good)	0.27(Very bad)	0.955(Medium)
J16	0.542(Bad)	0.528(Good)	1.097(Good)
M17	0.654(Medium)	1.098(Very good)	0.738(Bad)
M18	0.446(Very bad)	0.596(Good)	0.54(Very bad)
M19	0.416(Very bad)	0.426(Medium)	1.26(Good)
O21	1.351(Very good)	0.174(Very bad)	1.836(Very good)
P22	0.61(Medium)	0.394(Medium)	0.625(Very bad)
Q23	0.732(Medium)	0.325(Bad)	1.477(Good)
Q24	0.696(Medium)	0.308(Bad)	0.776(Bad)
S25	0.82(Good)	0.537(Good)	1.833(Very good)

S26	0.416(Very bad)	0.374(Medium)	0.619(Very bad)
T28	0.69(Medium)	1.203(Very good)	0.844(Medium)
T29	0.55(Bad)	0.575(Good)	0.672(Very bad)
W30	0.456(Very bad)	0.352(Medium)	1.122(Good)
W31	0.727(Medium)	0.352(Medium)	1.19(Good)
W32	0.428(Very bad)	0.374(Medium)	0.511(Very bad)
W33	0.473(Very bad)	0.314(Bad)	0.772(Bad)

In Table 5.1 and Figure 5.10, the in-sample set is one year data (from 01 January 2000 to 31 December 2000), and the out-of-sample set is one-month (from 01 January 2001 to 31 January 2001) just next to the in-sample set.

Literal Results

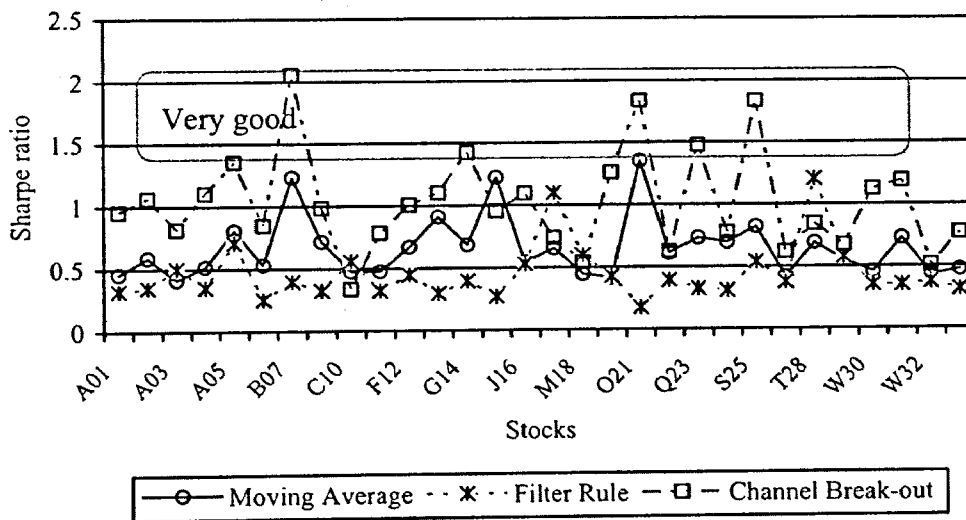


Figure 5.10 The comparison of Sharpe ratio of Moving average, Filter rule and Channel break-out rule.

From Table 5.1, the elements of “Very good” set are B07-MA, I15-MA, O21-MA, M19-Filter, T29-Filter, B07-CB (Channel Break-out), O21-CB and S25-CB. (This sorted pairs are ranked in one rule. It can also be ranked by all rules together.)

So, when we give a suggestion to a trader, we can give them the literal evaluation, such as B07-MA is “very good”. W33-Filter is “Bad” because its Sharpe ratio is less than the average value. If we only tell the trader that the Sharpe ratio of B07-MA is 1.229, the traders cannot get any idea whether it is a good or bad pair because the traders do not know its position in all pair list.

5.3.3 Evaluation and Conclusions

Through comparing the fuzzy value, we can classify pairs into some subsets: “very bad”, “bad”, “medium”, “good” and “very good”. It can be changed to more detailed subsets.

The criteria are the following rules and can be extended by different systems:

- (1) If the Sharpe ratio is in the average value, it is “Medium”;
- (2) If the Sharpe ratio is larger than medium and distributes in 10%-20% in the positive value, it is “Good”;
- (3) If the Sharpe ratio is more than square of a good Sharpe ratio, it is “Very good”;
- (4) If the Sharpe ratio is less than medium 10% to 20%, it is “Bad”;
- (5) If the absolute of Sharpe ratio is more than the square of a Sharpe ratio of “Bad”, and it is less than the Sharpe ratio of “Bad”, it is “Very bad”.

We need to clarify that “Good” or “Bad” of a stock is relative to other stocks. Evaluation results can be changed in different situations. When most Sharpe ratio values are negative, the “Good” or “Very good” pairs may be also negative. The reason is the “Good” and “Very Good” pairs still beat the market index return while others lose more money, the “Good” one loses less money. For example, if

the others lost 10 per cent, the trader who only lose 5 per cent is still a better trader. [Lin et al 2004b]

5.4 Multiple Criteria

In the stock market, different investors have different strategies, i.e. the “bigger” one only considers the volume, but the “smaller” one pays more attention to the price moving, so the criteria of evaluating a stock also considers the different aspects. Further, the algorithms (SGA and RGA) also need to consider both of them when we try to optimize trading rules. For example, some users want to make more profit, but, others only want to buy as many volumes as possible. For the different targets, users can choose different optimal function and fitness function for their different target. For the first one, the users can choose the profit as a fitness function, but for the second one, the users can set volume as a fitness function. Also, parameters, time duration, stock name, technical trading rules, the sizes of investments, fitness (profit, return, Sharpe ratio, etc), and so on. All of these can be set by a domain expert. In the next section, an example is given.

In this section, we give a detailed introduction of multi-objective optimization. [Christos et al 1999]

5.4.1 Background

It is frequently useful to select not just a single feature subset in any application fields including stock markets. The main problems are considering the subsets with different trade-offs between performance and complexity (i.e. we may tolerate lower performance in a model that also requires less features). Since the GA is population based, it seems natural to look for a method that produces a diverse range of such feature sets in the final population. This also helps to mitigate the problem of premature convergence, to which GAs are prone. We therefore use a multi-objective GA, where there are two objectives: to minimize

the number of stock-rule pairs in the subset, and to maximize return. Other multi-objective approaches are not presented in this thesis.

5.4.2 Solution

A solution is said to be Pareto optimal [Horn et al 1994] if it cannot be dominated by any other solution available in the search space. The use of a multiple criteria algorithm based on the concept of dominance can maintain population diversity, in order to allow the algorithm to discover a range of feature sets with different performance versus complexity trade-offs. The multi-objective GA employed in this work can be described as a niched Pareto GA with random sampling tournament selection. The algorithm uses a specialized tournament selection approach, based on the concept of dominance [Horn et al 1994]. The selection procedure is as follows:

1. Individuals are randomly selected from the population to form a dominance tournament group.
2. A dominance tournament sampling set is formed by randomly selecting individuals from the population.
3. Each individual in the tournament group is checked for domination by the dominance sampling group (i.e. if dominated by at least one individual).
4. If all but one of the individuals in the tournament group is dominated by the dominance tournament sampling group, the non dominated one is copied and included in the mating pool.
5. If all individuals in the tournament group are dominated, or if at least two of them are non-dominated, the winner which best seems to maintain diversity is chosen by selecting the individual with the smallest niche count. The niche count for each individual is calculated by following a typical sharing technique:

$$s(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_s}\right)^{\alpha_i} & \text{if } d_{ij} < \sigma_s \\ 0 & \text{otherwise} \end{cases}$$

$$m_i = \sum_{j=1}^N s(d_{ij}) \quad (5.4)$$

where m_i is the niche count of the i -th individual in the tournament group, s is calculated by the Hamming distances d_{ij} of the above individual with each of the N individuals already present in the mating pool and σ_s is the Hamming distance threshold, below which two individuals are considered similar enough to affect the niche count.

6. If the mating pool is full end tournament selection; otherwise go back to step 1.

Using some simple bitwise functions, Horn [Horn et al 1994] reported that this dominance sampling tournament selection was superior to a simple dominance tournament where the winner was chosen by checking the dominance among the members of the tournament group [Horn et al 1994]. Using Horn's approach, the domination pressure can be controlled by appropriate choice of the size of the dominance tournament sampling set.

The major computational cost associated with the use of GAs for feature selection is in the evaluation of the feature subsets. This involves building and evaluating a fuzzy model using a given feature subset. In order to avoid the computational costs associated with the wrapper approach, one can use a simple form of model that can be evaluated more quickly during the feature selection stage. Here we follow a fuzzy classifier design method based on cluster estimation [Chiu 1996]. The main characteristics of this approach are:

1. An initial fuzzy classification model is derived by cluster estimation.
2. The fuzzy rule base contains a separate set of fuzzy rules for each class.

3. Double-sided Gaussian membership functions are employed for the premise parts of the fuzzy rules. These are more flexible than the typical Gaussian kernel.
4. The classification outcome is determined by the rule with the highest activation.
5. Training is performed by a hybrid learning algorithm, which combines gradient-based and heuristic adaptation of the membership functions parameters. Only the rules with the maximum activation per class are updated for each pattern.

5.4.3 Conclusion

We have introduced an approach to perform feature selection for classification tasks, based on multi-criteria genetic algorithms. The multi-criteria GA method is justified as a feature selection approach when the number of features becomes large enough to make exhaustive evaluation infeasible or stepwise methods computationally more expensive. An additional benefit of the multi-criteria GA approach is that it can yield a range of solutions with different accuracy/complexity trade offs. Such information is potentially of critical importance, since it can guide decisions related to data acquisition for performing classification. Even though we have experimented with fuzzy models, the technique can actually be used with other classification methods as well. Further work should examine ways of reducing the possibility of missing some of the Pareto optimal solutions.

Our multi-objective genetic algorithm feature selection offers also the potential of naturally handling additional objectives, which are often problem dependent, such as costs of misclassification and data acquisition costs. We have only done some part of work and it shows the advantages of the multi-objective genetic algorithms, but, the whole experiment work has not been done yet.

5.5 Summary

In this chapter, we introduced some optimal algorithms focused on making more profit or getting a higher Sharpe ratio. The optimal algorithms are: standard genetic algorithms; robust genetic algorithms; Fuzzy set and multi-objective criteria. These algorithms make the system applicable and the overall performance is better than other result, for example the execution time (SGA).

Firstly, we introduce standard genetic algorithm which can reduce execution time and keep profit as high as the enumerate method at the same time. Genetic algorithms are adapted to overcome the execution time problem and keep the better performance. Our result also shows the comparison of execution time and performance (profit and return). It is the basic method for the further applications and makes many applications possible and practicable.

Secondly, in some cases, the maximal mathematical result is not a true result of finance. The reason is SGA does not consider domain knowledge. For example, in one year, it only generated two signals, but the result is the best in mathematics. So, we combine genetic algorithms with domain knowledge together in order to filter noisy signals. This algorithm is called robust genetic algorithm which can remove noisy signals and keep all the advantages of standard genetic algorithms. It also keeps the high performance and high execution efficiency.

Thirdly, fuzzy sets help us to rank the stocks and rules, so users can select the better stocks and rules easily. The Sharpe ratio, return and profit result is only a number, and it is relatively to others when we want to know whether it is better or worse. For example, if a stock-rule pair monthly return is 0.2%, we cannot determine it is better or not, because we do not know the other pairs performance. If all other pair monthly return is more than 0.3%, this pair is not a good one, but, if all other pair monthly return is negative, this pair is the best one. In order to give user a clarified result and output, we transfer the result into literal output,

such as “good”, “very good” and so on, so users can avoid the hesitation mentioned in the above example.

Lastly, we also try some research and applications for multi-objective genetic algorithm, such as it can divide the set by volume and price changing frequency. Sometimes, only one criterion is not enough in stock market data mining, because there are many different features, such as price; volume and liquidity. So, we need consider it together to select the best stocks and rules. One of the best solutions is adopting the multi-objective genetic algorithm. It has more than one fitness functions, and looks for the synthesized better parameters considering all the features: price; volume and liquidity, etc.

In this chapter, we give some optimal methods as the basic tools to implement the further data mining tasks.

Chapter 6 Applications

We have implemented some applications in this thesis. The applications are standard genetic algorithm (SGA) and robust genetic algorithm (RGA) in finding the best trading alert signals and combination of parameters; finding sub-domain of parameters for trading rules, ranking stock-rule pair list and finding the best size of investment for a number of stocks. We introduced them in the following sections respectively.

6.1 Optimal Parameter Combination

In stock market decision making systems, generally, data is very huge. The execution takes a lot of time if we test all possible combinations, hence the optimal algorithm is necessary and urgent. One of the optimal algorithms is genetic algorithm (GA), which is based on the evolutionary theory to find near-optimal solution from a huge solution set. [Robert 1999] [Section 5.1]

We have implemented several applications based on GA. Firstly; we find trading alert signals in order to get the most profit or the highest return. [Lin et al 2004a]

In stock market and other finance fields, GA has been applied to solve many problems. In quite a number of attempts GA has been used to acquire technical trading rules, both for Foreign Exchange Trading and for S&P500 market. One application is how to find the best combination values of each parameter. We know that there are many parameters in a trading rule. When we try to find the most profit, we must test parameter combinations one by one which is called enumerate algorithm.

Through analyzing stock markets, we get to know there are some combinations of parameters, which can produce a near-max profit and give some reasonable buy/sell suggestions. So our objective in this chapter is to find one of these near-max profit combinations efficiently.

First of all, we decide a sub-domain of each parameter with GA, then, in each sub-domain, we try to find a near optimal combination for each stock historical data.

After we find a sub-domain, sometimes we need to get only one value for our decision. Even from a sub-domain, we may spend 30 minutes computing the most profit value with enumerate algorithm, so we need to reduce the execution time. One applicable method is GA. See Figure 5.2.

6.2 Optimized Sub-Domain

After the robust results are found, we remove the single peak points by adding a soft filter onto SGA algorithms (see Algorithm 6.1). For each point, we compute the values of its neighborhood points. If the values are far from the central point, or negative, we discard them. While we are finding the most fitness one, we also consider its neighborhood points. If there is a small range, in which all the value can make more profit, it is “robust”.

Algorithm 6.1 Finding the best sub-domain algorithm. [Lin 2005c]

Step 1. For every parameter, set an initial size s_1 , and step t for the first parameter sub-domain;

Step 2. Computing the Sharpe ratio with GA in every sub-domain combination;

Step 3. If there is the best robust sub-domain, in which all the values are positive and better than in the others, then output the sub-domain and finish the algorithm; else execute step 4.

Step 4. Reset another sub-domain size $s_2 = \frac{1}{2}s_1$, repeat steps 2 and 3.

If $s_2 = t$, (in every sub-domain, there is the least size, generally only one value.) the algorithm becomes the standard genetic algorithm.

We can use the Algorithm 5.2 to find the robust optimized point for trading rules.

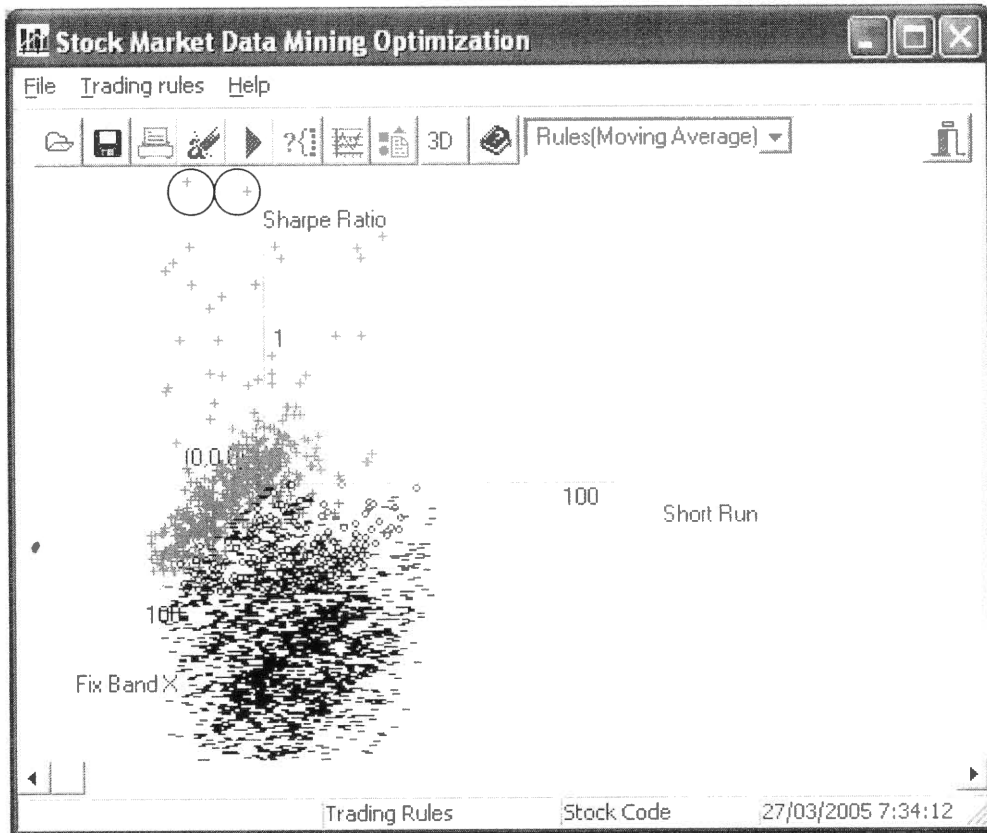


Figure 6.1 The result before optimization.

In Figure 6.1, the best points are “randomly distributed”, so we can not find a better range. (“+”: the positive value; “o”: zeroed value; “-”: negative value.)

