2009-01-01

New Algorithms for Optimal Portfolio Selection

Tanja Magoc
University of Texas at El Paso, t.magoc@gmail.com

NEW ALGORITHMS FOR OPTIMAL PORTFOLIO SELECTION

TANJA MAGOČ

Department of Computer Science

APPROVED:

François Modave, Ph.D., Chair

Vladik Kreinovich, Ph.D.

Oscar Varela, Ph.D.

Patricia D. Witherspoon, Ph.D.

Dean of Graduate School

NEW ALGORITHMS FOR OPTIMAL PORTFOLIO SELECTION

By

TANJA MAGOČ, M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science

THE UNIVERSITY OF TEXAS AT EL PASO

May 2009

Acknowledgements

I would like to thank a number of people who have encouraged me and made it possible for me to successfully come to the end of completing this dissertation and doctoral degree.

It is challenging to come up with words that would truly express all my gratitude to my Ph.D. adviser, Dr. François Modave. With his enthusiasm, his ideas, and his open-mindedness, he provided encouragement throughout the period of my research and dissertation–writing. His guidance and sound advices eased many steps on this long journey. His availability and effectiveness in discussions kept me going to reach success. Dr. Modave is one of the rare advisors that students dream that they will find.

I would also like to thank the members of my dissertation committee, Dr. Vladik Kreinovich and Dr. Oscar Varela, for accepting to be on my committee and allocating their time and attention to my work during a busy semester. In various occasions, Dr. Kreinovich's feedback helped me to improve my dissertation, while Dr. Varela's suggestions related to finance helped me to find data and methods appropriate for my work.

Moreover, I would like to thank the faculty and my peers in the Computer Science Department at the University of Texas at El Paso for their support during my studies.

Finally, I would like to thank my family for encouraging me and supporting me throughout my life.

Abstract

Over the past four thousand years, numerous techniques have been developed and used to address problems in Finance. These techniques include simple arithmetic calculations and probabilistic methods as well as intelligent systems techniques such as neural networks, genetic algorithms, multi-agent systems, and support vector machines. The techniques have been developed to accurately and quickly collect, validate, analyze, and integrate data that change dynamically.

The particular problem that we address in this dissertation is the construction of efficient algorithms for the problem of an optimal portfolio selection, that is, algorithms that would accurately and in real time determine the best distribution of wealth among several investment assets to achieve a specific goal. The main goal of making investments is to gain as much wealth as possible, therefore the goal is to maximize the return. However, the problem we face is that the investors do not know in advance what the return of each asset will be, but rather there are some educated predictions about the returns from each asset. These return predictions might turn out to be correct, but on the other hand, they might be completely wrong. Thus, there is a risk associated with each predicted return, and therefore with each asset. One of the goals of an investor is the minimization of risk. However, usually, a higher return is associated with a higher risk. Thus, it is impossible to maximize the return and minimize the risk independently. Moreover, additional characteristics, such as time to maturity, preferred portfolio structure, and reputations of companies that sell investments asset, among others, are also considered before locking money into an asset or a portfolio. This problem leads to the necessity of developing intelligent system techniques to find the best trade off between the return and the risk based on the preferences of an investor.

The techniques that are used to solve the problem of optimal portfolio selection all have their drawbacks. To solve these problems, we propose a new approach driven by utility-based multi-criteria decision making setting, which utilizes fuzzy measures and integration over intervals.

Table of Contents

List of Tables

1 Introduction

Numerous topics in finance have been studied over the past four thousand years. The earliest developments in the field of finance were based on arithmetic models, while later advancements have been heavily relying on the use of probabilistic techniques and stochastic models, as well as machine learning approaches [24].

Several computational techniques have been developed in order to collect data that change dynamically. Moreover, a wide variety of intelligent techniques have been built to validate, analyze, and integrate these data, and allow their use in different areas of finance such as determining a fair price of an asset or its derivative, selecting the best asset for money investment, or detecting the best distribution of wealth among several financial assets to diversify risk and achieve a high return at the same time. While pricing theory is mostly approached from a stochastic perspective, using models such as Black-Scholes [34], selection of an optimal portfolio requires development of intelligent systems techniques to analyze large financial data sets and extract information relevant to the problem ([36],[65]).

The particular problem that motivated this dissertation is a choice of an investment strategy that would lead to reaching a particular investment goal. Numerous investment assets are available for increasing wealth, ranging from low-risk bonds and high-risk stocks to their derivatives that include futures, forwards, options, and swaps among others. Each type of asset is defined by specific characteristics in terms of a method and timeline of payments, and thus selection of an optimal asset in not a straightforward process.

Moreover, all investment assets are characterized by numerous attributes. Among all the attributes, an investor is usually the most concerned about the return and the risk of the asset. However, other characteristics, such as the reputation of a company issuing stocks or bonds, the stability of a company, and the time to maturity of an asset, are also very important when making a decision about investments.

When considering a financial investment asset, an investor is mostly concerned with the return that this asset will yield. However, it is impossible to know in advance the exact return of an asset any time in future, therefore there is a risk of receiving a particular return associated with each asset. An investor would like to minimize the risk. However, typically, a higher value of the expected return of an asset yields a higher risk associated with that asset. This relationship between the return and the risk forces investors to invest money in several assets, that is the creation of an investment portfolio. A portfolio reduces the risk associated with an investment since the probability that multiple companies will perform poorly during a particular time interval is smaller than the probability of a single company failing. At the same time, the portfolio still receives a high expected return. In this dissertation, we particulary address the problem of an optimal portfolio selection.

Numerous techniques have been used to solve the problem of an optimal portfolio selection. Simple return-based strategies, methods based on stochastic processes, and intelligent systems techniques are the most often used approaches related to selecting an optimal portfolio. The return based strategies were the earliest methods for portfolio optimization, but are barely used anymore due to their neglect of the risk and other characteristics of an asset. Stochastic processes are useful tools in predicting the return of an asset, but are not used to determine how to distribute wealth among several assets.

Several intelligent systems techniques are widely used for selecting an optimal portfolio. Neural networks, genetic algorithms, rule-based systems including multi-agent systems, and support vector machines have all found applications in a portfolio selection. However, they all face certain drawbacks. With the exception of genetic algorithms, all the other mentioned intelligent systems rely on training data sets. Each element in a training data set is defined by its input(s) and output(s), which are known. The intelligent systems models (i.e., the coefficients in the models) are selected such that the models yield the correct outputs for given inputs in the training samples. Thus, these models are prone to overfitting to specific data used for training and might not perform well when other data are used. Moreover, in most cases, the intelligent systems approaches neglect dependencies among characteristics of assets such as return and risk and do not consider how changes in one criterion would influence changes in another criterion. Finally, the above methods assume that data are precise. However, the return and the risk of an asset are only predictions, and the reputation of a company is usually subjectively defined. Therefore, the data available in reality is not as precise as assumed in the models.

To address these drawbacks, we propose a novel method based on multi-criteria decision making setting and fuzzy integration over intervals. Fuzzy integration allows us to consider the dependencies among the characteristics of an asset, while data representation in terms of intervals rather than single numbers deals with imprecision faced in available and predicted data.

To address the problem of selecting an optimal portfolio, we start by briefly presenting basics of financial investments that outline existing options for investments along with their benefits and drawbacks. We also present the difficulties faced when making a decision on where to invest available wealth. We illustrate in more details the problem of selecting the best portfolio under various circumstances, and techniques used to solve this type of problem. We also describe hypotheses that are typically made when different methods are used as well as advantages and disadvantages of the most commonly used approaches in a portfolio selection. Then we present some tools leading to a novel approach, which involves multi criteria decision making setting from a utility perspective and intervals. The advantages of non-additive aggregation operators based on fuzzy integration over the other types of aggregation methods are also described. Lastly, we propose a novel approach to portfolio optimization which utilizes fuzzy integration and intervals to take into account dependence among characteristics of assets and imprecision of available data, characteristics of investment assets that are often neglected by other methods used to build an optimal portfolio. The theoretical model is followed by promising experimental results obtained through a set of testing data. We end with a few remarks on possible further improvements of the proposed method.

2 Finance Background

The field of finance covers numerous topics related to concepts of money and risk which evolve over time. It is people's nature to try to increase the amount of money that belongs to them without taking risk. The most commonly considered method to accomplish this task is to invest money in financial assets. However, investing money in assets requires a lot of thinking, calculations, and knowledge in order to select the assets that will lead to satisfying expectations of an investor.

Based on the methods that investors use to determine the best assets, we can divide investing in two main types: fundamental investing and technical (or chart-based) investing [35]. Fundamentalists analyze a company before investing money in it. They usually consider a company's price per earning ratio, competitive landscape, corporate initiatives, earning history, and dividend payout. On the other hand, chart investing just looks at the current chart of a company's performance. It is well-suited for short term investors as it is based on the assumption that a currently well-performing company will perform well in near future as well.

2.1 Investment Assets

Regardless of the type of investing, various investment assets are available for investors including stocks, bonds, derivatives (i.e., futures, forwards, options, and swaps), and commodities. We briefly present main characteristics of each of these assets in order to describe difficulties faced in deciding where to invest money.

2.1.1 Stocks

A stock (or a share) of a company is one of the simplest methods of investment. Stock-holders (i.e., investors in stocks) are owners of the company and therefore they receive a part of the earning of the company. This earning is usually in terms of dividends–monthly or yearly payments that can last indefinitely. The ownership of the company also ensures that if the company goes bankrupt, its owners share the inventory to compensate for their losses. However, there is no way to ensure that stock-holders will earn any money. A company might go bankrupt and not be able to compensate all investors, or company might be performing very purely and not earn enough money to pay dividends to its share-holders. Thus, there is an obvious risk of losing money or not gaining as much money as expected when making investments in stocks.

Various types of stocks exist, each of them carrying a different level of risk and return. Growth stocks are shares of companies that have steady growth even when economy is not doing well and therefore do not have as high risk as many other stocks. For example, stocks of food companies (e.g., Kellogg) and health-related companies (e.g., United Healthcare) fall into the category of growth stocks since people need to eat and might need to seek medical help even in times that are no so good for economy. On the other hand, cyclical stocks are shares of companies whose performance moves depending on the anticipated economic growth. When economy is growing, cyclical stocks tend to benefit as well. For example, technology companies (e.g., Motorola) and heavy machinery companies (e.g., Caterpillar) have much better results when economy allows potential buyers to spend money on these companies' instruments [35].

To determine which stock to buy, investors usually consider several factors including valuation, strategy, diversification, and willingness to risk [35]. Companies can be evaluated in several ways. The most common and the simplest method of evaluating a company is the company's earning since the profit that the company makes is either distributed to its stock-holders in terms of dividends or retained within the company. However, earning itself does not determine directly if buying a share of one company is better than a share of another company. Investors rather consider the price per earning ratio which determines the profit rather than the net income of an investor. Nevertheless, the price per earning ratio (i.e., the ratio of the price of a share and the earning per share over the last 12 months) is not a perfect measure of a company's performance either since small but growing companies often have a high starting price per earning ratio. Other common measures of a company's performance include the dividend yield (i.e., the dividend payment per share price), the price-to-book ratio (i.e., the ratio of the price of a share and the book value of the company, where the book value of a company is defined as the amount of money left if all assets would be sold and all the outstanding obligations would be paid), and the price-to-sales ratio (i.e., the ratio of the price per share and the revenue per share over the last 12 months).

Numerous stocks exist in markets, and information about them is available from several sources, one of them being Wall Street Journal. From the online resource of the Wall Street Journal, we can obtain information about different types of stocks [57]. For example, dividend stocks are separated based on the type of industry they belong to. Daily available data about dividend stocks from utility industries on April 3rd, 2009, are shown in the table 1. For each company, it shows the dividend amount and the dividend percent yield, the name and the symbol of the company, the market capitalization (i.e., the share price multiplied by the number of shares outstanding), the closing price, the change in price and the percent change in price, and the year to date percent change in price.

Table 1: Stocks being sold by utilities companies at Wall Street, April 3, 2009.

Div. amt.	Div. % yld.	Index/Core stock	Symbol	Mkt. cap.	Close	Chg.	% Chg.	YTD % Chg.
–	5.16	Utilities	–	–	1312.00	-10.06	-0.76	-8.7
$2.00	14.39	Ferrellgas Prtnrs	FGP	$941	$13.90	-0.17	-1.21	-5.2
2.58	11.76	Inergy LP	NRGY	1,126	21.94	-0.34	-1.53	28.8
2.72	10.29	Integrys Engy	TEG	2,021	26.44	-0.20	-0.75	-38.5
0.92	9.23	NiSource Inc	NI	2,731	9.97	-0.01	-0.10	-9.1
2.56	8.97	Amerigas Ptnrs	APU	1,621	28.54	-0.07	-0.24	1.5
1.28	8.80	Empire Dstrct	EDE	488	14.54	-0.09	-0.62	-17.4
1.08	8.80	Pepco Hldgs	POM	2,653	12.27	-0.10	-0.81	-30.9
2.70	8.44	Inergy HldgsLP	NRGP	647	32.00	-0.24	-0.74	47.7
1.24	8.41	Hawaiian Elc	HE	1,330	14.75	0.25	1.72	-33.4
1.42	7.83	Black Hills Corp	BKH	697	18.14	-0.23	-1.25	-32.7

Besides buying stocks, "borrowing" stocks is another type of investment practice involving this type of assets, which is known as *short-selling*. Short-selling of stocks is borrowing shares (i.e., instead of borrowing money) of a company by an investor who predicted that this company's performance will decline and the price of a share will decrease in price. The investor pays the "rent" on shares until he/she returns them to the company. If in the meantime the price of shares really decreased, the investor buys the shares he/she have borrowed from another investor at the lower price and earns the difference between the original ("borrowed selling") and the final ("buying") prices of the shares. The profit is decreased by the rent paid and the transaction fees. However, if the prediction by the investor was wrong and the price of shares did not decrease but rather increased, eventually the investor will need to buy these shares at a higher price and therefore loose some money. Thus, short-selling is not as simple as it might seem and is usually practiced only by experts.

2.1.2 Bonds

On the contrary to stock-holders who are partial owners of a company, bond-holders are lenders to a company. Companies sell bonds to pay off their debts or to invest into new businesses. In return for borrowed money, the company compensates the investors usually in terms of interest payments, called *coupons*, until the date of maturity (i.e., the date when all borrowed money, called *principal*, is returned).

The biggest seller of bonds in the United States is the U.S. government, which sells a large amount of Treasury bonds. States, cities, and counties sell so called municipal bonds to, for example, build a local school. Other companies sell other types of bonds to expand their businesses. For example, an electric company might sell bonds to build a new power plant. Since there are different types of bonds, each bond receives a rating that is based on the ability of the issuer's (i.e., the company borrowing money from investors) to repay the bond.

Treasury bond, if issued by a stable government, is usually the safest type of debt. There are several kinds of Treasury bonds based on the length of the time period before money is returned. The term "Treasury bond" usually refers to the bonds with the longest time to maturity, which is typically ten years or more. A "Treasury note" usually lasts two to ten years, while a "Treasury bill" (or a T-bill) is usually measured in weeks, most commonly thirteen to twenty-six weeks. Even though Treasury bonds (in the broad meaning of the term) represent the safest debt instrument for an investor, they are not without risk. The interest paid on treasury bonds is set in advance and is usually 4 to 5%. However, the inflation is a common characteristic of every economy. If the inflation during the bond's life increases to 6 or 7%, then the bond-holders are actually losing money since the interest they are receiving is lower than the inflation.

Municipal bonds usually pay a slightly higher interests to their buyers than Treasury bonds pay since states, cities, and counties are not as large and strong and stable as a government. Therefore they are more risky than Treasury bonds. Corporate bonds are bonds issued by numerous companies. They might be almost as safe as Treasury bonds, but they could also be very risky assets. They are divided into two main categories: investment-grade corporate bonds and junk bonds. The former are usually issued by companies with good history while the latter ones are sold by small unknown companies or companies with bad credit history. Since junk bonds lead to a high risk of not returning payments to the lenders, these bonds offer a higher return to attract investors.

A list of available bonds on the Amsterdam branch of Euronext Bonds market on April 3rd, 2009, is listed in the table 2. It shows whether currently the underlying company is increasing or decreasing its productivity, the name of the company, the market it belongs to, the volume of bonds traded in the last transaction, the coupon rate, and the maturity time given by date and time [18].

Table 2: Bonds available in Euronext–Amsterdam, April 3, 2009.

	Name		Volume	(%) Acc. Coupon	Maturity Time
↓	NEDER5, 5% 15JUL10	AMS	2,444,500	4.008	15/07/10 16:26
↑	NEDER 4% 15JUL18	AMS	2,396,239	2.915	15/07/18 15:43
↑	RB 4.75% 18	AMS	1,844,000	1.067	15/01/18 16:34
↓	CREDAGRIS, 8% 20	PAR	1,606,147	1.305	16/04/20 16:26
↑	NEDER 4% 15JUL16	AMS	1,601,550	2.915	15/07/16 15:30
↓	CASA 5.3% 11	PAR	1,282,154	1.207	15/07/11 16:26
↓	CASA, 5.2% 12	PAR	1,206,333	0.997	28/10/12 16:20
↑	NEDER 5% 15JUL12	AMS	1,192,500	3.643	15/07/12 16:26
↑	INGGROEP 8%	AMS	1,086,000	7.758	– 16:33
↑	NEDER 4% 15JUL19	AMS	978,000	0.580	15/07/19 16:08

Another type of bonds is a zero-coupon bond. Zero-coupon bonds do not pay an annual interest rate but are issued at a high discount. The investor gains from the difference between the discounted purchased price and the face value received at the maturity. In the other words, a zero-coupon bond is bought at a certain price and a higher return value is set in advance.

Finally, convertible bonds give an option to investors to convert the bond into a stock after some period of time. An investor benefits from this opportunity if the price of a share of the company is increasing, so rather than receiving interests and the principal invested, the investor owns a part of the company. Before the conversion is made, the investor receives interests as they would be paid on the bond. On the other hand, if the stock price of the company is not beneficial to the investor, he/she does not convert the bond into a stock and keeps the bond in its original state [35].

2.1.3 Derivatives

Even though stocks and bonds are the most common investment assets, coming in different flavors, there are other instruments in an investment market. Derivatives of these simple investment instruments are as important as stocks and bonds themselves. They are financial contracts between two parties and come in different forms. The most commonly used derivatives are futures, forwards, options, and swaps, which are defined thereafter.

Futures. Futures were first established in terms of trading goods for borrowed money [35]. For example, if a field worker needed money to raise corn, he would borrow money from a neighbor and would promise the neighbor to pay him back in corn when corn is ready for sale. They would make an agreement that the neighbor would buy corn at preset price with a transaction covering on a future date. This idea has been extended to stock and bond markets where a seller and a buyer of a futures agree on the price to be paid on a certain date in future for that particular financial asset. It is in the seller's interest to agree on the price that is higher than what the price of the asset will be on the agreed date for the trade. On the other hand, the buyer hopes to agree on the price that is lower than what the price will be on the expected date of trade. Considering the opposite interests of the seller and the buyer, the agreed price is usually the price of the asset on the trade date.

We can look at Liffe NYSE Euronext for available futures for certain regions. For example, the Netherlands stocks futures [62] on April 6^{th}, 2009, are listed in the table 3. Besides the name and the symbol of the company, the data include the type of the industry to which the company belongs, the last bid and the offer, and the price at which the transaction was made in the last period along with the number of shares traded in the last transaction.

Table 3: Stock futures in the Netherlands market as listed in Liffe NYSE Euronext, April 6, 2009.

Name	Symbol	Sector	Bid	Offer	Trade	Volume
ABN AMRO Holdings NV	AA	Banks	–	–	–	–
Aegon NV	AGN	Insurance	3.133	3.158	–	–
Akzo Nobel NV	AKZ	Health Care	30.567	30.673	–	–
ASML Holding NV	ASL	Technology	13.813	13.866	–	–
Fortis	FOR	Banks	–	1.384	–	–
Heineken NV	HEI	Food/Beverages	22.686	22.746	–	–
ING Groep NV	ING	Banks	5.153	5.204	5.118	8
Koninklijke Ahold NV	AHL	Retailers	8.361	8.409	–	–
Koninklijke DSM NV	DSM	Chemicals	19.889	19.988	–	–
Koninklijke Philips NV	PHI	Technology	12.122	12.162	–	–
Reed Elsevier	REN	Media	7.962	7.994	–	–
Royal Dutch SHELL PLC	RD	Oil & Gas	16.308	16.358	–	–
Royal KPN NV	KPN	Telecommunications	9.878	9.912	–	–
TNT NV	TPG	Personal&Household	13.21	13.264	–	–
UNIBAIL-RODAMCO SA	RCE	Financial Services	110.863	111.634	–	–
Unilever NV	UNA	Food/Beverages	15.227	15.274	–	–
Wolters Kluwer NV	WLS	Media	12.658	12.742	–	–

Forwards. Forwards are very similar to futures with the only difference being the method of trading. While futures fall under exchange trading, forwards are traded over-the-counter. Exchange trading involves the third party, that is a facility for trading, which is usually a bank. On the other hand, over-the-counter trading is performed directly between two parties. Thus, to undo a future action, it is possible to set an offset transaction, while undoing a forward contract requires both parties to agree on it, which makes undoing forward contracts much harder [17].

Options. Options are similar to futures and forwards in that they are also agreements on a price, called the *strike price*, to be paid on a certain day in the future if the buyer decides to exercise the option. The difference between a future and an option is that the buyer of a future contract has an obligation to buy the underlying asset on the expiration date, while a buyer of an option contract has the right but not the obligation to buy the underlying asset. Thus, an option buyer is accounted for paying a small fee, *a premium*, for creating an option, but on the day of maturity, he/she can decide if it is more worth buying the asset or just losing the initial fee that was paid for the contract [35].

The option contracts could be created for both–buying and selling investment assets. An option to buy an underlying assets is called a *call option*, while an option to sell an asset is called a *put option* [34], and the relation between their premiums is governed by put–call parity.

There are different types of options among which American and European options are two most commonly exercised ones [17]. An American option can be exercised on any day up to the expiration date (i.e., the agreed date for "trade"), while a European option may be exercised only on the expiration date. Both American and European options are called vanilla options as they are standardized and less interesting than so called exotic options. There are many different types of exotic options. However, a common characteristic among all of them is that the price of an exotic option usually depends on more than one value. For example, the price of an exotic option might depend on the average value of the values of an underlying asset at several points in time (i.e., Asian option), the value of reaching a certain level (i.e., barrier option), the value of foreign exchange rates (i.e., quanto option), etc.

Table 4: Stock options in Euronext market, April 3, 2009.

Name	Market	Code	Vol.	O.I.
ING GROEP	AMS	ING	101,573	3,536,027
ARCELOR MITTAL	AMS	MT	40,346	452,188
ROYAL DUTCH SHELL A	AMS	RD	34,329	2,167,220
KON. PHILIPS ELECTRONICS	AMS	PHI	29,028	1,208,611
KON. KPN	AMS	KPN	19,780	1,528,685
ALCATEL	PAR	CG1	16,301	743,703
EADS	PAR	EA1	15,883	83,373
VIVENDI	PAR	EX1	15,850	202,961
AEGON	AMS	AGN	14,097	2,231,199
AKZO NOBEL	AMS	AKZ	12,232	231,199

As an example, we can look at Euronext market to find information about available options in this market [56], which covers the markets of Amsterdam, Belgium, London, and Paris. The table 4 shows the data on April 3rd, 2009. For each option, we can find the name of the company, the market it belongs to (i.e., Amsterdam market, Paris market, etc.), the code of the company, the volume (i.e., the number of contracts traded in the most recent transaction), and the open interest (i.e., the outstanding long and short positions of the previous trading).

Table 5: Alcatel option–codes and classification, April 3, 2009.

Code	CG1	Market	Liffe Paris	Vol.	16,301	03/04/09
Exercise type	American	Currency	€	O.I.	743,703	02/04/09

Table 6: Alcatel option–underlying stock, April 3, 2009.

Name	ALCATEL	ISIN	FR0000130007	Market	Euronext Paris
Currency	€	Best bid	1.541 03/04/09 16:58	Best ask	1.542 03/04/09 16:58
Time	CET	Last	1.541 03/04/09 16:58	Last change %	1.38
Volume	17,208,344	High	1.595	Low	1.497

Table 7: Alcatel option–prices, April 3, 2009.

Calls								Puts						
Set.	Day vol.	Vol.	Time CET	Last	Bid	Ask	Str.	Bid	Ask	Last	Time CET	Vol.	Day vol.	Set.
0.24	–	–	–	–	0.25	0.29	1.3	0.01	0.03	–	–	–	–	0.02
0.16	1	1	10:25	0.19	0.17	0.20	1.4	0.02	0.05	–	–	–	–	0.04
0.06	4,007	7	12:16	0.09	0.06	0.09	1.6	0.10	0.13	0.12	11:57	100	200	0.14
0.01	14	14	15:41	0.03	0.01	0.03	1.8	0.24	0.29	–	–	–	–	0.29
0.01	–	–	–	–	–	0.02	2	0.43	0.47	–	–	–	–	0.48

Besides the basic information about all the investment assets available in a market, we can also obtain additional information about any individual asset. The data available about the option of Alcatel [1] are shown in the tables 5, 6, and 7.

Swaps. While futures, forwards, and options are contracts that involve selling or buying a simple asset, the swap is an agreement between two parties to exchange one stream of cash flow for another stream. The best known swap contract is the international currency exchange. The exchange rates between the currencies try to ensure that money in one country does not have higher power than money in another country. This is accomplished by following the *Law of one price* (i.e., purchasing power parity) which states that if p_A units of currency in a country A is traded at the exchange rate E for p_B units of currency in a country B, then the same items could be bought for the amounts p_A and p_B in the countries A and B, respectively [17]. However, if a country is going through an economic recession period or a high inflation period, it is worth investing domestic currency in outside markets.

As a simple example of the information available to public, we can look at the exchange rates of the most traded currencies. The table 8, which is updated on a daily basis and available online at [12], shows the information on the amount of one currency necessary to buy one unit of another currency on April 7th, 2009.

Table 8: Currency exchange rates, April 7, 2009.

	DOLLAR	EURO	POUND	SFRANC	PESO	YEN	CDNDLR
Canada	1.2392	1.6455	1.8288	1.0837	0.09116	0.01233	–
Japan	100.48	133.43	148.29	87.871	7.3915	–	81.085
Mexico	13.594	18.051	20.062	11.888	–	0.13529	10.970
Switzerland	1.1435	1.5185	1.6876	–	0.08412	0.01138	0.92277
U.K.	0.67760	0.89978	–	0.59257	0.04985	0.00674	0.54680
Euro	0.75307	–	1.1114	0.65856	0.05540	0.00749	0.60771
U.S.	–	1.3279	1.4758	0.87451	0.07356	0.00995	0.80697

2.1.4 Commodities

A commodity is something for which there is a demand, and its quality is the same regardless who produces it. For example, oil, gas, gold, and silver are commodities. The price of a commodity is universal and it fluctuates daily depending on global supply and demand. Thus, this price is a function of the universal market.

2.2 Selection of an Asset for Investment

Given numerous types of financial assets available for investment, the questions arises of how a particular investor selects an asset in which to invest money. The primary concern that comes to someone's mind is whether an investor could earn a large amount of money without taking any risk. More precisely, the matter is whether there is a new available stock that is underpriced, or whether an individual could earn money by converting money into different currencies until a circle is completed and the original currency is obtained, or whether there is any other way to earn money quickly.

Ensuring that arbitrage does not exist. To ensure that it is not possible to earn money by buying an asset and selling it right away, pricing of the asset needs to be correct. Assigning a price to an asset is not a simple process. The asset needs to be priced so that both the seller and the buyer will be happy. If an asset is underpriced, then the seller will not be happy since the buyer could immediately sell the asset for a higher price and earn a profit without doing anything. This situation is known as *arbitrage*. On the other hand, if an asset is overpriced, then not all shares will be sold and the price will have to decrease in order to sell them all. This situation will make the original buyers unhappy as they paid a higher price than the shares are worth. To ensure that both the seller and the buyer are happy, the correct price needs to be determined for the asset.

Various techniques exist for pricing distinct financial assets since different factors need to be considered when determining the price of basic assets compared to their derivatives. For example, when a company decides to go public and sell shares, the price of one share depends only on the company and this price is the current value of the stock. On the other hand, when an option is set for sale, the price of the option depends on the price of the underlying asset some time in future, and thus the prediction of the stock's price on (or before) the expiration date needs to be calculated. Furthermore, the currency exchange rates depend on the demand for a particular currency, and the economy of the involved countries.

Pricing of a stock is usually based on approximating how much investors would be willing to pay for a share [35]. Usually, a contract is made between the seller and an investment bank that sells all the shares. The contract contains the price, *initial price offer (IPO)*, at which a share will be paid to the seller, and the bank is responsible for selling all the shares that are listed in the contract.

However, this approach has a major drawback since the bank can take the advantage and price the stock low or sell the shares to its preferred customers. To solve this problem, Dutch auction model is often used. This model sets the IPO at the highest possible price that will result in selling all the shares. The price is determined by letting investors bid on the shares. The bids consist of the number of shares desired by an investor and the price the investor is willing to pay. The investors bid blindly and independently from each other. When all the bids are made, the investors that bid the highest are given the priority. If the investor with the highest bid did not bid on all shares, then the next highest bid is considered and the numbers of shares bid by both investors are added. If there still exist shares that are not distributed, another investor is added to the set of investors that will buy some shares. The process continues until all shares are distributed. Finally, the bid of the last investor added to this set of investors is the IPO, and all the investors from the set buy the requested shares at the IPO price.

Pricing of a stock seems pretty simple. However, pricing of its derivatives is much more complicated. To assign a correct price to an option, a future, or a forward, it is necessary to predict the behavior of the underlying company until the expiration date. Tree methods, Monte Carlo simulation, and Black-Scholes equation are often used to price an option [34].

Tree methods rely on determining all possible states of the company. It makes the predictions how many shares will be sold at each different price and ensures that whatever state occurs, the final output for the company will not change. One-step two-state tree is the simplest method of this type. It assumes that only two different states could be achieved, and usually an equal probability of reaching each state is assumed. More complex probabilistic models, multiple state models, and multiple step models are used to represent more precisely real life situations.

Compared to tree methods, which consider a fixed increase or decrease in the asset price, Monte Carlo simulation is based on considering a fixed increase or decrease in the logarithm of the price of an asset. The price of a share is repeatedly calculated from the log-distribution, and the results are averaged to derive the asset's price.

The most commonly used method for option pricing is the Black-Scholes model, which states that the price of an asset satisfies a certain partial differential equation known as the Black-Scholes equation:

$$\frac{\partial C}{\partial t} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} = rC, \tag{1}$$

where $C(S, t)$ is the price of the option at time t.

Finally, finding correct exchange rates for international currencies is based on the demand for the currency of a particular country. It is possible that an investor for a moment observes an arbitrage opportunity by converting money into a different currency. However, if a large amount of a particular currency is suddenly bought, it leads to a change in the exchange rate of the currency to close the gap for arbitragers.

Since there is no good chance for arbitrage in any particular market, local or international, investing money in assets requires a lot of thinking, calculations, and knowledge in order to select the assets that will lead to satisfying expectations of an investor. Of course, not all investors have the same goals, characteristics, and beliefs. Thus, the assets selected by distinct investors differ. However, all investors typically consider the same characteristics of assets before they make their decision. The most important features of an asset are its return and its risk. However, neither one of these attributes has a precise and universal definition. Therefore it is worth looking at different tools that have been used to characterize the return and the risk of an asset.

2.2.1 Return

The return of an asset is the earning that this asset brings to its owner. However, different assets collect gains at different times and often more than just once. For example, stocks receive dividends on a monthly, quarterly, or yearly basis, while bonds collect interests several times during their lifetime. Thus, we need to take into consideration earnings at all instances to accurately represent the return of an asset.

There are several common ways to calculate the return of an asset [17]. The simple return (or arithmetic return), R, is calculated over one time period. It is evaluated as the sum of the capital gain (or loss), C, and the dividend yield, D, and it is expressed as the percent in increase (or decrease) of the original price of the asset.

Definition 2.1. **[Simple return]**: *Simple return R is given by*

$$R = C + D = \frac{p_e - p_b}{p_b} + \frac{D}{p_b}, \tag{2}$$

where p_b and p_e represent the price of the asset at the beginning and at the end of the time interval, respectively.

However, investments are usually made over several consecutive time periods, where each period yields a simple return of R_i. If the total number of time intervals is T, we can define total return in several ways, among which arithmetic and geometric mean returns are the most commonly used ones [17].

Definition 2.2. **[Arithmetic mean return]**: *Arithmetic mean return AM of a series of T consecutive returns R_i is given by*

$$AM = \frac{R_1 + R_2 + \cdots + R_T}{T}. \tag{3}$$

Definition 2.3. **[Geometric mean return]**: *Geometric mean return GM of a series of T consecutive returns R_i is given by*

$$GM = ((1 + R_1) \cdot (1 + R_2) \cdot \cdots \cdot (1 + R_T))^{\left(\frac{1}{T}\right)} - 1 = \left(\frac{p_T}{p_1}\right)^{\left(\frac{1}{T}\right)} - 1, \tag{4}$$

where p_T and p_1 are the terminal and the initial prices of the asset, respectively.

Even though the arithmetic mean return is easy to calculate, its application is limited to cases in which we want to find the increase (or decrease) of capital on average during an entire time period. On the other hand, the geometric mean return represents the capital evolvement over an investment period much more accurately.

The return of an asset is the most important characteristic of an asset as investors want to gain as much earing as possible. We can easily calculate the return of an asset in the past, but we do not know the exact return of an asset in the future, which makes investment decisions much more complex. When making decisions, an investor relies on an expected (i.e., predicted) return of an assets. The price of an asset is predicted based on the current price of the underlying asset. Based on the *weak efficiency of markets* (or *Markov property*) states that all information about an asset is already encoded in its price and therefore, it is pointless to predict a future price of an asset based on its past prices [34]. However, not everyone believes in this theory, and some experts still use past data to predict the future price of an asset [28].

However, it is very difficult (if not impossible) to predict exactly the future behavior of an asset. Thus, it is hardly believable that the prediction of the return of an asset will always be accurate. However, the predicted return of an asset, which typically involves a probability distribution, is the best option that is available to an investor before the fact and is therefore the information that is used when making investments.

Prediction of the return of an asset. Different methods have been used to predict the return of an asset. Univariate [58] and multivariate [66] nearest neighbor methods, regression analysis [2], and non-linear analysis ([21],[42]) are a few commonly used methods to predict the return of an asset in the next time interval.

Univariate nearest neighbor method considers the time series of past returns [58]. It relies on the belief that if the last N time intervals had similar returns to N consecutive returns in past, then the return of the next time interval should be similar to the $(N+1)^{\text{st}}$ return from historic data. To apply this method, we considers the last N returns as a vector $\mathbf{v} = (v_1, v_2, ..., v_N)$, and find the similarity of this vector to all other vectors of N consecutive returns in past. The similarity is usually calculated as

$$d(\mathbf{v}, \mathbf{x}) = \left(\sum_{i=1}^{N} (v_i - x_1)^2 \right)^{\left(\frac{1}{2}\right)}, \tag{5}$$

where $\mathbf{x} = (x_1, x_2, ..., x_N)$ is the vector of N consecutive returns in the past. If the similarity, d, is smaller than a predefined threshold T, then vector $\mathbf{x}$ is considered a neighbor of vector $\mathbf{v}$. When all neighbors of vector $\mathbf{v}$ are found, the weight is distributed among all the neighbors. Usually a higher weight is given to a neighbor that has smaller distance from the vector $\mathbf{v}$. Then the weighted sum of the $(N+1)^{\text{st}}$ elements of the neighbors is calculated. The $(N+1)^{\text{st}}$ element of a neighbor is the return that happened in the very next time interval after the vector representing the neighbor ends. The weighted sum is the predicted return of the underlying asset for the next time interval.

Even though this method appears to be a good prediction method for the return of an asset, it presents a drawback. The theoretical problem with this approach lies in the framework of what are efficient markets, that is, if we found a historical replication of trends in the series of stock returns, then we should be able to take advantage of the trend. However, if everyone follows this principle, then by taking the advantage of the knowledge, the arbitragers would make the trend replication disappear. According to Fama's weak form of efficient market hypothesis [19], since stock prices reflect well historical information pertaining to it, we cannot use historical information to earn abnormal returns. Thus, by using this method, we can not predict exceptions (anomalies) in the returns.

Multivariate nearest neighbor method incorporates other characteristics with known values in the selection of the nearest neighbors. The distance function is defined as a weighted sum of distances between individual characteristics. In [66], trading volume is used as the second variable for the selection of the nearest neighbors.

A regressions analysis approach [2] compares the return of a particular asset to the return of the entire market in a specific time interval. A linear regression function is calculated based on all available data from past, where the market return is taken as the input (i.e., independent variable) and the asset return as the output (i.e., dependent variable) of the function. The constructed regression function is used to calculate the return of the asset for the following month.

The problem with this approach is that, to calculate the return of an asset in the next time interval, the return of the market for the next time interval is required, which is obviously not known. Thus, the market return has to somehow be predicted before the return of the asset could be anticipated. Usually, instead of predicting the market return for each upcoming time interval, the long historical average of the market return is used as the predicted market return.

Univariate and multivariate nonlinear methods ([21],[42]) extend the known linear methods, such as the nearest neighbor method and the regression analysis method to non-linear levels. Higher order polynomial functions and neural networks are trained to represent data more precisely than the linear methods. To gain more precision, these methods sacrifice the simplicity. They also suffer the same drawbacks as their respective linear models.

2.2.2 Risk

Since the actual return of an asset is known only after the fact and not at the moment when an investment is made, there is risk associated with the predicted return. The risk appears in several forms. First of all, there is a risk associated with the amount of return. Usually a company does not receive the same return during each time period, and therefore, there is a volatility in return. Thus, there is a risk whether the company will yield the same, a higher, or a lower return than during the previous time interval. Moreover, there are companies that go bankrupt and are therefore not able to pay thier owners or lenders. Usually Treasury bonds issued by a government of developed countries are low-risky assets since the government is likely to pay back the borrowed money. On the other hand, a newly established company is more likely to fold and not be able to pay the investors. Therefore, before considering the return of an asset, investors needs to take into consideration the risk of investing in a particular asset.

Risk could be defined as the uncertainty of the future return [34]. However, the return itself is not the only factor that determines the risk. The inflation is another factor that impacts the riskiness of an asset. If the actual return is equal to the predicted return, but the inflation is higher than the return, the investor would probably not be happy. On the other hand, if the return is lower than predicted but higher than the inflation, the investor would still not be happy, but would certainly appreciate more the positive impact of the return in the second case compared to the first case. Thus, the riskiness of an asset could be measured by looking if the asset's return is higher or lower than the current inflation. Based on Fisher Effect [38], a rise in the expected inflation rate will eventually cause an equal rise in the interest rate. However, the problem is that we can only estimate inflation. Thus, we need to find a different way of defining the risk of an asset.

The most common measure of risk is the volatility of returns. More volatile returns lead towards harder predictions of a future return and therefore a higher risk of a wrong prediction. The volatility of an asset is usually measured by standard deviation of returns [17]:

Definition 2.4. **[Standard deviation of returns]**: *Standard deviation of returns (i.e., the risk of an asset), SD, of a series of T consecutive returns R_i is given by*

$$SD = \sqrt{\left(\frac{1}{T}\right) \cdot \sum_{i=1}^{T} (R_i - AM)^2}, \tag{6}$$

where AM is the average mean return AM.

The return and the risk of an asset are certainly two very important characteristics that need to be considered when determining the best asset in which to invest money. However, return and risk are not the only features that need to be taken into consideration.

2.2.3 Other Characteristics that Influence the Selection of an Asset for Investment

Time horizon, time to maturity, fees and taxes, the reputation of a company, the stability of a company, the market to which company belongs, the location of an asset within the market, and the state of the economy are just a few of numerous possible features that can influence investment decisions.

Time horizon is the time that an investor has allocated to increase his/her wealth. Sometimes, an investor wants to quickly earn money to invest it in a business, or to buy a house. On the other hand, an investor wants to build a safe retirement living. Of course, time horizon in these two cases has different durations. In the first situation, the investor might allocate a few months or a few years to gain wealth, while the later situation is more likely to last decades. Thus, when determining an optimal portfolio, it is important to consider the goal as well as time limitation to achieve the goal.

Time to maturity is the time until an asset expires. For example, if money is invested into a bond, besides the interests, the investor will receive payments until the date on which the bond issuer will return all borrowed money. Thus, if an investor has a goal to increase wealth quickly, he/she needs to make sure that the bond expires before money is needed.

Fees are usually the costs associated with transactions for buying or selling an asset. The transaction fee could come in two forms: a fixed amount of money or a predetermined percent of money being invested. When deciding where to invest money, the amount of money that will be lost due to transaction fees needs to be considered. Rather than considering the pure return of an asset, transaction costs should be subtracted from the return to give a better idea of the profit gained by a particular investment.

The reputation of a company might have an impact on the decision about investments. The reputation could be described by different means. The most commonly used way to determine the reputation of a company is by the trust of its clients. Moreover, the trust of the clients could be based on different characteristics and behavior of the company. One possible way to describe the reputation of a company is to determine if the company has achieved the predicted returns in the past. Another option is to determine if a company belongs to the top 10% or 20% of the companies based on past returns. The type of the industry to which the company belong, the age of the company, and the courtesy of personnel employed in the company might all influence the reputation of a company. There are other ways to determine the reputation of a company. Regardless of the definition of a company's reputation, it is more likely that an investor wants to invest money in companies that have a good reputation rather than companies with a bad reputation. Even though the reputation of a company could be considered as a part of the risk to invest in a particular asset, usually risk is defined in different terms, so it is important to consider a company's reputation separately.

The stability of a company is also an important characteristic that should be considered when selecting an asset for an investment. The stability is closely related to the volatility of the return and therefore to the risk, but it differs from the risk in that, while a high risk might be desired by some investors in order to possibly obtain a high return, a high instability is not desirable since the investor would not know what to expect as an actual return. The stability of a company usually reduces the risk of investing money in this company and therefore attracts risk-averse individuals.

Another important aspect to consider when investing wealth is the location of a particular asset in a particular market as well as the strength of the market to which the asset belongs. A new company is more likely to have a weak reputation in a well-established market since there is no proof of its performance and therefore, the risk and the stability of the company are unknown, which makes the company not as attractive for investment as long lasting stable companies.

When considering the impact of a market on a particular asset, if a market is strong, it would be expected that all companies from this market will perform relatively well as they will be pushed by the market if they want to survive. On the other hand, in a poorly performing market as a whole, even a strong company might not receive as much respect by investors as the performance of the company might be expected to drop due to nonexistence of a healthy competition.

Finally, the current economic situation of a particular country or a specific region could have effects on the selection of particular stocks for investment. It is believed that the economy goes through business cycles, where each cycle consists of growth and recession [35]. Usually, growth takes longer time than recession. It is important to know the state of the economy to determine whether to invest money in a stock of, for example, *General Motors* or in a Treasury bond issued by a government. The behavior of the economy is determined by several factors including employment data, productivity, consumer inflation, factory capacity, food prices, inventories, and retail sales, among others.

As measures of employment, the number of jobs recently created and the unemployment rate are the most frequently used. If the economy is doing well, more jobs are created and we see a lower unemployment rate. On the other hand, if the economy is in a recession phase, job numbers shrink and unemployment rate increases.

The inflation is usually measured by the Consumer Price Index (CPI), which measures a basket of common goods that consumers purchase on a regular basis (e.g., food, energy, rent, clothes, transportation, medical care, etc.). The CPI calculates the increase/decrease in cost of living by calculating the expenses for the same basket of goods at different moments. Stock and bond holders do not like an inflation. For bond-holders, an inflation means that they might lose money since the inflation might be higher than the fixed interest they are getting paid. Similarly, stock-holders do not like an inflation because the inflation increases interest rates, which yields more expensive financing. Moreover, the inflation usually slows down economy, which leads to shrinking job opportunities and higher unemployment rate.

Total retail sales also give insights into economy. People who lost their jobs or are afraid that they might lose the jobs become more careful about their spending especially for big investments such as cars. If the number of cars sold drastically decreases, that is a sign of economy going down.

The return, the risk, time horizon, time to maturity, transaction fees, the reputation of a company, the stability of a company, the location of a company within a market, the strength of a market, and economy are just a few of numerous characteristics that could and should be considered when determining where to invest money.

2.3 Dependencies among the Characteristics of an Asset

Besides numerous characteristics of an asset that need to be considered when determining where to invest money, it is also important to notice that these characteristics are not independent from each other. We have already mentioned a few very obvious dependencies, such as that a higher return is usually accompanied by a higher risk. However, many other characteristics are also related.

For example, a higher time to maturity usually yields a higher return since there is more time to accumulate earnings, but there is also more time for the underlying company to default, thus creating a higher risk of investing money in a particular asset. The reputation of a company is partially based on the return of the company, and therefore a high return positively influences the reputation of a company. The stability of a company is directly related to the risk as a higher risk usually yields a higher instability of a company. The strength of a market is directly related to the reputation of a company in the market as companies in well-established markets tend to be pushed by the entire market to perform well in order to survive. The reputation of a company is directly related to the location of the company in a particular market as new companies are usually weak until they establish the trust of investors.

These and many other characteristics of an asset are mutually dependent and should not be considered completely independently.

2.4 Investment Portfolio

Taking into consideration all the characteristics of an investment asset and the dependence relations among them, it is not easy to establish which asset is the most promising one. The problem could easily be represented by just considering the return and the risk of an asset. An investor would like to maximize the return and at the same time minimize the risk. However, these two goals are contradictory since the return and the risk of an asset are not independent and furthermore, they tend to disperse in the opposite directions from each other. No one wants to invest in a high risk asset if the gain will be the same as from investing in a low risk asset. Thus, high risk assets need to offer higher return rates in order to attract investors. Therefore, higher the return of an asset usually implies higher the risk of the asset and vice versa. Thus, an investor needs to decide if he/she wants to invest in an asset that will result in the maximal return or in an asset that has the minimal risk. Another option available to an investor is to find the tradeoff between the return and the risk and invest money in an asset with an average return and an average risk, where the "average" return and risk could be defined in different ways by different investors.

However, these are not the only options that an investor has. An investor might decide to invest in more that one asset, some of the assets having a higher return while other assets having a higher risk. This structure is called an *investment portfolio*.

The creation of modern investment portfolio theory dates back to early 1950s, when Nobel prize winner in Economics Sciences, Harry M. Markowitz, determined that the expected return of a portfolio can not be increased without taking an additional risk. His initial *expected return-variance of return* model [45] is still widely used as a starting point in selecting an optimal portfolio. Markowitz noted that since the return of a portfolio is not guaranteed but rather approximated (i.e., expected), there exists a variance in this return causing the risk of receiving the expected return. Thus, the goal of an investor should be to maximize the return and minimize the risk, which is accomplished by distributing wealth among several investment assets with low correlations.

However, as Markowitz noted, returns from investment assets are intercorrelated, and therefore the complete elimination of risk is impossible. The portion of the risk that could be diversified by investing money in several assets rather than in one asset is company-related risk, so called *unsystematic risk*, while the *systematic risk* is the risk that depends on a market and could be reduced only by investing in other markets, particularly foreign markets that are not heavily dependent on the domestic market.

Since it is desirable to reduce the risk of an investment while increasing the return, it is natural to invest money in a portfolio, so now the questions becomes how to select an optimal portfolio. Before going into a further discussion about the selection of an optimal portfolio, we firstly define a portfolio and describe the meaning of the optimality of a financial portfolio.

Definition 2.5. **[Investment portfolio:]** *An investment portfolio is a vector of weights that represent the distribution of wealth among n investment assets*

$$\text{portfolio} = \mathbf{w} = (w_1, w_2, \ldots, w_n), \tag{7}$$

where w_i is the portion of wealth invested in the i^{th} asset.

The goal of an investor is to select an optimal portfolio, where the optimality is defined based on the investor's goal. Typical reasons for creating investment portfolios include, but are not limited to:

- Reducing the risk.
- Reducing the risk and increasing the return at the same time.
- Increasing the return given an acceptable level of risk.
- Maximizing the risk-adjusted return, which is defined as the ratio of the return and the risk.

To summarize, the two most commonly sought goals are maximization of the return for a given acceptable level of risk and minimization of the risk to obtain a predefined level of return. As also is the case with individual assets, it is also typical for portfolios that a higher value of the expected return implies a higher value of the risk associated with the portfolio, since one of the implicit rules of investments is that it is not possible to increase wealth without taking risks, and arbitrage ensures that the rule holds.

Moreover, regardless of the objective, there are usually several constraints imposed on the solution. Time horizon, time to maturity, transaction fees, and a preferable portfolio structure are just a few of possible constraints imposed on the problem. Most of these characteristics of a portfolio could be explained similarly to their meaning for individual assets. On the other hand, a preferable portfolio structure is relevant only for portfolios. For example, a person close to retirement is usually risk averse, so this individual would prefer to invest money in less risky government bonds and maybe only one high risk asset. On the other hand, an individual that needs to gain a large amount of money in a short period of time might be risk prone and want to invest money in more than one highly profitable but also highly risky stock.

The simplest and the most natural way to represent this type of problems is as a constrained optimization problem. The objective is to maximize or minimize an objective function (usually maximization of return or minimization of risk) subject to constraints such as non-negativity of weights (i.e., amount of money allocated to each asset), maximum amount of money available and (often) the requirement that exactly all money is invested, maximal level of risk acceptable, and minimal return required among others.

The problem can simply be represented as follows:

$$\text{maximize (or minimize)} \sum_{i=1}^{n} w_i x_i, \tag{8}$$

subject to constraints such as

$$w_i \geq 0 \; \forall i \in \{1, 2, \ldots, n\} \text{ (no short-selling is allowed)} \tag{9}$$

$$\sum_{i=1}^{n} w_i r_i \leq risk \text{ (maximum level of risk acceptable)} \tag{10}$$

$$\sum_{i=1}^{n} w_i R_i \geq return \text{ (minimum return required)} \tag{11}$$

$$\sum_{i=1}^{n} w_i = 1 \text{ (exactly all money is to be invested)} \tag{12}$$

where x_i is either return (in maximization problems) or risk (in minimization problems), n is the number of investment assets, w_i is the portion of wealth invested in the asset i, R_i is the return rate of the asset i, r_i is the risk of the asset i, $return$ is required level of return, and $risk$ is the level of risk acceptable by the investor. As presented above, selecting an optimal portfolio seems to be a straightforward linear programming problem. However, this representation is just the simplest problem that we can face when looking for an optimal portfolio. In general, many more constraints are imposed on the solution. Moreover, the objective function and the constraints are usually much more complex if more information, such as transaction cost, time period, preferable portfolio structure, relationships between characteristics of assets, etc., are taken into consideration. A real-life problem of portfolio selection is most commonly a non-linear optimization problem with constraints that are usually not (easily) solvable using general constraint solving techniques.

Regardless of the setting and its complexity, in the portfolio optimization problem, we aim at finding the vector of weights (i.e., amounts of wealth allocated into each asset), $\mathbf{w} = (w_1, w_2, \ldots, w_n)$, given all the other parameters.

2.5 Common Assumptions

Many of the above mentioned constraints are often taken into consideration when an algorithm is build to select an optimal portfolio. However, to make an algorithm for portfolio selection feasible, it is very rare that all the existing constraints are modeled. Moreover, in mathematical models, numerous assumptions are often made about the market and the assets even though these assumptions do not always hold. Nevertheless, they are good approximations of reality, and therefore still allow accurate and feasible modeling.

The most common assumptions that apply to many mathematical models representing financial problems include the following [34]:

- Market price is not affected by the actions performed. In general, this is not true since an increase in demand leads to an increase in price and vice versa. However, if trading happens in small quantities, the effects on the market will be insignificant.

- The liquidity assumption states that, at any time, an individual can buy or sell infinite amounts of any asset in a market. This might be true for the market of foreign currency exchange since these assets exist in (almost) infinite amounts, but this assumption is never true if an investor is trying to buy an "infinite" amount of stocks of a small company.

- Short-selling assumes that an investor can sell assets he/she does not hold, which will be represented by negative amounts in the portfolio distribution.

- Fractional amounts of assets could be bought.

- There is no transaction cost for trading assets.

Various other assumptions are often made in different models.

2.6 Return and Risk of a Portfolio

Building a portfolio requires that many characteristics of a portfolio be considered. As we have seen with a single asset, the return and the risk of a portfolio are the most important attributes to consider.

The return of a portfolio is usually defined as the weighted sum of returns of each individual asset, where the weight associated with each asset is the amount of wealth allocated to that asset, that is:

Definition 2.6. [Return of a portfolio]: *Given the returns R_i of each of n assets in the portfolio $\mathbf{w} = (w_1, w_2, \ldots w_n)$, the return of the portfolio $\mathbf{w}$ is given by*

$$R_p = w_1 \cdot R_1 + \ldots w_n \cdot R_n. \tag{13}$$

Of course, the main idea of a portfolio is to diversify risk so that the risk of a portfolio is lower than the sum of risks of individual assets. If the risk of an asset is defined as the standard deviation of its returns, then in a similar manner, the risk of a portfolio is defined as the standard deviation of the portfolio, where the standard deviation of the portfolio is defined as [17]:

Definition 2.7. [Standard deviation of a portfolio]: *Given the risks, r_i, of each of n assets and the correlation between each pair of assets i and j, $Corr_{ij}$, in the portfolio $\mathbf{w} = (w_1, w_2, \ldots w_n)$, the risk of the portfolio $\mathbf{w}$ is given by*

$$r_p = \left[\sum_{i=1}^{n} (w_i^2 \cdot r_i^2) + 2 \cdot \sum_{i=1}^{n} \sum_{j>i}^{n} w_i \cdot w_j \cdot r_i \cdot r_j \cdot Corr_{ij} \right]^{\frac{1}{2}}. \tag{14}$$

Even though this definition of the risk of a portfolio is widely used, it does not represent accurately the actual risk involved with the portfolio. To illustrate this problem, we can consider a portfolio consisting of only two assets with a correlation coefficient between them equal to -1. This situation would lead to the zero risk of the entire portfolio; however, in reality, this is not true as each asset is associated with some risk.

Moreover, this method of evaluating the risk of a portfolio penalizes returns that are above the average mean return, which might not be a bad thing. To solve this issue, semi-deviations [17] of returns are considered as another measure of a portfolio risk, where only returns below the mean are considered as volatility.

Another common measure involved with the measure of portfolio risk is *beta* [17]. It measures the contribution of an asset to the risk of portfolio:

Definition 2.8. [**Beta**]: *Beta of an asset i in the portfolio p of n assets is defined as*

$$\beta_i = \frac{SD_i}{SD_p} \cdot Corr_{ip}, \tag{15}$$

where SD_i and SD_p are variance of the asset i and the portfolio, respectively.

If we look at the formula for calculation of *beta*, we can conclude that a very volatile asset does not necessarily have to be a very risky because its correlation to portfolio might be very low.

Moreover, when selecting an optimal portfolio, we need to consider dependencies among the characteristics of a portfolio, which are very similar to dependencies among characteristics of an individual asset.

3 Related Work

Numerous models have been developed using a variety of computational techniques to solve the problem of optimal portfolio selection. Return-based strategies, methods involving stochastic processes, and intelligent systems techniques have been used under various assumptions to efficiently solve the selection of portfolio problem. Return-based strategies, as the name suggests, rely mostly on the return of the assets and aim at maximizing the return of a portfolio ([11],[16],[31],[55]). These techniques are the simplest models of financial investments and could usually be solved by using linear programming techniques or even classical calculus.