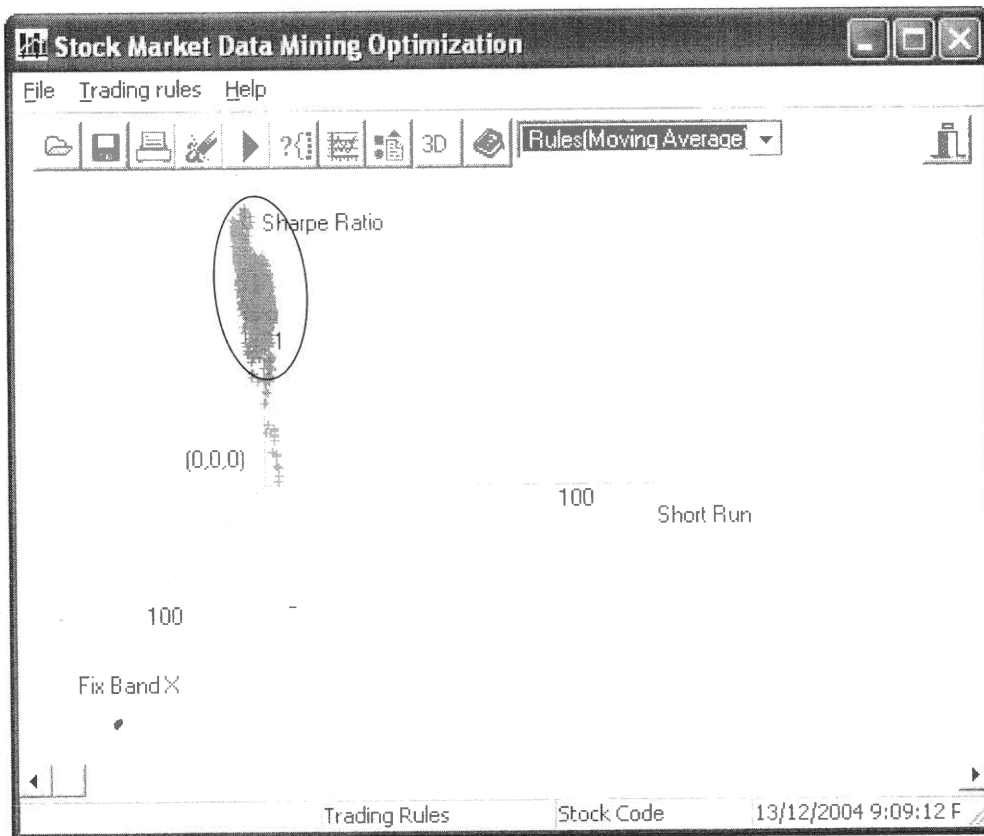


Figure 6.2 In the optimized sub-domain, the results of any parameter combination are positive.

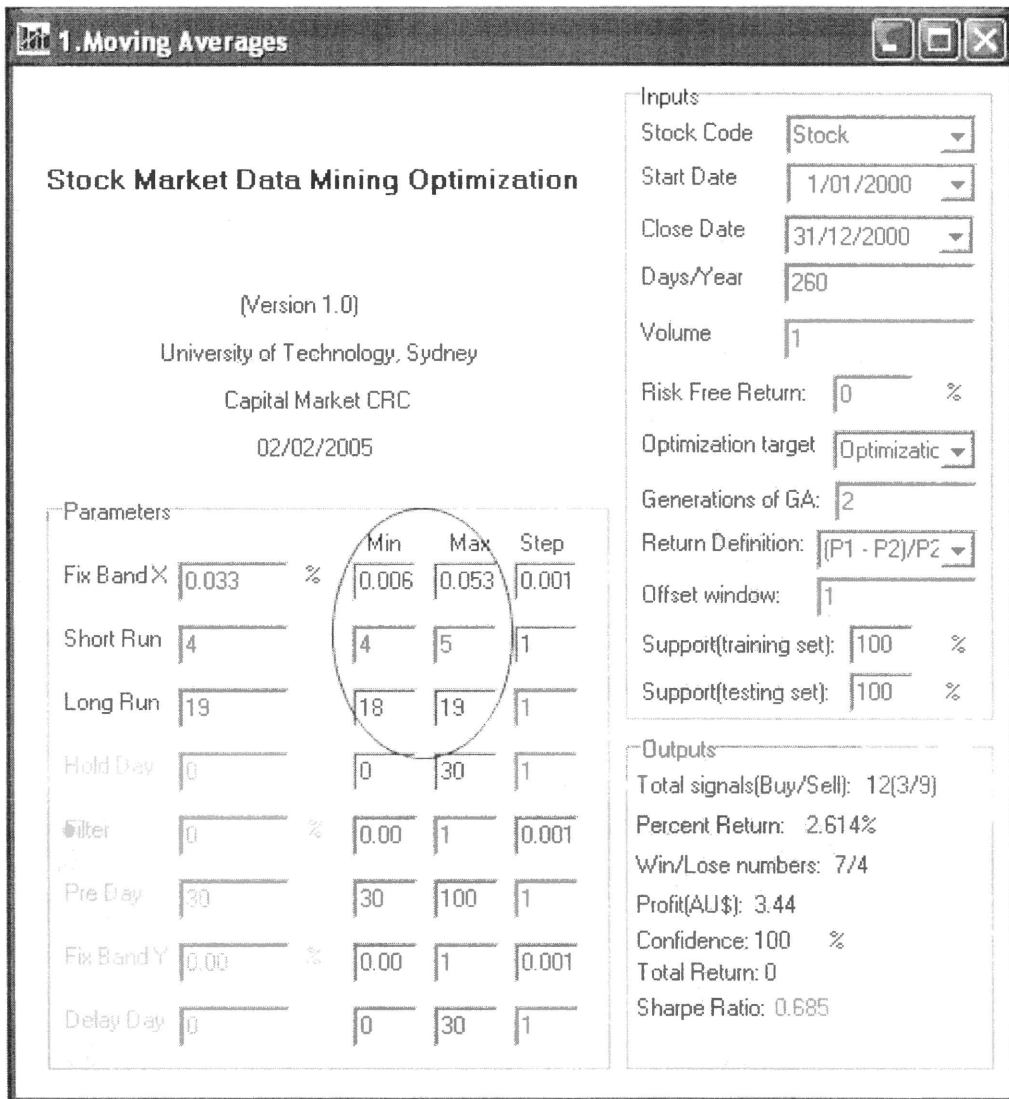


Figure 6.3 The optimized sub-domain with RGA.

6.3 Stock-Rule Pairs

For a single stock and a single rule, we can easily find the “best” robust results with RGA and domain knowledge. However, if there are many stocks and many rules, the method to select a stock-rule pair with a higher profit and lower risk is difficult.

6.3.1 Distribution of In-sample Set and Out-of-Sample Set

To simulate real stock markets, we divide historical data into two parts: in-sample data (training set) and out-of-sample data (testing set). Both sets are one month intraday order book data.

In the in-sample set, we sort stock-rule pairs by Sharpe ratio decreased. We keep the same parameter values which are obtained from in-sample set and use them in out-of-sample set so that the method is able to predict in the future trading suggestion. So, two hypotheses need to be overcome in order to guarantee this method is applicable and predictable. One is that the distribution patterns of in-sample and out-of-sample set should be similar. Another is that better pairs in-sample should be still better out-of-sample as well. To solve the first hypothesis, we draw the two graphs for both in-sample set and out-of-sample set and the results are almost the same in every month (see Figure 6.5). That means the pattern in-sample set and out-of-sample set are similar.

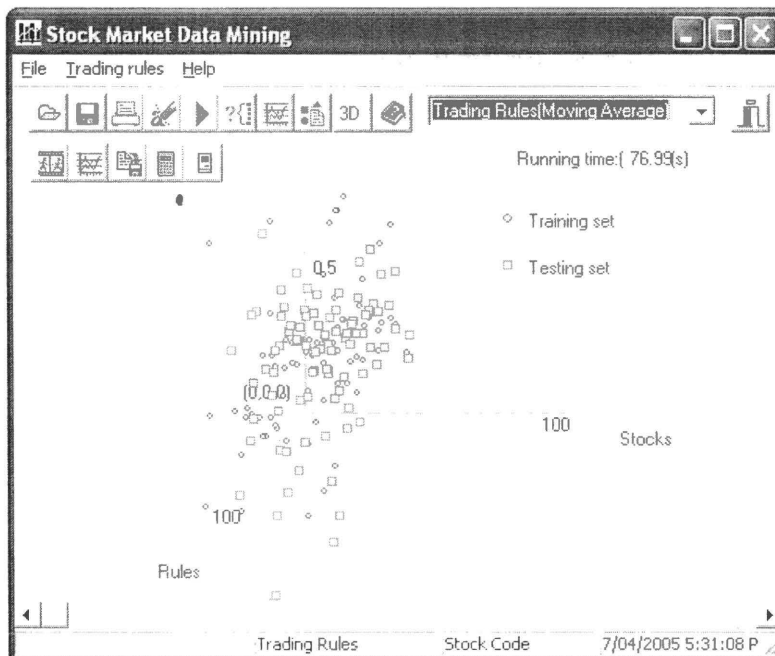


Figure 6.4 The comparison of in-sample (Training set) result and out-of-sample (testing set) result.

The next step is to prove the better pairs in-sample are still better out-of-sample.

6.3.2. Support and Confidence

The second hypothesis is support and confidence. The confidence is the percentage of the pairs, which are better in both in-sample set and out-of-sample set. The support is also a percentage which is the number of top pairs in out-of-sample set.

We choose the best top pairs from in-sample set and check whether the pairs are still better in out-of-sample set. The result is shown in Table 6.1 which proves that the better pairs in-sample set are still better out-of-sample set (see Figures 6.6 and 6.7). The confidence is about 80 per cent that means about 80 per cent better pairs from in-sample set are still better (profitable) in out-of-sample set (see Table 6.1).

Table 6.1 The percentage of confidence in out-of-sample set. The support is 30% in the out-of-sample set.

Top percent	July 2001	August 2001	September 2001
1%	78.12	100	100
2%	78.12	78.12	100
3%	78.12	78.12	100
4%	97.65	97.65	100
5%	93.75	93.75	100
6%	91.14	91.14	100
7%	78.12	89.28	89.28
8%	87.89	87.89	87.89
9%	78.12	86.80	86.80
10%	78.12	78.12	85.93
11%	85.22	78.12	78.12
12%	84.63	78.12	71.61

We have applied the in-depth data mining technology in the ASX data, and the results it gives show that the pairs rule exists in the trading market. When we get the best pair in-sample set (in-sample data), it also exists out-of-sample set (out-of-sample data). The confidence is more than 80%. (See Table 6.1)

```

Transaction cost 0.25 percent - Notepad
File Edit Format View Help
Training set(offset = 1, Investment = 1000.00($), In-sample date June 2001)
=====
0: Best Sharpe Ratio( 0.71), Rule Code( 2), Stock Code( 24), Best Return( 0.25)
1: Best Sharpe Ratio( 0.66), Rule Code( 2), Stock Code( 26), Best Return( 0.17)
2: Best Sharpe Ratio( 0.61), Rule Code( 1), Stock Code( 24), Best Return( 0.25)
3: Best Sharpe Ratio( 0.56), Rule Code( 1), Stock Code( 14), Best Return( 0.07)
4: Best Sharpe Ratio( 0.52), Rule Code( 3), Stock Code( 24), Best Return( 0.20)
5: Best Sharpe Ratio( 0.49), Rule Code( 1), Stock Code( 26), Best Return( 0.08)
6: Best Sharpe Ratio( 0.49), Rule Code( 1), Stock Code( 18), Best Return( 0.09)
7: Best Sharpe Ratio( 0.48), Rule Code( 3), Stock Code( 21), Best Return( 0.12)
8: Best Sharpe Ratio( 0.46), Rule Code( 2), Stock Code( 21), Best Return( 0.13)
9: Best Sharpe Ratio( 0.46), Rule Code( 3), Stock Code( 14), Best Return( 0.09)
10: Best Sharpe Ratio( 0.46), Rule Code( 1), Stock Code( 19), Best Return( 0.07)
11: Best Sharpe Ratio( 0.44), Rule Code( 1), Stock Code( 8), Best Return( 0.07)
12: Best Sharpe Ratio( 0.43), Rule Code( 1), Stock Code( 16), Best Return( 0.17)
13: Best Sharpe Ratio( 0.43), Rule Code( 3), Stock Code( 4), Best Return( 0.06)
14: Best Sharpe Ratio( 0.42), Rule Code( 2), Stock Code( 14), Best Return( 0.07)
15: Best Sharpe Ratio( 0.37), Rule Code( 2), Stock Code( 19), Best Return( 0.06)
16: Best Sharpe Ratio( 0.37), Rule Code( 2), Stock Code( 18), Best Return( 0.16)
17: Best Sharpe Ratio( 0.35), Rule Code( 3), Stock Code( 9), Best Return( 0.05)

```

Figure 6.5 The optimal pairs result from in-sample data (partly).

```

Transaction cost 0.25 percent - Notepad
File Edit Format View Help
Testing set(offset = 1, Investment = 1000.00($), out-of-Sample date July 2001)
=====
0: Best Sharpe Ratio( 0.49), Rule Code( 1), Stock Code( 14), Best Return( 0.07)
1: Best Sharpe Ratio( 0.46), Rule Code( 3), Stock Code( 13), Best Return( 0.08)
2: Best Sharpe Ratio( 0.42), Rule Code( 2), Stock Code( 14), Best Return( 0.08)
3: Best Sharpe Ratio( 0.42), Rule Code( 2), Stock Code( 26), Best Return( 0.06)
4: Best Sharpe Ratio( 0.42), Rule Code( 3), Stock Code( 17), Best Return( 0.11)
5: Best Sharpe Ratio( 0.40), Rule Code( 1), Stock Code( 13), Best Return( 0.09)
6: Best Sharpe Ratio( 0.37), Rule Code( 2), Stock Code( 24), Best Return( 0.12)
7: Best Sharpe Ratio( 0.37), Rule Code( 1), Stock Code( 17), Best Return( 0.11)
8: Best Sharpe Ratio( 0.36), Rule Code( 3), Stock Code( 26), Best Return( 0.05)
9: Best Sharpe Ratio( 0.33), Rule Code( 3), Stock Code( 16), Best Return( 0.07)
10: Best Sharpe Ratio( 0.31), Rule Code( 1), Stock Code( 26), Best Return( 0.06)
11: Best Sharpe Ratio( 0.30), Rule Code( 1), Stock Code( 11), Best Return( 0.07)
12: Best Sharpe Ratio( 0.30), Rule Code( 1), Stock Code( 1), Best Return( 0.08)
13: Best Sharpe Ratio( 0.29), Rule Code( 3), Stock Code( 14), Best Return( 0.05)
14: Best Sharpe Ratio( 0.29), Rule Code( 3), Stock Code( 9), Best Return( 0.08)
15: Best Sharpe Ratio( 0.28), Rule Code( 3), Stock Code( 24), Best Return( 0.73)
16: Best Sharpe Ratio( 0.25), Rule Code( 1), Stock Code( 23), Best Return( 0.12)
17: Best Sharpe Ratio( 0.24), Rule Code( 1), Stock Code( 22), Best Return( 0.14)

```

Figure 6.6 The optimal pairs result from out-of-sample (partly).

Figures 6.5 and 6.6 show the sorted pairs in both in-sample set and out-of-sample set respectively. From the figures, we can also find the in-sample better pairs are still better in out-of-sample set. Such as the top three better in-sample pairs are 2-24, 2-26 and 1-24. Their positions in out-of-sample set are 6, 3 and 1. Figures 6.5 and 6.6 are the matrix of stock-rule pair performance. It equals to a two-dimension table, one dimension is stock and the other is rule. The value is the best performance for each stock-rule pair (under the possible best parameter combination).

In the next sections, all experiments are done following the same rules: the parameter values are gotten from in-sample set which is one month length. The results including Sharpe ratio, return and profit are gotten from out-of-sample set so that the method can be used in prediction.

To simulate real stock markets, we divide historical data into two parts: in-sample data and out-of-sample data. In the first data, we use trading rules and stocks to sort the pairs by Sharpe ratio decreased. In the second data, we find all best signals for each pair. When a user fixes the amount to invest, such as AUD 1K (=AUD 1000), AUD10K, AUD 100K, AUD 1M (=AUD 1000000) or AUD 10M, etc, we give the user the best return for how many pairs the user need to trade. At the same time, we also give the user the stock-rule pairs and the alert trading signals.

6.4 Relationship between Investment and the Number of Pairs

6.4.1 Profit, Return, Top Percentage Pairs and the Size of Investment

In some experiments or markets, we do not know the transaction cost or the transaction cost changed from time to time. There are also many different kinds of transaction costs for different investors. For example, the brokers only pay 0.1 per cent transaction cost, but the dealers pay 0.25 per cent transaction cost in ASX. So, we give one result without transaction cost (see Figures 6.7 and 6.8)

and another result with transaction cost (0.25% per transaction value, see Figures 6.9, 6.10 and 6.11) under the same conditions (Jan 2001 is in-sample data and Feb 2001 is out-of-sample data. There are the same 27 stocks and 3 trading rules, totally 81 stock-rule pairs)

Figures 6.7 and 6.8 display the result of monthly return and profit without transaction. It shows the five relationships listed in the follow paragraphs.

(1) The average return per share is higher when the number of pairs is less, because the pairs are sorted from higher return to lower return. So, when we choose top-percentage pairs, their average performance is better than the average performance of all pairs. (In Figures 6.7, 6.9, 6.15, 6.16 and 6.17, the return is average return per share. That is the total return divided by the number of shares. The total return is the whole profit divided by the investment, so the figure of total return is same as that of the profit except times the number of investment. So, in this thesis, we do not give the whole return graph.)

(2) When investment is more, return is less. The reason is when we invest fewer amounts, the most money is used to purchase better stocks so return is higher. However, when we invest more money, we have to purchase more stocks even if some of the stocks are not as good as the top percentage stocks.

(3) The monthly profit is higher when the number of pairs is more (see Figure 6.8), but, it does not go upwards as rapidly as the investment. That proves when we trade more stocks we can get more profit. However, the profit is not growing as quickly as the investment. The reason is much more “no-good” stocks are traded.

(4) The monthly profit increases with the investment increases. The profit is more when the investment is more although the return is lower. That clarifies when the investment is less, it can only buy or sell the better signals but lose the “good” or “no-good” signals. However, the “good” signals can also make a little bit profit. So, if we want to get more profit, we shall invest more money and

select more stocks. However, if we want to get a higher return, we should only invest less money and select less top-percentage stocks. The result is consistent with real markets and other experiments also show the same conclusion when we consider the transaction cost.

(5) When the investment is more than a threshold, the profit and return do not increase any more. Because the stocks and signals are limited, if the investment is more than the market liquidity, the excess money cannot be traded. This is the reason why when the investment is very “big”, but the profit and return keep steady. See Figure 6.10.