Methods involving stochastic processes ([7],[15],[23],[29],[54]) usually focus on predicting the behavior of the assets rather than finding an optimal distribution of wealth among the assets. Of course, predicting the values of the return and the risk of assets is a very important part when determining in which assets to invest money.

Intelligent systems techniques are used when simpler techniques such as linear programming are not applicable due to complexity of the problem, namely the complexity of the objective function (i.e., the main goal of an investor) and the constraints. Commonly used intelligent systems include genetic algorithms, rule-based expert systems, neural networks, and support vector machines. Genetic algorithms ([39],[40],[67]) usually apply two stages towards the selection of an optimal portfolio. In the first stage, the best performing assets are selected based solely on their returns, while the second stage determines the best distribution of wealth among the selected assets and is based on the return and the risk of the assets. Rule-based expert systems ([9],[52],[60]) are still only a theoretical tool for portfolio selection that is designed to imitate all the steps used by investment consultants. Neural networks ([4],[6],[41],[68],[69]) have found several applications in a portfolio optimization ranging from forecasting the behavior of investment assets to optimizing the distribution of wealth among assets. In either case, the risk and the return of the assets are the only characteristics considered when making a decision. Finally, support vector machines ([20]) have found the application in classification of assets into one of two classes: the assets with exceptional high returns and the assets with unexceptional returns, thus analyzing the performance of the assets only based on the returns.

In this chapter, we focus on computational intelligence techniques and present an extensive selection of intelligent systems methods such as genetic algorithms, neural networks, supports vector machines, and expert-based systems to select optimal portfolio strategies.

3.1 Genetic Algorithms

In this section, we first give a general description of genetic algorithms, and then explain how these algorithms work in a portfolio management framework. We also compare how genetic algorithm approaches perform versus other approaches, e.g., greedy algorithms.

3.1.1 Theoretical Background

A genetic algorithm (GA) is an optimization method (see e.g., [25],[30],[48],[49]) that imitates the biological process of natural survival of the fittest individuals in a population. Each individual is characterized by a sequence of genes, which constitute a chromosome. The fittest individuals are selected for mating. Through exchange of chromosomal material between selected pairs and through mutations, a new generation is produced. Thus, generating a new population follows a three-step process: selection, crossover–exchange of genetic material between two individuals to produce one or more offsprings, and mutation in genes. Genetic algorithm simulates all three steps of the natural evolution process.

A genetic algorithm starts by defining its optimization variables and the fitness function. Each variable represents a gene, and all genes of an individual represented a chromosome:

Definition 3.1. **[Chromosome]** *A chromosome representing an individual i is the vector of all genes of this individual*:

$$\text{chromosome}_i = (p_1, \ldots, p_N), \tag{16}$$

where N is the number of genes (variables), and p_j $(\forall j = 1, \ldots, N)$ is a gene (i.e., the value of a *variable) of the individual i.*

Since the variables could include qualitative as well as quantitative values of different ranges, each of them needs to be encoded into a finite set of distinct values, usually represented using binary digits.

The next step is to define the fitness function used in the algorithm.

Definition 3.2. [Fitness function] *The fitness function*

$$f(\text{chromosome}) = f(p_1, \ldots, p_N) \tag{17}$$

represents an optimization criterion that defines the fitness of each individual.

Usually, the fitness is to be maximized, so that the fittest individuals are selected for the next step. However, the fitness function could be defined as a cost function in which case the fittest individuals are the ones with the lowest cost.

After defining the variables and the fitness function, an initial population is generated either by a random number generator or by encoding values of variables for specific individuals. Initializing the population ends the preparation part of a genetic algorithm and denotes the beginning of the iterative steps. The first of three iterative steps is selection. A proportion of the population is selected to proceed to the next step and the remainder of the population is discarded. Most commonly, the generational gap, that is the percentage of the population selected to continue process, is 50%, but any other percentage could be used. The selection process is based on the fitness level of individuals and could be performed mainly in two different ways. The first method ranks all individuals based on their fitness level and selects top ranked individuals. The second method of selection relies on random selection in which a higher probability of selection is given to fitter individuals.

The selected individuals proceed to the crossover step that chooses two individuals for mating in order to produce one or two offsprings. The most commonly used method for mating is the one-point crossover technique that picks a random point r between the first and the last position in a chromosome, a point called crossover point, and produces two offsprings in the following way. The first offspring copies the genes 1 to r from the first parent and the genes $r + 1$ to N from the second parent, while the second offspring is produced by changing the order of the parents. The crossover using different parents continues until the number of individuals is increased to the original size of the population. Note however, that there are some variations in how to perform the crossover step.

The crossover step is followed by a mutation. A proportion of genes is chosen for mutation. The mutating genes are selected randomly. The selected genes take random values from the domain of the variable. The mutation process is very important since it slows down the quick convergence of the population in a small search area. It also allows the current best solution to jump away from a local optimum that is not a global optimum. However, it is desired that the current best individual is not mutated in order to not lose the current best solution, so many GAs apply the elitist strategy to protect the individual with the highest fitness from being mutated.

Finally, the fitness of each individual in the population is calculated again and the convergence criterion is checked. Ideally, at the end, all individuals in the population have the same genes, representing the optimal solution. However, a genetic algorithm is usually stopped after a predefined number of iterations, which results in a set of optimal values rather than just a single solution, a characteristic that suits portfolio selection problems very well.

3.1.2 Applications to Portfolio Management

As a computational intelligence technique, GAs have found different applications in portfolio management. In [39], the authors developed a two-stage algorithm to allocate wealth among numerous investment assets to reach an investor's goal. The first step, first described in [67], uses a GA to select the highest performing assets among thousands of available assets, while the second step utilizes another GA to find an optimal wealth distribution among chosen assets.

The choice of the assets to proceed to the second stage of the algorithm is based purely on the return of assets. Each asset is represented as a chromosome containing four genes. Each of the four genes is a representation of one financial indicator used as an input variable. The four variables are:

- Return on capital employed: $\text{ROCE} = \frac{\text{profit}}{\text{shareholder's equity}} \cdot 100\%$.
- Price per earning ratio: $\text{P/E} = \frac{\text{profit}}{\text{earnings per share}} \cdot 100\%$.
- Earning per share: $\text{EPC} = \frac{\text{net income}}{\text{the number of ordinary shares}}$.
- Liquidity ratio: $\frac{\text{current assets}}{\text{current liabilities}} \cdot 100\%$.

Each financial indicator is rated and takes one of eight values (0-7), where 0 stands for a poor performance of the asset and 7 represents a very good performance. These values are encoded as binary numbers so that each gene is a three-digit binary number.

Next, "fitness" of each asset is determined. To find the fitness of an asset, all assets are ranked based on the annual price return (APR):

$$APR_n = \frac{ASP_n - ASP_{n-1}}{ASP_{n-1}}, \tag{18}$$

where APR_n is the annual price return for the year n and ASP_n is the annual stock price for the year n. The assets with a high APR are considered good assets. Thus, all the assets are ranked from 1 to N where the asset with the highest APR is ranked 1, and the asset with the lowest APR is ranked N. The asset's ranking, r, is then mapped into the range $0 - 7$ using the linear mapping $R_{\text{actual}} = 7 \cdot \frac{N-r}{N-1}$, where N is the number of assets. Finally, a fitness function, which determines the optimization criterion, is designed. The most commonly used fitness function is the mean square error between the estimated ranking and next year's actual ranking:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(R_{\text{derived}} - R_{\text{actual}})^2}. \tag{19}$$

The goal is to minimize the value of $RMSE$.

After defining the variables and the fitness function, the selection step of GA is performed. Chromosomes are selected randomly for crossover with a higher probability for selection being given to chromosomes with a higher fitness. The one-point crossover technique, which is used to combine two parents to produce two offsprings, picks a position in a chromosome and interchanges the values of two parents at this position. Finally, a random mutation in each gene changes 0 to 1 or vice versa with a probability equal to 0.005.

The generation produced by this method is either accepted as a final population or another iteration of selection, crossover, and mutation is performed. The process stops when one of the following three conditions is satisfied:

- A predefined number of iterations is reached.
- A defined fitness is reached.
- A convergence criterion of the population is reached. In an ideal case, all the chromosomes of the final generation have same genes, representing the optimal solution.

At the end of the first step of two-stage portfolio optimization algorithm, the assets are ranked based on their return and the best m assets are considered for an investment. The second stage of the algorithm determines the wealth distribution among these m assets. It takes into consideration the risk as well as the return with the goal to minimize the risk for the expected level of the return.

This step of the algorithm is based on yet another genetic algorithm. Before applying the second GA, the expected return of each asset and the covariance between each pair of assets are calculated. The expected return of the asset i after n time intervals is calculated as

$$E(R_i) = \sum_{t=1}^{n} \frac{R_{it}}{n}, \tag{20}$$

where

$$R_{it} = \frac{SCP_{it} - SCP_{i(t-1)}}{SCP_{i(t-1)}} \tag{21}$$

is the return of the asset i for time t and SCP_{it} is the closing price for the asset i at time t. The covariance between assets i and j is given by

$$\sigma_{ij} = \frac{1}{n}\sum_{t=1}^{n}(R_{it} - E(R_i)) \cdot (R_{jt} - E(R_j)). \tag{22}$$

The algorithm designs chromosomes using the binary representation of asset's weight, w_i, which is the amount of wealth allocated to the asset i. The weight of the asset i is normalized by

$$x_i = \frac{w_i}{\sum\limits_{j=1}^{m} w_j} \tag{23}$$

to fit into 8-bits allocated for representation of each chromosome. The weights are adjusted through the GA algorithm until the optimal weights are achieved.

Next, a fitness function, defined by

$$Fitness = \sum_{i=1}^{m}\sum_{j=1}^{m}\sigma_{ij}x_i x_j + \left(\sum_{i=1}^{m} E(R_i)x_i - R_p^*\right)^2, \tag{24}$$

is designed to take into consideration the tradeoff between the risk and the return. The optimal solution is obtained by minimizing this function. The first term of the fitness function minimizes the risk, which is defined as the volatility of the assets included in the portfolio, while the second term minimizes the difference between the portfolio's overall return and the pre-defined required return, R_p^*. The fitness function for each chromosome determines the assets chosen for the selection, crossover, and mutation, which are performed similarly to the processes in the first GA. The results of these processes determine the generation for the next iteration. The final generation determines the distribution of wealth among the chosen assets.

The algorithm was tested on data obtained from Shanghai Stock Exchange during a period ranging from January 2001 to December 2004. The test used monthly and yearly available information. After the first stage of the algorithm, top 10, 20, and 30 stocks were selected for three different experiments. The results showed that the greater the number of stocks selected to be included in the portfolio, the lower the return of the portfolio. The portfolio with 10 stocks produced by the genetic algorithm was also tested against the equally weighted portfolio, which allocates equal amount of money in each of 10 stocks. The investment portfolio that resulted from the GA constantly outperformed the equally weighted portfolio.

A similar two-stage genetic algorithm was build in [40]. The only differences are the details of an asset representation and selection, crossover, and mutation processes. An asset is again represented as a chromosome, which is an $n-$dimensional vector consisting of n parameters called genes. If the initial population contains m assets, the selection process picks exactly $[\frac{m}{2}]$ assets with the highest fitness and discards all other assets. For the crossover stage, a random positive integer, $r \leq n$, is selected and two offsprings are produced by the following procedure. The first offspring copies the first r genes from the first parent and the last $n-r$ genes from the second parent, while the second offspring is created by copying the first r genes from the second parent and the last $n-r$ genes from the first parent. Formally, two parents P_1 and P_2 yield two offsprings O_1 and O_2 by the following rules:

$$O_1 = \{g_i | g_i \in P_1 \text{ if } i \leq r \text{ else } g_i \in P_2\} \tag{25}$$

and

$$O_2 = \{g_i | g_i \in P_2 \text{ if } i \leq r \text{ else } g_i \in P_1\} \tag{26}$$

where g_i represents the i^{th} gene.

Finally, the mutation is performed by randomly selecting another positive integer r, $r \leq n$. All the genes except the r^{th} gene are copied, and the gene r takes a random value that represents a possible mutation.

The algorithm was tested on data obtained from the Australian Stock Exchange. The results were compared against a Greedy algorithm and the comparison showed that the genetic algorithm performed only slightly more weakly than the Greedy algorithm but ran much faster.

3.2 Rule-based Expert Systems

Even though genetic algorithms showed good results when applied to portfolio management, other intelligent systems have been used as well to optimize the distribution of wealth among assets. Rule-based expert system is one of these techniques, so we review the basics of expert systems and then describe their application to portfolio selection.

3.2.1 Theoretical Background

A rule-based expert system simulates the decision making ability of a human expert in a field of interest (see e.g., [47]). The system is designed to allow "communication" between a user and itself in order to obtain some information that is necessary for solving a problem. The communication is performed through a user interface, which consists of a pseudo-natural language processing component that allows interaction between the user and the system using a limited form of natural language. Another role of the user interface is to display the solution of the problem being considered to the user along with a possible explanation for the decision actually made.

The "brain" of an expert based system consists of two parts – the knowledge base and the inference engine. The knowledge base contains the facts and the rules of the subject at hand. The rules are usually the rules of predicate calculus. The inference engine consists of processes that manipulate the knowledge base to make inferences.

The rules are usually directly entered in the system's knowledge base. However, sometimes the rules are inferred through training samples. The process of building an expert system this way usually iterates through many cycles until human experts are satisfied by its performance. The test cases are run on the system to ensure that the system provides the same results as would a human expert in the field.

There are several methods to make inferences from the given rules, but forward chaining and backward chaining are the most commonly used ones [22]. Forward chaining, as the name suggests, starts with the facts and deduces the conclusion by applying rules to the facts. On the other hand, backward chaining involves reasoning in the opposite direction. It starts with the hypothesis and tries to induce the facts to support the hypothesis.

An expert-based system is the simplest example of rule-based systems that has been applied to the selection of optimal portfolio [9]. However, portfolio management involves numerous tasks that, in real life, would not be performed by a single expert. To better simulate the behavior of human experts, a single expert systems have been used as a base for development of multi-agent systems. Multi-agent systems simulate tasks of several experts and combine their expertise to make a final decision [52]. This kind of system allows communication among temporally and spatially separated experts, which is why they have found application in lots of different areas.

3.2.2 Applications to Portfolio Management

The first attempt to design an expert system to assist portfolio managers is described in [9]. The basic idea of this system, called *Folio*, is to interview a user and use an expert's knowledge to first determine the user's investment goals and then build the portfolio that best suits the situation. The algorithm consists of three steps: the interview of the investor, the inference of the goals of the investor, and the optimization of distribution of wealth to reach goals of the investor.

The interview contains a set of questions that help the expert to derive the correct goal of the investor. The simple questions determine the user's acceptance of the level of risk, the desired return, and the user's tax bracket among others. Based on the obtained answers, the algorithm infers the rules of a user's goal, and the rules are used to determine the goals of the investor.

Folio recognizes 14 different goals for investment including acquiring a required level of return, reducing risk by investing into different assets, and minimizing risk while attaining the desired return. Each goal is characterized by five parameters: a target value, a penalty for exceeding the target value, a penalty for falling short of the target value, a lower bound under which the penalty becomes infinite, and an upper bound above which the penalty becomes infinite. These parameters are established from the inferred rules. About 50 rules (derived from interview) are used to infer one or more parameters of the goals. Sometimes, a parameter has more than one possible value, in which case a heuristics is used to determine the most certain value.

When the goal and its parameters are specified, *Folio* uses a goal programming algorithm to determine the distribution of wealth among assets that best fits the goal parameters. The goal programming algorithm used by *Folio* is a linear programming solver whose objective function is set to calculate the differences between the user's target values and the obtained values for each of the parameters. The algorithm minimizes the sum of the deviations of obtained values from desired values. The optimal wealth distribution among classes of assets is suggested. The algorithm considers nine classes of assets which include different levels of low-risk to high-risk assets. However, the distribution of wealth among each class is not given by this algorithm for several reasons. First, this method does not require *Folio* to consider thousands of investment assets that exist in the market and therefore, reduces the complexity of the algorithm. Second, *Folio* requires only the aggregate knowledge about the properties of each asset class and not the knowledge of individual assets. Moreover, the aggregate values change less slowly than an individual asset's characteristics, so *Folio* stays current for longer time period. Finally, picking the exact assets for an investment is the responsibility of an investment advisor and not *Folio*.

Even though performance of *Folio* has not been tested on real data, this algorithm is the foundation for the further development of expert based systems which evolved into multi-agent systems for portfolio management. The advantages of multi-agent systems (MAS) over the single-agent systems include [52]:

- The ability to avoid performance bottlenecks due to one stage in the multi-stage process.
- Possibility for interconnection and interoperation of multiple systems.
- Natural distribution of tasks among different agents.
- Possibility to connect spatially and temporally distributed experts.
- Enhancement of overall system performance including reliability, computational efficiency, maintainability, flexibility, and reuse among others.

A multi-agent system for portfolio monitoring, called *Warren*, was developed in [60] and further improved and implemented in [52]. *Warren* was designed to monitor portfolio rather than manage it. Monitoring portfolio is a continuous picture of an existing portfolio, which helps to determine if reallocation of assets is necessary, but does not suggest how to redistribute the wealth. The goal of *Warren* is to provide an overall picture of the existing portfolio based on the numerous available data from different sources.

Warren is composed of several types of agents: interface, task, middle, and information agents. The interface agent, *Warren Interface*, communicates with investor. This type of agent interviews the user and collects all necessary data that determine the goals of the investor. It also displays a comprehensive summary of the user's current portfolio and allows the user to buy and sell assets. The middle agent, *MatchMaker*, helps match agents that request services with agents that provide those services.

The task agents, *RiskCritical* agent and *Comptroller*, perform tasks. The tasks are performed by formulating problem-solving plans and carrying them out in collaboration with other agents. *RiskCritical* agent calculates the risk of the portfolio, while *Comptroller* agent is in charge of buying and selling assets.

The information agents monitor and collect financial information about companies of interest when requested by a middle agent. Warren contains four information agents: *FdsHistory* agent, *iYahooStocks*, *iEdgar*, and *TextMiner*. *FdsHistory* agent provides a historical view of financial data summary, *iYahooStocks* provides stock prices, *iEdgar* provides financial data summaries from SEC's Edgar web site, and *TextMiner* provides financial news analysis. *FdsHistory*, *iYahooStocks*, and *iEdgar* provide quantitative data about companies of interest, while *TextMiner* provides qualitative data available from numerous news agencies.

TextMiner is designed as a text classification agent to sort data available from a high volume of news reports about financial assets since only useful details should be considered when monitoring portfolio. *TextMiner* sorts the news from Reuters, CNN, Business Wire, etc. by first separating financial from non-financial news in articles. The financial news cover the reports on company's earnings, movements of stock price, revenues, and similar information, while the news about corporate control and legal and regulatory issues are considered non-financial. To separate financial from non-financial data, *TextMiner* was trained on a set of 1,239 news articles, which were labeled manually. The selection process is based on the weighting of commonly used terms (words or phrases) in the following way. First, each news article is represented in a high-dimensional space, where each dimension corresponds to a term. Then, the set of news articles is represented by the term-by-document matrix $M = T \times N$, where T is the number of terms and N is the number of articles. The set of terms $T = \{t_1, ..., t_t, ..., t_T\}$ is constructed by eliminating the words whose frequency is higher than *frequent threshold* (words that are considered to be just features) and the words whose frequency is lower than *infrequent threshold*. Each term has its weight w_t, which indicates how important the term is for a given document. All the weights are scaled from 0 to 1 with a higher weight being given to terms that appear often in one article but less frequently in other documents. Precisely, the weight of a word is determined by

$$w_t = \frac{(1 + \log(f_{it})) \cdot \log \frac{N}{d_t}}{\sqrt{\sum_{s \neq t} (log(f_{is}) + 1)^2}}, \tag{27}$$

where f_{it} is the number of times the term t occurs in the document i, and d_t is the number of documents in which the word t occurs. The weight is normalized by the document's length. After the weights for each term are determined, the article d is compared to one of the classes, $C = \{\text{financial}, \text{non-financial}\}$. A class is determined by the mean vector of all documents in the class,

$$\mathbf{c} = \frac{1}{|c|} \sum_{d \in c} \mathbf{d}, \tag{28}$$

and the calculation of similarity is the measure of the cosine of the angle between the class vector and the document vector

$$s(\mathbf{d}_i, \mathbf{c}_j) = \arg\max_{c_j \in C} \frac{\mathbf{d}_i \cdot \mathbf{c}_j}{||\mathbf{d}_i|| \cdot ||\mathbf{c}_j||}. \tag{29}$$

When financial news are separated from non-financial news, they are classified into one of five groups: good, good–uncertain, neutral, bad–uncertain, bad. Here the 'good' news are the ones clearly showing a company's good financial standing whereas 'bad' news are the ones clearly showing the bad financial standing of a given company. 'Neutral' news mention financial facts but do not give enough information to determine whether the facts indicate the good or the bad financial state of a given company. Two 'uncertain' categories refer to the prediction of future behavior of the company. The classification into one of five classes is performed by co-locating phrases, that is looking for words in the article that are usually in the same order in a sentence but not necessarily next to each other, such as 'earning' and 'increased'. The selection of useful co-located phrases is based on the training set of data.

Finally, a step-by-step description of the performance of *Warren* follows. First, the *MatchMaker* initializes the virtual work-space for agent-naming and resources for *Warren*, and all the other agents register their services with *MatchMaker*. The *Warren Interface* displays the current portfolio of the investor, and allows the user to buy/sell assets. If the investor requests the financial information about a particular company, the interface agent sends the request to the middle agent, and the middle agent invokes information agents to provide requested information. The information agents look for the information on the requested company and provide it to the interface agent. Warren Interface displays the gathered information and the *RiskCritical* agent calculates the risk of new portfolio. Finally, the *Comptroller* agent updates the investor's portfolio if he/she decides to buy/sell an asset.

Even though the entire model has not been tested on real-life data, *TextMiner* showed great results when compared to traditional Bayesian approaches to classify articles. With this in mind, *Warren* gives a promising tool for portfolio monitoring.

3.3 Neural Networks

Neural networks are tools that model human learning to some extent. They are relatively efficient for classification purposes and can be used to identify an optimal portfolio. Thus, we touch the basics of neural networks that are necessary to understand the several approaches based on neural networks that we are presenting in the following sections.

3.3.1 Theoretical Background

An artificial neural network, or just neural network (NN), is designed to imitate the actions of the human neural system, which consists of neurons and axons, the links between neurons (see e.g.,[61]). Similarly, a neural network consists of nodes and directed links between nodes. A NN is based on the ability to learn from training data sets in order to yield accurate results when applied to real data.

Several types of neural networks exist, the simplest one being the perceptron [50].

Definition 3.3. **[Perceptron]** *The perceptron is a neural network that consists of two layers of nodes: input and output layers. The input nodes,* $\mathbf{x} = (x_1, \ldots, x_n)$*, represent the input values, and the output nodes,* $\mathbf{y} = (y_1, \ldots, y_m)$*, carry out mathematical calculations and output the results.*

The function used to calculate the outputs is called the activation function. The most commonly used activation function in a perceptron is the *sign* function:

$$\hat{y} = \text{sign}\left(\sum_{i=1}^{n} w_i x_i\right) = \text{sign}(\mathbf{w} \cdot \mathbf{x}), \tag{30}$$

where $\mathbf{w} = (w_1, \ldots, w_n)$ is the vector of weights assigned to the links from input to output nodes. The weights represent the strength of the connection between the nodes and are determined by a learning process using the training data set for which the expected outcome is known. The weights are updated after each training example by

$$w_j^{(k+1)} = w_j^{(k)} + \lambda \left(y_i - \hat{y}_i^{(k)}\right) x_{ij}, \tag{31}$$

where $w^{(k)}$ is the weight of the i^{th} input link after the k^{th} iteration, x_{ij} is the value of the j^{th} attribute of the training example $\mathbf{x}_i$, and λ is the learning rate that is determined by user. The value of λ belongs to interval $[0, 1]$ and is designed to control the amount of adjustment after each training sample. The learning rate is either a constant that stays small throughout the entire training process to avoid overfitting to a specific training data element, or λ is adaptable, in which case it starts with a large value, but its size gets smaller during the training process.

The perceptron model is the simplest kind of neural networks and is used only for classification purposes. It is very efficient when data are linearly separable, that is we can create a hyperplane that clearly separates two classes. However, more complex multilayer networks are much more powerful and applicable to other types of problems.

Definition 3.4. **[Multilayer network]** *A multilayer network is a neural network that contains one or more hidden layers of nodes that perform calculations and influence more accurate weight adjustments.*

In a multilayer network, the links between nodes can go either only from a lower layer to a higher layer (input being the lowest layer and output the highest layer), which is the case in feed-forward networks, or the links can connect nodes in the same layer or even be directed towards the previous layers, which is the case in recurrent networks. The multilayer networks can use different activation functions, such as linear, sigmoid, and hyperbolic tangent function among others. These functions allow more complex situations to be modeled by multilayer networks.

A neural network learning algorithm works by minimizing the sum of the squared errors:

$$Err(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{32}$$

where $\hat{y}$ depends on $\mathbf{w}$. If $\hat{y}$ is replaced by $\mathbf{w} \cdot \mathbf{x}$, then the error function becomes quadratic in its parameters, and a global minimum can be easily found. However, if a non-linear function is used as an activation function, hidden and output layers produce non-linear outputs, so finding the solution for $\mathbf{w}$ becomes more difficult. Usually this problem is solved using a gradient descent method, which basically increases the weights in a direction that reduces the overall error function:

$$w_j^{k+1} = w_j^k - \lambda\frac{\partial Err(\mathbf{w})}{\partial w_j}, \tag{33}$$

where λ is the learning rate. This method can be successfully used to learn weights for the output layer. However, it might not be as easy to perform the computation for hidden layers since it is not possible to know their error term, $\frac{\partial Err}{\partial w_j}$, without knowing what their output values should be. To solve this problem, the backpropagation algorithm is used. It forces two phases in each iteration of the algorithm: the forward and the backward phases. In the forward phase, the weights computed in the previous iteration are used to compute the outputs of each node in the network and the computations follow in forward direction. In the backward phase, the weights are updated in the reverse order–the weight update formula is applied to the last layer first, and then for each previous layer one-by-one going towards the first layer, which allows the use of the output at the next level to compute the error at the previous layer.

3.3.2 Applications to Portfolio Management

Neural networks have found several applications in portfolio management ([4],[6],[41],[68],[69]) ranging from forecasting the behavior of investment assets to optimizing the distribution of wealth among assets.

Lowe [41] used an analog NN to find the optimal distribution of wealth among investment assets. The optimal portfolio is constructed by minimizing the risk function defined by

$$\text{risk} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left[y(t) - \sum_{i=1}^{m} w_i x_i(t)\right]^2}, \tag{34}$$

where m is the number of assets, T is the number of iterations, $y(t)$ is the market portfolio return, $x_i(t)$ is the return of the asset i at time t, and w_i is the proportion of wealth invested into the asset i. The risk function is subject to a non-negativity constraint of the weights $w_i \geq 0$ for every asset i and the normalization constraint $\sum_{i=1}^{m} w_i = 1$. This linear constraint optimization problem could be transformed into a nonlinear unconstrained optimization problem that minimizes the cost function

$$E = \frac{1}{T}\sum_{t=1}^{T}\left[y(t) - \sum_{i=1}^{m} w_i x_i(t)\right]^2 + \lambda\left[\sum_{i=1}^{m} w_i - 1\right]^2 + \mu\sum_{i=1}^{m}\frac{1}{1+e^{\beta w_i}}. \tag{35}$$

The first term of this equation corresponds to minimizing the risk; the second term replaces the normalization constraint; and the third term transforms non-negativity constraint into a barrier potential term, which has the form of a logistic. The parameters λ and μ are penalties that are used when constraints are not satisfied, and could be adjusted based on investors preferences.

The minimization of the cost function could be performed by using any standard gradient based method, such as the Runge-Kutta integration algorithm with a possibility to adapt step size based on the results form a previous iteration. The performance of the analog neural network in portfolio management was tested on seven stocks in the electricity sector of the UK market starting on the 26^{th} of September 1989. The model would lead to earning money in periods when the market was performing well, however, when the market was on downside, the model did not perform well either.

Another application of neural networks in portfolio management is described by Casas [4] to predict which of three considered classes of assets will outperform the other two. The three classes in consideration are: stocks, bonds, and money markets. The idea is to invest all wealth into one class of assets for a given time interval, and then re-evaluate the performance of the asset classes and make a new decision for the next time interval. This approach does not diversify the portfolio in order to reduce risk, and is based purely on the return of three classes of assets rather than performance of individual assets. Hence it is not likely to be useful in practical applications.

A neural network, that uses fundamental factors such as earnings, price per earning ratios, interest rates, and inflation, as input values, is trained with backpropagation algorithm to predict behavior of three classes of assets. The relative performance of classes of assets is measured by the risk premium. The risk premium between two asset classes A and B is calculated as

$$\Gamma_{AB} = E(A) - E(B), \tag{36}$$

where $E(x)$ is the expected return of the class x. Assuming that risk premium follows normal distribution, the probability that class A outperforms the class B is given by

$$P(A > B) = CND(\Gamma_{AB}, \mu_{AB}, \sigma^2_{AB}), \tag{37}$$

where CND is cumulative normal distribution function, μ_{AB} is mean risk premium, and σ^2_{AB} is standard deviation of risk premium. The algorithm calculates the probabilities that stocks will outperform bonds, bonds will outperform money markets, and stocks will outperform money markets.

The performance of this algorithm was tested against a buy-and-hold strategy that buys and holds S&P500 Index for the entire time period under consideration, which was 12 months in this case study for the year 1999. The NN approach outperformed the buy-and-hold strategy at the end of 12 months. Moreover, it predicted correctly 92% of the time which asset class would outperform the other two classes.

Another example of forecasting ability of NN was tested in [69]. In this paper, the authors presented a portfolio management algorithm that consists of three parts. The first part uses error correction neural network (ECNN) to forecast the behavior of assets. The second step uses a higher-level feedforward network to compute the excess return of one asset over another asset. Finally, the third part determines the optimal wealth distribution based on the excess returns.

The forecasting behavior of each asset in future is based on the expected return of the asset, which depends on the previous state of the asset, s_t, external influences, u_t, and the difference between the predicted output, y_t, and the observed output, y_t^d, at the previous iteration. Thus,

$$s_{t+1} = f(s_t, u_t, y_t - y_t^d), \tag{38}$$

where $y_t = g(s_t)$ is determined based on the current state. In the suggested model, the expected return is predicted based on an error correction neural network, which uses weight matrices of appropriate dimensions, $A, B, C,$ and D, to transform the problem into the following set of equations:

$$s_{t+1} = \tanh(As_t + Bu_t + D\tanh(Cs_t - y_t^d)) \tag{39}$$

$$y_t = Cs_t. \tag{40}$$

The optimization of parameters is obtained by finite unfolding in time using shared weights, which solves

$$\min_{A,B,C,D} \frac{1}{T} \sum_{t=1}^{T} (y_t - y_t^d)^2. \tag{41}$$

After the parameters are established by an ECNN, the expected return is calculated for each asset, f_i. Next, the difference between expected returns of two assets is calculated for each pair of assets, $e_{ij} = f_i - f_j$. Finally, the cumulative excess return of each asset is calculated as weighted sum of excess returns,

$$e_i = \sum_{j=1}^{k} w_{ij} e_{ij}, \tag{42}$$

where $w_{ij} \geq 0$ is the assigned weight to the pair (i, j) of assets. Based on cumulative excess returns, the proportion of wealth that should be invested into the asset i is calculated by

$$a_i = \frac{e^{e_i}}{\sum\limits_{j=1}^{k} e^{e_j}} = a_i(\mathbf{w}, f_1, ..., f_k). \tag{43}$$

This form guarantees that exactly all wealth is distributed ($\sum a_i = 1$) and the proportions of investment are non-negative ($a_i \geq 0$).

However, there are sometimes market share constraints given by the asset manager, and they are usually given in the form of an interval with a lower bound and an upper bound. If the mean of the available allocation for asset i is denoted by m_i, and the admissible spread is given by Δ_i, then the proportion a_i should fall into the interval $[m_i - \Delta_i, m_i + \Delta_i]$. The vector of means, $\mathbf{m} = (m_1, \ldots, m_k)$, is used as a benchmark distribution. To comply with the requirements of the manager, the excess return is adjusted by a bias vector $\mathbf{v}$ so that

$$e_i = v_i + \sum_{j=1}^{k} w_{ij}(f_i - f_j), \tag{44}$$

where the vector $\mathbf{v} = (v_1, \ldots, v_k)$ could be determined in advance by setting the excess returns to zero and solving the system of nonlinear equations

$$m_1 = a_1(v_1, \ldots, v_k)$$

$$\vdots$$

$$m_k = a_k(v_1, \ldots, v_k). \tag{45}$$

The non-unique solution of the form

$$v_i = \ln(m_i) + c \tag{46}$$

could be simplified by setting $c = 0$.

Since the interval $[m_i - \Delta_i, m_i + \Delta_i]$ represents a constraint for parameters $w_{i1}, ..., w_{ik}$, the optimal portfolio selection defined as the return maximization problem can be solved by solving a penalized maximization problem

$$\max_{\mathbf{w}} \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{k} \left[r_{it} a_i(f_{it}, ..., f_{kt}, \mathbf{w}) - \lambda ||a_i - m_i||_{\Delta_i} \right], \tag{47}$$

where r_{it} is the actual return of the asset i at time t and

$$||x||_{\Delta} = \begin{cases} 0 & \text{if } |x| \leq \Delta \\ |x| - \Delta & \text{otherwise} \end{cases}. \tag{48}$$

The proposed model was tested on the basis of monthly data from the G7 countries markets. Data from September 1979 to June 1993 were used to train the network, and based on the produced coefficients, the model was tested during the period from July 1993 to May 1995. The results showed that the neural network based model outperformed the benchmark model by almost 10%.

The modification of asset allocation step of this algorithm is presented in [68] and shows how to incorporate the risk of investing into selected assets rather than determining the optimal portfolio only based on the return. The authors use an ECNN (developed in [69]) to forecast the return of assets, r_i, which is used to calculate the risk-adjusted expected excess return rather than the expected return that does not consider risk related to the assets. The risk-adjusted excess return is defined by

$$\rho_i = \sum_t \frac{r_{it} - r_f}{|r_{it} - r_{it}^d|}, \tag{49}$$

where r_f is the risk-free asset return and r_{it}^d is the actual return at time t. Based on the risk-adjusted excess returns, the assets are ranked from the highest to the lowest, and all assets whose risk-adjusted excess return is higher than a pre-defined threshold value ρ^* are selected to be included into the portfolio.

If we denote the set of assets included in the portfolio by A, the proportion of wealth invested in each of the selected assets is determined by

$$w_i = \frac{\rho_i}{\sum_A \rho_j}. \tag{50}$$

This model used weekly data from 68 stocks from the German stock market during the period ranging from November 1994 to June 1999 in order to train the considered neural network. The algorithm's performance was tested on data from July 1999 to June 2000. Four portfolios were built with different number of stocks included: 5, 10, 15, and 20 stocks. The results of the algorithm were compared to the performance of the benchmark, which included all 68 stocks whose weights were chosen based on the shares of the stocks in the market. All portfolios produced by the NN algorithm outperformed the benchmark portfolio. Among the four derived portfolios, the portfolio with the smallest number of assets performed better than all the other portfolios.

Finally, [6] shows another application of NN as a forecast model as well as a decision model for wealth allocation. The multilayer perceptron (MLP) with one hidden Tanh layer (with H hidden units) and a linear output layer is considered. The function represented by MLP is given by

$$f(\mathbf{x};\theta) = A_2 \tanh(A_1\mathbf{x} + b_1) + b_2, \tag{51}$$

where $\mathbf{x}$ is the current distribution of wealth among assets, A_1 is an $H \times M$ matrix (with M being the dimension of the input vector $\mathbf{x}$), A_2 is an $N \times H$ matrix (with N being the dimension of the output vector), b_1 is an H-element vector, b_2 is an N-element vector, and $\theta = (A_1, A_2, b_1, b_2)$ is the vector of parameters. The parameters represented by the vector θ are found by training the network to minimize a cost function; the cost function differs for two types of the model–the forecast and the decision model. The optimization is performed by using a conjugate gradient descent algorithm. The gradient of the parameters with respect to the cost function is computed using the backpropagation algorithm for MLP.

In the forecast model, a neural network is used to predict the returns of assets in the next time period, $\mu_{t+1|t}$, given explanatory variables u_t, which belong to the set of the available information, I_t. The network is trained to minimize the prediction error of returns of assets in the next time period by using a quadratic loss function

$$C_F(\theta) = \frac{1}{T}\sum_{t=1}^{T} ||f(u_t;\theta) - r_{t+1}||^2 + C_{WD}(\theta) + C_{ID}(\theta), \tag{52}$$

where $||\cdot||$ is the Euclidian distance, $f(\cdot;\theta)$ is the function computed by MLP, and $C_{WD}(\theta)$ and $C_{ID}(\theta)$ are terms used for regularization purposes. The regularization is needed to prevent over-fitting by specifying *a priori* preferences on weights in the neural network. $C_{WD}(\theta)$ is the weight decay. It tries to reduce magnitude of the weights in the network by setting a penalty on the squared norm of all network weights. On the other hand, $C_{ID}(\theta)$ is the input decay. It tries to utilize useful inputs to train the network by penalizing the inputs that turn out to be unimportant.

The neural network decision model uses a NN to directly determine the distribution y_t of wealth among assets based on the explanatory variables u_t. The NN is trained to minimize the negative of the financial performance evaluation criterion

$$C_D(\theta) = -\frac{1}{T}\sum_{t=1}^{T} W_t + C_{WD}(\theta) + C_{ID}(\theta) + C_{\text{norm}}, \tag{53}$$

where C_{norm} is a preferred norm of the neural network. The preferred norm is important since two vector solutions that differ only by a constant multiple would be considered as different solutions without the use of preferred norm. The result would be that, for each vector θ, there would be a direction with (almost) zero gradient, so there would be no local minimum. The preferred norm variable, which is given by user, re-scales the parameters so that the norm constraint is achieved.

Training MLP for the decision problem is more complex than for the forecast model. It includes a feedback loop, which induces a recurrence by inputting the distribution y_{t-1} to determine the output y_t. Also the backpropagation through time algorithm is used to compute the gradient by going back in time, starting from the last time step until the first one.

Sometimes, the user has an idea of the optimal portfolio or has *a priori* preferences of the portfolio structure (i.e., the proportion of wealth invested into stocks versus the proportion invested into bonds). In this case, instead of the preferred norm, the preferred portfolio is considered. Deviation from the preferred portfolio is penalized by

$$C_{\text{ref.port.}} = \frac{1}{T}\sum_{t=1}^{T} \text{penalty}_{\text{ref.port.}}(y_t), \tag{54}$$

where penalty is calculated as the squared distance between the network output and the reference portfolio.

For testing purposes, Toronto Stock Exchange market data from January 1971 to July 1996 were used, and the results proved to outperform the benchmark algorithms. It was also shown that the decision model is preferred to the forecast model as it relies on fewer assumptions.

3.4 Support Vector Machines

Finally, we give a general description of support vector machines and their application to portfolio selection.

3.4.1 Theoretical Background

A support vector machine (SVM) is one of the most commonly used classification techniques (see e.g., [51],[61]). It classifies data into one of two groups by constructing a hyperplane that separates these two groups. The simplest situation is when the data are linearly separable. In this case, usually more than one hyperplane could be constructed to represent the boundary between two classes that will result in a zero error. However, instead of minimizing the empirical error (or error produced by training data), the best hyperplane should minimize the generalization error, that is the error that could result from classifying real data based on the model developed from the training set. To explain how to minimize the generalization error, we first define the margin hyperplanes. We consider a hyperplane b and create two other hyperplanes, b_1 and b_2, such that they are parallel to b and as far as possible from b (going into opposite direction from b) so that they do not touch any training data element. The distance between the hyperplanes b_1 and b_2 is called the margin of hyperplane. Since several non-parallel hyperplanes usually exist in the linearly separable case, we select the pair of parallel hyperplanes that yield the highest margin of hyperplane. The decision boundary is represented by the hyperplane going straight through the middle between two selected margin hyperplanes.

To formally define the best hyperplane, we consider a set of N training examples, each of them denoted by $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})$ corresponds to the attribute set for the i^{th} training example and $y_i \in \{-1, 1\}$ is the class label. Given this notation, the decision boundary is given by $\mathbf{w} \cdot \mathbf{x} + b = 0$, where $\mathbf{w}$ and b are the parameters of the model which are determined through training. Based on the calculated parameters, the decision for a new data sample $\mathbf{z}$, which is not in the training set, is determined by

$$y = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{z} + b > 0 \\ -1 & \text{if } \mathbf{w} \cdot \mathbf{z} + b < 0 \end{cases} . \tag{55}$$

Since the margin hyperplanes are defined as

$$\mathbf{w} \cdot \mathbf{x} + b = \pm 1, \tag{56}$$

each training data sample satisfies the conditions

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1 \text{ if } y_i = 1 \tag{57}$$

and

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \text{ if } y_i = -1. \tag{58}$$

These two conditions could be simplified to

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1. \tag{59}$$

Furthermore, we denote the margin hyperplanes by $\mathbf{w} \cdot \mathbf{x} + b = +1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$, which implies that the margin, d, of the decision hyperplane is $d = \frac{2}{||\mathbf{w}||}$. To simplify calculations necessary to find the best hyperplane, $||\mathbf{w}||$ is usually replaced by $||\mathbf{w}||^2$. Thus, maximizing the margin is equivalent to minimizing

$$f(\mathbf{w}) = \frac{||\mathbf{w}||^2}{2}. \tag{60}$$

We can formally define the objective of the learning process in SVM training phase as follows:

Definition 3.5. [Linear SVM: separable case]: *The learning task in a SVM can be formalized as the following constraint optimization problem:*

$$\min_{\mathbf{w}} \frac{||\mathbf{w}||^2}{2} \tag{61}$$

$$\textit{subject to } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \;\; \forall i = 1, 2, \ldots, N. \tag{62}$$

The problem of solving for $\mathbf{w}$ and b is a convex optimization problem (since the objective function is quadratic and the constraints are linear) that could be solved by using the standard Lagrange multiplier method, which rewrites the objective function in terms of a Lagrangian

$$L_P = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} \lambda_i(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1), \tag{63}$$

where the parameters λ_i are called the Lagrange multipliers. The first term tries to minimize the objective function, while the second term replaces the constraint and must be minimized in order to reduce the penalty of not satisfying the constraint. When solving for the Lagrange multipliers, many of them are equal to zero. However, a few Lagrange multipliers that are non-zero correspond to the training examples that lie exactly on one of the margin hyperplanes and thus represent support vectors, which are used to find the values of $\mathbf{w}$.

The Lagrangian problem could be transformed into a dual problem that involves finding only Lagrange multipliers. The problem maximizes the dual Lagrangian

$$L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j, \tag{64}$$

where the Lagrangian multipliers must be non-negative.

The solution to this problem can be found using numerical techniques such as quadratic programming. The solution for $\mathbf{w}$ is calculated by

$$\mathbf{w} = \sum_{i=1}^{N} \lambda_i y_i \mathbf{x}_i \tag{65}$$

and b is obtained by solving

$$\lambda_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b)] = 0, \tag{66}$$

while the decision boundary can be expressed as

$$\left(\sum_{i=1}^{N} \lambda_i y_i \mathbf{x}_i \cdot \mathbf{x} \right) + b = 0. \tag{67}$$

The previous description to find the optimal decision boundary works well if the training data are linearly separable. However, it is not always the case. Very often, any decision boundary would misclassify some training examples. The problem could be approached by introducing positive slack variables ξ_i that represent the error of the decision boundary for the training sample i [10]. Thus, the new objective function

$$f(\mathbf{w}) = \frac{||\mathbf{w}||^2}{2} + C\left(\sum_{i=1}^{N} \xi_i\right) \tag{68}$$

tends to minimize the error besides minimizing the original objective function. Here C represents the penalty for misclassification, and is determined by user. The new objective function and the inequality constraints

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1 - \xi_i \text{ if } y_i = 1,$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq 1 + \xi_i \text{ if } y_i = -1, \tag{69}$$

could be easily transformed into the Lagrangian where each Lagrangian value is bounded above by the value of the parameter C:

$$0 \leq \lambda_i \leq C. \tag{70}$$

This problem could be approached by using quadratic programming.

In some instances, however, a better solution exists than reducing the misclassification. A non-linear decision bound might exist to correctly classify training data that are not separable by the linear method. The idea is to transform the original coordinates of the training sample $\mathbf{x}$ into a new space $\Phi(\mathbf{x})$ so that a linear decision bound can be used to correctly separate data in the new space. The problem with this approach is to determine the mapping that will lead to desired results. Now, the problem of learning from training data becomes:

Definition 3.6. **[Nonlinear SVM]**: *The learning task in a non-linear SVM can be formalized as the following constraint optimization problem:*

$$\min_{\mathbf{w}} \frac{||\mathbf{w}||^2}{2} \tag{71}$$

$$\textit{subject to } y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1 \ \ \forall i = 1, 2, \ldots, N. \tag{72}$$

The attempt to solve this problem by transforming it into Lagrangian is usually not easy due to need for calculation of the dot product between the new spaces $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$, which might be very complicated. However, since the dot product is a measure of similarity between two instances $\mathbf{x}_i$ and $\mathbf{x}_j$, we can solve this problem by applying the kernel trick, which computes the similarity between two instances in the transformed space by using the original attribute set [3]. The kernel function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{73}$$

is the function that calculates the similarity of instances $\mathbf{x}_i$ and $\mathbf{x}_j$ by using the attributes in the original space, which simplifies the computation of the dot product. The use of the kernel trick also does not require the knowledge of the exact transformation Φ because the kernel function used in non-linear SVM must satisfy Mercer's theorem [61]:

Theorem 3.7. **[Mercer's theorem]**: A kernel function K can be expressed as

$$K(\mathbf{u}, \mathbf{v}) = \Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}) \tag{74}$$

if and only if, for any function $g(x)$ such that $\int g(\mathbf{x})^2 d\mathbf{x}$ is finite, then

$$\int K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0. \tag{75}$$

Support vector machines represent the most commonly used classifier. They have been used extensively in various applications, in particular portfolio optimization.

3.4.2 Applications to Portfolio Management

A support vector machine (SVM) can be used to classify stocks into one of two classes–the stocks with exceptional high returns (Class+1) and the stocks with unexceptional returns (Class-1) [20]. An SVM tries to minimize a bound on the generalization error rather than the empirical error as many other approaches do. It uses several financial indicators to determine the performance of each asset. The n financial indicators of the asset i are represented as a vector $\mathbf{x}_i = (x_1, ..., x_n)$. The expected future return of the stock is a binary dependent variable $y_i = \pm 1$, where $+1$ represents the Class+1 asset and -1 represents the Class-1 asset. Thus, the training set of m companies consists of pairs $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\} \subset R^N \times \{\pm 1\}$. The classifier (SVM) tries to learn from the training set, and it behaves as a function that maps the input variables $\mathbf{x}$ into an output value y. The misclassification is reduced by adjusting parameters.

An SVM is a classifier that tries to construct an optimal separating hyperplane between two classes by minimizing the bound on the misclassification risk. To solve linearly separable patterns, traditional approach using quadratic programming is utilized to maximize the dual Lagrangian.

In the case of non-separable patterns, different kernel functions could be used. In the test case, the Radial Basis Kernel, $K(\mathbf{x}, \mathbf{y}) = exp(-||\mathbf{x} - \mathbf{y}||^2)$, was used.

The method was tested on the Australian Stock Exchange using data ranging from 1992 to 2000. The data from the earliest three years were used for training and validation of estimated parameters that were then used to predict the performance of stocks in the next year. Eight groups of financial indicators were used to calculate the performance of stocks: return on capital, profitability, leverage, investment, growth, short term liquidity, return on investment, and risk. The data for each stock were converted into an eight-element input vector. For the training samples, the output of each stock was determined by its annual actual performance with the top 25% of stocks being selected into the Class+1, and the remaining stocks being assigned the Class-1. For testing purposes, the stocks selected into the Class+1 group were given equal weights in the portfolio. The created portfolio's return outperformed the equally weighted portfolio consisting of all available stocks, which was used as the benchmark portfolio.

3.5 Summary of the Existing Techniques

We have presented several attempts to use intelligence systems in the portfolio management. Genetic algorithms, rule-based expert systems, neural networks, and support vector machines have all contributed towards finding an optimal distribution of wealth among available assets. With the exception of genetic algorithm, all other methods are based on the ability to learn from examples and approximation of algorithm's parameters due to training samples. This approach could lead to overfitting of the parameters to specific type of data or a specific sample, which might not be applicable in other situations.

Moreover, all of the presented approaches do not consider the relationships between the characteristics of an asset. For example, the return and the risk are known to usually move in the same direction, that is higher the return of an asset, higher the risk of that asset. However, the presented approaches do not take into consideration this and many other existing dependencies.

Furthermore, the return, the risk, and other characteristics of an asset are assumed to be precisely known for each asset in consideration. In reality, this is not always the case as the best we can do is to predict the future return and the risk. Sometimes, these predictions are not correct, but all the presented techniques rely on the precise knowledge of these values.

To face the drawbacks of the presented approaches, we propose a new approach to portfolio optimization. The novel approach utilizes a utility-based multi-criteria decision making setting and fuzzy integration over intervals.

4 Multi-criteria Decision Making and Fuzzy Integration

Before going into the details of portfolio optimization problem based on the multi-criteria decision making framework, we review the basics of multi-criteria decision making, fuzzy measures, and fuzzy integration.

A multi-criteria decision making (MCDM) problem seeks the optimal choice among a (usually finite) set of alternatives. It can formally be defined as a triple $(X, I, (\succeq_i)_{i\in I})$ where

- $X \subset X_1 \times \cdots \times X_n$ is the set of alternatives with each set X_i representing a set of values of the attribute i.
- I is the (finite) set of criteria (or attributes).
- $\forall i \in I$, $\succeq_i$ is a preference relation (a weak order) over X_i.

The next task is to "combine" the partial preference relations $\succeq_i$ of an alternative into a global value for the alternative such that the final order of the alternatives is in the agreement with the decision maker's partial preferences. The natural way to construct a global preference is by using utility function for each attribute to reflect partial preferences of a decision-maker, and then combine these monodimensional utilities into a global utility function using an aggregation operator. The utility functions

$$u_i : X_i \to \mathbb{R} \tag{76}$$

such that

$$\forall x_i, y_i \in X_i, u_i(x_i) \geq u_i(y_i) \text{ if and only if } x_i \succeq_i y_i \tag{77}$$

map the values of all attributes onto a common scale. The existence of monodimensional utility functions is guaranteed under relatively loose hypotheses by the work presented in [37].

Numerous aggregation operators could be used to combine monodimensional utilities into a single number that represents the value of an alternative. Two simple approaches that correspond to optimistic and pessimistic behavior of the decision maker are maximax and maximin strategies, respectively, assuming that the goal of a decision-maker is to maximize the utility. The maximax method compares the utilities of all attributes of an alternative and chooses the highes utility value, $\max_i u_i(x_i)$, to represent the global utility of the alternative $\mathbf{x} = (x_1, ..., x_n)$. This approach reflects an optimistic behavior of the decision-maker since he/she is concerned only with the attribute that has the highest utility for the given alternative. The maximax method tries to maximize the best criterion:

$$\max_{\mathbf{X} \in X}(\max_{i \in \{1,...,n\}} u_i(x_i)). \tag{78}$$

On the contrary, the maximin method reflects a pessimistic behavior of the decision-maker as the decision-maker is concerned only with the attribute that could result in the worst value. This method compares the utilities of all attributes of an alternative and chooses the lowest utility value, $\min_i u_i(x_i)$, to represent the global utility of the alternative $\mathbf{x}$. The decision-maker tries to maximize the value of the worst case scenario:

$$\max_{\mathbf{X} \in X}(\min_{i \in \{1,...,n\}} u_i(x_i)). \tag{79}$$

To allow for a position between these extremes when making a decision, a simple combination of maximax and maximin approaches is achieved by a weighted aggregation operator, named Hurwicz criterion [32],

$$\max_{\text{all alternatives}}(\alpha \max_i u_i(x_i) + (1 - \alpha) \min_i u_i(x_i)), \tag{80}$$

where $\alpha \in [0, 1]$ is the weight given by the decision maker. This approach simplifies to the optimistic case if $\alpha = 1$ and to the pessimistic case if $\alpha = 0$.

These simple approaches are very tempting to use for quick decisions. However, they focus only on a few criteria and ignore the impact of other characteristics of alternatives, which often does not suit a given situation. Thus, we usually need to consider more complex aggregation operators that take into consideration all attributes. The simplest and most natural of them is a weighted sum approach, in which the decision-maker is asked to provide weights, w_i, that reflect the importance of each criterion. Thus, the global utility of the alternative $\mathbf{x} = (x_1, ..., x_n) \in X$ is given by

$$u(\mathbf{x}) = \sum_{i=1}^{n} w_i u_i(x_i). \tag{81}$$

The best alternative is the one that maximizes this value. Even though this approach is attractive due to its low complexity, it can be shown that using an additive aggregation operator, such as weighted sum, is equivalent to assuming that all the attributes are independent [46]. In practice, this is usually not realistic and therefore, we need to turn to non-additive approaches, that is to aggregation operators that are not linear combinations of partial preferences.

Before approaching non-additive methods, we give the definition of a non-additive measure, a tool for building non-additive aggregation operators. For practical purposes, we restrict ourselves to measures defined on a finite set I.

Definition 4.1. **[Non-additive measure]**: *Let I be the set of attributes and $\mathcal{P}(I)$ the power set of I. A set function $\mu : \mathcal{P}(I) \to [0, 1]$ is called a non-additive measure (or a fuzzy measure) if it satisfies the following three axioms:*

(1) $\mu(\emptyset) = 0$: the empty set contains no information.

(2) $\mu(I) = 1$: the maximal set contains all the information.

(3) $\mu(B) \leq \mu(C)$ if $B,\ C \subset \mathcal{P}(I)$ and $B \subset C$: a new criterion added cannot make the importance of a coalition (a set of criteria) diminish.

Of course, any probability measure is also a non-additive measure. Therefore non-additive measure theory is an extension of traditional measure theory. Moreover, a notion of integral can also be defined over such measures.

A non-additive integral, such as the Choquet integral [8], is a type of a general averaging operator that can model the behavior of a decision maker. The decision-maker provides a set of values of importance, this set being the values of the non-additive measure on which the non-additive integral is computed from. Formally, the Choquet integral in finite case is defined as follows:

Definition 4.2. **[Choquet integral]**: *Let* μ *be a non-additive measure on* $(I, \mathcal{P}(I))$ *and an application* $f : I \to \mathbb{R}^+$*. The Choquet integral of* f *w.r.t.* μ *is defined by:*

$$(C)\int_I f d\mu = \sum_{i=1}^{n}(f(\sigma(i)) - f(\sigma(i-1)))\mu(A_{(i)}), \tag{82}$$

where σ *is a permutation of the indices in order to have* $f(\sigma(1)) \leq \cdots \leq f(\sigma(n))$*,* $A_{(i)}$ *is* $A_{(i)} = \{\sigma(i), \ldots, \sigma(n)\}$*, and* $f(\sigma(0)) = 0$*, by convention.*

It can be shown that many aggregation operators can be represented by Choquet integrals with respect to some fuzzy measure. However, note that there are other non-additive approaches to decision making besides the Choquet integral, one of them being the Sugeno integral [59]:

Definition 4.3. **[Sugeno integral]**: *Let* μ *be a fuzzy measure on* $(I, \mathcal{P}(I))$ *and an application* $f : I \to [0, +\infty]$*. The Sugeno integral of* f *w.r.t.* μ *is defined by:*

$$(S)\int f \circ \mu = \bigvee_{i=1}^{n}(f(x_{(i)}) \wedge \mu(A_{(i)})), \tag{83}$$

where $\vee$ *is the supremum and* $\wedge$ *is the infimum.*

Even though the Choquet and the Sugeno integrals are structurally similar, their applications are very different. The Choquet integral is generally used in quantitative measurements, while Sugeno integral has found more applications in qualitative approaches. However, we restrict ourselves to quantitative approaches.