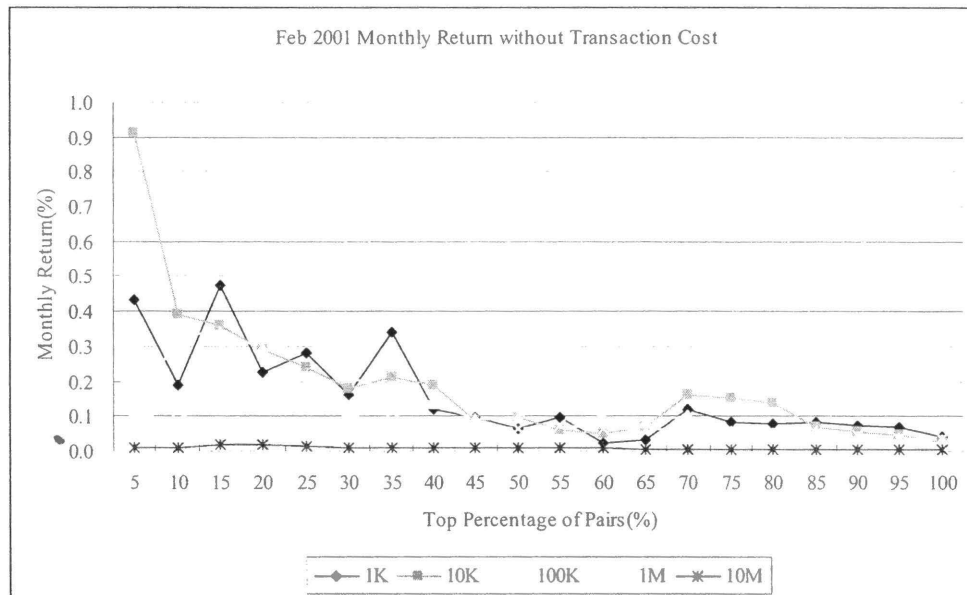


Figure 6.7 Feb 2001 monthly average return to the top percentage pairs without transaction cost. (In-sample data is Jan 2001, and the out-of-sample data is Feb 2001.)

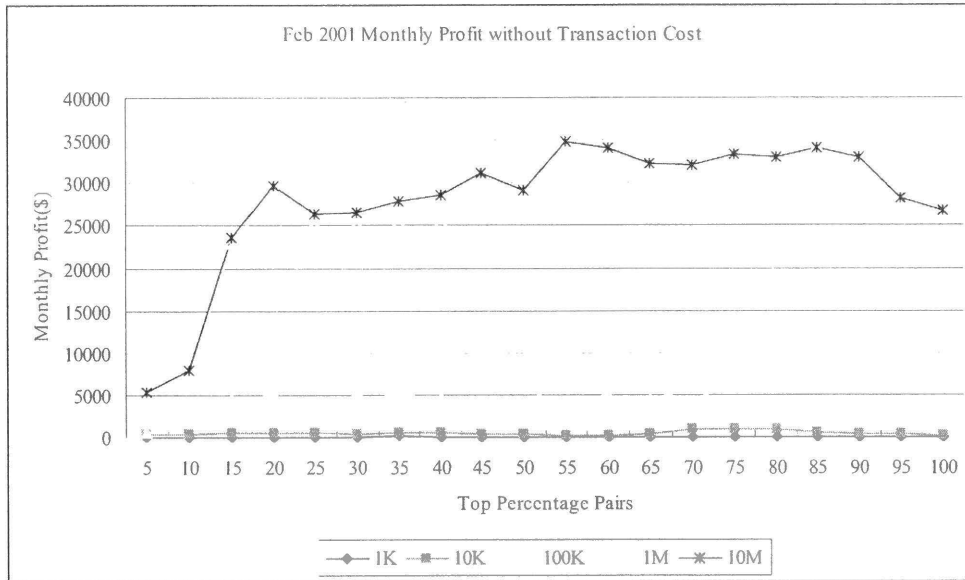


Figure 6.8 Feb 2001 Monthly profit without transaction cost.

Figures 6.9, 6.10 and 6.11 draw the monthly return and profit with transaction cost.

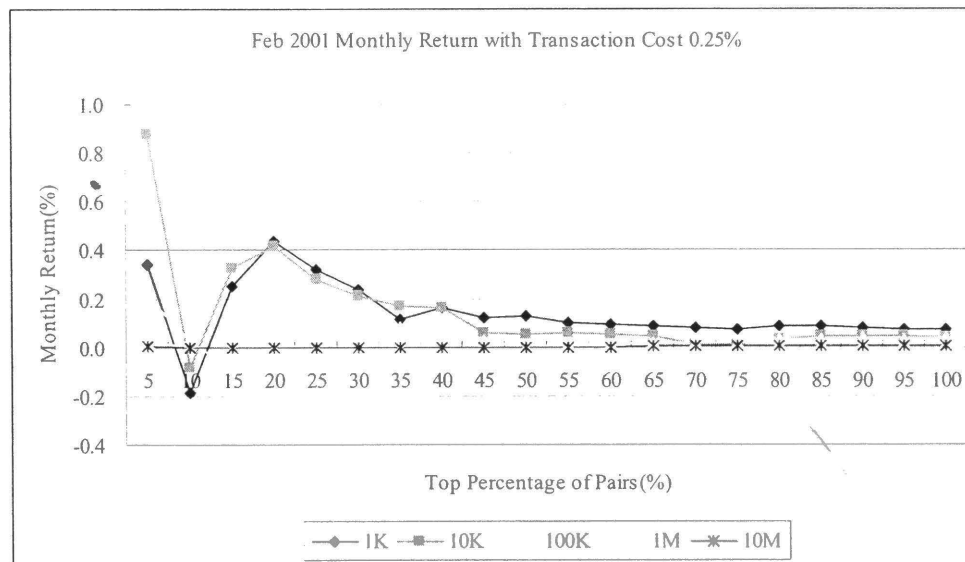


Figure 6.9 The monthly average return to the per cent top pairs with transaction costs. (The transaction cost is 0.25% of transaction value).

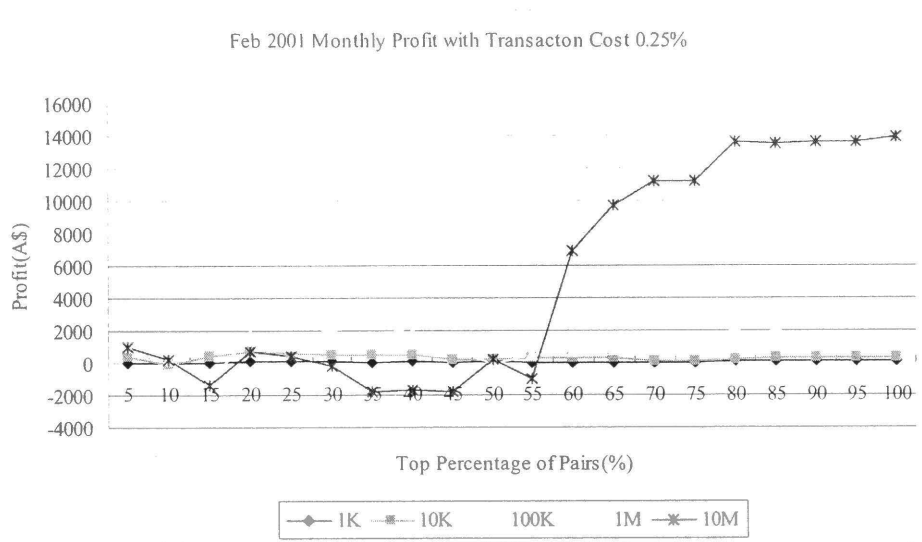


Figure 6.10 Feb 2001 monthly profit with transaction cost 0.25%. (The profit of 1M and 10M are completed same)

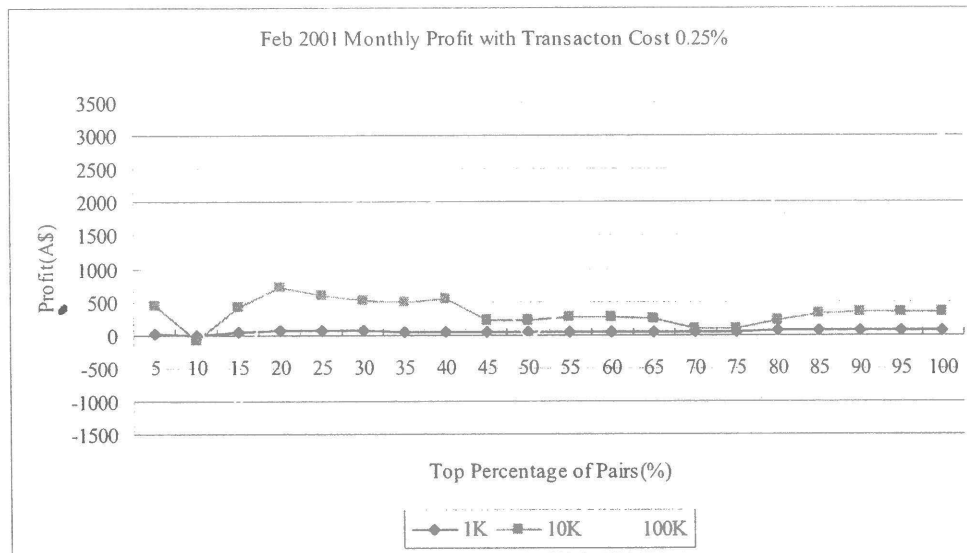


Figure 6.11 Feb 2001 monthly profit with transaction cost 0.25%.

However, in the real market, sometimes, we must consider the transaction cost, so we also draw other graphs with 0.25% transaction cost in Figures 6.9, 6.10 and 6.11. We have done 2001 and 2002 two years experiments with 0.25% transaction cost. The return and profit reduce a little, but the trends are the same

as without transaction cost. It also matches the results without transaction cost and better than market index return. However, it is more reasonable and convincing when we consider transaction cost.

In the following section, we discuss the relationship between the sizes of investments and the number of pairs in order to make an optimal combination of investment and pairs to get a higher return.

6.4.2 Investment and the Number of Pairs

When a trader wants to invest, the trader must consider the investment and the amount of the shares. When a trader invests a great deal of money, the trader should trade more shares to reduce the dead money and make a higher return. However, when a trader invests a little money (such as A\$10K) the trader should only concern the little top per cent pairs. In this section, we show some experiments to find the relationship between the sizes of investments and the number of pairs.

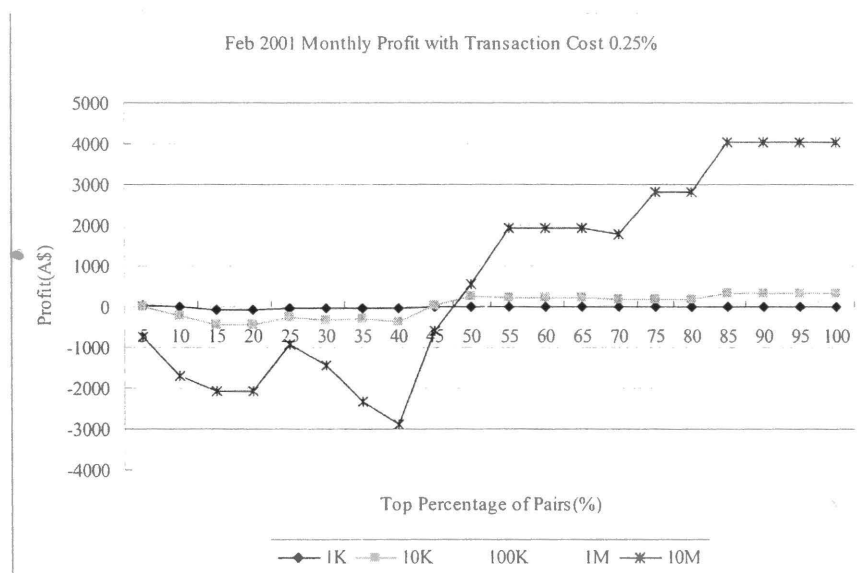


Figure 6.12 Feb 2001 monthly profit with transaction cost 0.25%. The number of stocks is 27 and the number of rule is 1. The number of stock-rule pairs is totally 27.

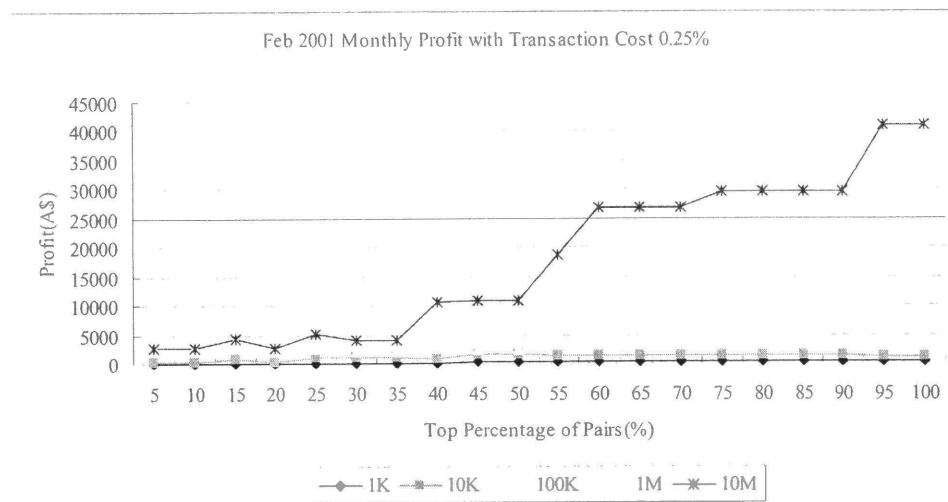


Figure 6.13 Feb 2001 monthly profit with transaction cost 0.25%. The number of stocks is 9 and the number of rules is 3. The number of stock-rule pairs is totally 27.

Figures 6.12 and 6.13 show the profit for different investments. (The return to investment is the same result. Here, we choose profit because the profit is larger and clearer than return.). When the number of pair is less, the big investment has not more profit, because the number of pairs (signals) is not enough to consume all investment, especially the number of pair is less. From Figures 6.12 and 6.13, the line of 10M is exactly the same as the line of 1M, and part of line 1M is also the same as the line of 100K.

6.4.3 Stock-Rule Pair Return and Market Index Return

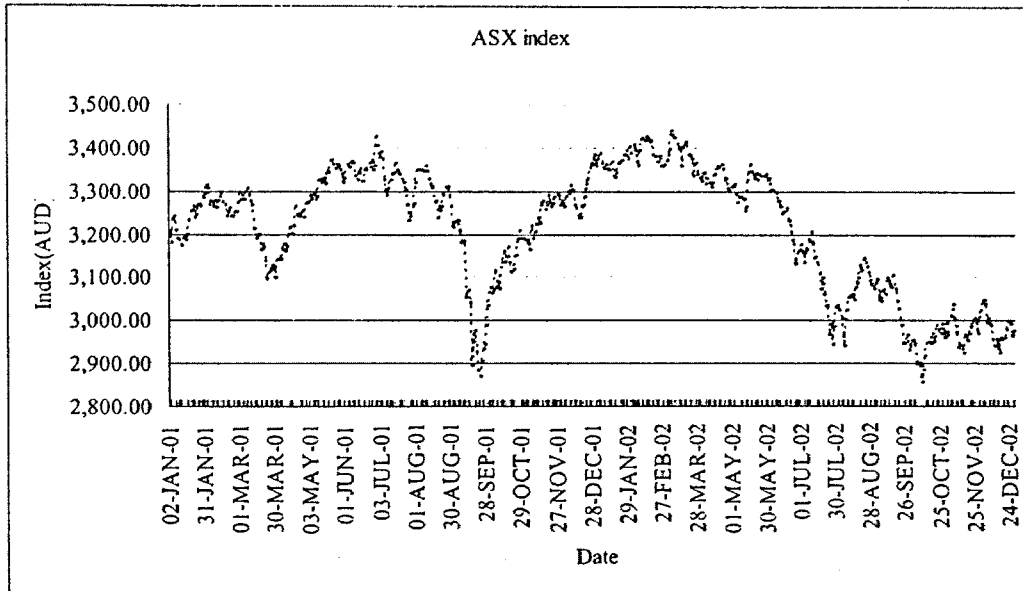


Figure 6.14 ASX market index from 1 January 2001 to 31 December 2002.

The market index changes all the time, so, to evaluate whether some strategies are better or worse, we should consider not only its return and profit but also the market index return. That means, when the market index return is positive, the better strategies should be to make a higher return than the market return, not only positive. When the market index gets a negative return, the strategies are also better if the recommended pairs lose less money than market index. This criterion is called “beat the market”.

Figure 6.14 is the ASX index from 1 January 2001 to 31 December 2002.

Comparison of Index Return and Pair Return

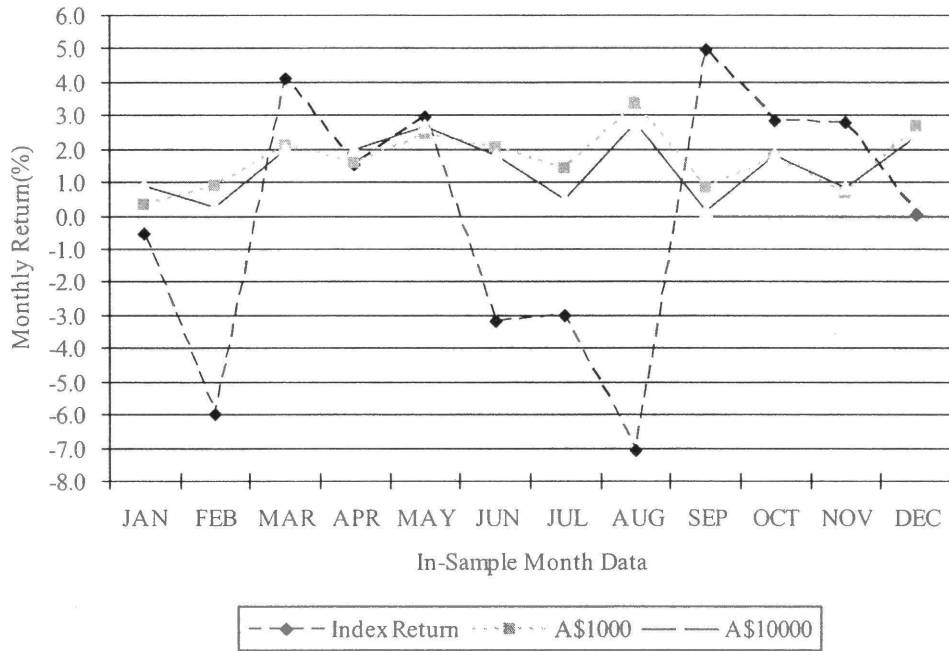


Figure 6.15 The comparison of out-of-Sample monthly return and market monthly return (with transaction cost 0.25 %, year 2001, considering the top 5% pairs only. Out-of-sample month is one month after in-sample month).