Although the Choquet integral is well suited for quantitative measurements, it has a major drawback. The decision maker needs to input a value of importance of each subset of attributes, that is total of 2^n values. More precisely, since the value of the empty set and the entire set are known by the definition of a fuzzy measure, the exact number of values required from the decision-maker is $2^n - 2$. This still leads to an exponential complexity and is therefore intractable. However, we can overcome intractability by using 2-additive measure to limit the complexity to a $O(n^2)$ (as shown in [5]) and still get accurate results.

Before giving the definition of a 2-additive measure, we need to define notion of interaction indices of orders 1 and 2. The importance of an attribute (or the interaction index of degree 1) is best described as the value this attribute brings to each coalition it does not belong to. It is given by the Shapley value [53]:

Definition 4.4. **[Shapley value]**: *Let μ be a non-additive measure over I. The Shapley value of index i is defined by:*

$$v(i) = \sum_{B \subset I \setminus \{i\}} \gamma_I(B)[\mu(B \cup \{i\}) - \mu(B)] \tag{84}$$

with

$$\gamma_I(B) = \frac{(|I| - |B| - 1)! \cdot |B|!}{|I|!} \tag{85}$$

and $|B|$ denoting the cardinal of B.

While the Shapley value gives the importance of a single attribute to the entire set of attributes, the interaction index of degree 2 represents the interaction between two attributes, and is defined by ([14],[27]):

Definition 4.5. **[Interaction index of degree 2]**: *Let μ be a non-additive measure over I. The interaction index between i and j is defined by:*

$$I(i,j) = \sum_{B \subset I \setminus \{i,j\}} (\ \xi_I(B) \cdot (\mu(B \cup \{i,j\}) - \mu(B \cup \{i\}) - \mu(B \cup \{j\}) + \mu(B)) \tag{86}$$

with

$$\xi_I(B) = \frac{(|I| - |B| - 2)! \cdot |B|!}{(|I| - 1)!}. \tag{87}$$

The interaction indices belong to the interval $[-1, +1]$ and

- $I(i, j) > 0$ if the attributes i and j are complementary;
- $I(i, j) < 0$ if the attributes i and j are redundant;
- $I(i, j) = 0$ if the attributes i and j are independent.

Even though we can define interaction indices of any order, defining the importance of attributes and the interaction indices between two attributes is generally enough in MCDM problems. Thus, 2-additive measures constitute a feasible and accurate tool in this setting. The formal definition of a 2-additive measure follows [14]:

Definition 4.6. [2-additive measure]: *A non-additive measure μ is called 2-additive if all its interaction indices of order equal to or larger than 3 are null and at least one interaction index of degree two is not null.*

We can also show [26] that the Shapley values and the interaction indices of order two offer us an elegant way to represent a Choquet integral. Therefore, in a decision-making problem, we can ask the decision maker to give the Shapley values, I_i, and the interaction indices, I_{ij}, and then use the Choquet integral with respect to a 2-additive measure, μ, to obtain the aggregation operator:

$$(C)\int_I f d\mu = \sum_{I_{ij}>0} (f(i) \wedge f(j)) I_{ij} + \sum_{I_{ij}<0} (f(i) \vee f(j)) |I_{ij}| + \sum_{i=1}^{n} f(i)(I_i - \frac{1}{2} \sum_{j \neq i} |I_{ij}|). \tag{88}$$

This form of the Choquet integral is a practical approach to many situation, one of them being portfolio management. Nevertheless, it has a major drawback when applied to to the selection of an optimal portfolio since it assumes that the exact values for each interaction index of orders 1 and 2 are known. However, we can not assume that a decision maker can give us the precise values for each index. To face this problem, we introduce intervals to allow the decision maker to give us ranges of values rather then the exact values for each Shapley value and each interaction index of second order.

5 Intervals

We review basics of intervals and interval arithmetic in order to ease the understanding of how this concept could help us in the selection of optimal portfolio problem. Interval Arithmetic (IA) is an arithmetic over sets of real numbers called *intervals*. It had started the development in fifties in order to model uncertainty, and to tackle rounding errors of numerical computations. For a complete presentation of interval arithmetic, we refer the reader to [33].

Definition 5.1. **[Interval]**: *A closed real interval is a closed and connected set of real numbers. The set of closed real intervals is denoted by* $\mathbb{IR}$. *Every* $\boldsymbol{x} \in \mathbb{IR}$ *is denoted by*

$$[\underline{x}, \overline{x}], \tag{89}$$

where its bounds are defined by $\underline{x} = \inf \boldsymbol{x}$ *and* $\overline{x} = \sup \boldsymbol{x}$.
For every $a \in \mathbb{R}$, *the interval point* $[a, a]$ *is also denoted by* a.

The *width* of a real interval $\boldsymbol{x}$ is the real number $w(\boldsymbol{x}) = \overline{x} - \underline{x}$. Given two real intervals $\boldsymbol{x}$ and $\boldsymbol{y}$, $\boldsymbol{x}$ is said to be *tighter than* $\boldsymbol{y}$ if $w(\boldsymbol{x}) \leq w(\boldsymbol{y})$.

Interval Arithmetic operations are set theoretic extensions of the corresponding real operations. These operations can be implemented by real computations over the bounds of intervals. Given two intervals $\boldsymbol{x} = [a, b]$ and $\boldsymbol{y} = [c, d]$, we have, for example:

- $\boldsymbol{x} + \boldsymbol{y} = [a + c, b + d]$.
- $\boldsymbol{x} - \boldsymbol{y} = [a - d, b - c]$.
- $\boldsymbol{x} \times \boldsymbol{y} = [\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\}]$.
- $\boldsymbol{x}^n = \begin{cases} [a^n, b^n] & \text{if } n \text{ is an odd natural number} \\ [0, \max\{|a|, |b|\}^n] & \text{if } n \text{ is even and } 0 \in [a, b] \\ [\min\{|a|, |b|\}^n, \max\{|a|, |b|\}^n] & \text{if } n \text{ is even and } 0 \notin [a, b] \end{cases}.$

Moreover, the associative law and the commutative law are preserved over $\mathbb{IR}$. However, the distributive law does not hold. In general, only a weaker law is verified, called semi-distributivity. For all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{IR}$, we have:

- associativity:

  $(\boldsymbol{x}+\boldsymbol{y})+\boldsymbol{z}=\boldsymbol{x}+(\boldsymbol{y}+\boldsymbol{z})$.

  $(\boldsymbol{x}\boldsymbol{y})\boldsymbol{z}=\boldsymbol{x}(\boldsymbol{y}\boldsymbol{z})$.

- commutativity:

  $(\boldsymbol{x}+\boldsymbol{y})=(\boldsymbol{y}+\boldsymbol{x})$.

  $\boldsymbol{x}\boldsymbol{y}=\boldsymbol{y}\boldsymbol{x}$.

- sub-distributivity:

  $\boldsymbol{x}\times(\boldsymbol{y}+\boldsymbol{z})\subseteq\boldsymbol{x}\times\boldsymbol{y}+\boldsymbol{x}\times\boldsymbol{z}$.

5.1 Intervals of preferences

As mentioned earlier, to define preferences over multi-dimensional alternatives, the user is required to provide importance and interaction indices, but is more likely to provide intervals of values $\boldsymbol{I_i}$ and $\boldsymbol{I_{ij}}$, $\forall i, j \in \{1, \ldots, n\}$, which leads to evaluation of a Choquet integral over intervals using IA [5]:

$$(C_I)\int_I f d\mu = \sum_{\boldsymbol{I_{ij}}>0}(f(i)\wedge f(j))\boldsymbol{I_{ij}} + \sum_{\boldsymbol{I_{ij}}<0}(f(i)\vee f(j))|\boldsymbol{I_{ij}}| + \sum_{i=1}^{n} f(i)\left(\boldsymbol{I_i} - \frac{1}{2}\sum_{j\neq i}|\boldsymbol{I_i j}|\right), \quad (90)$$

where the annotation $(C_{\mathbb{I}})$ means that the interpretation of this formula is performed using IA. As a consequence, the value of the integral is an interval.

5.2 Strategies of preference

To determine which interval Choquet integral yields the best results, we need to compare intervals. When comparing intervals, the ideal case is when the intervals do not intersect. In this case, if the alternative I is evaluated with values that are all better than those of the alternative J, the preference is clearly given to the alternative I.

However, the above case is very specific and unfortunately does not happen often. More common is that two intervals intersect and we need to choose a better of two overlapping intervals. The strategies to make decisions in such cases are described below.

A simple naive strategy offers a straightforward solution that compares only the upper bounds and gives the preference to the interval with the highest upper bound (which corresponds to an optimistic behavior of a decision-maker as he/she is only interested in the highest potential values rather than all the values that could be reached), or compares the lower bounds and gives preference to the highest lower bound (which corresponds to a pessimistic behavior).

However, many alternatives between the very optimistic case and the very pessimistic case are possible. They require us to look simultaneously at the upper and the lower bounds as well as the width of the intervals, which highlights the degree of uncertainty of an alternative's value. To combine these variables, a degree of preference was introduced [5]. A degree of preference, $d(I, J)$, intended to express the extent to which a better value of the Choquet integral is likely to lie in the interval I, rather than in the interval J.

It is defined as a function $d : \mathbb{I}^2 \rightarrow [0, 1]$, where:

$$d(I,J) = \begin{cases} \frac{\overline{I}-\overline{J}}{|(\overline{I}-\overline{J})+(\underline{J}-\underline{I})|} & \text{if } \overline{I} > \overline{J} \text{ and } \underline{J} \geq \underline{I} \\ \\ 1 & \text{if } \overline{I} = \overline{J} \text{ and } \underline{I} > \underline{J} \\ & \text{or: if } \overline{I} > \overline{J} \text{ and } \underline{I} \geq \underline{J} \\ & \text{or: if } \overline{I} = \overline{J} \text{ and } \underline{I} = \underline{J} \\ \\ 1 - d(J, I) & \text{otherwise} \end{cases} . \tag{91}$$

The higher the value of the degree of preference, the greater the chance that the optimal interval is the interval I, while lower value of the degree of preference implies that the interval J would more likely contain higher value of the Choquet integral.

The degree of preference, as described above, assumes that a decision-maker is risk-neutral, that is the person is not willing to undergo an extreme risk nor he/she believes that there is a reason to be too careful. However, sometimes, a person exhibits a risk-prone attitude and leans towards optimistic behavior, or on the other hand, the decision-maker could be more risk-averse especially if there is a reason to expect pessimistic results. If a decision-maker could provide the level of risk that he/she is willing to take in order to maximize the utility, then we can modify the degree of preference to include this fact.

Let us assume that the level of risk a person wants to take is expressed by a real value in the interval $[0, 1]$, where naturally, values close to 0 represent pessimistic situations, and values closer to 1 mean more optimistic expectations. Now, we can tighten the considered interval to better suit this level of risk. The shrinking of the interval $[\underline{X}, \overline{X}]$ based on the risk level $r \in [0, 1]$ is done in the following way [43]:

- First, calculate the proportion of the interval that is considered important by the decision-maker:

$$p = 2 \cdot \min\{r - 0, 1 - r\}. \tag{92}$$

- Next, calculate the size of the interval that corresponds to the given proportion:

$$\text{size} = p \cdot (\overline{X} - \underline{X}). \tag{93}$$

- Finally, calculate the interval of importance, $[\underline{N}, \overline{N}]$:

$$[\underline{N}, \overline{N}] = \begin{cases} [\underline{X}, \underline{X} + size] & \text{if } r \leq 0.5 \\ [\overline{X} - size, \overline{X}] & \text{otherwise} \end{cases}. \tag{94}$$

This approach clearly returns a single point instead of an interval in cases when the level of risk is at extreme points, i.e., the interval of importance is the upper bound of the original interval when the risk level is 1, and the lower bound when the risk level is 0. In both cases, the problem is reduced to comparison of single (extreme) points rather than intervals, the situation that corresponds to the naive strategy.

Once we have tightened the intervals to reflect the level of risk the decision-maker is willing to take, we apply equation (91) to new intervals of importance to calculate the degree of preference, which determines the better of two intervals.

The presented approach to determine the better of two intervals given the level of risk works well if the decision-maker can provide the exact degree of risk he/she is willing to take. However, in reality it is hard to describe the level of risk by a single number [43]. More probable is that a person could define the level of risk by an interval $r = [\underline{r}, \overline{r}]$. In this case, the calculation of the interval of importance that encounters for the optimism/pessimism of a person is a bit more complicated. Instead of a precise interval, the result is an interval whose bounds are themselves intervals (2nd order interval), and therefore, the degree of preference would result in an interval, $d(I, J) = [\underline{d}, \overline{d}]$ rather than a single number. Three situations could occur:

- $\overline{d} < 0.5$ in which case the preferable choice is interval J.
- $\underline{d} > 0.5$ in which case the preferable choice is interval I.
- $0.5 \in [\underline{d}, \overline{d}]$ in which case the preferable choice is
  (1) interval I if $(\overline{d} - 0.5) \geq (0.5 - \underline{d})$
  (2) interval J otherwise.

All of the above rankings of intervals suppose uniform probability distribution, which is a reasonable assumption if no additional information is available. However, sometimes more information is accessible and more accurate probability distribution over an interval could be considered. Typically, if the width of interval is not limited, it is common that a decision-maker would give an interval bigger than what he/she really believes the interval should be to cover any possible extreme value even though the extreme values would very rarely happen. Thus, it is not uncommon that the values within an interval would not follow uniform distribution, but rather a form of Gaussian distribution (possibly screwed). In this situation, it is reasonable to assume that the interval Choquet integral would also not follow uniform distribution but would rather have higher probability of values in the interior of the interval than those close to bounds.

In the case where more information is available about the probability distribution over an interval, we can slightly modify the approach used to calculate the degree of preference [43]. As before, we start by tightening the given interval based on the level of risk, r, that a person is willing to take. Thus, we need to determine the value, s, within the given interval $[\underline{X}, \overline{X}]$ such that

$$s = \begin{cases} P(x \leq 2r) & \text{if } r < 0.5 \\ P(x \geq (2r-1)) & \text{if } r > 0.5 \end{cases} . \tag{95}$$

Thus, the interval of importance is:

$$[\underline{N}, \overline{N}] = \begin{cases} [\underline{X}, \underline{X} + s] & \text{if } r \leq 0.5 \\ [\overline{X} - s, \overline{X}] & \text{otherwise} \end{cases} . \tag{96}$$

Note that the above formula when applied to a uniform distribution leads exactly to the equation (94), with s replacing the variable *size*.

The next step is to calculate the degree of preference between two intervals given their new bounds. Taking into consideration the probability distribution, the degree of preference is given by:

$$d(I,J) = \begin{cases} \frac{P(\overline{J} \leq x \leq \overline{I})}{P(\overline{J} \leq x \leq \overline{I}) + P(\underline{I} \leq x \leq \underline{J})} & \text{if } \overline{I} > \overline{J} \text{ and } \underline{J} \geq \underline{I} \\ \\ 1 & \text{if } \overline{I} = \overline{J} \text{ and } \underline{I} > \underline{J} \\ & \text{or: if } \overline{I} > \overline{J} \text{ and } \underline{I} \geq \underline{J} \\ & \text{or: if } \overline{I} = \overline{J} \text{ and } \underline{I} = \underline{J} \\ \\ 1 - d(J,I) & \text{otherwise} \end{cases} . \quad (97)$$

When applied to uniform distribution, this equation simplifies to the equation (91).

6 New Algorithms for Portfolio Selection

We propose two different algorithms that make use of multi-criteria decision making approach to find the optimal portfolio allocation. A two-stage algorithm uses a multi-criteria decision making setting to rank all assets. Based on the rank, good assets are selected among thousands of assets that exist in a market and wealth is invested in these selected assets only. The second step of the algorithm utilizes another MCDM setting to determine the exact wealth allocation among the assets to best suit the goals of a particular investor.

The second algorithm utilizes a similar multi-criteria decision making settings and starts by clustering all assets into three groups based on their risk. Based on the investor's acceptable level of risk, distribution of wealth among three groups of assets is determined and MCDM setting is created to determine the exact allocation of wealth within each cluster.

We first define a multi-criteria decision making problem by considering the set of all asset as the set of alternatives. We determine a finite set of criteria that characterize investment assets–return (R), risk (r), time to maturity (t), transaction cost (c), etc., and define a utility function for each of them. The simplest method to choose a rational utility functions is to provide mappings from the values of an alternative onto the interval $[0, 1]$, $f : X_i \to [0, 1]$. For the return of an asset, this could mean that the highest realistic return is mapped into 1, the lowest return to 0, and the other returns are proportionally mapped into values between 0 and 1. The utility of the risk could be defined in a similar fashion except that the highest risk is mapped to 0 and the lowest risk to 1 since a high value of risk is less desired than a low value of risk. Similar arguments hold for time to maturity and transaction cost. Once the utility function for each criterion is defined, we proceed to calculation of the global value of each asset.

If the decision maker (the investor) is concerned only with the return or only with the level of risk, then the maximax strategy could be used to rank all the assets with a high importance given to return in the first case and to the risk in the second case, and low importance given to all the other attributes. However, usually an investor wants to maximize the return for a given level of risk or minimize the risk while attaining the required return level in a certain time period.

Thus, all the criteria have some influence on the decision. The decision-maker is asked to input the Shapley value of each criterion, that is the importance of each criterion relative to other criteria. Since the attributes are mutually dependent (e.g., a higher return usually implies a higher risk, a longer time to maturity usually means a higher return, etc.), the weighted sum approach does not promise to give accurate results. However, we can approximate the interaction indices for each pair of attributes by understanding their dependencies within a market, and use the Choquet integral with respect to a 2-additive measure, defined by Shapley values and interaction indices of order 2, to calculate the global value of an asset. The Choquet integral values are used to order the assets giving a higher rank to the assets with the higher value of the Choquet integral.

Top n assets are chosen to proceed to the second stage of the algorithm. The number n is either pre-defined by the investor, or all the assets with the Choquet value above a threshold specified by the investor are selected. We denote the set of all assets that are used to create a portfolio by A. The second stage of the algorithm tends to find the optimal distribution of wealth among the n selected assets, $\mathbf{w} = (w_1, \ldots, w_n)$, by considering another multi-criteria decision making setting. The set of alternatives is now defined as the set of all possible portfolios using only the assets selected based on their rank. The set of criteria is unchanged from the first stage of the algorithm. However, the values of the criteria for a portfolio are defined in terms of the criteria values for each asset in the portfolio and the weights assigned to each asset (i.e., proportions of wealth allocated to the asset). For example, we can define the following values for the return, the risk, time to maturity, and the transaction cost:

- The return of the portfolio is

$$R(\mathbf{w}) = \sum_{i=1}^{n} R_i w_i. \tag{98}$$

- The risk of the portfolio is

$$r(\mathbf{w}) = \sum_{i=1}^{n} r_i w_i. \tag{99}$$

- Time to maturity of the portfolio, however, is not the weighted sum of the individual assets' maturity times. It is the maximum time to maturity of all assets included in the portfolio:

$$t(\mathbf{w}) = \underset{j}{max}\, t_j, \text{where } j \text{ is such that } x_j \in A. \tag{100}$$

Note that if all assets are included in every portfolio, then the time to maturity will be same for all portfolios.

- The transaction cost of the portfolio is

$$c(\mathbf{w}) = \sum_{i=1}^{n} c_i v_i, \tag{101}$$

where $v_i = w_i$ if the transaction cost of the asset i is a proportion of wealth invested into the asset, and $v_i =$ constant s if the transaction cost of the asset j is equal to s for any amount invested.

- Similarly, the values of other attributes characterizing a portfolio could be defined in terms of values of individual assets included into the portfolio and the distribution of wealth allocated to the assets.

Keeping the same Shapley values for all attributes and interaction indices of degree 2 for each pair of attributes as given in the first step of the algorithm, we maximize the Choquet integral of the alternatives. Thus, this stage of the algorithm reduces to a constrained optimization problem where we look for the vector $\mathbf{w} = (w_1, \ldots, w_n)$ that maximizes the objective function

$$\max_{\mathbf{w}} \sum_{I_{ij}>0} I_{ij}[u_i(x_i) \wedge u_j(x_j)] + \sum_{I_{ij}<0} |I_{ij}| \cdot [u_i(x_i) \vee u_j(x_j)] + \sum_{i=1}^{n} \left(u_i(x_i) - \frac{1}{2} \sum_{i \neq j}^{n} I_{ij} \right). \tag{102}$$

Here, x_i and x_j represent criteria of the portfolio (e.g., risk, return, time to maturity, transaction cost, etc.), which are defined in terms of w_i, and one of the characteristics of portfolio (r_i, R_i, t_i, c_i, or others).

The maximization problem could be subject to the following (and/or similar) constraints:

$$\sum_{i=1}^{n} w_i R_i \geq R \text{ or } \sum_{i=1}^{n} w_i r_i \leq r \text{ (portfolio satisfies the main goal of the investor)} \tag{103}$$

$$\sum_{i=1}^{n} w_i = 1 \text{ (exactly all wealth is invested)} \tag{104}$$

$$w_i \geq 0 \;\; \forall i = 1, \ldots, n \text{ (money can not be borrowed to be invested in an asset).} \tag{105}$$

This problem involving constraints could be solved using standard optimization techniques. Since all constraints are linear, the choice of the optimization technique depends on the form of the objective function. Using the simple utility functions described in this section, the objective function is linear as well, which allows us to use the simplex method to determine the optimal solution. However, if some complex utility functions are used to execute the multi-criteria decision process, the objective function might not be linear and other methods must be used to find the solution. Since optimization methods applied to complex objective functions usually guarantee to find only a local optimum but not the global one, we can iterate the algorithm several times with different starting points to find, in theory, a better solution.

To reduce the complexity of the presented algorithm, we developed another algorithm that utilizes MCDM setting in portfolio selection. It starts by ordering all the assets based only on their risk. Using this ranking, the algorithm clusters all the assets into three groups: high, middle, and low risk assets. The clustering is performed such that one third of assets with the highest risk constitutes group 1, group 2 contains the middle risk assets, and group 3 is the third of assets with the lowest risk. Next, we calculate the Choquet integral of each asset following the same MCDM setting as in the first stage of the first algorithm. We select top $n_1 > 0$, $n_2 > 0$, and $n_3 > 0$ assets respectively from high, middle, and low risk clusters to be included into the portfolio. The values of n_1, n_2, and n_3 are either all equal and predetermined, or they are such that the values of assets selected from each cluster are higher than a predefined threshold value.

Based on the investor's level of risk aversion, the proportion of wealth invested in each cluster is determined and denoted by p_1, p_2, and p_3 respectively for groups 1, 2, and 3. If the decision-maker is highly risk-averse, p_1 will be much smaller than p_2 and p_3, while for a risk-prone individual, p_3 will be smaller than p_1 and p_2. However, none of the numbers will be equal to zero in order to diversify portfolio. This is necessary in order to reduce unsystematic risk, that is, the risk that depends on the company.

Finally, the wealth allocated to each cluster is distributed among the assets that belong to the group, so that the optimal portfolio is selected. Each cluster is considered separately from the other two and the best distribution of wealth is determined by maximizing the Choquet integral of the portfolios built by selected assets in each group

$$\max_{\mathbf{w}} \sum_{I_{ij}>0} I_{ij}[u_i(x_i) \wedge u_j(x_j)] + \sum_{I_{ij}<0} |I_{ij}|[u_i(x_i) \vee u_j(x_j)] + \sum_{i=1}^{n} \left(u_i(x_i) - \frac{1}{2} \sum_{i \neq j}^{n} I_{ij} \right) \tag{106}$$

subject to

$$\sum_{i=1}^{n} w_i = 1 \text{ (exactly all wealth is invested)} \tag{107}$$

and

$$w_i \geq 0 \;\; \forall i = 1, \ldots, n \text{ (money can not be borrowed to be invested in an asset).} \tag{108}$$

Note that this algorithm does not explicitly require satisfaction of the main goal of the investor (e.g., required return level, maximum risk rate, etc.), but this requirement is implicitly accounted for in the distribution of wealth among three clusters. We can again apply one of the standard optimization techniques to solve this problem.

Even though the utility based multi-criteria decision making setting and its solution based on the use of a Choquet integral with respect to a 2-additive measure is a feasible and accurate solution for values given by the decision maker, this approach faces another problem. We cannot expect a decision maker to give precise values for the importance and interaction indices. In order to overcome this hurdle, it was shown [5] that the use of intervals provides a nice solution in MCDM problem.

Intervals allow the problem of portfolio management to be presented more realistically as the investor is asked to provide the ranges of values of the importance and interaction indices of order 2 instead of the exact values. It is reasonable to believe that an investor can determine whether, for example, minimization of risk is more important than the return from an asset, or whether the time period in which an amount could be obtained is more important than risk. However, it is more realistic that the investor can give the interval of how much one criterion is more important than the other criterion rather than giving the exact values of the relative importance among the criteria. Thus, the intervals provide a rational way to solve the portfolio optimization problem by following the same procedures as the non-interval based methods and evaluating the Choquet integral over intervals and extending the optimization techniques to intervals as well.

7 Testing

The proposed algorithm was tested on data from the Shanghai stock market in the period ranging from 2000 to 2007, and which were obtained from the *Wharton Research Data Services* [63]. Only stocks that were available during the entire period from 2000 to 2007, and for which all data were available in the mentioned period were used in the study, which accounted for a total of 365 stocks. The monthly available data were used to calculate yearly data, which were necessary for testings. The entire data sets used for testing the algorithm are available in the Appendix A.

When designing the experiments, some of the common assumptions on a market were made. These assumptions include:

- Market price is not affected by the actions performed.
- An individual can buy or sell infinite amounts of any asset in a market.
- Fractional amounts of assets could be bought.
- There is no transaction cost for trading assets.

For the purpose of building a multi-criteria decision making environment, two quantitative criteria (return and risk) and one qualitative criterion (reputation) of an asset were chosen to be examined. A return of an asset was predicted using the regression method. The returns from the five years immediately preceding the year in which the algorithm performance was tested were used to find a linear regression function, and data available from all the years were used to find the average long term market return, which was used as the predicted market return for the following year. The risk of an asset was calculated as a standard deviation of returns in the last five years.

Finally, the reputation of a company owing a stock was calculated based on the longevity of the company and the proportion of positive and negative returns during the last five years. Since investors have more trust in companies that have existed for longer time period, the reputation of a good performing company is higher if the company has existed for a longer period of time. In our experiment, we assigned the following weights to reputation with respect to the age of a company:

- age of a company: 0-2 years $\rightarrow$ $weight = 0.2$;
- age of a company: 3-5 years $\rightarrow$ $weight = 0.4$;
- age of a company: 6-8 years $\rightarrow$ $weight = 0.6$;
- age of a company: 9-12 years $\rightarrow$ $weight = 0.8$;
- age of a company: over 12 years $\rightarrow$ $weight = 1$.