Figure 6.15 and Table 6.2 show the monthly return on stock-rule pair and market index monthly return. In the total 12 months out-of-sample data of year 2001, 7 monthly returns of the stock-rule pair strategy (investment is 1K) are better than index return. Only 5 monthly returns are less than index return. In 5 months, the index return is minus, but, for stock-rule pair strategy, there is only one month (Sep 2001) with negative return. The average monthly return of stock-rule pair strategy is better than index return. See Table 6.3.

Table 6.2 Comparison of market index return and monthly returns for different investments. (Transaction cost is 0.25%. See Figure 6.15)

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Index	-0.516	-6.017	4.106	1.499	2.945	-3.168	-3.014	-7.089	4.976	2.857	2.815	0.005
A\$1K	0.341	0.916	2.092	1.572	2.46	2.064	1.424	3.38	-1.225	1.766	0.688	2.658
A\$10K	0.879	0.242	2.01	2.009	2.655	1.792	0.507	2.787	-0.225	1.833	0.846	2.365
A\$100K	0.208	0.292	1.347	1.316	0.857	0.711	1.338	0.67	0.511	0.966	0.299	0.689
A\$1M	0.021	0.015	0.139	0.208	0.086	0.095	0.161	0.022	0.045	0.15	0.03	0.061
A\$10M	0.002	0.002	0.014	0.021	0.009	0.01	0.016	0.002	0.005	0.015	0.003	0.006

Table 6.3 Annual return and average monthly return.

	Annual return (%)	Average monthly return (%)
Index	2.747	0.228
A\$1K	18.136	1.511
A\$10K	17.7	1.457
A\$100K	9.204	0.767
A\$1M	1.033	0.086
A\$10M	0.105	0.008

Table 6.2 displays the result in year 2001. In-sample data is one month and out-of-sample data is consecutively one month (In the figure, the month is in-sample data, the result is calculated in out-of-sample data). The return is derived from out-of-sample data whose trading rule parameter is calculated in-sample data. In Table 6.2, the month is in-sample data. Table 6.2, Table 6.3 and Figure 6.15 show that seven monthly returns of stock-rule pair strategy are better than that of market index return. 11 out-of 12 returns of stock-rule pair strategy are positive, but the market index return has 5 negative results. The annual return (1 February 2001 to 31 January 2002) and the average return of stock-rule pair strategy are better than the market index return. The annual return and monthly return are

18.136%, 17.7%, 9.204% and 1.511%, 1.457%, 0.767% when the investments are A\$1K, A\$10K and A\$100K respectively. The three returns are better than market return. The annual return and average market index return is 2.747% and 0.228% (see Table 6.3). For the result of 1M and 10M, the investment is much more than the shares and signals so there is much “dead money” (never used to buy or sell) in hand (see Section 6.4.2), so the return is relatively lower.

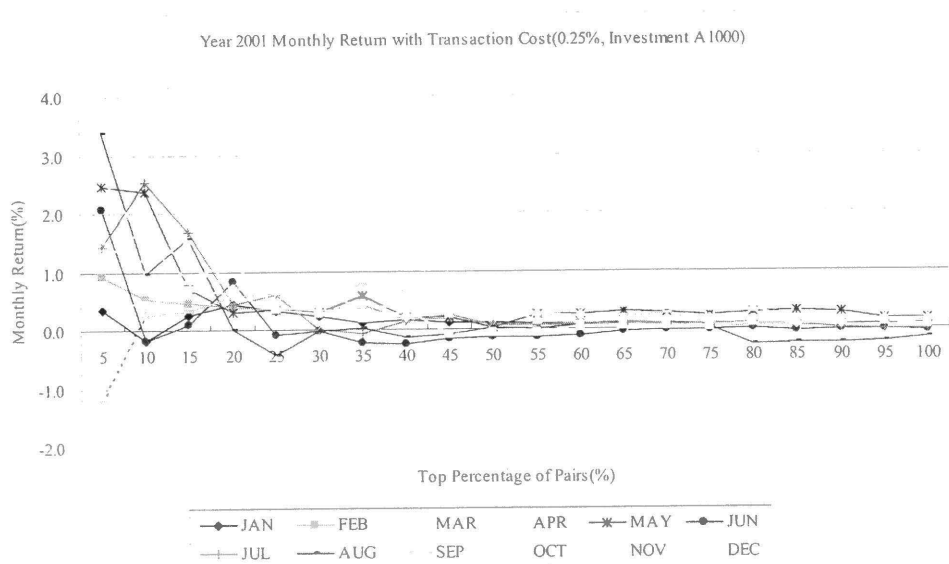


Figure 6.16 Year 2001 monthly average return with transaction cost 0.25%.

Figures 6.16 and 6.17 are year 2001 and 2002 monthly average return. Comparing to the market index graph (see Figure 6.14), we can draw the relationship between monthly return and index value.

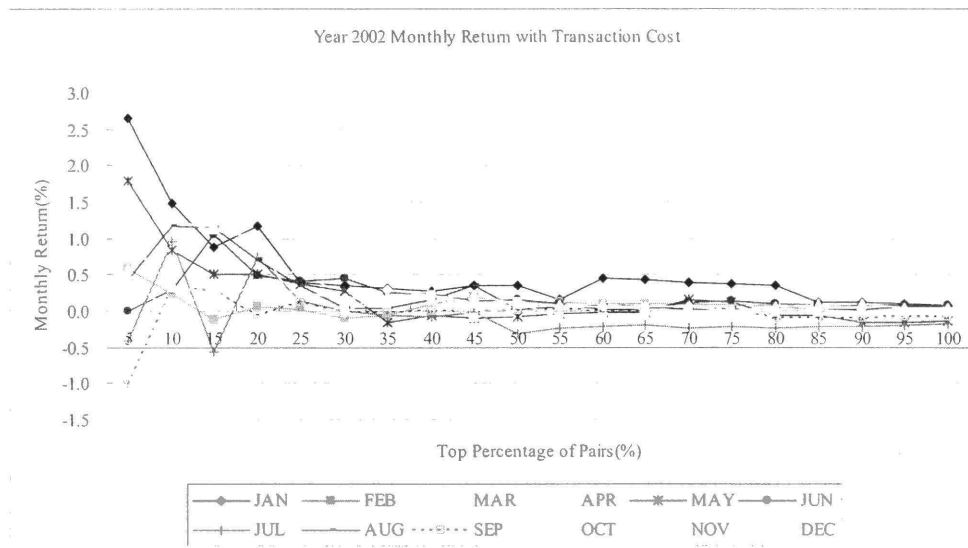


Figure 6.17 Year 2002 monthly average return with transaction cost 0.25%. (There are some negative monthly returns.)

The monthly return is negative when the index decreases suddenly. Such as, Jun to Aug of 2001 and May to Jul, Sep 2002. Because, we use one month as training set (in-sample set) and one month as testing set (out-of-sample set), if the two months trend are contrary, sometimes, the monthly return is negative. The reason is the training and testing sets are not consistent. However, the whole return and profit are still positive, so the stock-rule pair strategy is still valuable in real market trading.

6.4.4 Maximal Return Point

If some traders invest different money in stock markets, the traders should decide the number of pairs depending on the investment. So for different investment, it is important to decide the best number of pairs and top percentage for different investments, because it can make more profit with less money without trading.

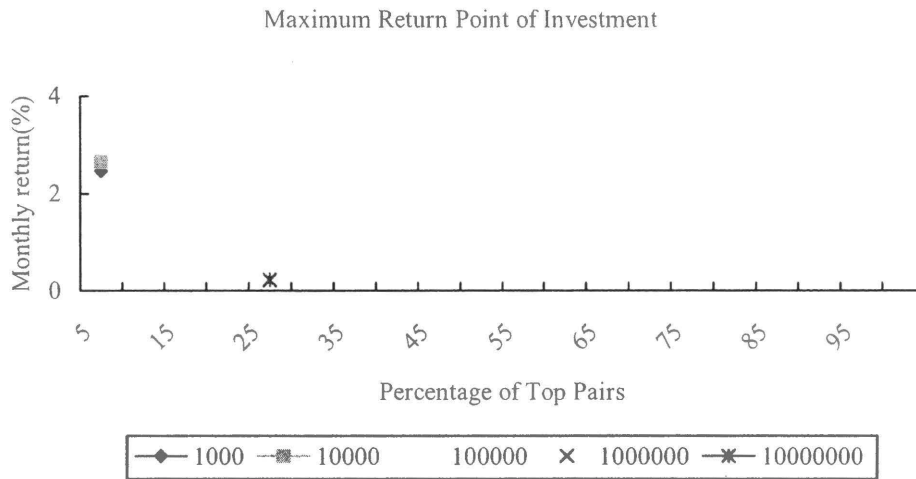


Figure 6.18 Jun 2001 the maximum points for investment.

From Figure 6.18, we can see the relationship between the investment and the percentage of top pairs. When the investment increases, the return decreases, but the percentage of top pair increases. That means when the investment increases, it needs more stocks and signals to be traded to make a higher return. However, when the investment decreases, it can only trade the limited stocks and signals. For the excess stocks and signals, it cannot trade any more because the money is not enough to trade all stocks and signals.

Figure 6.16 gives the following information: when the investment is increasing, the highest return can be reached when the number of pair is larger, but, when the investment is higher, the return is lower. It also shows that the suggested return is higher than the market index return which proves the return of stock-pair strategy is higher than the average market index return.

It is consistent with the real stock market. Here, we give an experiment result, and the graph of the investment and the top percentage of pairs. See Figures 6.18, 6.19 and 6.20.

6.4.5 Determination of Investment

For the different number of shares (pairs), the return must be different depending on the amount of the investments. When the investment is little, we can get a high return if we only consider a few top-percentage better stocks, but, when the investment is much, we should trade more stocks in order to reduce the “dead money”(never used to buy or sell) in hand to make a higher return. The following is the result we have got through the experiments.

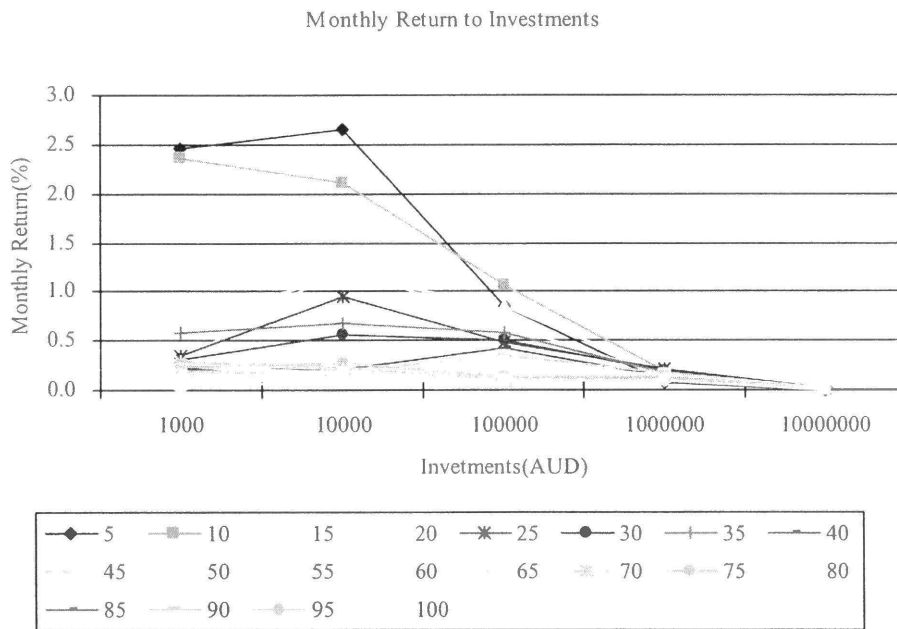


Figure 6.19 Jun 2001 monthly return to the investment and the number of pairs (Transaction cost is 0.25%).

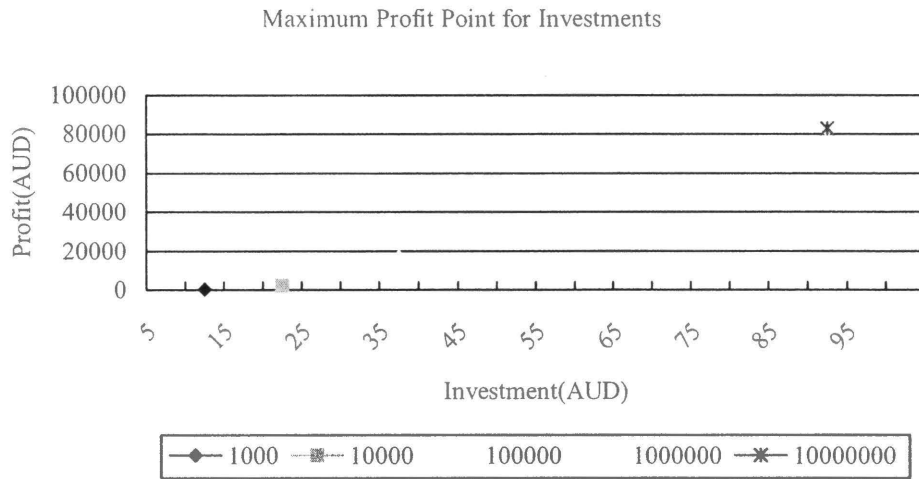


Figure 6.20 Jun 2001 maximum points of profit to the investment.

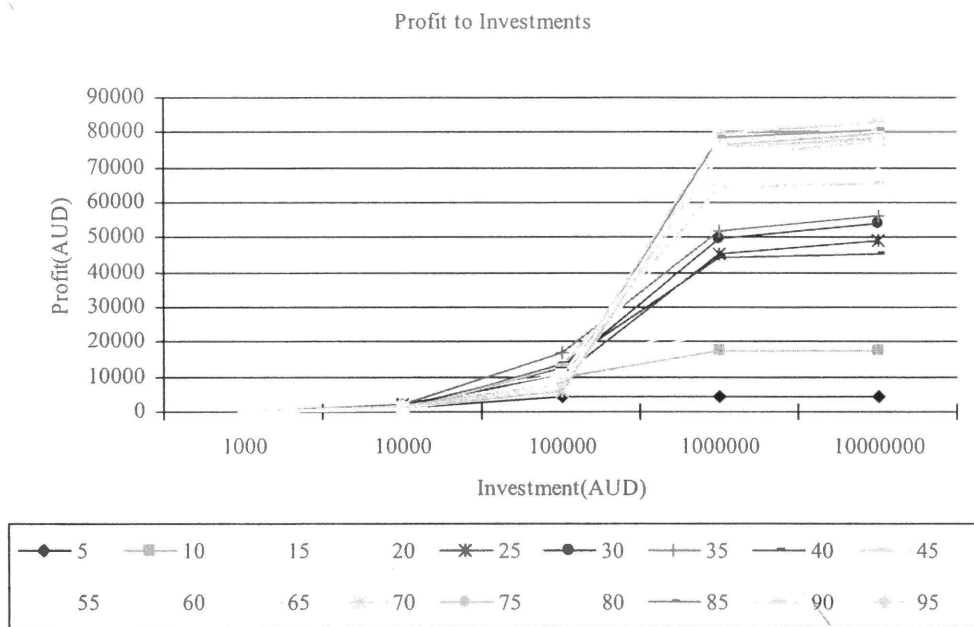


Figure 6.21 Profit to the investment and the number of pairs (from 5 per cent to 100 per cent, step is 5 per cent).

In Figure 6.21, ASX in-sample data is May 2001. Out-of-sample data is June 2001. Transaction cost is 0.25%.

Figure 6.20 shows that when the investment is less, it can get a maximum point at the little number of pairs selected (top 5% or 20% per cent pairs are enough). However, when investment is big, the maximum points are archived at the big percentage (90%) because it needs more shares to be traded.

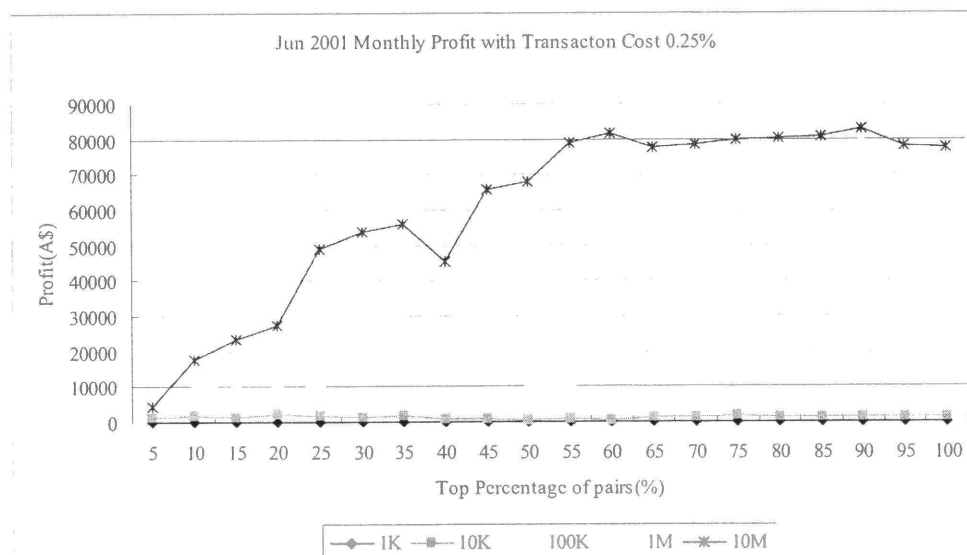


Figure 6.22 Jun 2001 monthly profit with transaction cost 0.25%.

From Figures 6.19, 6.21 and 6.22, we can see the result is consistent with the real market, and it answers the question of the relationship between the size of investment and the number of pairs. When the investment is more, the profit is higher absolutely, but the average return per share is relatively lower. When the investment is less, the profit is lower absolutely, but the average return per share is relatively higher. That explains why the investment is more, the average return is lower. The reason is that a large portion of money is not used to buy or sell any stocks, so the excess money does not make any contribution to return.

Figure 6.23 is the part signal series. It comes from out-of-sample top percentage pairs and sorted by time. The data fields are: serial number of signal, rule number, stock number, date, time, price, volume, trading action (Buy/Sell signal).

```

025signal jun - Notepad
File Edit Format View Help

Transaction Cost (0.25%), In-sample set date : (20010601 -- 20010630 ),
Out-of-sample set date: (20010701 -- 20010731)
= Total signals of all stock-rule pairs =
0: Rule( 1), Stock( 1), 20010702, 10: 0:57, 8.42, 1000, BUY
1: Rule( 1), Stock( 4), 20010702, 10: 1:26, 16.70, 900, BUY
2: Rule( 1), Stock( 1), 20010702, 10: 1:33, 8.38, 100, SELL
3: Rule( 1), Stock( 9), 20010702, 10: 2:49, 6.28, 1000, BUY
4: Rule( 1), Stock( 9), 20010702, 10: 4:15, 6.25, 500, SELL
5: Rule( 3), Stock( 1), 20010702, 10: 5: 8, 8.38, 4505, BUY
6: Rule( 1), Stock( 11), 20010702, 10: 5:21, 3.95, 252, BUY
7: Rule( 1), Stock( 11), 20010702, 10: 7:32, 4.00, 280, SELL
8: Rule( 2), Stock( 24), 20010702, 10: 9:16, 27.11, 57, BUY
9: Rule( 3), Stock( 24), 20010702, 10: 9:16, 27.50, 500, BUY
10: Rule( 3), Stock( 1), 20010702, 10: 9:22, 8.35, 2800, SELL
11: Rule( 3), Stock( 26), 20010702, 10: 9:45, 10.75, 1280, BUY
12: Rule( 2), Stock( 26), 20010702, 10:10: 5, 10.80, 600, BUY
13: Rule( 1), Stock( 26), 20010702, 10:10: 5, 10.80, 400, BUY
14: Rule( 2), Stock( 24), 20010702, 10:11:24, 27.20, 500, SELL
15: Rule( 1), Stock( 24), 20010702, 10:11:24, 27.20, 500, BUY
16: Rule( 1), Stock( 16), 20010702, 10:12:19, 2.96, 668, BUY
17: Rule( 3), Stock( 23), 20010702, 10:13: 9, 5.39, 600, BUY
18: Rule( 2), Stock( 1), 20010702, 10:13:53, 8.45, 1550, BUY
19: Rule( 1), Stock( 23), 20010702, 10:14: 5, 5.40, 400, BUY
20: Rule( 2), Stock( 23), 20010702, 10:14:29, 5.42, 1000, BUY
21: Rule( 3), Stock( 9), 20010702, 10:14:51, 6.23, 600, BUY
22: Rule( 1), Stock( 1), 20010702, 10:15:30, 8.45, 304, BUY
23: Rule( 2), Stock( 24), 20010702, 10:16: 9, 27.49, 1580, BUY
24: Rule( 1), Stock( 24), 20010702, 10:16: 9, 27.20, 574, SELL
25: Rule( 3), Stock( 24), 20010702, 10:16: 9, 27.20, 574, SELL
26: Rule( 2), Stock( 11), 20010702, 10:16:38, 4.01, 7000, BUY
27: Rule( 2), Stock( 24), 20010702, 10:17:30, 27.15, 500, SELL
28: Rule( 3), Stock( 1), 20010702, 10:18: 0, 8.46, 600, BUY
29: Rule( 2), Stock( 6), 20010702, 10:20: 2, 47.49, 679, BUY
30: Rule( 1), Stock( 23), 20010702, 10:20:33, 5.41, 19000, SELL
31: Rule( 2), Stock( 23), 20010702, 10:20:33, 5.41, 6000, SELL
32: Rule( 1), Stock( 24), 20010702, 10:21:33, 27.22, 900, BUY
33: Rule( 3), Stock( 26), 20010702, 10:21:44, 10.86, 967, SELL
34: Rule( 2), Stock( 2), 20010702, 10:22:26, 6.62, 2831, BUY
35: Rule( 1), Stock( 23), 20010702, 10:25:19, 5.44, 1500, BUY

```

Figure 6.23 The signals of the real time best-pair trading alert

6.5 Summary

In this chapter, some applications are presented in finding near optimal parameters, finding optimal sub-domain of parameters, searching for stock-rule pairs and the relationship between the investment and the return. All of these applications help investors to make more profit but take lower risk.

Firstly, we undertook SGA and RGA to get optimal combination of parameters to get a better Sharpe ratio for any stock and rule efficiently. It is important and it

is the foundation of further work because it makes execution time be an endurable time. Meanwhile, the result is near-optimal.

Secondly, in-depth rules are given by sub-domain combination of parameters. Because sometimes in stock markets, it is difficult to find the best one value, and users also want to make a little micro-tune by themselves, so an optimal sub-domain is better than a single value. In the sub-domain, every combination has the ability to output a positive Sharpe ratio leading to a higher profit even if it is not the highest one. So it is useful for investors.

Thirdly, we presented an effective rank list of stock-rule pairs. For different stock, we can combine it with a different rule and the optimal parameters. From the result, the pair list has a high percentage intersection of in-sample set and out-of-sample set.

Finally, we found the relationship between the sizes of investments and the number of pairs. When both pairs and investment are less, the return is higher but the profit is lower, when both of pairs and investment are higher, the profit is higher but the return is lower. It proves the less pair is the top performance ones, but, it can not make a large investment tradable for less volume. In this application, for the special stocks and rules, we gave an exact explanation and relationship. The result is consistent with real stock markets.

All of the above applications are tested in-sample set and verified out-of-sample set. The experiments are similar to real stock trading so that the result is more convincing.

Chapter 7 Evaluation

7.1 Financial Profitability

It is easy to observe high performance in-sample for trading rules and genetic algorithms, because both are executed under user control. The users can train trading rules until results are satisfactory. However, in real market trading, it is hard and important to get a high performance out-of-sample. The predictability and applicability are key factors, so we discuss the performance out-of-sample to evaluate our algorithms and strategies.

7.1.1 Economic Profitability

Profit is a target of most companies and traders, so the profit is the first basic criterion for our trading strategies. When we use standard genetic algorithms and robust genetic algorithms to compute parameters of trading rules, the profit is near to the result of enumerate algorithms which is the best one. Moreover, the execution time of our algorithms is far less than that of enumerate algorithms. [See Chapter 5.1 and 5.2]. We use RGA as the fundamental algorithm for all other in-depth algorithms, so we can get the near best profit with less execution time.

In real stock markets, we must consider all real situations and conditions to evaluate the performance of a stock or a trading rule, for instance, order book data (price, time and volume) and trading costs. The data and trading costs are important factors to decide the performance. Data comes from order book which can be gotten from CMCRC and SIRCA. Trading costs include not only transaction costs and taxes, but also hidden costs involved in the collection and analysis of information. According to Sweeney [Sweeney 1988], large institutional investors are able to achieve one-way transaction costs in the range of 0.1 to 0.2 per cent. However, for general situation and considering ASX market, 0.25 per cent transaction cost is acceptable. In this dissertation, all

experiments are considering 0.25 per cent transaction costs unless mentioned different.

To evaluate trading rules or genetic algorithm profitability, an appropriate benchmark is necessary. Since there is not such similar system which consider stock-rule pairs and investments when compute the profit, we prefer market return (market index return or index return), which is the return of all ordinary stocks of ASX, as the benchmark. In this thesis, the experimental results are the comparison of our system (stock-rule pair methodology, etc) to the index return.

Figure 7.1 gives the comparison of market return and different investment return with 0.25 per cent transaction cost. The investments are one thousand dollars and ten-thousand dollars, respectively. This method is used to rank stock-rule pairs.

Figure 7.1 and Table 7.1 demonstrate that the index return changes significantly, that means, sometimes, we can get a high return, but we also take a high risk. The return of our stock-rule pair methodologies is always positive and changes in a small range. That means the risk is lower. The index return is -0.055; the average returns of our methodology are 1.592 and 1.427, when the investments are one thousand dollars and ten-thousand dollars, respectively. The index return is higher than monthly return when investment is more than 100 thousand, because the number of stock-rule pairs is not enough for so much investment (see Chapter 6.4). The experimental results show that our methodologies can make a higher return at a lower risk. Figure 7.1 shows that the index returns are less than that of stock-rule pair methodologies.

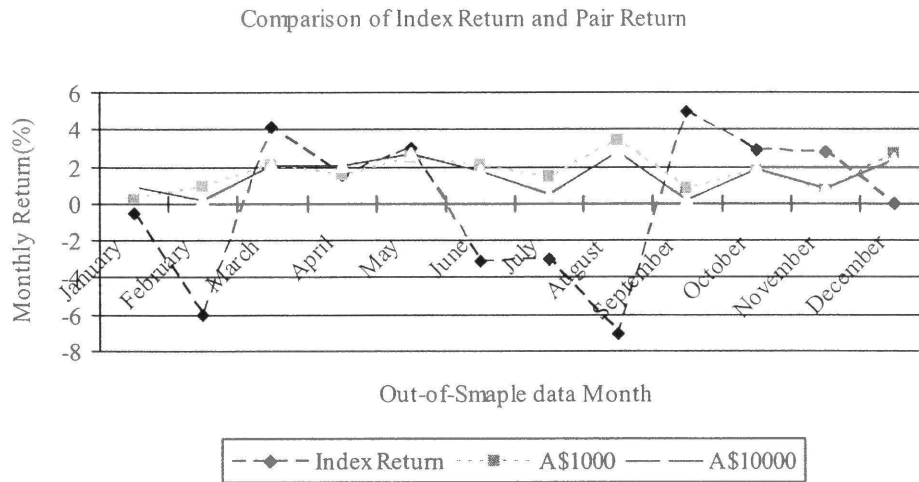


Figure 7.1 The comparison of index return and Stock-rule pair methodologies returns for different investments. (Monthly return with 0.25% transaction cost).

Table 7.1 Index return and Stock-rule pair methodologies return.

	Annual return (%)	Average monthly return (%)
Index	2.747	0.228
A\$1K	18.136	1.511
A\$10K	17.7	1.457
A\$100K	9.204	0.767
A\$1M	1.033	0.086
A\$10M	0.105	0.008

7.1.2 Sharpe Ratio

Another criterion is Sharpe ratio (in briefly, S/R), which is the ratio of return to risk. If S/R is higher that means return is high and risk is low. If S/R is lower that means return is low but risk is high [Investopedia]. Table 7.2 presents the best S/R result for the stock-rule pair methodologies. (In-sample date is year 2000, and year 2001 is out-of-sample set).

Table 7.2 The Sharpe ratio of stock-rule pairs.

Stocks	Rules (Sharpe ratio)		
	Moving Average	Filter rule	Channel Break-out
A01	0.458	0.323	0.957
A02	0.588	0.347	1.066
A03	0.411	0.5	0.817
A04	0.516	0.349	1.104
A05	0.804	0.706	1.353
B06	0.529	0.251	0.846
B07	1.229	0.395	2.053
C09	0.715	0.322	0.984
C10	0.476	0.557	0.334
C11	0.475	0.319	0.784
F12	0.669	0.446	1.007
F13	0.91	0.296	1.103
G14	0.684	0.395	1.425
I15	1.223	0.27	0.955
J16	0.542	0.528	1.097
M17	0.654	1.098	0.738
M18	0.446	0.596	0.54
M19	0.416	0.426	1.26
O21	1.351	0.174	1.836
P22	0.61	0.394	0.625
Q23	0.732	0.325	1.477
Q24	0.696	0.308	0.776
S25	0.82	0.537	1.833
S26	0.416	0.374	0.619
T28	0.69	1.203	0.844

T29	0.55	0.575	0.672
W30	0.456	0.352	1.122
W31	0.727	0.352	1.19
W32	0.428	0.374	0.511
W33	0.473	0.314	0.772

7.1.3 Predictability

Since t -tests are widely used in statistics to judge the mean value of an algorithm, this tests method can be developed to test our result. When we want to investigate the statistical significance of the forecasting power of buy and sell signals, we can use traditional t -tests to examine whether trading rules issue profitable or not. The method is its buy (or sell) signals on days when the return on the market is on average higher (or lower) than unconditional mean return for the market.

The t -statistic used to test the predictability of the buy signals is: [Robert 1999]

$$t_{buy} = \frac{\bar{r}_{buy} - \bar{r}_m}{\sigma \sqrt{\frac{1}{N_{buy}} + \frac{1}{N}}} \quad (7.1)$$

where \bar{r}_{buy} represents the average daily return following a buy signal and N_{buy} is the number of days that the trading rule returns a buy signal. The null and alternative hypotheses can be stated as:

$$H_0 : \bar{r}_{buy} \leq \bar{r}_m \quad (7.2)$$

$$H_1 : \bar{r}_{buy} > \bar{r}_m \quad (7.3)$$

Also, a t -statistic can be reused to test the predictability of the sell signals. In order to test whether the difference between the mean return on the market following a buy signal and the mean return on the market following a sell signal is statistically significant, a t -test can be specified as: [Robert 1999]

$$t_{buy-sell} = \frac{\bar{r}_{buy} - \bar{r}_{sell}}{\sigma \sqrt{\frac{1}{N_{buy}} + \frac{1}{N_{sell}}}} \quad (7.4)$$

where the null and alternative hypotheses are:

$$H_0 : \bar{r}_{buy} - \bar{r}_{sell} \leq 0 \quad (7.5)$$

$$H_1 : \bar{r}_{buy} - \bar{r}_{sell} > 0. \quad (7.6)$$

Through the entire dissertation, all returns, profits and Sharpe ratios are derived from out-of-sample set. The results are shown that our algorithms to be profitable and predictable already, so we did not use the other methods to prove the predictability again, but, the t -test methods can be used in the future research.

7.2 Computational Performance

7.2.1 Execution Time

In stock market, execution time is usually a long time, because the order book data is usually very huge and the combination is also very large. If we test generic algorithms for one trading rule, enumerate algorithm may compute 10,000,000 times for all possible combinations. It may cost 60 minutes. It is $O(m^n)$, m is the number of a parameter domain value, n is the number of total parameters. The execution time is almost impossible to give a real time outcome and result. So an optimal algorithm to save execution time is the precondition for all real time analysis systems.

In this thesis, we implement standard genetic algorithm (SGA) to do some experiments and get a near-optimal result. Execution time is reduced almost by 99 per cent (see Figure 5.2). The executing complex of genetic algorithm is about $O(Gn)$, G is the number of generations, n is the number of parameters. Sometimes, the executing complex is varied, because ending conditions is changed. Such as, a genetic algorithm gets a near-optimal result rapidly.

The result of genetic algorithm is more than 90 per cent near the optimal value. However, the randomly selected result cannot get the similar result. Sometimes the randomly selected parameter only results a negative profit. [Lin et al 2004a]

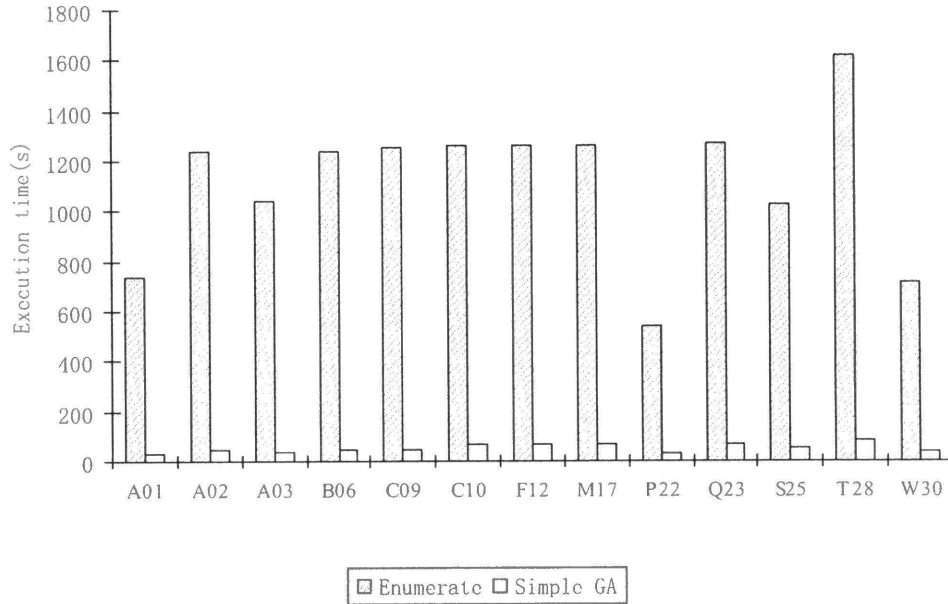


Figure 7.2 The execution time of enumerate algorithm and GA.

In Figure 7.2, the 13 shares are selected randomly from ASX. Filter rule, order book data period is from 2 January 1998 to 20 Feb 2001.

7.2.2 Memory

All the experiments are run under the same conditions (see Chapter 1). All experiments can be run and output correct results. Memory optimization can be improved by other algorithms, which is not the target of this dissertation. The in-depth methodologies do not need more memory, and the in-depth methodologies can be implemented in an ordinary computer configuration. No excess memory is needed for our method. Of course, high speed and big memory computer will helpful.

Chapter 8 Conclusions and Future Work

8.1 Conclusions

In this chapter, we conclude the evaluation metrics both in finance and information technology. In either side, the performance is improved. We did not find the similar systems, so we only compare the result before and after the optimization, for instance, the execution time with to without genetic algorithms, the market average return to the ranked pair return.

Firstly, we build a knowledge data base to store the domain knowledge (expert experience and domain constraints), which are very important in real stock market trading system. Domain knowledge can help to get an improved result efficiently and effectively. Currently, most systems have not considered domain knowledge. In our system, we built a domain knowledge database to integrate domain knowledge into the system. For the two kinds of knowledge, one comes from experts and another comes from system output. We can keep both of them and store domain knowledge into the knowledge database. Such as, in-sample and out-of-sample data size, which can be set by experts or computed by the system. The suitable sizes can avoid a noise and keep predictability.

Secondly, we use an optimized sub-domain to instead of a single value to filter a noise and make sure our result is really a good one, even if the result is not the best one for making profit. The advantages of the sub-domain are the result has little possibility of being a noise and traders can micro-tune the parameter combination in the sub-domain. The result shows that performance is always good in the sub-domain.