Furthermore, the number of positive returns, *number*, is simply the number of positive returns of a company within the last five years. The reputation of a company is calculated as the product of the weight assigned to reputation and the number of positive returns

$$reputation = weight \cdot number. \tag{109}$$

Next, a utility function for each criterion was created by finding the maximum (*max*) and the minimum (*min*) value of each criterion within the database. A linear function was designed to map all the values of a criterion into the interval [0,1] by mapping the *max* into 1 and the *min* into 0 in the cases of return and reputation, while the opposite was done in the case of risk.

In the next step, an investor is asked to input the Shapley values of each criterion. Each Shapley value is entered as an interval that can takes values in the range [0,1]. Tests were performed using three different sets of the Shapley values representing three different behaviors of investors. The first set of the Shapley values represents an individual that equally cares about all three considered characteristics. The second set of the Shapley values represents an individual that considers the return and the risk of an asset equally important but does not care much about the reputation of a company. Finally, the third set of the Shapley values represents an investor for whom the return is the most important, the risk is much less important than the return, and the reputation has very little importance.

Before calculating the Choquet integral of each asset, the interaction indices of order 2 between each pair of criteria were determined using an expert's knowledge about markets. It was determined that a return and a risk of an asset are redundant criteria since a high return will have the same impact on an asset as a high risk and vice versa. The interaction level between these two criteria is relatively high, so the interaction index of [-0.9,-0.8] was assigned to the pair (return,risk). Since a high reputation allows for a lower return and a high return leads towards a high reputation, these two criteria are complementary, but they are not as highly interactive as the pair (return, risk). Thus, the interaction index of [0.4,0.5] was assigned to the pair (return, reputation). Finally, a high risk yields a low reputation and vice versa, which results in risk and reputation being redundant. The interaction index of [-0.6,-0.5] was assigned to the pair (risk,reputation).

Finally, given all the criteria and the corresponding utility values as well as the Shapley values and the interaction indices of the second order, the Choquet integral with respect to a 2-additive measure of each asset is calculated over intervals. The top 10, 20, and 30 assets were selected to proceed to the second stage of the algorithm as three separate test cases.

In the second stage of the algorithm, portfolios were considered as the alternatives out of which the best one is to be selected. The Shapley values and the interaction indices from the first stage of the algorithm were used in the second stage as well.

The utility functions for each criterion of a portfolio are defined similarly to the utility values in the first stage of the algorithm except that a utility value of a criterion of a portfolio is defined as a weighted sum of the individual values of the assets that compose the portfolio. The weights are assigned based on the distribution of wealth among the assets in the portfolio, where finding the optimal weights is the task of the second stage of the algorithm. The optimal weights are defined by maximizing an optimization function, which consists of the Choquet integral decreased by penalties that result from not satisfying constraints.

The first of two constraints considered in the problem is that no more than all money and no less than 50% of all money should be invested in a portfolio. If these conditions are not satisfied, a penalty to the optimization function is applied. The penalty applied should be a value that is greater or equal to any value that the Choquet integral with respect to a 2-additive measure could possibly take in this setting. The reason for this penalty is that if the constraint is not satisfied, we do not want to consider these distributions as a possible solution since we can not invest money that we do not have and therefore, the problem becomes impractical if the constraint is not satisfied. To determine the value of the penalty, let us consider the formula of the Choquet integral w.r.t. a 2-additive measure:

$$(C)\int_I f d\mu = \sum_{I_{ij}>0}(f(i)\wedge f(j))I_{ij} + \sum_{I_{ij}<0}(f(i)\vee f(j))|I_{ij}| + \sum_{i=1}^{n} f(i)(I_i - \frac{1}{2}\sum_{j\neq i}|I_{ij}|). \tag{110}$$

If we consider the first two sums, we conclude that each interaction index of order 2 is covered by at most one of the sums (the interaction indices of 0 are not covered at all). Each element of each summation is a minimum or a maximum of two utility values multiplied by the absolute value of the corresponding interaction index. Each utility value and the absolute value of each interaction index belong to the interval [0,1]. Thus, each element of any of the first two summation is at most equal to one. The maximum number of elements in two summations is the number of interaction indices of order 2, which is $\frac{n\times(n-1)}{2}$, where n is the number of criteria considered. Further, the third summation is the multiplication of a Shapley value and a utility value decreased by a certain amount. Thus, to find its maximum value, we can disregard the subtracted part, and only find the maximum value of the utility value multiplied by a Shapley value. As both of these values fall into the interval [0,1], the maximum value of each element of the third summation is one, and thus, the value of the summation can not exceed n. The maximum value that the Choquet integral can theoretically take is therefore

$$\frac{n\cdot(n-1)}{2} + n = \frac{n^2+n}{2} < \frac{n^2+n^2}{2} = n^2. \tag{111}$$

Thus, the penalty that should be applied for a violation of the first constraint should be equal to n^2. Moreover, since some unfeasible solutions are closer to satisfying the constraint than other solutions, this penalty is increased based on the distance from satisfying the constraint. The distance is calculated as the Euclidean distance between the sum of weights and the value of one (the optimal situation is to invest all wealth), and this distance is added to the previous penalty.

The second constraint considered is a soft constraint on short selling. We allow short selling, but we would prefer that no short selling or minimal short selling occurs. We allow that only a value equal to 10% or less of the investors total wealth is borrowed from an asset. However, each short selling is penalized by the penalty equal to the number of criteria. Thus, a small penalty is assigned if money is borrowed only from a small number of assets, while a higher penalty is accumulated if many assets are used for borrowing money from them.

Finally, genetic algorithm technique is used as an optimization technique to find the optimal distribution of weights among the selected assets that would maximize the value of the Choquet integral decreased by the penalties. The particular genetic algorithm creates the population of 500 individuals, where each gene of an individual represents the portion of wealth allocated to a particular asset. Possible values of each gene range from -0.1, which represents a possibility of borrowing the amount of money equal to 10% of the total investor's wealth from one asset in terms of short selling, up to 0.4, which means that no more than 40% of the entire wealth could be invested in one asset. The initial population is generated mostly randomly. Two extreme individuals are created. The first one is generated by forcing each gene to take the value of -0.1, while the second one represents an equally weighted portfolio. These individuals are each selected for the crossover in the initial fifteen iterations in order to create more individuals close to the values that are more probable to be contained in genes of an optimal individual.

The selection process is designed to give a higher probability to the individuals with a higher fitness to be selected for crossover and mutation.

The crossover procedure is modified compared to the typical crossover technique used in genetic algorithms. Instead of selecting a crossover point and coping all the exact genes from one of two parents to produce an offspring, a random number, r, in the interval [0,1] is generated. The value of the gene of the first parent is multiplied by r, and this value is added to the value of the corresponding gene of the second parent multiplied by $1 - r$. The second offspring is produced by the reverse process of multiples of the genes form the same two parents. Thus, given two parents

$$P_1 = (x_1, x_2, \ldots, x_n) \text{ and } P_2 = (y_1, y_2, \ldots, y_n), \tag{112}$$

the offsprings are produced as

$$O_1 = (r_1 \cdot x_1 + (1 - r_1) \cdot y_1, \ldots, r_n \cdot x_n + (1 - r_n) \cdot y_n \tag{113}$$

and

$$O_2 = ((1 - r_1) \cdot x_1 + r_1 \cdot y_1, \ldots, (1 - r_n) \cdot x_n + r_n \cdot y_n. \tag{114}$$

Finally, individuals for mutation are randomly selected with the probability of 0.05. A randomly selected gene receives a randomly generated value in the interval [-0.1,0.4].

An optimal portfolio was build based on the predicted values of the return, the risk, and the reputation of each asset. For testing purposes, the portfolio's return was calculated using the actual returns for the following year, and the return of the portfolio was compared to actual returns of several benchmark portfolios. The first benchmark is an equally weighted portfolio using all 365 assets. In an equally weighted portfolio, an equal amount of wealth is invested in each asset. The second benchmark portfolio is the equally weighted portfolio over the assets selected for the second stage of the algorithm based on the multi-criteria decision making setting from the first stage of the algorithm. The third benchmark is an equally weighted portfolio over N assets chosen based on the predicted risk-adjusted returns, where N is the number of assets selected to proceed to the second stage of the algorithm.

The proposed algorithm as well as benchmark portfolios were coded in C++, and the entire code could be found in the Appendix B.

7.1 Results

The results of each run of the algorithm are shown in the Appendix C. It is easy to see from the results presented in the Appendix C that our algorithm outperformed the benchmark portfolios in almost every case. To test the significance of these results, we performed Wilcoxon one-tailed pairwise test. The Wilcoxon test only makes the assumption that data are symmetric about the mean [64]. Since we are testing that the return of the portfolio obtained by using fuzzy integration over intervals is greater than the return of benchmark portfolios, we used one-tailed test to measure the significance of our results. The statistical analysis was performed using R, an environment for statistical computing [13].

The Wilcoxon test was performed separately for each benchmark and for each set of resulting values corresponding to portfolios containing 10, 20, and 30 assets. The obtained p-values are presented in the tables 9, 10, and 11.

Table 9: Results of Wilcoxon one-tailed pairwise test for a portfolio containing 10 assets.

benchmark	1	2	3
p-value	0.005859	0.003906	0.01953

Table 10: Results of Wilcoxon one-tailed pairwise test for a portfolio containing 20 assets.

benchmark	1	2	3
p-value	0.001953	0.001953	0.001953

Table 11: Results of Wilcoxon one-tailed pairwise test for a portfolio containing 30 assets.

benchmark	1	2	3
p-value	0.001953	0.001953	0.003906

According to the results, except for the ten-asset portfolio's performance against the third benchmark, for all the other cases, the p-value is smaller than 0.01, and thus, we can conclude at a 99% confidence level that the results obtained by our algorithm are significantly greater than the performances of the benchmarks. In the case of the ten-asset portfolio, we can conclude at a 98% confidence level that the performance of our algorithm is better than the performance of the third benchmark.

8 Conclusion and Future Work

The novel approach for an optimal portfolio selection has shown significant impact on the improvements of the existing techniques both theoretically and experimentally.

As results presented in the previous section suggest, the proposed algorithm significantly outperformed all benchmark algorithms. Furthermore, as it could be seen from the data presented in the Appendix C, the new approach yields a higher return on almost every situation. Moreover, in many cases in which the benchmark portfolios would lead to a loss of money, the new algorithm would have positive returns and therefore, it reduces the risk of losing money.

From a theoretical perspective, the approach based on fuzzy integration over intervals has solved many drawbacks that exist in the currently used methods as well as added some new features that give options to investors to express different behaviors.

The proposed method does not depend on any training samples. Thus, once developed, the model could be applied to any market. Moreover, if a new piece of information becomes available, it is easy to adopt the model to take into consideration the new information by just adding one more criterion into the model.

The approach takes care of dependencies among the criteria by evaluating the portfolios through the use of fuzzy measures and fuzzy integration. It also successfully deals with imprecise data by using intervals. Thus, the new algorithm effectively copes with these two big drawbacks of other approaches.

Finally, the proposed model allows for risk-prone and risk-averse behavior of investors. A particular behavior is represented by an investor's choice of the Shapley values of each criterion, which represents how important each criterion is for a particular investor when making a decision. The level of risk taken by an investor is also accounted for during the comparison of intervals of the values of the Choquet integrals.

8.1 Future Work

Even though the proposed approach yields significantly greater returns than the benchmark portfolios, this approach is not perfect. There is some room for improvement in several areas.

First of all, the interaction indices of the second order among each pair of criteria are based on our knowledge of market behavior. However, a deeper study of interaction indices in a relationship with a market is needed to more precisely define these values. By using an interval representation of the indices, we are capable to represent the market; nevertheless, tighter intervals would represent a market even better. Moreover, an extraction of a fuzzy measure could solve the problem even more precisely.

Second, a better global optimization technique needs to be developed. A genetic algorithm guarantees only that a local optimum will be found rather than a global optimum. We have improved and adopted a genetic algorithm to our needs by forcing the existence of some individuals in the initial population and by forcing that some individuals of the initial population are close to the positions that we expect to yield the optimum value. However, the algorithm still does not guarantee that a global optimum will be found.

Third, the algorithm does not take into consideration dependencies among the assets. However, in reality, a performance of a company is dependent on performances of other companies. Thus, the algorithm needs to consider the interconnections among the assets in a market and include these dependencies using into the practical application.

Fourth, a more dynamic setting could be designed, where trading could be performed on a more frequent basis. Monthly, daily, or even hourly investment trading could be designed using a similar algorithm.

Finally, we need to examine more the structure of the Choquet integral and determine why this particular method performs better than other approaches in order to take even more advantage of its structure and improvements that it could bring.

Regardless of all the possible improvements, the novel approach, which is based on fuzzy integration over intervals, showed to be a very good model for the selection of an optimal portfolio and a practical tool for real-life applications.

References

[1] Alcatel, `http://euronext.com/trader/summarizedmarketderivatives`, April 3, 2009.

[2] Baker, R. and Belgorodskiy, A., "Predicting stock returns and assessing prediction performance," *IMA Journal of Management Mathematics*, no. 18, pp. 413-433, 2007.

[3] Boser, B. E., Guyon, I. M., and Vapnik, V. N., "A training algorithm for optimal classifiers," in 5$^{\text{th}}$ *Annual ACM Workshop on COLT*, D. Haussler (Editor), pp. 144-152, ACM Press, Pittsburgh, PA, 1992.

[4] Casas, C. A., "Tactical asset allocation: an artificial neural network based model," *Proceedings of International Joint Conference on Neural Networks*, no. 3, pp. 1811-1816, 2001.

[5] Ceberio, M. and Modave, F., "Interval-based multicriteria decision making," in *Modern Information Processing: from Theory to Applications*, B. Bouchon-Meunier, G. Coletti, and R. R. Yager (Editors), Elsevier ed., 2006.

[6] Chapados, N. and Bengio, Y., "Cost functions and model combination for VaR-based asset allocation using neural networks," *IEEE Transactions on Neural Networks*, no. 1, pp. 890-906, 2001.

[7] Chiu, K.-C. and Xu, L., "Arbitrage pricing theory-based Gaussion temporal factor analysis for adaptive portfolio management," *Decision Support Systems*, no. 37, pp. 485-500, 2004.

[8] Choquet, G., "Theory of capacities," *Annales de l'Institut Fourier*, no. 5, 1953.

[9] Cohen, P. R. and Lieberman, M. D., "A report on FOLIO: an expert assistant for portfolio managers," *Investment Management: Decision Support and Expert Systems*, pp. 135-139, 1990.

[10] Cortes, C. and Vapnik, V., "Support-Vector Networks," in *Machine Learning*, vol. 20, 1995.

[11] Cover, T. M. and Ordentlich, E., "Universal portfolios with side information," *IEEE Transactions on Information Theory*, vol. 42, no. 2, 1996.

[12] Currncies – Wall Street Journal, http://online.wsj.com/public/page/news-currency-currencies-trading.html, April 7, 2009.

[13] Dalgaard, P., *Introductory statistics with R*. Springer, 2008.

[14] Denneberg, D. and Grabisch, M., “Shapley value and interaction index,” *Mathematics of Interaction Index*, 1996.

[15] Desai, R., Lele, T., and Viens, F., “A Monte-Carlo method for portfolio optimization under partially observed stochastic volatility,” *Proceedings of Computational Intelligence for Financial Engineering*, 2003.

[16] El-Yaniv, R., “Competitive solutions for online financial problems,” *ACM Computer Surveys*, vol. 30, no. 1, 1998.

[17] Estrada, J., *Finance in a nutshell: A no-nonsense companion to the tools and techniques of finance*. Prentice Hall, 2005.

[18] Euronext Bonds, http://www.euronext.com/landing/landingBond-20663-EN.html, April 3, 2009.

[19] Fama, E. F., “Efficient capital markets: a review of theory and empirical work,” *Journal of Finance*, 1970.

[20] Fan, A. and Palaniswami, M., “Stock selection using support vector machines,” *Proceedings of International Joint Conference on Neural Networks*, no. 3, pp. 1793-1798, 2001.

[21] Farmer, J. D. and Sidorowich, J. J., “Predicting chaotic time series,” *Physical Review Letters*, vol. 59, no. 8, pp. 845-848, 1987.

[22] Giarratano, J. and Riley, G., *Expert systems: principles and programming*. PWS Publishing Company, Boston, MA, 1994.

[23] Goel, M. and Kumar, S. K., “A risk-sensitive portfolio optimization problem with stochastic interest rate,” *Journal of Emerging Market Finance*, vol. 5, 2006.

[24] Goetzmann, W. N. and Rouwenhorst, K. G., *Origins of Value: A Document History of Finance*. Oxford Universuty Press, 2005.

[25] Goldberg, D. E., *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, MA, 1989.

[26] Grabisch, M., "The interaction and Mobius representation of fuzzy measures on finite spaces, k-additive measures: a survey," in *Fuzzy measures and integrals: Theory and applications*, M. Grabisch, T. Murofushi, and M. Sugeno (Editors), Physica Verlag, 2000.

[27] Grabisch, M. and Roubens, M., "Application of the Choquet integral in multicriteria decision making," in *Fuzzy Measures and Integrals: Theory and Applications*, M. Garbisch, T. Murofushi, and M. Sugeno (Editors), Physica Verlag, 2000.

[28] Graham, B., *The intelligent investor*. Harper Business Essntials, 1949.

[29] Greyserman, A., Jones, D. H., and Strawderman, W. E., "Portfolio selection using hierarchical Beyesian analysis and MCDM methods," *Journal of Banking & Finance*, vol. 30, pp. 669-678, 2006.

[30] Haupt, R. L. and Haupt, S. E., *Practical Genetic Algorithms*, 2nd edition. Wiley, New York, 2004.

[31] Hung, K. K., Cheung, C. C., and Xu, L., "New Sharpy-ratio-related methods for portfolio selection," *Proceedings of Computational Intelligence for Financial Engineering*, pp. 34-37, 2000.

[32] Jaffray, J.-Y. and Jeleva, M., "Information processing under imprecise risk with the Hurwicz criterion," *Proceedings of International Sumposium on Imprecise Probability: Theories and Applications*, Prague, Czeck Republic, 2007.

[33] Jaulin, L., Kieffer, M., Didrit, O., and Walter, E., *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.

[34] Joshi, M., *The Concepts and Practice of Mathematical Finance*. Cambridge, 2003.

[35] Kansas, D., *The Wall Street Journal: Complete money & investing guide book*. Three Rivers Press, New York, 2005.

[36] Kovalerchuk, B. and Vityaev, E., *Data Mining in Finance: Advances in Relational and Hybrid Methods*. Kluwer Academic Publishers, 2000.

[37] Krantz, D., Luce, R., Suppes, P., and Tverski, A., *Foundations of measurement*. Academic Press, 1971.

[38] Kwong, M., Bigman, D., and Taya, T., *Floating exchange rates and the state of world trade and payments*. Beard Books, 2002.

[39] Lai, K. K., Yu, L., Wang, S., and Zhou, C., "A double-stage genetic optimization algorithm for portfolio selection," in *ICONIP 2006*, part III, LNCS 4234, I. King et al. (Editors), pp. 928-937, Springer-Verlag Berlin Heidelberg, 2006.

[40] Lin, L., Cao, L., Wang, J., and Zhang, C., "The applications of genetic algorithms in stock market data mining optimization," in *Data Mining V*, A. Zanasi, N. F. F. Ebecken, and C. A. Brebbia (Editors), WIT Press, 2004.

[41] Lowe, D., "Novel exploitation of neural network methods in financial markets," *IEEE International Conference on Neural Networks*, no. 6, pp. 3623-3628, 1994.

[42] Ma, J. and Liu, L., "Multivatiate nonlinear analysis and prediction of Shangai stock market," *Discrets Dynamics in Nature and Society*, vol. 2008, 2008.

[43] Magoc, T., Ceberio, M., and Modave, F., "Interval-based multi-criteria decision making: strategies to order intervals," *Proceedings of North American Fuzzy Information Processing Society*, 2008.

[44] Magoc, T., Modave, F., Ceberio, M., and Kreinovich, V., "Computational methods for investment portfolio: the use of fuzzy measures and constraint programming for risk management,"

in *Foundation on Computational Intelligence*, A.-E. Hassanien and A. Abraham (Editors), Springer Verlag, 2008.

[45] Markowitz, H. M., “Portfolio Selection”, *Journal of Finance*, vol. 7, no. 1, pp. 77-91, 1952.

[46] Modave, F. and Grabisch, M., “Preferential independence and the Choquet integral,” 8^{th} *International Conference on the Foundations and Applications of Decision Under Risk and Uncertainty*, Mons, Belgium, 1997.

[47] Nilsson, N. J., *Artificial intelligence: a new synthesis*. Morgan Kaufman, 1998.

[48] Prugel-Bennett, A. and Shapiro, J. L., “An analysis of genetic algorithms using statistical mechanics,” *Physycal Review letters*, vol. 72, no. 9, pp. 1305-1309, 1994.

[49] Reeves, C. R. and Rowe, J. E., *Genetic algorithms-Principles and Perspective: A Guide to GA Theory*. Kluwer Academic Publisher, London, 2003.

[50] Rosenblatt, F., “The perceptron: a probabilistic model for information storage and organization in the brain,” in *Psychological Review*, vol. 65, no. 6, pp. 386-408, 1958.

[51] Scholkopf, B., Burges, C., and Vapnik, V., “Extracting support data for a given task,” *Advances in Neural Information Processing Systems*, 1995.

[52] Seo, Y., Giampapa, J., and Sycara, K., “Financial news analysis for intelligent portfolio management,” Tech. report CMU-RI-TR-04-03, Robotics Institute, Carnegie Mellon University, 2004.

[53] Shapley, L. S., “A value for n-person games,” in *Contributions to the Theory of Games*, H. W. Kuhn and A. W. Tucker (Editors), no. 2, pp. 307-317, Princeton University Press, 1953.

[54] Shreve, S. E., *Stochastic Calculus for finance I: the binomial asset pricing model*. Springer, 2005.

[55] Singer Y., “Switching portfolios,” *International Journal of Neural Systems*, pp. 488-495, 1998.

[56] Stock options, http://www.euronext.com/landing/landingInfo-1834-EN.html, April 3, 2009.

[57] Stock Scan, http://online.wsj.com/mdc/public/page/2_3022-scandiv.html?mod=topnav_2_3002, April 3, 2009.

[58] Stone, C. J., "Consistent nonparametric regression," *Annals of Statistics*, vol. 5, pp. 595-645, 1977.

[59] Sugeno, M., "Theory of fuzzy integrals and its applications," PhD Thesis, Tokyo Institute of Technology, 1974.

[60] Sycara, K., Decker, K., and Zeng, D., "Intelligent agents in portfolio management," in *Agent Technology: Foundations, Applications, and Markets*, N. Jennings and M. Wooldridge (Editors), Springer, 1998.

[61] Tan, P., Steinbach, M., and Kumar, V., *Introduction to Data Mining*. Addison Wesley, 2006.

[62] Universal stock futures, http://www.universal-stockfutures.com/, April 6, 2009.

[63] Wharton Research Data Services, http://wrds.wharton.upenn.edu/wrdsauth/members.cgi.

[64] Wilcoxon, F., "Individual comparisons by ranking methods," *Biometrics*, no. 1, pp. 80-83, 1945.

[65] Williams, G. J. and Simoff, S. J., *Data Mining: Theory, Methodology, Techniques, and Applications*. Lecture Notes in Artificial Intelligence. Springer, 2006.

[66] Yoon, J.-H., Kim, T. S., and Lee, H.-K., "The information of trading volume in the prediction of stock index returns: a nonparametric investigation," *Informs & Korms*, pp. 605-611, 2000.

[67] Zhou, C., Yu, L., Huang, T., Wang, S., and Lai, K. K., "Selecting valuable stocks using genetic algorithm," in *SEAL 2006*, LNCS 4247, T.-D. Wang et al. (Editors), pp. 688-694, Springer-Verlag Berlin Heidelberg, 2006.

[68] Zimmermann, H. G. and Grthmann, R., "Optimal asset allocation for a large number of investment opportunities," *Intelligent systems in Accounting, Finance, and Management*, no. 13, pp. 33-40, 2005.

[69] Zimmermann, H. G., Neuneier, R, and Grthmann, R., "Active portfolio management based on error correction neural networks," *Proceedings of Neural Network Information Processing systems*, Vancouver, Canada, 2002.

Appendix A. Data Used for Testing.

The following three tables contain the data used for the testing of the proposed algorithm. The table 12 contains the predicted return for the year 2005, the predicted risk, and the reputation of each company that offered stocks on the market at the end of the year 2004. These values represent the data known to investors at the time of making a decision of where to invest money. Moreover, the last column also provides the actual return of each company in the year 2005. This value is known only after the fact and was not used for selection the optimal portfolio but rather for calculating the return obtained by the selected portfolio, which was necessary for testing the performance of the algorithm. The tables 13 and 14 contain similar data for the years 2006 and 2007, respectively.

Table 12: Data for the year 2005.

Company	Predicted return	Predicted risk	Reputation	Exact return
AEROSPACE COMMUNICATIONS	0.0154396922	0.0510693824	0.8	-0.0173895185
ALONG TIBET	0.0095093819	0.0487030000	1.2	-0.0356112471
AMOI ELECTRONICS	-0.0078918953	0.0381232971	1.2	-0.0658452998
ANHUI GOLDEN SEED WINERY	0.0061979325	0.0358389602	0.6	-0.0469867602
ANHUI HELI	0.0177246005	0.0283475046	2.4	0.0178280364
ANHUI QUANCHAI ENGINE	-0.0093963077	0.0244966560	1.2	-0.0255875983
ANHUI WANWEI	0.0039715049	0.0261689443	0.6	-0.0190027265
BAIDA	0.0125074966	0.0336913745	0.8	0.0264547976
BAOTOU HUAZI	-0.0024091552	0.0183466465	0.6	-0.0265965967
BAOTOU TOMORROW TECH	-0.0080174283	0.0196843062	0.6	-0.0238003379
BEIJING C & W TECH	0.0195354961	0.0516760093	2	-0.0521168782
BEIJING DOUBLE-CRANE	0.0187310611	0.0746198766	2.4	0.0204062870
BEIJING TRADE	0.0036377885	0.0233226781	0.8	-0.0089112877
BEIJING TIANTAN	0.0033999252	0.0252405816	1.2	-0.0228328957
BEIJING TONGRENTANG	0.0119906266	0.0149412937	2.4	0.0000269691
BEIJING CONSTRUCTION	0.0106602760	0.0334881598	0.6	0.0057671866
BEIJING WANDONG	0.0150578364	0.0429510182	0.6	-0.0065727064
BEIJING WANGFUJING	0.0093555364	0.0240054585	1.6	0.0130197753
BEIJING XIDAN	0.0054260084	0.0283026280	0.8	-0.0104828098
BEIQI FOTON MOTOR	-0.0057816194	0.0349935894	1.2	-0.0667134996
BEIREN PRINTING	0.0133735149	0.0311834377	0.8	-0.0312665661
CHANG CHUN EURASIA	-0.0036491240	0.0167357497	0.8	0.0077009515
CHANGCHUN ECONOMICS	-0.0086473272	0.0233960910	0.8	-0.0115290482
CHANGCHUN AUTOMOBILE	0.0033143937	0.0367305058	1.6	-0.0339503753
CHANGCHUN YIDONG CLUTCH	-0.0019481125	0.0216951326	0.6	-0.0299453206
CHANGLIN	0.0278022036	0.0578229323	1.8	-0.0557097160
CHENGDU B-RAY	0.0076221691	0.0179195721	2.0	0.0140481374
CHENGDU DR. PENG TELECOM	0.0088670704	0.0313484576	1.0	-0.0510054006
CHENGDU QIANFENG	0.0016147593	0.0440603867	1.6	-0.0435400913
CHENGSHANG	0.0045905810	0.0300088490	0.8	-0.0095276988

Table 12: Data for the year 2005.

Company	Predicted return	Predicted risk	Reputation	Exact return
CHINA ANIMAL	0.0110369589	0.0357526979	1.2	-0.0191046412
CHINA CYTS TOURS	0.0086352184	0.0374011603	1.2	-0.0225441667
CHINA DONGFANGHONG	0.0322579381	0.0538537401	0.6	-0.0008651333
CHINA ENTERPRISE	0.0166721410	0.0328431461	0.8	0.0180739612
CHINA FIBERGLASS	0.0220651833	0.0471079061	0.8	0.0217393775
CHINA FIRST PENCIL	0.0053922775	0.0276051985	0.8	-0.0036939107
CHINA GEZHOUBA	-0.0004545065	0.0197001855	1.2	-0.0198121511
CHINA HI-TECH	-0.0024068990	0.0198940145	0.8	-0.0253863971
CHINA JIALING	0.0021963261	0.0221161738	1.0	-0.0232649677
CHINA RAILWAY	-0.0031620811	0.0200338720	1.6	0.0243966710
CHINA SATCOM	0.0040672307	0.0348589770	0.8	-0.0260773080
CHINA SHIPPING	0.0101699300	0.0267390287	0.8	-0.0191403071
CHINA SPORTS	0.0101647955	0.0303587126	0.6	-0.0086867004
CHINA SHIPBUILDING	0.0104907939	0.0258692226	1.2	0.0338274766
CHINA TELEVISION	0.0077596577	0.0301303535	1.8	-0.0180243984
CHINA WORLD TRADE	0.0088898095	0.0240527048	0.6	-0.0057317302
CHINA-KINWA HIGH TECH	0.0203989723	0.0367818873	1.6	-0.0279937285
CHONGQING BREWERY	-0.0069443423	0.0132769874	1.6	0.0034585050
CHONGQING DEPARTMENT STORE	-0.0071385615	0.0173856766	0.8	0.0179026607
CHONGQING ROAD	0.0044497563	0.0228328642	1.2	-0.0159757978
CHONGQING TAIJI	0.0050913193	0.0568054888	2.4	-0.0236970401
CHONGQING WATER AND ELECTRIC	-0.0001158558	0.0258019201	0.8	-0.0124412738
CHONGQING WANLI	0.0041649566	0.0497449955	0.8	-0.0198004960
CHONGQING SWELL	0.0155680385	0.0368039308	1.2	-0.0265849664
CITYCHAMP DARTONG	0.0160909364	0.0215396907	3.0	-0.0078974953
CNTIC TRADING	0.0075947071	0.0346203191	0.6	0.0249815073
COFCO XINJIANG TUNHE	0.0156658917	0.0928539629	3.2	0.0417132373
CRED HOLDING	0.0262910513	0.0941644557	3.2	-0.0249888860
CSC NANJING	0.0171895748	0.0414505154	1.6	0.0065383763
CSSC JIANGNAN	0.0043376589	0.0197533437	0.6	-0.0298226547
DALIAN DAXIAN	0.0076493587	0.0264798060	2.4	-0.0457753271
DALIAN THERMAL POWER	0.0045761989	0.0259935566	0.8	-0.0284087156
DASHANG	0.0208239604	0.0283397186	1.6	0.0501317824
DATANG HUAYIN ELECTRIC	0.0037146864	0.0231731864	1.6	-0.0522924309
DATANG TELECOM TECH	0.0035251722	0.0353556078	1.2	-0.0244643703
DAZHONG TRANSPORTATION	-0.0027434554	0.0167162064	0.8	-0.0132082204
DONG FANG BOILER	0.02983425353	0.0326316459	4.0	0.0009795740
DONGAN HEIBAO	-0.0095950705	0.0285477129	0.6	-0.0082062244
DONGFANG ELECTRIC	0.0166620783	0.0269330326	2.4	0.0023236007
DONGFENG AUTOMOBILE	0.0144236218	0.0377268973	3.0	-0.0127300223
DONGFENG ELECTRONIC	0.0272039786	0.0551510229	1.2	-0.0680103180
DOUBLE COIN HOLDINGS	0.0116824256	0.0349113203	0.8	0.0139607481
EASTERN COMMUNICATIONS	-0.0089720288	0.0332124284	0.6	-0.0340880603
FOUNDER TECH	0.0132390825	0.0273846990	3.0	-0.0379791199
FUJIAN CEMENT	0.0093672585	0.0295724793	1.6	-0.0447301228
FUJIAN DONGBAI	0.0048077329	0.0297450875	0.8	-0.0026646908
FUJIAN QINGSHAN PAPER	0.0103624890	0.0304681210	0.8	-0.0324718972
FUYAO GLASS	0.0383263810	0.0465334069	3.0	-0.0200611329
GANSU QILIANSHAN	0.0266899984	0.0482574736	0.6	-0.0421010809
GANSU TRISTAR	0.0125038388	0.0210893868	1.6	-0.0222070132
GANSU YASHENG	0.0154764682	0.0289719429	1.6	-0.0446311703
GD POWER DEVELOPMENT	0.0309833510	0.0426156655	1.6	0.0010680230

Table 12: Data for the year 2005.

Company	Predicted return	Predicted risk	Reputation	Exact return
GINWA ENTERPRISE	-0.0035452459	0.0244206935	0.6	-0.0297482881
GRINM SEMICONDUCTORs	-0.0099610312	0.0260373303	0.4	-0.0310803774
GUANGDON MEIYAN	0.0074834413	0.0186868928	1.6	-0.0426427214
GUANGDONG SHENGYI	0.0309720912	0.0624510601	2.4	-0.0096323015
GUANGZHOU DEVELOPMENT	0.0155105597	0.0308676268	1.2	-0.0037036237
GUANGZHOU IRON AND STEEL	0.0226412788	0.0345428727	1.6	-0.0493555705
GUANGZHOU PEARL	0.0000546717	0.0240706477	0.8	-0.0112456001
GUANGZHOU SHIPYARD	0.0135749855	0.0281847102	1.6	-0.0239132069
GUIZHOU CHANGZHENG	0.0104817937	0.0317817147	1.2	-0.0174898286
GUIZHOU LIYUAN	0.0077078101	0.0316310540	0.6	-0.0090361695
GUODIAN NANJING	-0.0096674001	0.0190389661	0.8	-0.0043590808
HABIN GONG	-0.0071991387	0.0195540377	0.8	-0.0372460003
HAINAN AIRLINES	0.0117976973	0.0297274827	1.0	-0.0364374066
HAINAN YEDAO	-0.0003618436	0.0314637977	0.8	-0.0091537293
HANDAN IRON & STEEL	0.0017409132	0.0168102894	1.2	0.0033200535
HANG ZHOU IRON & STEEL	0.0199980136	0.0341362553	1.2	-0.0383483546
HANGZHOU JIEBAI	0.0153958121	0.0376514817	0.8	0.0063610181
HANGZHOU TIAN-MU-SHAN	-0.0019840686	0.0275949132	1.0	-0.0118691978
HARBIN AIR CONDITIONING	0.0236263898	0.0462960376	1.6	0.0060747268
HARBIN DONGAN	0.0039520777	0.0256308792	0.6	0.0053129276
BAIDA	0.0121297281	0.0392516164	0.8	0.0152273178
HARBIN HIGH-TECH	-0.0139159589	0.0151575788	0.8	-0.0204947610
HARBIN PHARMACEUTICAL	0.0269084063	0.0350718310	2.4	-0.0037062252
HARBIN PHARMACEUTICAL (group)	0.0178474848	0.0437247996	2.0	-0.0055460125
HEBEI WEIYUAN	0.0047693041	0.0287421982	1.6	-0.0262335580
HEILNGJIANG HEIHUA	0.0027693739	0.0234839834	1.2	-0.0371329017
HENAN ANCAI HI-TECH	0.0180422799	0.0513444910	1.2	-0.0610028543
HENAN HUANGHE	0.0019559349	0.0282936634	1.2	-0.0108399449
HENAN PHARMACEUTICAL	-0.0003918719	0.0120644158	1.2	-0.0341575893
HENAN ORIENTAL SILVER STAR	-0.0106486687	0.0292997942	0.6	0.0021503423
HENAN YINGE	-0.0070918204	0.0156486122	0.8	0.0106175551
HISENSE ELECTRIC	0.0000339574	0.0215981066	0.6	0.0081252945
HIT SHOUCHUANG TECH	0.0006895741	0.0310463715	0.8	-0.0320645308
HUADIAN ENERGY	0.0104771162	0.0325489196	1.6	-0.0313748743
HUALIAN SUPERMARKET	0.0138704639	0.0364587979	0.8	-0.0350833583
HUANGSHAN TOURISM	0.0015786954	0.0235283574	1.2	0.0179271373
HUAXIN CEMENT	0.0168766036	0.0279334542	1.6	-0.0103845824
HUBEI EASTERN GOLD	0.0178148305	0.0674385954	1.6	-0.0411112420
HUBEI MAILYARD	-0.0089272363	0.0213616429	0.8	-0.0168326550
HUBEI XINGFA	-0.0001320607	0.0272183753	0.8	-0.0091538168
HUNAN HAILI	0.0089430511	0.0337431428	0.8	-0.0284950613
HUNAN HUASHENG	-0.0046953631	0.0290831533	0.6	-0.0231538116
HUNAN JINJIAN	0.0067511578	0.0344342376	0.6	-0.0389112514
INNER MONGOLIA BAOTOU	0.0159529615	0.0412530302	0.6	-0.0247149267
INNER MONGOLIA JINYU	0.0072236207	0.0208052877	0.8	-0.0303951822
INNER MONGOLIA MENGDIAN	0.0156762824	0.0366010009	2.4	-0.0279287534
INNER MONGOLIA YILI	0.0075023779	0.0227139017	1.6	0.0420846831
INSIGMA TECH	0.0123777760	0.0291006153	2.4	-0.0179938122
IRICO DISPLAY	0.0007272462	0.0314972830	1.6	-0.0546212225
JIANGSU CHENGXING	0.0159441650	0.0230807688	2.4	-0.0161099454
JIANGSU CHUNLAN	-0.0066937863	0.0248402458	0.8	-0.0056415844
JIANGSU HOLLY	0.0214702586	0.0583630891	0.8	-0.0123340985

Table 12: Data for the year 2005.

Company	Predicted return	Predicted risk	Reputation	Exact return
JIANGSU HONGTU HIGH TECH	0.0036739396	0.0337646602	0.6	-0.0172636484
JIANGSU SOPO	0.0098901217	0.0467223115	0.6	-0.0245754772
JIANGSU SUNSHINE	0.0070181827	0.0277512286	0.4	-0.0343913589
JIANGSU WUZHONG	0.0241487671	0.0243175743	3.2	0.0258638142
JIANGSU YONGDING	-0.0026013829	0.0199692484	0.8	-0.0153941820
JIANGSU YUEDA	0.0002756255	0.0341949476	2.0	-0.0288412487
JIANGSU ZONGYI	-0.0105567669	0.0248961179	1.6	-0.0116897357
JIANGXI CHANGJIU	0.0202427546	0.0697594910	0.8	-0.0368038461
JIANGXI JIANGZHONG	-0.0019033197	0.0198153461	0.6	-0.0028844363
JILIN FOREST	0.0060931411	0.0214385942	1.2	0.0009310565
JILIN YATAI	0.0043310600	0.0246206227	2.0	-0.0091632232
JINGNENG PROPERTY	0.0041748692	0.0291731339	0.8	-0.0086578980
JINZHOU PORT	0.0102537397	0.0286553651	1.6	-0.0365614835
JONJEE HIGH & NEW TECH	0.0030614608	0.0250665247	0.8	-0.0275657851
LANZHOU GREAT WALL	-0.0021150966	0.0239923213	0.6	-0.0070251986
LANZHOU MINBAI	0.0135740096	0.0446992686	0.8	-0.0158835349
LAWTON DEVELOPMENT	0.0089693899	0.0395216031	0.8	-0.0464246698
LESHAN ELECTRIC	0.0288524581	0.0566170412	1.0	-0.0139260856
LIAONIN GUONENG	-0.0213618513	0.0124146058	0.0	-0.0115797744
LIAONING CHENG	0.0083365509	0.0420959899	1.6	0.0014504112
LIAONING HONGYANG	0.0010544147	0.0334057880	0.8	0.0038071080
LIAONING TIMES GARMENTS	0.0081667979	0.0313271950	0.8	-0.0474844668
LINHAI	0.0094271500	0.0497550547	1.2	-0.0188570912
LONG MARCH	0.0222029945	0.0320509174	2.0	0.0090039448
LUCKYFILM	0.0024223412	0.0270903950	1.2	-0.0566646088
LUOYANG GLASS	0.0083515633	0.0325944342	0.8	-0.0302604039
MAANSHAN IRON & STEEL	-0.0032273860	0.0228557131	0.8	-0.0242911187
MEIDU HOLDING	-0.0020456628	0.0348073933	0.8	0.0094284751
MIANYANG GAOXIN	-0.0067443381	0.0345414281	1.0	-0.0178493707
MINMETALS DEVELOPMENT	0.0157069312	0.0340652128	1.2	-0.0024556970
NANJING CHEMICAL	0.0166509110	0.0363143383	1.6	-0.0385649502
NANJING MEDICAL	0.0029028561	0.0285908884	0.8	-0.0050483154
NANJING PANDA ELECTRONICS	0.0031444233	0.0343513383	0.8	-0.0013368594
NANJING XINGANG HIGH-TECH	-0.0101851051	0.0180544190	0.8	-0.0093707820
NANJING XINJIEKOU	0.0056401628	0.0230143617	1.6	0.0037256021
NANNING	-0.0012007711	0.0207462597	0.8	-0.0233020435
NANZHI	0.0001159467	0.0271181473	0.6	-0.0178655320
NEUSOFT	0.0008365523	0.0315203533	2.0	-0.0223916655
NING XIA HENG LI	0.0092369483	0.0509582479	1.2	-0.0370923311
NINGBO FUBANG JINGYE	0.0054031814	0.0170134152	2.4	-0.0569879814
NINGBO FUDA	0.0359020205	0.0517058133	1.6	0.0007309126
NINGBO MARINE	0.0063989338	0.0245110803	0.6	-0.0207622040
NINGBO SHANSHAN	-0.0010202413	0.0170473478	1.6	-0.0249273477
NINGBO UNITED	0.0069514990	0.0360057474	0.8	-0.0015509757
NINGBO VEKEN ELITE	0.0002564613	0.0188799375	0.8	-0.0246223668
NORTH CHINA	0.0111180907	0.0237451379	1.6	-0.0493947824
NORTHEAST EXPRESSWAY	0.0022824553	0.0190829961	2.0	-0.0200819065
ORIENT INCORPORATION	0.0127178670	0.0319831458	1.6	-0.0015653440
PHENIX OPTICAL	0.0209116247	0.0412171770	1.2	-0.0389554902
PHOENIX	0.0169074458	0.0667538246	2.4	-0.0171635165
QINGDAO HAIER	0.0074842169	0.0282747782	1.0	-0.0133980468
QINGHAI JINRUI MIMERAL	0.0226928460	0.0600766844	1.8	-0.0106001182

Table 12: Data for the year 2005.

Company	Predicted return	Predicted risk	Reputation	Exact return
S & P PHARMACEUTICAL	-0.0004379233	0.0326941368	0.8	0.0085160455
SAIC MOTOR CORPORATION	0.0088457714	0.0463481896	1.2	-0.0011634127
SGSB	0.0109749122	0.0304114259	0.8	-0.0254760964
SH FRIENDSHIP	0.0208573570	0.0320220553	1.6	-0.0028753577
SH JINJIANG HOTELS	0.0085980029	0.0278134704	1.6	0.0107509888
SH JINQIAO EXPORT	0.0016411353	0.0206401984	0.8	0.0138962826
SH POWER TRANSMISSION	0.0019259563	0.0201381796	0.8	0.0322518642
SHAANXI BROADCAST	0.0173507591	0.0471876206	2.4	-0.0033040613
SHAANXI QINLING	0.0241667683	0.0567803291	2.4	-0.0528208114
SHAN DONG HEUNGKONG	0.0174937696	0.0337388514	1.6	-0.0319116902
SHANDONG DACHENG	0.0007632056	0.0258132954	1.0	-0.0247766675
SHANDONG JIANGQUAN	-0.0028718478	0.0251804457	0.8	-0.0207022513
SHANDONG JIUFA EDIBLE	-0.0020703775	0.0305075318	1.8	-0.0238752732
SHANDONG LANGCHAO	0.0448178998	0.0666790663	1.6	-0.0289801981
SHANDONG LUBEI	-0.0150433492	0.0148977231	0.0	-0.0198902995
SHANDONG LUKANG	0.0041984441	0.0335785569	0.8	-0.0140281810
SHANDONG NANSHAN	0.0056136405	0.0272742947	1.6	-0.0118810876
SHANDONG TYAN HOME	0.0116409389	0.0572807254	0.8	-0.0383120654
SHANDONG LUXIN HIGH-TECH	-0.0017284639	0.0299406900	2.0	-0.0215979903
SHANG HAI YA TONG	-0.0018272354	0.0205629651	1.6	-0.0021458140
SHANGHAI 3F NEW MATERIALS	0.0152613105	0.0245006081	2.4	0.0200789903
SHANGHAI AEROSPACE	0.0124844108	0.0359826126	0.6	0.0294390237
SHANGHAI AJ	0.0037890198	0.0228088133	1.6	-0.0506289247
SHANGHAI BAOSIGHT	0.0178213116	0.0340673606	1.6	-0.0073185934
SHANGHAI BELLING	0.0005565718	0.0207647335	0.6	-0.0312791739
SHANGHAI CHENGTOU	0.0057930926	0.0185871190	1.6	-0.0091436455
SHANGHAI CHLOR-ALKALI	0.0115604198	0.0289239029	0.8	-0.0299078139
SHANGHAI CONSTRUCTION	0.0067443883	0.0213704543	0.6	-0.0010356781
SHANGHAI DAZHONG	0.0065543147	0.0283353222	1.0	-0.0234394624
SHANGHAI DINGLI TECH	0.0197332582	0.0545871894	0.8	-0.0071417957
SHANGHAI DRAGON	0.0052680296	0.0328120519	0.8	-0.0244178702
SHANGHAI DUOLUN	0.0104983642	0.0343326419	0.8	-0.0316036516
SHANGHAI EAST-CHINA COMPUTER	0.0220299250	0.0520083763	0.8	-0.0240656049
SHANGHAI ELECTRIC	0.0152037248	0.0304887830	1.6	-0.0329842578
SHANGHAI ERFANGJI	0.0058620699	0.0294839547	1.0	-0.0115159666
SHANGHAI FEILO ACOUSTICS	0.0379102152	0.0535390043	2.0	-0.0187188873
SHANGHAI FEILO	0.0047800178	0.0264976292	1.0	-0.0153706080
SHANGHAI FIRST PROVISIONS	0.0283276827	0.0259640640	3.2	0.0310512757
SHANGHAI FOSUN	0.0107478418	0.0282266811	1.8	0.0001026467
SHANGHAI FUDAN	0.0038670019	0.0274039422	1.6	-0.0362689042
SHANGHAI HAIBO	0.0033136350	0.0161494897	0.8	0.0029247551
SHANGHAI HAINIAO	0.0011602501	0.0388616898	0.8	0.0119190023
SHANGHAI HAIXIN	0.0165268991	0.0263186242	1.6	-0.0307032650
SHANGHAI HIGHLY	0.0140745310	0.0337887431	1.6	0.0142284521
SHANGHAI HONGSHENG TECH	0.0233056137	0.0489167538	0.8	0.0221489827
SHANGHAI HUITONG	-0.0005445953	0.0241316949	1.0	-0.0119518028
SHANGHAI PHARMACEUTICAL	0.0064639944	0.0230787761	1.0	0.0032024230
SHANGHAI AIRPORT	0.0108059983	0.0204092040	2.4	-0.0039642640
SHANGHAI JIABAO	0.0036208084	0.0308895800	0.8	-0.0094483384
SHANGHAI JIAO DA NAN YANG	0.0043375720	0.0341931188	0.8	0.0145210163
SHANGHAI JIAO YUN	0.0044780362	0.0211744202	1.6	-0.0129961983
SHANGHAI JIELONG	0.0052952961	0.0458658922	1.6	-0.0112317019

Table 12: Data for the year 2005.

Company	Predicted return	Predicted risk	Reputation	Exact return
SHANGHAI JIN JIANG	0.0142418960	0.0317578518	1.6	0.0089393754
SHANGHAI JINFENG	0.0082419463	0.0370289355	0.8	-0.0100796523
SHANGHAI JINLING	0.0054461478	0.0344277137	1.6	-0.0127350682
SHANGHAI JOIN BUY	0.0015487590	0.0211458148	0.8	-0.0390292154
SHANGHAI LANSHENG	-0.0003200655	0.0331672429	1.6	-0.0154466397
SHANGHAI LUJIAZUI	-0.0044620960	0.0169355784	0.8	-0.0080981316
SHANGHAI MALING AQUARIUS	0.0192503330	0.0309251867	1.2	-0.0327926656
SHANGHAI MATERIAL TRADING	0.0108243521	0.0349691030	1.6	0.0035879394
SHANGHAI MET	0.0089913782	0.0565318231	2.4	-0.0238372756
SHANGHAI NEW HUANG	-0.0078368715	0.0172699420	0.8	-0.0111616601
SHANGHAI NEW WORLD	0.0139715203	0.0245402467	2.0	0.0321424081
SHANGHAI ORIENTAL PEARL	0.0060781119	0.0233297259	2.4	0.0129953062
SHANGHAI PHARMACEUTICAL	0.0079280823	0.0273983427	0.8	-0.0167462876
SHANGHAI POTEVIO	0.0118961131	0.0402683133	0.8	-0.0060153292
SHANGHAI PUDONG	-0.0132328080	0.0156075610	0.8	0.0294837597
SHANGHAI QIANGSHENG	0.0152747657	0.0322699768	1.6	0.0017782576
SHANGHAI SANMAO	0.0028596010	0.0311745852	0.8	-0.0167738638
SHANGHAI SHENDA	0.0042151921	0.0235243960	1.6	-0.0047430832
SHANGHAI SHENHUA	0.0048356931	0.0203419059	2.0	-0.0482854864
SHANGHAI SHENTONG METRO	0.0269585999	0.0558571945	0.8	0.0319099052
SHANGHAI SHIMAO	0.0176685226	0.0268469434	1.6	-0.0405279557
SHANGHAI TIANCHEN	-0.0013478272	0.0335400425	1.6	-0.0229862383
SHANGHAI TONGDA	0.0050509823	0.0283529139	1.6	-0.0138048576
SHANGHAI TONGJI	-0.0065138604	0.0204250199	0.8	-0.0137444753
SHANGHAI TUNNEL	0.0009754267	0.0240450381	1.6	-0.0096986075
SHANGHAI WANYE	-0.0036965170	0.0314910629	1.0	0.0069742552
SHANGHAI WHITECAT	0.0015839000	0.0265945231	2.4	-0.0005128345
SHANGHAI WORLDBEST	-0.0034211084	0.0215041001	0.6	-0.0257621818
SHANGHAI XINGYE	-0.0077286306	0.0317917900	1.0	-0.0263415586
SHANGHAI YAOHUA	-0.0033675872	0.0185282044	2.4	-0.0339882046
SHANGHAI YIMIN	0.0104355870	0.0269950743	0.8	0.0373110578
SHANGHAI YUYUAN	0.0076958088	0.0273240059	0.8	0.0095255883
SHANGHAI ZHANGJIANG	0.0177295042	0.0242681344	1.8	-0.0334392984
SHANGHAI ZI JIANG	0.0124411222	0.0223601705	0.8	-0.0173164520
SHANXI COKING	0.0186619919	0.0381622101	1.2	-0.0587869602
SHANXI LANHUA SCI-TECH	0.0143283345	0.0270950754	1.8	0.0033982876
SHANXI TOP ENERGY	0.0212483790	0.0335557719	2.4	-0.0290171014
SHANXI XINGHUACUN	0.0127313541	0.0259495636	1.6	0.0264739061
SHEN MA INDUSTRY	0.0058858254	0.0259033092	0.8	-0.0088782654
SHENERGY	0.0123769352	0.0360493950	1.6	0.0098061346
SHENJI KUNMING	0.0149428809	0.0445857887	0.8	-0.0240867529
SHENYANG JINBEI	0.0052473338	0.0353948094	2.0	-0.0484807884
SICHUAN CHANGHONG	-0.0169467677	0.0223147052	1.0	0.0056927741
SICHUAN CHUANTOU	0.0298501010	0.0551018407	1.0	-0.0211460326
SICHUAN GOLDEN SUMMIT	0.0068711846	0.0294777520	1.0	-0.0345857959
SICHUAN HEJIA	0.0059413731	0.0263029130	1.2	-0.0411990676
SICHUAN MINGXING	-0.0004633026	0.0209752502	2.0	-0.0380703853
SICHUAN MINJIANG	0.0095985217	0.0234506138	1.6	-0.0140904354
SICHUAN SWELLFUN	-0.0067705021	0.0317786497	0.8	-0.0413796443
SILVERTIE HOLDING	0.0104805792	0.0323098225	1.6	-0.0218379307
SINO CONSTRUCTION	0.0121498168	0.0289125576	0.8	-0.0321435787
SINOPEC SHANGHAI	0.0135494913	0.0319688914	1.6	-0.0103273357

Table 12: Data for the year 2005.

Company	Predicted return	Predicted risk	Reputation	Exact return
SINOPEC YIZHENG	0.0069569780	0.0226994739	1.6	-0.0372038141
SINOTEX INVESTMENT	0.0234230955	0.0462219121	0.6	-0.0226472237
SONGLIAO AUTOMOTIVE	0.0050570234	0.0614352777	1.6	-0.0095883556
SOUTHWEST	0.0090198528	0.0345037117	0.8	-0.0125327689
STAR LAKE BIOSCIENCE	0.0225568620	0.0443034543	1.6	-0.0296695274
SUNNY LOAN TOP	0.0021847169	0.0205753035	1.6	-0.0243575486
SUNTEK TECHNOLOGY	0.0005413380	0.0316062101	0.8	-0.0636272704
SUZHOU NEW DISTRICT HI-TECH	-0.0063824502	0.0178501408	0.8	0.0038417755
SVA ELECTRON	0.0107320280	0.0279235575	2.0	-0.0609404698
SVA INFORMATION	-0.0046951836	0.0189324558	0.8	-0.0375278722
TAIYUAN HEAVY INDUSTRY	0.0109208129	0.0357639407	1.2	-0.0403243364
TIANJIN ENVIRONMENTAL	0.0171662930	0.0351826064	1.6	-0.0370071681
TIANJIN HI-TECH	0.0021263551	0.0154384030	0.8	-0.0232045305
TIANJIN PORT	0.0278508551	0.0242782356	3.2	0.0003022200
TIANJIN QUANYE BAZAAR	0.0005566655	0.0245211580	0.8	-0.0151378967
TIBET RHODIOLA	-0.0105541975	0.0228225563	0.4	0.0210192450
TIBET TOURISM	0.0051597454	0.0413918746	1.2	0.0047659560
TONGHUA DONGBAO	-0.0126336377	0.0647687404	0.8	-0.0210728542
TOPSUN SCIENCE AND TECH	0.0023227817	0.0278172360	0.6	-0.0175982722
TSINGHUA TONGFANG	0.0118239510	0.0299318620	1.2	-0.0348324090
TSINGTAO BREWERY	0.0101644548	0.0187512719	3.2	-0.0111385803
TUOPAI YEAST LIQUOR	0.0043102862	0.0368766957	0.8	-0.0002250985
WEIFANG BEIDA	0.0047882730	0.0474964489	0.8	-0.0351189026
WINOWNER	-0.0043981049	0.0487793790	0.8	-0.0108343254
WINSAN	0.0076718028	0.0390714521	0.8	-0.0256554736
WUHAN EAST LAKE HIGH TECH	0.0029400441	0.0314712477	1.6	-0.0314848177
WUHAN HANSHANG	0.0036302854	0.0381609557	1.0	0.0066275681
WUHAN HUMANWELL HI-TECH	0.0207905715	0.0505213666	0.8	-0.0115097217
Wuhan LINUO SOLAR ENERGY	0.0051281348	0.0316461571	0.8	0.0362052332
WUHAN SANZHEN	-0.0003570590	0.0233498883	0.6	-0.0109625443
WUHAN STEEL	0.0212878099	0.0216622223	1.8	-0.0088389852
WUHAN XIANGLONG	0.0017361416	0.0246824294	0.8	-0.0341664603
WUXI TAIJI	0.0022814829	0.0227542520	1.0	-0.0393423856
XIAMEN C & D	0.0036795026	0.0251289197	1.2	-0.0146066209
XIAMEN ENGINEERING	0.0129552907	0.0339768194	1.6	-0.0353674449
SHANGHAI FEILO	0.0063953306	0.0214593232	0.6	0.0168530684
XIAMEN INTERNATIONAL TRADE	0.0028767948	0.0190944625	1.6	0.0015972232
XIAMEN KING LONG MOTOR	0.0159579344	0.0268229961	0.8	0.0154246970
XIAMEN ELECTRONICS	-0.0226843733	0.0043682308	0.0	-0.0076963455
XIAMEN PROSOLAR	0.0091185370	0.0352406353	0.4	-0.0172609270
XI'AN SEASTAR	-0.0118810132	0.0208346445	0.4	-0.0283366272
XINING SPECIAL STEEL	0.0003614106	0.0229849262	1.2	-0.0292096408
XINJIANG FRIENDSHIP	0.0101047374	0.0361011337	0.8	-0.0360771402
XINJIANG JOINWORLD	0.0127710236	0.0285825119	0.6	-0.0026290056
XINJIANG TALIMU	0.0079785258	0.0358320180	0.8	-0.0199135363
XINJIANG TIANYE	0.0104235012	0.0328261260	1.2	0.0012336941
XINJIANG YILITE	0.0034106696	0.0211134000	0.4	-0.0079189560
XINYU IRON & STEEL	0.0058683209	0.0402752299	1.8	-0.0064029164
Y.U.D. YANGTZE RIVER	-0.0015257198	0.0296050213	1.8	-0.0194446715
YANTAI HUALIAN	0.0041063506	0.0501704214	2.0	-0.0376587264
YANTAI XINCHAO	0.0084514201	0.0324674061	1.0	-0.0255050323
YANZHOU COAL MINING	0.0183101115	0.0216652625	1.8	-0.0203574519

Table 12: Data for the year 2005.