Thirdly, we implement genetic algorithms to improve computing efficiency with a near-optimal performance (more than 90 per cent of enumerate algorithm result, but, only 0.01 to 0.1 per cent of total execution time). Genetic algorithms are the fundamental tools for in-depth data mining applications, otherwise the further

research and computation becomes impossible. For example, for ranking all stock-rule pairs in ASX, standard genetic algorithms costs 5 to 6 hours. The enumerate algorithm costs 2-3 days to compute only one investment. Moreover, we have embedded domain constraints into genetic algorithms so that genetic algorithms can filter noisy signals. We call genetic algorithms with domain knowledge as robust genetic algorithms (RGA).

Fourthly, we discuss fuzzy set and multiple criteria methods to evaluate stocks. A numeric result is not clear to describe whether a stock is good or bad. One value does not mean good always, since stock market usually changes time by time. Sometimes, a stock with a positive return may be not good, but, on the contrary, sometimes, a stock with a negative return may be good. Because we should compare to other stocks, market index return and risk free return, etc. Once we considered all these factors, the output stock list should be much more useful.

Fifthly, we build stock-rule performance table to rank pairs. The performance of a stock changed when the stock combined with different trading rules. So we cannot determine a stock is good or bad without trading rules. We consider stock-rule pairs performance rather than stocks performance to overcome this problem. So, we build the stock-rule performance table. From the table [Chapter 6.4], we can see the performance changes for the different stock-rule combination. Our future work is based on the stock-rule pair table to select stock-rule pairs and decide investments.

Finally, we consider different investments and the number of stock-rule pairs. We draw profit and monthly return graphs for the investments, number of stock-rule pairs, and compare the return of our method to the stock market index return. The result shows the pattern among these factors, in which, we can get the more detailed and exact relationship among these factors, so our method is more practicable in real market investing. [Chapter 6.4]

In summary, we discuss and implement some algorithms and ideas for real stock market trading methodology. All of these problems come from current trading

platforms and these algorithms and ideas enhance the current platforms [F-TRADE]. These algorithms and ideas are essential and necessary for real stock trading. In our optimal trading system, both efficiency and effectiveness are improved. Meanwhile, our system also keeps profitability and predictability. The performance of our system is better than that of the current system and better than the market return. Moreover, it is more reasonable, applicable and actionable. All of these are the key issues in a real market trading system.

8.2 Future Work

In this thesis, we have presented some problems and solutions, but, there are still some new in-depth researches need to be considered in the future.

First, the problem we discuss is how to search the best stock-rule pairs. Currently, we build a stock-rule performance table and sort them by performance. We can imagine that stocks can be divided into some sub-sets with similar special patterns. In different sub-sets, the stocks should be having the same patterns and the stocks can combine with the same trading rules. For example, we separate all stocks into different sub-sets by some defined patterns, such as, “increasing rapidly-decreasing slowly”, “increasing slowly-decreasing slowly”, “keeping steady”, “increasing rapidly-decreasing rapidly”. All stocks have the same pattern in one sub-set. For one sub-set, we can confirm one best trading rule for the sub-set, so we need not compute them one by one. This method can both improve efficiency and classify new stocks without any computation.

Second, we want to consider investments when we rank stock-rule pairs. Currently, when we rank the stock-rule pairs, we do not consider the investments. We just consider a fixed investment, such as 1000 dollars. For different investments, the better pair may be become a worse one. So the best way is when we rank the pairs, we consider the investments together. The expected result is: when investment is 1000 dollars, we get a sorted stock-rule pair list; when

investment changes to 1000000 dollars, we get another maybe different sorted stock-rule pair list.

Third, we focus the algorithm to find the top best pair groups. Currently, when we want to select the best top percentage pairs (group), the method is we sort all stock-rule pairs and select top percentage pairs. When we sort the stock-rule pairs, we discard the influence of these pairs each other. The problem is the whole return of a group does not equal to the sum of each pair return in the group. Because these signals maybe influence each other, and volume and available money are different. The result is not guaranteed to be the best any more even if all the single pair is the best one. For example, one pair buy signal may make other pair buy signal cannot be realized because the available money is not enough to buy. So, our future work is to get the best combination pairs considering signals distribution and volume. In the best group, each pair may be not the best one, but their combination is the best one.

Fourth, we improve the robust genetic algorithm furthermore. We combine genetic algorithms to the finance problems and make genetic algorithms more efficiently and effectively. We have implemented genetic algorithms and robust genetic algorithms into our system, but, we do not make any improvements for genetic algorithms (traditional genetic algorithms added the domain constraints), In the future, we will consider how to upgrade the standard genetic algorithms and make the standard genetic algorithms more efficiently and effectively, for instance, to add some new parameters, operations or to change probabilities of crossover and mutation to a suitable value.

Bibliography

- [Aarts et al 2005] Aarts, F., and Lehnert, T. On Style Momentum Strategies. *Applied Economics Letters* 12, 795-799. 2005.
- [AC3] <http://www.ac3.com.au>
- [Acar et al 1997] Acar, E. and S. Satchell, (eds.). *Advanced Trading Rules*, Butterworth-Heinemann. 1997.
- [Acharya et al 1997] S. Acharya, M. J. Franklin, and S. B. Zdonik. Balancing Push and Pull for Data Broadcast. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 183–194, 1997.
- [Achelis 1995] Achelis, S.B. *Technical Analysis from A to Z*. Probus Publishing, Chicago, Illinois. 1995.
- [Alexander 1961] Alexander, S., Price Movements in Speculative Markets: Trends or Random Walks, *Industrial Management Review*, v2 n2, May, 7-26.1961.
- [Alexander 1964] Alexander, S., Price Movements in Speculative Markets: Trends or Random Walks, No. 2. In Cootner, P. (ed.). *The Random Character of Stock Price Prices*, Massachusetts Institute of Technology Press, Cambridge MA. Pp 338-372. 1964.
- [Allen et al 1993] Allen, P.M., and H.K. Phang. Evolution, Creativity and Intelligence, in H. Haken and A. Mikhailov (eds.), 1993, *Interdisciplinary Approaches to Nonlinear Complex Systems*, Springer-Verlang, Berlin.
- [Allen et al 1994] Allen, P.M., and H.K. Phang. Managing Uncertainty in Complex Systems: Financial Markets, in L. Leyesdorff, P. Van den Besselaar (eds.), 1994, *Evolutionary Economics and Chaos Theory*, Pinter, London.
- [Allen et al 1999] Allen, Franklin and Karjalainen, Risto. Using Genetic Algorithms to Find Technical Trading Rules, *Journal of Financial Economics*, 51, 245-271. 1999.
- [Alonso et al 1988] Rafael Alonso, Daniel Barbar'a, Hector Garcia-Molina, and Soraya Abad. Quasi-Copies: Efficient Data Sharing for Information Retrieval

- Systems. In *Proceedings of the International Conference on Extending Database Technology*, pages 443–468, 1988.
- [Amir et al 1998] E. Amir, S. McCanne, and R. Katz. An Active Service Framework and Its Application Real-Time Multimedia Transcoding. In *Proceedings of the ACM SIGCOM Conference*, pages 178–189, 1998.
- [Andrada-Felix, et al 2003] Andrada-Felix, J., Fernandez-Rodriguez, F., Garcia-Artiles, M. D., and Sosvilla-Rivero, S.. An Empirical Evaluation of Non-linear Trading Rules. *Studies in Nonlinear Dynamics and Econometrics* 7. 2003.
- [Andrew et al 1990] Lo, Andrew W. and A. Craig MacKinlay. Data-Snooping Biases in Tests of Financial Asset Pricing Models, *The Review of Financial Studies*, 1990 3, 431–467.
- [Angoss] <http://www.angoss.com>
- [Ankenbrandt 1991] Ankenbrandt CA. An Extension to the Theory of Convergence and a Proof of the Time Complexity of Genetic Algorithms. *Foundations of Genetic Algorithms*. Ed. Rawlins GJE, Morgan Kaufmann Publishers Inc. 1991.
- [Anna C. Gilbert, et al 2005] Anna C. Gilbert, Yannis Kotidis, S. Muthukrishnan, Martin J. Strauss, Domain-Driven Data Synopses for Dynamic Quantiles. *IEEE Transactions on Knowledge and Data Engineering*. July 2005 (Vol. 17, No. 7) pp. 927-938.
- [Arrow 1963] Arrow, K. J. Comment, *Review of Economics and Statistics* 45 (Supplement: February), 1963, 24-27.
- [ASX] <http://www.asx.com.au>
- [Aytug et al 1996] Aytug, H. and G. J. Koehler. Stopping Criteria for Finite Length Genetic Algorithms. *ORSA Journal on Computing*. 1996.8, No. 2, pp. 183-191.
- [Aytug et al 1997] Aytug, H., S. Bhattacharyya and G. J. Koehler. A Markov Chain Analysis of General Cardinality Genetic Algorithms with Power of 2 Cardinality Alphabets. *European Journal of Operational Research*. 1997, 96, pp. 195-201.

- [Aytug et al 2000] Aytug, H. and G. J. Koehler. New Stopping Criteria for Genetic Algorithms, *European Journal of Operational Research*, 2000, 126, 662-674.
- [Bajeux et al 1993] Bajeux-Besnainou, I., and Portait, R. Dynamic Asset Allocation in a Mean-Variance Framework, Working Paper, ESSEC (revised January 1995).1993.
- [Ball 1978] Ball, R. Filter Rules: Interpretation of Market Efficiency, Experimental Problems and Australian Evidence. *Accounting Education*, 18(2), 1-17. 1978.
- [Banavar et al 1999] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman. An Efficient Multicast Protocol for Content-Based Publish-Subscribe Systems. In *Proceedings of the 19th International Conference on Distributed Computing Systems*, 1999.
- [Bauer 1994] Bauer, R.J. *Genetic Algorithms and Investment Strategies*, Wiley Finance Edition, John Wiley and Sons, New York. 1994.
- [Baviera et al 2002] Baviera, R., Pasquini, M., Serva, M., Vergni, D., and Vulpiani, A.. Antipersistent Markov Behavior in Foreign Exchange Markets. *Physica a-Statistical Mechanics and Its Applications* 312, 565-576. 2002.
- [Bessembinder et al 1997] Bessembinder, H. and K. Chan. The Profitability of Technical Trading Rules in the Asian Stock Markets, *Pacific-Basin Finance Journal*, July, 1997, 257-284.
- [Bessembinder et al 1998] Bessembinder, H., and K. Chan. Market Efficiency and the Returns to Technical Analysis, *Financial Management*, 27 (2), 5-17. 1998.
- [Bestavros 1996] A. Bestavros. Peculative Data Dissemination and Service to Reduce Server Load, Network Traffic and Service Time in Distributed Information Systems. In *Proceedings of International Conference on Data Engineering*, pages 180–189, 1996.
- [Bhide et al 2002] Manish Bhide and Krithi Ramamritham and Prashant Shenoy. Efficiently Maintaining Stock Portfolios Up-to-Date on the Web. In *Proceedings*

of the 12th International Workshop on Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems, page 60, 2002.

[Bird 1985] Bird, P. J.W.N. The Weak Form Efficiency of the London Metal Exchange, *Applied Economics*, 17, 571-587. 1985.

[Black et al 1973] Black, F., and Scholes, M. The Pricing of Options and Corporate Liabilities, *Journal of Political Economy* 1973, 81, 637-54.

[Bo et al 2005] Bo, L., Sun, L. Y., and Mweene, R.. Empirical Study of Trading Rule Discovery in China Stock Market. *Expert Systems with Applications* 28, 531-535. 2005.

[Bookstaber et al 1985] Bookstaber, R., and Clarke, R. Problems in Evaluating the Performance of Portfolios with Options, *Financial Analysts Journal* 1985, 41, (January/February), 48-62.

[Bradley 1979] Efron, Bradley. Bootstrap Methods: Another Look at the Jackknife, *Annals of Statistics* 1979, 7, 1–26.

[Breedon et al 1978] Breedon, D., Litzenberger, R. Prices of State Contingent Claims Implicit in Option Prices, *Journal of Business*, 1978, 51, 621-652.

[Brennan et al 1981] Brennan, M., and Solanki, R. Optimal Portfolio Insurance, *Journal of Financial and Quantitative Analysis*, 1981, 16, 279-300.

[Brock et al 1992] Brock, W., Josef L. and Blake L.. Simple Technical Trading Rules and the Stochastic Properties of Stock Returns, *Journal of Finance* 1992, 47, 1731–1764.

[Brooks et al 2005] Brooks, C., and Katsaris, A.. A Three-Regime Model of Speculative Behaviour: Modelling the Evolution of the S&P 500 Composite Index. *Economic Journal* 115, 767-797. 2005.

[Brown et al 1985] Brown, D., and Gibbons, M. A Simple Econometric Approach for Utility-Based Asset Pricing Models, *Journal of Finance* 1985, 40, 359-381.

[Cao et al 1997] P. Cao and S. Irani. Cost-Aware WWW Proxy Caching Algorithms. In *Proceedings of the 1997 USENIX Symposium on Internet Technologies and Systems*, pages 193–206, 1997.

- [Cao et al 2004a] Longbing Cao, Chao Luo, Dan Luo, Li Liu. Ontology Services-Based Information IntegRation in Mining Telecom Business Intelligence. *Proceeding of PRICAI04*, Springer Press, 85-94, 2004.
- [Cao et al 2004b] Longbing Cao, Jiaqi Wang, Li Lin, and Chengqi Zhang. Agent Services-Based Infrastructure for Online Thesis of Trading Strategies. *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04)*, IEEE Computer Society Press, 345-348, 2004.
- [Cao et al 2004c] Longbing Cao, Jiarui Ni, Jiaqi Wang, Chengqi Zhang. Agent Services-Driven Plug-and-Play in F-TRADE. *Australian Conference on Artificial Intelligence 2004*: 917-922.
- [Cao et al 2005] Longbing Cao, Schurmann, R. and Zhang, C. Q. Domain-Driven In-depth Pattern Discovery: A Practical Methodology, *Proceedings of AusDM*, pp. 101-114. 2005.
- [Cate 1992] V. Cate. Alex - A Global File System. In *Proceedings of the 1992 USENIX File System Workshop*, pages 1–12, 1992.
- [Chang et al 1994] Chang, P.H.K. and C.L. Osler. Evaluating Chart-based Technical Analysis: The Head and Shoulder Pattern in Foreign Exchange Markets, Research Paper, Federal Reserve Bank of New York. 1994.
- [Chankhunthod et al 1996] A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz, and K. J. Worrel. A Hierarchical Internet Object Cache. In *Proceedings of the 1996 USENIX Technical Conference*, pages 153–164, 1996.
- [Cheung et al 1997] Cheung, Y., and C.Y. Wong. The Performance of Trading Rules on Four Asian Currency Exchange Rates, *Multinational Finance Journal*, v1 n1., 1-22. 1997.
- [Chiu 1996] Chiu S. L. Selecting input variables for fuzzy models. *Journal of Intelligent and Fuzzy Systems*. 4. 243-256. 1996.
- [Christos et al 1999] Christos E., Andrew H, John M. Chris C. Multiple –Criteria Genetic Algorithms for Features Selection in Neurofuzzy Modeling. *Proc. of the IJCNN'99*, 10-16 July Washington, USA, 4387-4392. 1999.
- [CMCRC] <http://www.cmcrc.com>

- [Conrad et al 1998] Conrad, J. and G. Kaul. An Anatomy of Trading Strategies, *Review of Financial Studies*, V.11, No.3, 489-519. 1998.
- [Cornell et al 1978] Cornell, W.B., and J.K. Dietrich. The Efficiency of the Market for Foreign Exchange Under Floating Exchange Rates, *Review of Economics and Statistics*, 60, 111-120. 1978.
- [Corrado et al 1992] Corrado, C.J., and S.H. Lee. Filter Rule Tests of the Economic Significance of Serial Dependencies in Daily Stock Returns, *Journal of Financial Research*, 15(4), 369-387. 1992.
- [CRISP-DM] <http://www.crisp-dm.org>
- [Curcio et al 1997] Curcio, R., C. Goodhart, D. Guillaume, and R. Payne. Do Technical Trading Rules Generate Profits? Evidence from the Intra-Day Foreign Exchange Market, *Int. J. Fin. Econ*, v2 n4. Special Issue on Technical Analysis and Financial Markets. 1997.
- [Dacorogna 1995] Dacorogna, M.M.. The Main Ingredients of Simple Trading Models for Use in Genetic Algorithm Optimization, Research Paper, Olsen Associates. 1995.
- [Davis 1987] Davis, L. Genetic Algorithms and Simulated Annealing. London, Pitman. 1987.
- [Dempster et al 2001] Dempster, M. A. H., Payne, T. W., Romahi, Y., and Thompson, G. W. P.. Computational Learning Techniques for Intraday FX Trading Using Popular Technical Indicators. *IEEE Transactions on Neural Networks* 12, 744-754. 2001.
- [Diebold et al 1995] Diebold, F. X. and R. S. Mariano. Comparing Predictive Accuracy, *Journal of Business and Economic Statistics* 1995, 13, 253–265.
- [Diebold 1998] Diebold, F. X.. *Elements of Forecasting* (South-Western College Publishing, Cincinnati, Ohio) 1998.
- [Dimitris et al 1994] Politis, Dimitris, and Joseph Romano. The Stationary Bootstrap, *Journal of the American Statistical Association* 1994, 89, 1303–1313.
- [Dooley et al 1983] Dooley, M.P., and J. Shafer, Analysis of Short Run Exchange Rate Behavior: March 1973 to November 1981, in D. Bigman and T.

- Taya (eds.), *Exchange Rate and Trade Instability: Causes, Consequences and Remedies*, pp. 43-69. Ballinger, Cambridge, MA. 1983,
- [Dorian 1999] Pyle, Dorian. *Data Preparation for Data Mining*. Morgan Kaufmann, 1999.
- [Dowjones] <http://www.dowjones.com>
- [Dybvig et al 1982] Dybvig, P., and Ingersoll, J. Mean Variance Theory in Complete Markets, *Journal of Business* 1982, 55, 233-252.
- [Dybvig et al 1985] Dybvig, P., and Ross, S. Differential Information and Performance Measurement Using a Security Market Line, *Journal of Finance* 1985, 40, 383-399.
- [Edwards et al 1992] Edwards, Robert D. and John Magee. *Technical Analysis of Stock Trends* (John Magee, Inc., Boston).1992.
- [E-IntelligenceGroup] <http://datamining.it.uts.edu.au>
- [Emery et al 2002] Emery, G. W., and Liu, Q. F.. An Analysis of the Relationship between Electricity and Natural-Gas Futures Prices. *Journal of Futures Markets* 22, 95-122. 2002.
- [English 2002] John English. *The New Australian Stock Market Investor*. Allen & Unwin, 2002.
- [Eric E. 2004] Eric Evans, *Domain-Driven Design*. Addison-Wesley, 2004.
- [Fama et al 1966] Fama, E.F. and M.E. Blume. Filter Rules and Stock Market Trading, *Journal of Business*, 39, 226-241. 1966.
- [Fama et al 1996] Fama, Eugene and Marshall Blume. Filter Rules and Stock-Market Trading, *Journal of Business*. 1996, 39, 226–241.
- [Ferson et al 1996] Ferson, W., and Schadt, R. Measuring Fund Strategy and Performance in Changing Economic Conditions, *Journal of Finance* 1996, 51, 425-461.
- [FOREX] <http://www.l38.net>
- [Foster et al 1997] Foster, F. Douglas, Tom Smith, and Robert E. Whaley. Assessing Goodness-of-Fit of Asset Pricing Models: The Distribution of the Maximal R², *Journal of Finance* 1997, 52, 591–607.

- [Frenkel et al 2004] Frenkel, M., and Stadtmann, G.. Trading Rule Profitability and Central Bank Interventions in the Dollar-Deutschmark Market. *Jahrbucher Fur Nationalokonomie Und Statistik* 224, 653-672. 2004.
- [F-TRADE] <http://datamining.it.uts.edu.au:8080/tsap>
- [F-TRADE Features] <http://www-staff.it.uts.edu.au/~lbcao/fttrade/fttrade.htm>
- [F-TRADE Manual] <http://www-staff.it.uts.edu.au/~lbcao/fttrade/manual.pdf>.
- [Galai et al 1984] Galai, D., and Geske, R.. Option Performance Measurement, *Journal of Portfolio Management*, 1984, 42-46.
- [Gartley 1935] Gartley, H. M. *Profits in the Stock Market* (Lambert-Gann Publishing Company, Pomeroy, Washington), 1935.
- [Gencay 1996a] Gencay, R.. Non-linear Prediction of Security Returns with Moving Average Rules, *Journal of Forecasting*, 15, 165-174. 1996.
- [Gencay 1996b] Gencay, R.. The Predictability of Security Returns with Simple Technical Trading Rules, Working Paper, U. of Windsor. 1996.
- [Gencay et al 1996c] Gencay, R., and T. Stengos. Technical Trading Rules and the Size of the Risk Premium in Security Returns, Working Paper, U. of Windsor. 1996.
- [Gencay et al 1997] Ramazan Gencay & Thanasis Stengos. Technical Trading Rules and the Size of the Risk Premium in Security Returns, *Studies in Nonlinear Dynamics & Econometrics*, Berkeley Electronic Press, vol. 2(2), pages 23-34. 1997.
- [Gencay 1998] Gencay, R.. Optimization of Technical Trading Strategies and the Profitability in Security Markets, *Economics Letters*, 59, 249-254. 1998.
- [Gencay 1999] Gencay, R.. Linear, Nonlinear and Essential Foreign Exchange Rate Prediction with Simple Technical Trading Rules, *Journal of International Economics*, 47(1), 91-107. 1999.
- [Glosten et al 1994] Glosten, L., and Jagannathan, R. A Contingent Claim Approach to Performance Evaluation, *Journal of Empirical Finance* 1994, 1, 133-160.

- [Goldbaum 1996] Goldbaum, D.. A Nonparametric Examination of Market Information: Application to Technical Trading Rules, Working Paper, U. of Wisconsin. 1996.
- [Goodhart et al 1992] Goodhart, C.A.E., and R. Curcio. When Support/Resistance Levels Are Broken, Can Profits Be Made? Evidence from the Foreign Exchange Market, LSE Financial Markets Group Discussion Paper Series, L.142, July. 1992.
- [Greenhalgh et al 2000] Greenhalgh, D. and S. Marshall. Convergence Criteria for Genetic Algorithms. *SIAM Journal on Computing*. 2000, 30, 1, pp.269-282.
- [Grinblatt et al 1989] Grinblatt, M., and Titman, S. Portfolio Performance Evaluation: Old Issues and New Insights, *Review of Financial Studies* 1989, 2, 393-421.
- [Grinblatt et al 1994] Grinblatt, M., and Titman, S. A Study of Mutual Fund Returns and Performance Evaluation Techniques, *Journal of Financial and Quantitative Analysis* 1994, 29, 419-44.
- [Grinblatt et al 2004] Grinblatt, M., and Moskowitz, T. J.. Predicting Stock Price Movements from Past Returns: The Role of Consistency and Tax-loss Selling. *Journal of Financial Economics* 71, 541-579. 2004.
- [Gwertzman et al 1995] J. Gwertzman and M. Seltzer. The Case for Geographical Push Caching. In *Proceedings of the 5th Annual Workshop on Hot Operating Systems*, 1995.
- [Gwertzman et al 1996] J. Gwertzman and M. Seltzer. World-Wide Web Cache Consistency. In *Proceedings of the 1996 USENIX Technical Conference*, pages 141–152, 1996.
- [Haas et al 1996] P.J.Haas. Hoeffding Inequalities for Join-selectivity Estimation and Online Aggregation. Technical Report, IBM Almaden Research Center, 1996.
- [Halbert 1997] White, Halbert. A Reality Check for Data Snooping, Technical Report, NRDA, San Diego, CA.1997.

- [Han et al 2000] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Technologies*. Morgan Kaufmann Publishers, 2000.
- [Han et al 2001] Jiawei Han and Micheline Kamber. *Data Mining: Technologies, Techniques, Tools, and Trends*. Morgan Kaufmann Publishers, 2001.
- [Harrison et al 1979] Harrison, J.M., and Kreps, D. Martingales and Arbitrage in Multiperiod Security Markets, *Journal of Economic Theory*, 1979, 20, 381-408.
- [He et al 1993] He, H., and Leland, H. On Equilibrium Asset Price Processes, *Review of Financial Studies*, 1993, 6, 593-617.
- [Hendrik et al 1995] Bessembinder, Hendrik and Chan, Kalok. The Profitability of Technical Trading Rules in the Asian Stock Markets, *Pacific-Basin Finance Journal*, Elsevier, vol. 3(2-3), pages 257-284. 1995.
- [Henriksson et al 1981a] Henriksson, R., and Merton, R. On Market Timing and Investment Performance I. An Equilibrium Theory of Value for Market Forecasts, *Journal of Business* 1981, 54, 363-406.
- [Henriksson 1981b] Henriksson, Merton R. C.. On Market Timing and Investment Performance II: Statistical Procedures for Evaluating Forecasting Skills, *Journal of Business*, 54, 513-533. 1981
- [Hexton 1994] Richard Hexton. *Technical Analysis in the Options Market: The Effective Use of Computerized Trading Systems*. Wiley, 1994.
- [Hoeffding 1963] Wassily Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *American Statistical Journal*, pages 13–30, March 1963.
- [Hong et al 2003] Hong, Y. M., and Lee, T. H.. Inference on via Generalized Spectrum and Nonlinear Time Series Models. *Review of Economics and Statistics* 85, 1048-1062. 2003.
- [Horn et al 1994] J. Horn, N. Nafpliotis and D.E. Goldberg. A Niche Pareto Genetic Algorithm for Multiobjective Optimisation. *Proc. of the IEEE Conference on Evolutionary Computation, ICEC'94*. 1. 82-87, 1994.
- [Hudson et al 1996] Hudson, R., M. Dempsey, and K. Keasey. A Note on the Weak Form Efficiency of Capital Markets: The Application of Simple Technical

Trading Rules to UK Stock Prices - 1935 to 1994, *Journal of Banking and Finance*, 20, 1121-1132. 1996.

[IBM IM] IBM Intelligent Miner: <http://www.software.ibm.com/data/iminer/>

[Investionary] <http://www.investionary.com>

[Investopedia] <http://www.investopedia.com>

[Irwin et al 1984] Irwin, S.H., and J.W. Uhrig. Do Technical Analysts have Holes in Their Shoes? *Review of Research in Futures Markets*, 3: 264-277. 1984.

[Jack 1996] Schwager, Jack D. *Schwager on Futures: Technical Analysis* (John Wiley & Sons, Inc., New York), 1996.

[Jackwerth 1997] Jackwerth, J. Do We Live in a Lognormal World? Finance Working Paper, London Business School. 1997.

[Jaeger et al 1996] Jaeger Manfred, Mannila Heikki, Weydert Emil. Data Mining as Selective Theory Extraction in Probabilistic Logic. In R. Ng, editor, *SIGMOD'96 Data Mining Workshop*, The University of British Columbia, Department of Computer Science, TR 96-08, 41-46, 1996.

[James 1968] James, F.. Monthly Moving Averages: An Effective Investment Tool, *Journal of Financial and Quantitative Analysis*, 3, 315-526. 1968.

[JDP] <http://www.javaworld.com/columns/jw-java-design-patterns-index.shtml>

[Jensen 1970] Jensen, M., and G. Bennington. Random Walks and Technical Theories: Some Additional Evidence, *Journal of Finance*, 25, 469-482. 1970.

[John 1986] Murphy, John J. *Technical Analysis of the Futures Markets: A Comprehensive Guide to Trading Methods and Applications* (New York Institute of Finance, New York), 1986.

[Joseph et al 1997] Joseph M. Hellerstein, Peter J. Haas, and Helen J. Wang. Online aggregation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 171-182, 1997.

[Kahn et al 1996] Kahn, R., and Stefek, D. Heat, Light, and Downside Risk, BARRA Research Memo. 1996.

[Kenneth 1996] West, Kenneth D. Asymptotic Inference about Predictive Ability, *Econometrica* 1996, 64, 1067-1084.

- [Kevin 2000] Dowd, Kevin. Adjusting for Risk: An Improved Sharpe ratio, *International Review of Economics & Finance*, Elsevier, vol. 9(3), pages 209-222. 2000.
- [Kho 1996] Kho, B.C.. Time-Varying Risk Premia, Volatility and Technical Trading Rule Profits: Evidence from Foreign Currency Futures Markets, *Journal of Financial Economics*, 41:249-290. 1996.
- [Knez et al 1996] Knez, P.J., and M.J. Ready. Estimating the Profits from Trading Strategies, *Review of Financial Studies*, Vol. 9(4), 1996.
- [Koza 1994] Koza, J. R. Genetic Programming II: Automatic Discovery of Reusable Programs. Cambridge, Mass; London, MIT Press. 1994.
- [Kraus et al 1976] Kraus, A., and Litzenberger, R. Skewness Preference and the Valuation of Risk Assets, *Journal of Finance* 1976, 31, 1085-1100.
- [L38] <http://www.L38.com>
- [Lajbcygier et al 2003] Lajbcygier, P., and Lim, E.. Trading Futures with the Largest Equity Drawdown Method. *Intelligent Data Engineering and Automated Learning* 2690, 929-933. 2003.
- [Lam et al 2000] Kin Lam, K.C. Lam. Forecasting for the Generation of Trading Signals in Financial Markets. *Journal of Forecasting*, 19(2000), pp. 39-52.
- [Lamsweerde et al 1998] van Lamsweerde, R. Darimont and E. Letier, Managing Conflicts in Goal-Driven Requirements Engineering, *IEEE Transactions on Software Engineering*, Special Issue on Managing Inconsistency in Software Development, Vol. 24 No. 11, November 1998, pp. 908 - 926.
- [Lawrence et al 1994] Blume, Lawrence, David Easley, and Maureen O'Hara. Market Statistics and Technical Analysis: The Role of Volume, *Journal of Finance*. 1994, 49, 153-181.
- [LeBaron 1992] LeBaron, B.. Do Moving Average Trading Rule Results Imply Nonlinearities in the Foreign Exchange Markets?. Working Paper, U. Wisconsin. 1992.
- [LeBaron 1993] LeBaron, B.. Nonlinear Diagnostics and Simple Trading Rules for High-Frequency Foreign Exchange Rates, in Gershenfeld, N., and A.

- Weigend, (eds.), 1993, *Predicting the Future and Understanding the Past: A Comparison of Approaches*, Addison-Wesley. 1993.
- [LeBaron 1996] LeBaron, B.. Technical Trading Rule Profitability and Foreign Exchange Intervention, NBER Working Paper 5505. 1996.
- [LeBaron 1998] LeBaron, B.. Technical Trading Rules and Regime Shifts in Foreign Exchange, in Acar E. and Satchell S. (eds.), 1997, *Advanced Trading Rules*, Butterworth Heinemann. 1998
- [Lee 1996] Lee, Ch.I., and I. Mathur. Trading Rule Profits in European Currency Spot Cross-rates, *Journal of Banking and Finance*, 20, 949-962. 1996.
- [Leigh et al 2002] Leigh, W., Modani, N., Purvis, R., and Roberts, T.. Stock Market Trading Rule Discovery Using Technical Charting Heuristics. *Expert Systems with Applications* 23, 155-159. 2002.
- [Leitch 1991] Leitch, G., and J.E. Tanner. Economic Forecast Evaluation: Profits Versus the Conventional Error Measures, *AER*, 81(3), 580-590. 1991.
- [Leland 1980] Leland, H. Who Should Buy Portfolio Insurance? *Journal of Finance* 1980, 35, 581-94.
- [Leuthold 1972] Leuthold, R.. Random Walk and Price Trends: The Live Cattle Futures Market, *Journal of Finance*, 27: 879-889. 1972.
- [Levich et al 1993] Levich, R. and L. Thomas. The Significance of Technical-Trading Rules Profits in the Foreign Exchange Market: A Bootstrap Approach, *Journal of International Money and Finance*, 12(5), 451-474. 1993.
- [Levy 1967] Levy, R.A.. Relative Strength as a Criterion for Investment Strategies, *Journal of Finance*, 22, 595-610. 1967.
- [Levy 1971] Levy, R.A.. The Predictive Significance of Five-Point Chart Patterns, *Journal of Business*, Vol. 44, No. 3, July: 316-323. 1971.
- [Li et al 2002] Li, W., and Lam, K.. Optimal Market Timing Strategies Under Transaction Costs. *Omega-International Journal of Management Science* 30, 97-108. 2002.
- [Likhatchev et al 2003] Likhatchev, A., G. Ratzer, et al. Financial Trading Systems: Neural and Genetic Algorithms: 111. 2003.

- [Lin et al 2004a] Li Lin, Longbing Cao, Jiaqi Wang, Chengqi Zhang. The Applications of Genetic Algorithms in Stock Market Data Mining Optimization. *Proceedings of Fifth International Conference on Data Mining, Text Mining and Their Business Applications*, 273-280. Malaga, Spain. September 15-17, 2004.
- [Lin et al 2004b] Li Lin, Chengqi Zhang. The Application of Fuzzy Sets in Finding the Best Stock-Rule Pairs. *The Proceedings of 5th International Conference on Recent Advances in Soft Computing*. Nottingham, United Kingdom. 472-476, December 16-18, 2004.
- [Lin et al 2005a] Li Lin, Longbing Cao, Chengqi Zhang. The Fish-eye Visualisation of Foreign Currency Exchange Data Streams. *The 4th Asia Pacific Symposium on Information Visualisation*, Sydney, Australia. 87-92, January 27-29, 2005.
- [Lin et al 2005b] Li Lin, Longbing Cao, Chengqi Zhang. The Visualization of Large Database in Stock Market. *The 25th IASTED International Conference on Databases and Applications (DBA 2005)*, Innsbruck, Austria, 163-166, February 14 - 16, 2005.
- [Lin et al 2005c] Li Lin, Longbing Cao, Chengqi Zhang. Genetic Algorithms for Robust Optimization in Financial Applications, *The 4th IASTED International Conference on Computation Intelligence (CI 2005)*, 387-391, July 4-6, Calgary Canada, 2005.
- [Lin et al 2005d] Li Lin, Dan Luo, Li Liu. Mining Domain-Driven Correlations in Stock Market. *The 18th Australian Joint Conference of Artificial Intelligence*. 979-982. 5-9 Dec 2005, Sydney, Australia.
- [Liu et al 1997] C. Liu and P. Cao. Maintaining Strong Cache Consistency in the World-Wide-Web. In *Proceedings of the 17th International Conference on Distributed Computing Systems*, pages 12-21, 1997.
- [Logue et al 1978] Logue, D., R. Sweeney, and T. Willett. The Speculative Behaviour of Foreign Exchange Rates During the Current Float, *Journal of Business Research*, Vol. 6, May: 159-74. 1978

- [Lopez-Fernandini 2001] Lopez-Fernandini, C.. Regulation FD of the SEC's Selective Disclosure and Insider Trading Rule: Finally, Full and Fair Disclosures. *Administrative Law Review* 53, 1353-1374. 2001.
- [Lotfi 1969] Lotfi A. Zadeh. Fuzzy Sets. *Information and Control*. 8: 338-353, 1969.
- [Lucke 2003] Lucke, B.. Are Technical Trading Rules Profitable? Evidence for Head-and-Shoulder Rules. *Applied Economics* 35, 33-40. 2003.
- [Lukac et al 1988] Lukac, L.P., B.W. Brorsen and S.H. Irwin. Similarity of Computer Guided Technical Trading Systems, *Journal of Futures Markets*, 8, 1-13. 1988.
- [Lukac et al 1989] Lukac, L.P., B.W. Brorsen and S.H. Irwin. The Usefulness of Historical Data in Selecting Parameters for Technical Trading Systems, *Journal of Futures Markets*, 9, 55-65. 1989.
- [Lukac et al 1990] Lukac, L.P., and B.W. Brorsen. A Comprehensive Test of Futures Market Disequilibrium, *The Financial Review*, v.25 n.4, 593-622. 1990.
- [Lyon 1990] Lyon, A.B.. Capital Gains Tax Rate Differentials and Tax Trading Strategies, Working Paper, U. of Maryland. 1990.
- [Malan et al 1997] G. R. Malan, F. Jahanian, S. Subramanian, and Salamander. A Push Based Distribution Substrate for Internet Applications. In *Proceedings of the 1997 USENIX Symposium on Internet Technologies and Systems*, 1997.
- [Mark 1992] Taylor, Mark. The Use of Technical Analysis in the Foreign Exchange Market, *Journal of International Money and Finance* 1992, 11, 304-314.
- [McFadden et al 1999] Fred R. McFadden, Jeffrey A. Hoffer, Mary B. Prescott. *Modern Database Management*. Ed 5. Addison-Wesley, c1999.
- [Meton 1973] Merton, R. An Intertemporal Capital Asset Pricing Model, *Econometrica* 1973, 41, 867-87.
- [Mihael 2002] Mihael Ankerst. The Perfect Data Mining Tool: Interactive or Automated? *Report on the SIGKDD-2002 Panel*. Edmonton Canada, 2002.

- [Narowcki 1984] Narowcki, D.. Adaptive Trading Rules and Dynamic Market Disequilibrium, *Applied Economics*, 16, 1-14. 1984.
- [Neely et al 1996] Neely, C., R. Dittmar, and P. Weller. Is Technical Analysis in the Foreign Exchange Market profitable? A Genetic Programming Approach, CEPR Discussion Paper 1480. 1996.
- [Neely 1997a] Neely, C. Technical Analysis in the Foreign Exchange Market: A Layman's Guide, *Federal Reserve Bank of St. Louis Review*, September/October 1997.
- [Neely et al 1997b] Neely, C. and P. Weller. Technical Analysis and Central Bank Intervention, Working Paper, *Federal Reserve Bank of St. Louis*. 1997.
- [Neely 1998a] Neely, C. Technical Analysis and the Profitability of US Foreign Exchange Intervention, *Federal Reserve Bank of St. Louis Review*, July/August 1998, 3-17.
- [Neely et al 1998b] Neely, C. and P. Weller. Technical Trading Rules in the European Monetary System, Working Paper, Federal Reserve Bank of St. Louis. 1998.
- [Neely et al 1999] Neely, Christopher J. and Weller, Paul A. Technical Trading Rules in the European Monetary System, *Journal of International Money and Finance*, Elsevier, vol. 18(3), pages 429-458. 1999.
- [Neely et al 2001] Neely, Christopher J. & Weller, Paul A., 2001. Technical Analysis and Central Bank Intervention, *Journal of International Money and Finance*, Elsevier, vol. 20(7), pages 949-970. 2001.
- [Neely 2002] Neely, C. J.. The Temporal Pattern of Trading Rule Returns and Exchange Rate Intervention: Intervention does not Generate Technical Trading Profits. *Journal of International Economics* 58, 211-232. 2002.
- [Neely et al 2003] Neely, C. J., and Weller, P. A.. Intraday Technical Trading in the Foreign Exchange Market. *Journal of International Money and Finance* 22, 223-237. 2003.

- [Neftci et al 1981] Neftci, S.N. and A.J. Policano. Can Chartists Outperform the Market? Market Efficiency Tests for Technical Analysis, *Journal of Futures Markets*, 4, 465-478. 1981.
- [Ng et al 1998] Ng. R., Lakshmanan, L., Han J. and Pang A. Exploratory Mining and Pruning Optimizations of Constrained Association Rules. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, ACM press Seattle, Washington, 13-24, 1998.
- [Ninan et al 2003] Anoop Ninan and Purushottam Kulkarni and Prashant Shenoy and Krithi Ramamritham and Renu Tewari. Cooperative Leases: Scalable Consistency Maintenance in Content Distribution Networks. In *Proceedings of the 11th WWW Conference*, Honolulu, Hawaii, USA, May 2003.
- [Nix et al 1992] Nix, A. and M. D. Vose. Modeling Genetic Algorithms with Markov Chains. *Annals of Mathematics and Artificial Intelligence*. 1992, 5, pp. 79-88.
- [NYSE] <http://www.nyse.com>
- [Okunev et al 2003] Okunev, J., and White, D.. Do Momentum-based Strategies still Work in Foreign Currency Markets? *Journal of Financial and Quantitative Analysis* 38, 425-447. 2003.
- [Olston et al 2000] Chris Olston and Jennifer Widom. Offering a Precision-Performance Tradeoff for Aggregation Queries over Replicated Data. In *Proceedings of the International Conference on Very Large Data Bases*, pages 144–155, 2000.
- [Olston et al 2001] Chris Olston, Boon Thau Loo, and Jennifer Widom. Adaptive Precision Setting for Cached Approximate Values. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 355–366, 2001.
- [Olston et al 2002] Chris Olston and Jennifer Widom. Best-Effort Cache Synchronization with Source Cooperation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 73–84, 2002.

- [O'Reilly 2005] O'Reilly, U.-M. *Genetic Programming Theory and Practice II*. New York, Springer Science+Business Media. 2005.
- [Olson 2004] Olson, D.. Have Trading Rule Profits in the Currency Markets Declined over Time? *Journal of Banking & Finance* 28, 85-105. 2004.
- [Osler et al 1995] Osler, C. L. and P. H. Kevin Chang. Head and Shoulders: Not Just a Flaky Pattern, *Federal Reserve Bank of New York Staff Report* 4. 1995.
- [Osler 1998] Osler, C.L.. Identifying Noise Traders: The Head and Shoulders Pattern in US Equities, Research Paper, *Federal Reserve Bank of New York*. 1998.
- [Pau 1991] Pau, L.F.. Technical Analysis for Portfolio Trading by Symmetric Pattern Recognition, *Journal of Economic Dynamics and Control*, 15, 715-730. 1991.
- [Pedersen et al 2004] Pedersen, H. H., and de Zwart, G. J.. Uncovering the Trend-Following Strategy - To Help Currency Managers. *Journal of Portfolio Management* 31, 94. 2004
- [Perry 1987] Kaufman, Perry J. *The New Commodity Trading Systems and Methods* (John Wiley & Sons, Inc., New York), 1987.
- [Peter 1997] Coy, Peter. He Who Mines Data May Strike Fool's Gold, *Business Week* June 16, 1997, 40.
- [Pictet et al 1995] Pictet, O.V., M.M. Dacorogna, B. Chopard, et al. Using Genetic Algorithms for Robust Optimization in Financial Applications, Olsen & Associates, Research Institute for Applied Economics. Switzerland. April 27, 1995.
- [Pictet et al 1996] Pictet, O.V., M.M. Dacorogna, R.D. Dave, B. Chopard, R. Schirru, and M. Tomassini. Genetic Algorithms with Collective Sharing for Robust Optimization in Financial Applications, *Neural Network World*, 5(4), 573-587. 1996.
- [Pratt 1964] Pratt, J. Risk Aversion in the Small and in the Large, *Econometrica* 1964, 32, 122-36.

- [Prinzie et al 2005] Prinzie, A., and Van den Poel, D. Constrained Optimization of Data-mining Problems to Improve Model Performance: A Direct-Marketing Application. *Expert Systems with Applications* 29, 630-640. 2005.
- [Pruitt et al 1989] Pruitt, S.W. and R.E. White. Exchange-Traded Options and CRISMA Trading, *Journal of Portfolio Management*, 15:4. 1989.
- [Ramazan 1998a] Gencay, Ramazan. Optimization of Technical Trading Strategies and the Profitability in Security Markets, *Economics Letters*, Elsevier, vol. 59(2), pages 249-254. 1998.
- [Ramazan 1998b] Gencay, Ramazan. The Predictability of Security Returns with Simple Technical Trading Rules, *Journal of Empirical Finance*, Elsevier, vol. 5(4), pages 347-359. 1998.
- [Ready 1997] Ready, M.J.. Profits from Technical Trading Rules, Working Paper, U. of Wisconsin. 1997.
- [Richard 1986] Sweeney, Richard J. Beating the Foreign Exchange Market, *Journal of Finance* 1986, 41, 163–182.
- [Richard 1988] Sweeney, Richard J. Some New Filter Rule Tests: Methods and Results, *Journal of Financial and Quantitative Analysis*, 1988, 23, 285–300.
- [Richard et al 1993] Levich, Richard and Lee Thomas, III. The Significance of Technical Trading-Rule Profits in the Foreign Exchange Market: A Bootstrap Approach, *Journal of International Money and Finance* 1993, 12, 451–474.
- [Robert 1932] Rhea, Robert. *The Dow Theory* (Fraser Publishing Co., Burlington, Vermont), 1932.
- [Robert 1987] Merton, Robert. On the State of the Efficient Market Hypothesis in Financial Economics. In R. Dornbusch, S. Fischer, and J. Bossons, eds.: *Macroeconomics and Finance: Essays in Honor of Franco Modigliani* (MIT Press, Cambridge, Mass.), 1987, 93- 124.
- [Robert 1999] Robert Pereira. Forecasting Ability but no Profitability: an Empirical Evaluation of Genetic Algorithm-Optimised Technical Trading Rules. Discussion Papers. Series A. LaTrobe University, Bundoora, Victoria, Australia, 99.06. July, 1999.

- [Roll 1978] Roll, R. Ambiguity when Performance is Measured by the Securities Market Line, *Journal of Finance* 1978, 33, 1051-1069.
- [Rubinstein 1976] Rubinstein, M. The Valuation of Uncertain Income Streams and the Pricing of Options, *Bell Journal of Economics and Management Science* 1976, 7, 407-25.
- [Rubinstein et al 1981] Rubinstein, M., and Leland, H. Replicating Options with Positions in Stock and Cash, *Financial Analysts Journal* 1981, 37, 63-75.
- [Ruiz et al 2002] Ruiz, E., and Pascual, L.. Bootstrapping Financial Time Series. *Journal of Economic Surveys* 16, 271-300. 2002.
- [Ryan et al 1999] Sullivan, Ryan, Allan Timmermann, et al: Data-Snooping, Technical Trading Rule Performance, and the Bootstrap. *Journal of Finance*, 54 (1999), pp: 1647-1692.
- [Saacke 1998] Saacke, P.. Technical Analysis and the Effectiveness of Central Bank Interventions, Working Paper, U. of Hamburg. 1998.
- [Saacke 2002] Saacke, P.. Technical Analysis and the Effectiveness of Central Bank Intervention. *Journal of International Money and Finance* 21, 459-479. 2002.
- [Salih 1991] Neftci, Salih. Naive Trading Rules in Financial Markets and Wiener-Kolmogorov Prediction Theory: A Study of 'Technical Analysis', *Journal of Business* 1991, 64, 549-571.
- [Sapp 2004] Sapp, S.. Are all Central Bank Interventions Created Equal? An Empirical Investigation. *Journal of Banking & Finance* 28, 443-474. 2004.
- [SAS EM] SAS Enterprise Miner: <http://www.sas.com/technologies/analytics/datamining/miner/>
- [Sharpe et al 1995] Sharpe, W., Alexander, G., and Bailey, J., *Investments*, 5th edition, Prentice Hall, Englewood Cliffs, N.J. 1995.
- [Shaughnessy et al 1997] O'Shaughnessy, James P.. What Works on Wall Street: A Guide to the Best- Performing Investment Strategies of All Time (McGraw-Hill, Inc., New York), 1997.
- [SHOE] <http://www.cs.umd.edu/projects/plus/SHOE/>

- [Silber 1993] Silber, W.L.. Technical Trading: When it Works and when it Doesn't, *Journal of Derivatives*, 1, Spring, 39-44. 1993.
- [SIRCA] <http://www.sirca.org.au>
- [Skouras 1997] Skouras, S. Analysing Technical Analysis, Working Paper, European University Institute. 1997
- [Smarts] <http://www.smarts.com.au>
- [Smidt 1965] Smidt, S. A Test of the Serial Dependence of Price Changes in Soybeans Futures, *Food Research Institute Studies*, 5: 117-136. 1965.
- [Sortino et al 1991] Sortino, F., and Vandermeter, R. Downside Risk, *Journal of Portfolio Management* 1991, 17, 27-32.
- [SPSS Clementine] SPSS clementine: <http://www.spss.com/clementine/>
- [Srinivasan et al 1998] Raghav Srinivasan, Chao Liang, and Krithi Ramamritham. Maintaining Temporal Coherency of Virtual Data Warehouses. In *Proceedings of 19th IEEE Real-Time Systems Symposium*, 1998.
- [Standardpoors] <http://www.standardpoors.com>
- [Stephen 1994] Taylor, Stephen. Trading Futures Using a Channel Rule: A Study of the Predictive Power of Technical Analysis with Currency Examples, *Journal of Futures Markets* 1994, 14, 215–235.
- [Stephen et al 1998] Stephen J. Brown, William N. Goetzmann and Alok Kumar. The Dow Theory: William Peter Hamilton's Track Record Re-Considered, New York University, Leonard N. Stern School Finance Department Working Paper Seires 98-013, 1998.
- [Stevenson et al 1970] Stevenson, R., and R. Bear. Commodity Futures: Trends or Random Walk, *Journal of Finance*, 25, 65-81. 1970.
- [Stoll 2003] Hans R. Stoll. Market Microstructure. *Working Paper, Financial Markets Research Center*, 2003.
- [Storey 1993] V.C. Storey. Understanding Semantic Relationships. *The Very Large Data Bases Journal*. 2(4):455-488, 1993.
- [Studer et al 1996] Studer R, Eriksson H, Gennari JH, Tu SW, Fensel D, Musen M. Ontologies and the Configuration of Problem-Solving Methods. In Gaines

BR and Musen MA (eds) *Proceeding of the 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada, 1996.

[Sweeney 1986] Sweeney, R.J.. Beating the Foreign Exchange Market, *Journal of Finance*, 41, 163- 182. 1986.

[Sweeney 1988] Sweeney, R.J.. Some New Filter Tests: Methods and Results, *Journal of Financial and Quantitative Analysis*, 23, 285-300. 1988.

[Sweeney et al 1989] Sweeney, R. J. and P. Surajaras. The Stability of Speculative Profits in the Foreign Exchanges, in R.M.C. Guimaraes, B.G. Kingsman and S.J. Taylor, *A Reappraisal of the Efficiency of Financial Markets*, New York: Springer-Verlag. 1989.

[Sweeney et al 1990] Sweeney, R.J., and E.J. Lee. Trading Strategies in the Forward Exchange Markets, in Raj Aggarwal and C.F. Lee (eds.), *International Dimensions of Securities and Currency Markets, Advances in Financial Planning and Forecasting Series Vol.4, part A.*, JAI Press, Greenwich, Conn., 55-79. 1990.

[Szakmary et al 1997] Szakmary, A.C. and I. Mathur. Central Bank Intervention and Trading Rule Profits in Foreign Exchange Markets, *Journal of International Money and Finance*, 16, 513-535. 1997.

[Tatur 2005] Tatur, T.. On the Trade off between Deficit and Inefficiency and the Double Auction with a Fixed Transaction Fee. *Econometrica* 73, 517-570. 2005.

[Taylor 1983] Taylor, S.J.. Trading Rules for Investors in Apparently Inefficient Futures Markets, in *Futures Markets – Modelling, Managing and Monitoring Futures Trading*, Basil Blackwell, Oxford, 165-198. 1983.

[Taylor et al 1989a] Taylor, S.J. and A. Tari. Further Evidence Against the Efficiency of Futures Markets, in R.M.C. Guimaraes, B.G. Kingsman and S.J. Taylor, *A Reappraisal of the Efficiency of Financial Markets*, New York:Springer-Verlag. 1989.

[Taylor 1989b] Taylor, S.J.. Profitable Currency Futures Trading: a Comparison of Technical and Time-Series Trading Rules, in Lee R. Thomas III (Ed.), *The Currency Hedging Debate*, IFR Publishing, London. 1989.

- [Taylor 1992] Taylor, S.J.. Rewards Available to Currency Futures Speculators: Compensation for Risk or Evidence of Inefficient Pricing, *Supplement to the Economic Record 1992: Special issue on Futures Markets*. 1992.
- [Taylor 1994] Taylor, S.J.. Trading Futures Using a Channel Rule: A Study of the Predictive Power of Technical Analysis with Currency Examples, *Journal of Futures Markets*, 14(2), 215-235. 1994.
- [Thomas 1990] Thomas, L.R.. Random Walk Profits in Currency Futures Trading, in Lee R. Thomas III (Ed.), *The Currency Hedging Debate*, IFR Publishing, London. 1990.
- [Thuraisingham 1998] Bhavani Thuraisingham. *Data Mining: Technologies, Techniques, Tools, and Trends*. CRC Press, 1998.
- [Thomas et al 1998] Thomas Hellstrom and Kenneth Holmstrom. Predicting the Stock Market. Dept. of Mathematics and Physics, Malardalen University. Technical Report Series IMA-TOM-1997-07, August 9, 1998.
- [TradeStation] <http://www.tradestation.com>
- [TriCom] <http://www.tricom.com.au>
- [Twocrows] <http://www.twocrows.com>
- [Van et al 1968] Van Horne, J.C., and G.C.C. Parker. Technical Trading Rules: A Comment, *Financial Analysts Journal*, XXIII, November-December 1967: 87-92.
- [Veit 2003] Daniel Veit. *Matchmaking in Electronic Markets*. Springer, 2003.
- [Vose et al 1991] Vose, M. D., and G.E. Liepins. Punctuated Equilibria in Genetic Search. *Complex Systems*. 1991, 5, no. 1, 31-44.
- [Vose 1996] Vose, M. D. Logarithmic Convergence of Random Heuristic Search. *Evolutionary Computation*. 1996, 4, No. 4, pp. 395-404.
- [Vose 1999] Vose, M.D. *The Simple Genetic Algorithm*. The MIT Press, Cambridge, Massachusetts. 1999.
- [White 2000] White, H.. A Reality Check for Data Snooping, *Econometrica* Vol. 68, No.5, 1097-1126. 2000.

- [William 1922] Hamilton, William P.. *The Stock Market Barometer* (Harper and Brothers Publishers, New York), 1922.
- [William et al 1992] Brock, William, Josef Lakonishok, and Blake LeBaron. Simple Technical Trading Rules and the Stochastic Properties of Stock Returns, *Journal of Finance*, 1992, 47, 1731–1764.
- [William et al 1998] William N. Goetzmann and Stephen J. Brown & Alok Kumar. The Dow Theory: William Peter Hamilton's Track Record Re-Considered, *Yale School of Management Working Papers ysm85*, Yale School of Management. 1998
- [Wright et al 1999] Wright A. H., Y. Zhao. Markov Chain Models of Genetic Algorithms, In *Proceedings of the Genetic and Evolutionary Computation Conference*, Orlando, Florida. 1999.
- [Yu et al 2000] Haifeng Yu and Amin Vahdat. Efficient Numerical Error Bounding for Replicated Network Services. In *Proceedings of the International Conference on Very Large Data Bases*, pages 123–133, 2000.
- [Zhang et al 2005] Chengqi Zhang and Shichao Zhang, In-Depth Data Mining and Its Application in Stock Market. *Proceedings of ADMA-04*, Wuhan, China, 2005 (Keynote Speech).