Company	Predicted return	Predicted risk	Reputation	Exact return
YIBIN PAPER	0.0018534285	0.0285115772	2.0	-0.0252794852
YOUNGOR	0.0149172345	0.0262143505	1.6	0.0008887094
YUNNAN BOWIN TECH	0.0054212735	0.0243059238	1.0	-0.0038521551
YUNNAN FREETRADE	0.0050324887	0.0411350365	1.6	-0.0349282611
YUNNAN MALONG	0.0229548152	0.0275052233	0.6	-0.0420053942
YUNNAN YUNTIANHUA	0.0164177327	0.0251431543	1.8	-0.0128327085
YUNNAN YUNWEI	0.0031913302	0.0224108216	0.6	0.0014720542
ZHE JIANG DONG RI	0.0002616010	0.0266423202	0.6	-0.0104952931
ZHEJIANG CHINA LIGHT	0.0053295307	0.0252579494	1.6	-0.0358232466
ZHEJIANG FURUN	0.0049106039	0.0294422793	0.8	-0.0224679119
ZHEJIANG GUYUELONGSHAN	-0.0028430711	0.0252923652	1.2	0.0095183791
ZHEJIANG HOLLEY TECH	0.0014786538	0.0247433768	1.2	-0.02322351112
ZHEJIANG HSD INDUSTRIAL	0.0245790636	0.0524959830	0.8	-0.0130911263
ZHEJIANG JUHUA	0.0011211441	0.0186538275	0.6	0.0213251649
ZHEJIANG MEDICINE	0.0108841063	0.0437631796	1.2	0.0306595930
ZHEJIANG ORIENT	0.0064789523	0.0354569985	0.8	-0.0252615295
ZHEJIANG QIANJIANG	0.0038340761	0.0277793064	1.6	0.0105467077
ZHEJIANG SHENGHUA BIOK	0.0181247400	0.0541339321	0.8	-0.0136542794
ZHEJIANG XINHU VENTURE	0.0098064036	0.0358807956	2.0	-0.0227396352
ZHENGZHOU COAL & ELECTRIC	0.0112173084	0.0492843472	0.6	-0.0209907220
ZHENGZHOU YUTONG BUS	0.0083774770	0.0134802337	2.4	0.0039035503
ZHONGCHU DEVELOPMENT STOCK	-0.0026576853	0.0218424216	1.2	-0.0044211009
ZHONGLU	0.0150940994	0.0475489999	1.6	0.0473285192

Table 13: Data for the year 2006.

Company	Predicted return	Predicted risk	Reputation	Exact return
AEROSPACE COMMUNICATIONS	0.02423422154	0.0171245668	0.0	0.0922584137
ALONG TIBET	-0.1448548471	0.0332372422	0.8	0.0472296149
AMOI ELECTRONICS	-0.1916903579	0.0474180862	1.2	0.0402277653
ANHUI GOLDEN SEED WINERY	-0.0499529177	0.0100161207	0.0	0.0893158194
ANHUI HELI	0.0626166827	0.0178189077	2.4	0.1282559716
ANHUI QUANCHAI ENGINE	0.0228904624	0.0188233897	0.6	0.0392961001
ANHUI WANWEI	-0.0328547667	0.0042844513	0.0	0.0711528730
BAIDA	0.0881517150	0.0209420048	1.0	0.0521177223
BAOTOU HUAZI	0.0034821554	0.0097014611	0.0	0.0574008571
BAOTOU TOMORROW TECH	0.0291426979	0.0108923362	0.0	0.02427862649
BEIJING C & W TECH	-0.0950513602	0.0324205437	1.0	0.01999701155
BEIJING DOUBLE-CRANE	0.0759444244	0.0650597380	2.4	0.0571292290
BEIJING TRADE	-0.0019394558	0.0047453997	0.0	0.0565170949
BEIJING TIANTAN	0.0251300936	0.0171798003	0.6	0.1409751046
BEIJING TONGRENTANG	0.0069568420	0.0152989984	2.4	0.0191180509
BEIJING CONSTRUCTION	0.0267804205	0.0166211470	0.6	0.0949166891
BEIJING WANDONG	0.0413161222	0.0180466521	0.0	0.0288032125
BEIJING WANGFUJING	0.0528289122	0.0139361610	1.6	0.1404787046
BEIJING XIDAN	-0.0009928524	0.0065684281	0.0	0.0664203144
BEIQI FOTON MOTOR	-0.0297905654	0.0449073814	1.6	0.0895179143
BEIREN PRINTING	-0.0363928703	0.0082047020	0.0	0.0304179886
CHANG CHUN EURASIA	0.0631710428	0.0142176106	1.0	0.0769414441
CHANGCHUN ECONOMICS	0.0235239653	0.0120622024	0.0	0.0525676853
CHANGCHUN AUTOMOBILE	-0.1233206301	0.0185023836	0.8	0.0449410297
CHANGCHUN YIDONG CLUTCH	-0.0452318090	0.0040662068	0.0	0.0470633795
CHANGLIN	-0.0607338261	0.0444053676	1.6	0.0775117704
CHENGDU B-RAY	0.0647831779	0.0131086535	2.0	0.0569385773
CHENGDU DR. PENG TELECOM	-0.1171562774	0.0161648056	0.0	0.1584980064
CHENGDU QIANFENG	0.0167695531	0.0265052242	1.0	0.0568218148
CHENGSHANG GROUP	-0.0050130464	0.0076917016	0.0	0.0922450101
CHINA ANIMAL	-0.0607932096	0.0230306222	0.6	0.1156410708
CHINA CYTS TOURS	-0.0037224863	0.0201538659	0.6	0.0971821768
CHINA DONGFANGHONG	0.0230894994	0.0131494670	0.0	0.1025899504
CHINA ENTERPRISE	0.0343405054	0.0165972250	0.8	0.0864294354
CHINA FIBERGLASS	0.1685674255	0.0359940791	1.2	0.0627846057
CHINA FIRST PENCIL	0.0070250763	0.0087747363	0.0	0.0668645995
CHINA GEZHOUBA	0.0128064199	0.0122200132	0.6	0.0687106785
CHINA HI-TECH	-0.0043110230	0.0111562199	0.0	0.0246617890
CHINA JIALING	-0.0227579983	0.0056603055	0.0	0.0536097864
CHINA RAILWAY	0.1042028490	0.0241433097	1.6	0.0595619231
CHINA SATCOM	0.0209846813	0.0109008530	0.0	0.0699039558
CHINA SHIPPING	0.0158516194	0.0084352854	0.0	0.0499548902
CHINA SPORTS	0.0412012105	0.0127846481	0.0	0.0740837964
CHINA SHIPBUILDING	0.0421323550	0.0265843542	1.2	0.1206060808
CHINA TELEVISION	0.0655274781	0.0232637317	1.2	0.0648826422
CHINA WORLD TRADE	0.0332086260	0.0088429324	0.0	0.0862386452
CHINA-KINWA HIGH TECH	-0.0378188691	0.0272715261	0.8	0.0611587819
CHONGQING BREWERY	-0.0277890958	0.0124022766	2.4	0.1054604333
CHONGQING DEPARTMENT STORE	0.0886016673	0.0200150433	1.0	0.0850678269
CHONGQING ROAD	0.0006858662	0.0114022485	0.6	0.0282572473
CHONGQING TAIJI	-0.1306169750	0.0426901825	1.6	0.0387633427

Table 13: Data for the year 2006.

Company	Predicted return	Predicted risk	Reputation	Exact return
CHONGQING WATER AND ELECTRIC	-0.0086587952	0.0080477920	0.0	0.0289437594
CHONGQING WANLI	-0.0686745519	0.0239255685	0.0	0.0594550115
CHONGQING SWELL	-0.0422944793	0.0149413952	0.6	0.0500313053
CITYCHAMP DARTONG	0.0015350219	0.0111083230	2.0	0.0611755523
CNTIC TRADING	0.0911359596	0.0243453729	0.6	0.0425812448
COFCO XINJIANG TUNHE	0.0202317019	0.0834320988	3.2	0.0872275342
CRED HOLDING	-0.1307112257	0.0768884632	2.4	0.0563605054
CSC NANJING	0.1122024301	0.0317205206	1.6	0.0416874295
CSSC JIANGNAN	-0.0820445141	0.0111304140	0.0	0.0845625580
DALIAN DAXIAN	-0.0549839867	0.0261804062	1.6	0.0629406671
DALIAN THERMAL POWER	0.0015239547	0.0107639560	0.0	0.0310942181
DASHANG	0.1440403816	0.0297343584	1.6	0.0642978030
DATANG HUAYIN	-0.0332611841	0.0207218727	0.8	0.0410680565
DATANG TELECOM TECH	0.0135769479	0.0237103472	0.6	0.0667125589
DAZHONG TRANSPORTATION	-0.0014856466	0.0105427489	0.0	0.0670953560
DONG FANG BOILER	0.1084634514	0.0320300196	4.0	0.0685702767
DONGAN HEIBAO	-0.0008648489	0.0175514872	0.0	0.0322305760
DONGFANG ELECTRIC	0.1174180671	0.0243982225	2.4	0.0913535890
DONGFENG AUTOMOBILE	-0.1420932335	0.0341036599	2.0	0.0378609699
DONGFENG ELECTRONIC	-0.1266911274	0.0305004165	0.6	0.0349325924
DOUBLE COIN HOLDINGS	0.0241072489	0.0175568118	1.0	0.0189004450
EASTERN COMMUNICATIONS	-0.0171538889	0.0127575488	0.0	0.1319190975
FOUNDER TECHN	0.0255229525	0.0269140753	2.0	0.0231616079
FUJIAN CEMENT	-0.0129399280	0.0172687417	0.8	0.0622426992
FUJIAN DONGBAI	0.0257795648	0.0123250631	0.0	0.1096486224
FUJIAN QINGSHAN	-0.0197955032	0.0094158319	0.0	0.0446979527
FUYAO GLASS	0.0833962701	0.0301381750	2.0	0.0983563945
GANSU QILIANSHAN	-0.0430746328	0.0128797768	0.0	0.0641901014
GANSU TRISTAR	-0.0100875114	0.0089207395	0.8	0.0375609912
GANSU YASHENG	-0.0417591204	0.0181450848	0.8	0.0791708304
GD POWER DEVELOPMENT	0.1184760506	0.0256857560	2.0	0.0187815812
GINWA ENTERPRISE	-0.0233166618	0.0028459257	0.0	0.0487428746
GRINM SEMICONDUCTORs	0.0169527210	0.0141280156	0.0	0.1078732403
GUANGDONG MEIYAN	-0.0318411193	0.0198554385	0.8	0.0633218727
GUANGDONG SHENGYI	0.1089087335	0.0388449618	1.6	0.0739022495
GUANGZHOU DEVELOPMENT	0.0441598278	0.0131660361	0.6	0.0310072744
GUANGZHOU IRON AND STEEL	-0.0337300012	0.0225048648	0.8	0.0373265773
GUANGZHOU PEARL	-0.0121003904	0.0062710866	0.0	0.0588505533
GUANGZHOU SHIPYARD	0.0345084887	0.0173353106	0.8	0.1604232482
GUIZHOU CHANGZHENG	0.0559068730	0.0165178395	0.6	-0.0031858467
GUIZHOU LIYUAN	-0.0088986758	0.0072995141	0.0	0.0976750329
GUODIAN NANJING	0.0704835655	0.0180792935	0.6	0.0640074673
HABIN GONG	-0.0354030651	0.0104556545	0.0	0.0318487833
HAINAN AIRLINES	-0.0310714619	0.0123421357	0.0	0.0570576053
HAINAN YEDAO	0.0309545883	0.0165063482	0.0	0.0515368784
HANDAN IRON & STEEL	0.0718163091	0.0143152188	1.2	0.0393080150
HANG ZHOU IRON & STEEL	0.0149992604	0.0245915064	0.6	0.0303251212
HANGZHOU JIEBAI	0.0373442725	0.0130161187	1.0	0.0671287402
HANGZHOU TIAN-MU-SHAN	-0.0132525898	0.0110764883	0.0	0.0508517792
HARBIN AIR CONDITIONING	0.1147959259	0.0265040797	1.6	0.0134386531
HARBIN DONGAN	0.0126544951	0.0146028406	0.6	0.0295228480
BAIDA	0.1113964168	0.0211731540	0.8	0.0834456223

Table 13: Data for the year 2006.

Company	Predicted return	Predicted risk	Reputation	Exact return
HARBIN HIGH-TECH	0.0026791952	0.0051131090	0.0	0.0524630085
HARBIN PHARMACEUTICAL	-0.0708089052	0.0247873871	1.6	0.0745680725
HARBIN PHARMACEUTICAL (group)	-0.1204348387	0.0377436792	1.0	0.0275799186
HEBEI WEIYUAN	-0.0617312864	0.0143267123	1.0	0.0672088346
HEILNGJIANG HEIHUA	-0.0073931325	0.0170869109	0.6	0.0539714646
HENAN ANCAI HI-TECH	0.0208190901	0.0316806888	0.6	0.0053190197
HENAN HUANGHE	-0.0205165328	0.0197177322	0.6	0.0291560549
HENAN PHARMACEUTICAL	-0.0240887783	0.0144821490	0.6	0.0369851717
HENAN ORIENTAL SILVER STAR	0.0237800778	0.0208364822	0.8	0.0365905510
HENAN YINGE	0.0666890109	0.0170250146	0.8	0.0264097273
HISENSE ELECTRIC	0.0910558525	0.0176116832	0.6	0.0164270562
HIT SHOUCHUANG TECH	-0.0293193499	0.0143601619	0.0	0.0261632705
HUADIAN ENERGY	0.0079713933	0.0201121723	0.8	0.0210095511
HUALIAN SUPERMARKET	-0.0516020174	0.0124760198	0.0	0.1691777261
HUANGSHAN TOURISM	0.0420286812	0.0221420106	1.6	0.0752901624
HUAXIN CEMENT	0.0131139559	0.0065957059	0.8	0.0552722402
HUBEI EASTERN GOLD	-0.0250672576	0.0441770266	0.8	0.1434032503
HUBEI MAILYARD	0.0112888504	0.0064735454	0.0	0.0527139279
HUBEI XINGFA	0.0037470659	0.0128287294	0.0	0.0633027090
HUNAN HAILI	-0.0212713622	0.0080035659	0.0	0.0234363488
HUNAN HUASHENG	-0.0408562352	0.0070241387	0.0	0.0655464986
HUNAN JINJIAN	-0.0141868353	0.0120880280	0.0	0.0852357784
INNER MONGOLIA BAOTOU	-0.0058717151	0.0080057201	0.0	0.1035080615
INNER MONGOLIA JINYU	-0.0406094574	0.0114607427	0.0	0.0723986029
INNER MONGOLIA MENGDIAN	0.0342213968	0.0350391262	1.6	0.0220865788
INNER MONGOLIA YILI	0.1552183013	0.0282235977	1.6	0.0819090349
INSIGMA TECHN	-0.0255301946	0.0184227216	1.6	0.0816578193
IRICO DISPLAY	-0.0275539129	0.0320345576	1.0	0.0352782312
JIANGSU CHENGXING	0.0297930896	0.0131994922	1.6	0.0278735110
JIANGSU CHUNLAN	0.0584138681	0.0145960665	0.0	0.0199414919
JIANGSU HOLLY	0.0349142714	0.0162479721	0.0	0.0499448649
JIANGSU HONGTU HIGH TECH	0.0233361845	0.0148762964	0.0	0.0663798050
JIANGSU SOPO	-0.0357265559	0.0065376922	0.0	0.0502949097
JIANGSU SUNSHINE	-0.0083676907	0.0158714164	0.0	0.0679022743
JIANGSU WUZHONG	-0.0002431861	0.0162185628	3.2	-0.0538685129
JIANGSU YONGDING	0.0080992410	0.0069477736	0.0	0.0241103769
JIANGSU YUEDA	0.0510918002	0.0271815082	1.0	0.0397515068
JIANGSU ZONGYI	-0.0026853062	0.0248981699	2.0	0.0251923993
JIANGXI CHANGJIU	-0.2925427160	0.0508333179	0.6	0.0356895443
JIANGXI JIANGZHONG	-0.0045897384	0.0096618247	0.0	0.0956128491
JILIN FOREST	0.0465850165	0.0105467944	1.2	0.0732699720
JILIN YATAI	0.0415542581	0.0131512426	1.0	0.1425837192
JINGNENG PROPERTY	0.0117876761	0.0091945667	0.0	0.0475957720
JINZHOU PORT	0.0247396411	0.0225730304	0.8	0.0436237994
JONJEE HIGH & NEW TECH	-0.0183393166	0.0067786482	0.0	0.0784898116
LANZHOU GREAT WALL	0.0551748074	0.0128777372	0.0	0.0473437998
LANZHOU MINBAI	-0.0075368105	0.0093190431	0.0	0.0601229291
LAWTON DEVELOPMENT	0.0251437494	0.0212684043	0.0	0.0319454624
LESHAN ELECTRIC	0.0378141463	0.0149879628	0.0	0.0376585159
LIAONIN GUONENG	0.0076114454	0.0120384852	0.0	0.0685039333
LIAONING CHENG	-0.0156693482	0.0250562424	1.6	0.1713054906
LIAONING HONGYANG	0.0179526834	0.0174460169	0.8	0.0364092420

Table 13: Data for the year 2006.

Company	Predicted return	Predicted risk	Reputation	Exact return
LIAONING TIMES GARMENTS	-0.0364870793	0.0149570280	0.0	0.0279810901
LINHAI	0.0219514204	0.0336053131	0.6	0.0450958313
LONG MARCH	0.1230401389	0.0215758000	2.0	0.0751148327
LUCKYFILM	-0.0310022778	0.0257777918	0.6	0.0265947568
LUOYANG GLASS	-0.0437386086	0.0106973489	0.0	0.0546268646
MAANSHAN IRON & STEEL	0.0325836432	0.0248646393	0.8	0.0733228607
MEIDU HOLDING	0.0557437035	0.0210013162	0.8	0.0395254100
MIANYANG GAOXIN	0.0169954687	0.0192662775	0.0	0.0337714315
MINMETALS DEVELOPMENT	0.0649351302	0.0129598870	0.6	0.0470062572
NANJING CHEMICAL	-0.0432463374	0.0179230004	1.0	0.0436786946
NANJING MEDICAL	0.0242201081	0.0089028141	0.0	0.0853767878
NANJING PANDA ELECTRONICS	0.0761295218	0.0185628586	0.0	0.0332092289
NANJING XINGANG HIGH-TECH	0.0305108520	0.0104031507	0.0	0.0939184283
NANJING XINJIEKOU	0.1090435782	0.0208093061	2.0	0.0378704237
NANNING	-0.0527622325	0.0093783571	0.0	0.0496257699
NANZHI	-0.0085095482	0.0034729349	0.0	0.0353893573
NEUSOFT	0.0267570003	0.0202626681	1.0	0.1200267419
NING XIA HENG LI	0.0073220830	0.0357578186	0.6	0.0257559794
NINGBO FUBANG JINGYEV	-0.0890896744	0.0303285824	1.6	0.0113547137
NINGBO FUDA	0.0979139544	0.0215147434	1.6	-0.0105831441
NINGBO MARINE	0.0080219539	0.0093209038	0.0	0.0293589840
NINGBO SHANSHAN	0.0031264144	0.0141205246	1.0	0.0431543836
NINGBO UNITED	0.0236889017	0.0125367055	0.0	0.0757239819
NINGBO VEKEN ELITE	-0.0109544886	0.0055793327	0.0	0.0347507701
NORTH CHINA	-0.0643315688	0.0197093884	1.0	0.0478492937
NORTHEAST EXPRESSWAY	-0.0002962240	0.0109725053	1.0	0.0326030099
ORIENT INCORPORATION	0.0645523532	0.0197073594	1.0	0.0939581479
PHENIX OPTICAL	-0.0417701500	0.0175242506	0.6	0.0716242131
PHOENIX	-0.0545176018	0.0514077491	1.6	0.0795393596
QINGDAO HAIER	0.0449345389	0.0154835564	0.0	0.0804185620
QINGHAI JINRUI MIMERAL	-0.0660344170	0.0418219495	1.6	0.0220962154
S & P PHARMACEUTICAL	0.0114514791	0.0197369078	0.8	0.0346482859
SAIC MOTOR CORPORATION	0.1020608481	0.0447946461	0.6	0.0828282849
SGSB	-0.0660715221	0.0096491350	0.0	0.0480175477
SH FRIENDSHIP	-0.0201363820	0.0065547880	0.8	0.0337652366
SH JINJIANG HOTELS	0.0203216655	0.0208994995	1.6	0.0576637739
SH JINQIAO EXPORT	0.0018653256	0.0157660211	1.0	0.0835811827
SH POWER TRANSMISSION	0.0729915977	0.0231168664	1.0	0.0518243869
SHAANXI BROADCAST	0.0413001598	0.0401392090	2.0	0.0895759462
SHAANXI QINLING	-0.1824245604	0.0484236260	2.4	0.0412011750
SHAN DONG HEUNGKONG	-0.0173495243	0.0217667418	0.8	0.0528399578
SHANDONG DACHENG	-0.0227963731	0.0051475867	0.0	0.0485830048
SHANDONG JIANGQUAN	0.0214851406	0.0117855869	0.0	0.0506724682
SHANDONG JIUFA EDIBLE	0.0320346158	0.0262514193	1.2	0.0262240929
SHANDONG LANGCHAO	0.0418026260	0.0183461859	0.8	-0.0011356866
SHANDONG LUBEI	0.0296139770	0.0101314494	0.0	0.0374523301
SHANDONG LUKANG	0.0247951991	0.0171319428	0.0	0.0589103027
SHANDONG NANSHAN	0.0498902356	0.0133809359	0.8	0.0983893165
SHANDONG TYAN HOME	-0.0556152180	0.0124953684	0.0	0.0715163137
SHANDONG LUXIN HIGH-TECH	0.0572640205	0.0233561464	1.0	0.0250177906
SHANG HAI YA TONG	-0.0504962099	0.0140865638	0.8	0.0299822935
SHANGHAI 3F NEW MATERIALS	0.0047079029	0.0203130498	3.0	0.0249021129

Table 13: Data for the year 2006.

Company	Predicted return	Predicted risk	Reputation	Exact return
SHANGHAI AEROSPACE	0.0866914852	0.0228495995	0.6	0.0896082599
SHANGHAI AJ	-0.0591634831	0.0220949269	1.0	0.0773996364
SHANGHAI BAOSIGHT	-0.0109783066	0.0131186169	0.8	0.0302362163
SHANGHAI BELLING	-0.0557869950	0.0124179683	0.0	0.0170589642
SHANGHAI CHENGTOU	0.0295056472	0.0137834059	1.0	0.0245462900
SHANGHAI CHLOR-ALKALI	-0.0589797537	0.0111647711	0.0	0.0192226779
SHANGHAI CONSTRUCTION	0.0020593916	0.0087213626	0.0	0.0327099709
SHANGHAI DAZHONG	0.0076420016	0.0088988788	0.0	0.0506382799
SHANGHAI DINGLI TECH	-0.0048997300	0.0146492105	0.0	0.0198070606
SHANGHAI DRAGON	-0.0050014382	0.0046132082	0.0	0.0394098430
SHANGHAI DUOLUN	-0.0782616691	0.0141316566	0.0	0.0436570330
SHANGHAI EAST-CHINA COMPUTER	-0.0237294234	0.0132633906	0.0	0.0161121520
SHANGHAI ELECTRIC	-0.0341661957	0.0152766924	0.8	0.0891967631
SHANGHAI ERFANGJI	-0.0081071838	0.0065911633	0.0	0.0168636800
SHANGHAI FEILO ACOUSTICS	0.0210465215	0.0134621746	1.0	0.0119019864
SHANGHAI FEILO	0.0283881255	0.0101481141	0.0	0.0076005462
SHANGHAI FIRST PROVISIONS	0.0872413492	0.0155272077	4.0	0.0533537769
SHANGHAI FOSUN	0.0798192895	0.0174912374	1.8	0.0360419128
SHANGHAI FUDAN	-0.0391489004	0.0159948698	1.0	0.0190419798
SHANGHAI HAIBO	0.0106876514	0.0071214971	0.8	0.0186603472
SHANGHAI HAINIAO	-0.0150919265	0.0250596256	1.0	0.0163651771
SHANGHAI HAIXIN	-0.0971917580	0.0261588880	0.8	0.0362404654
SHANGHAI HIGHLY	0.0258064895	0.0244667994	2.0	-0.0242890786
SHANGHAI HONGSHENG TECH	0.0787739754	0.0250033928	1.0	0.0144676705
SHANGHAI HUITONG ENERGY	0.0129580096	0.0097563606	0.0	0.0690967650
SHANGHAI PHARMACEUTICAL	0.0392991728	0.0094426224	1.0	0.0448331380
SHANGHAI AIRPORT	0.0496049814	0.0212877647	1.8	0.0437159563
SHANGHAI JIABAO	0.0612685409	0.0129114078	0.0	0.0473232266
SHANGHAI JIAO DA NAN YANG	0.0602669469	0.0219942672	1.0	0.0129598395
SHANGHAI JIAO YUN	0.0172369485	0.0123783493	0.8	0.0440234383
SHANGHAI JIELONG	-0.1603689951	0.0353647323	0.8	0.0657260194
SHANGHAI JIN JIANG	-0.0122989505	0.0194013838	1.6	0.0285981941
SHANGHAI JINFENG	0.0426008048	0.0125778377	0.0	0.1013553995
SHANGHAI JINLING	-0.0377999500	0.0218100623	1.0	0.0201763880
SHANGHAI JOIN BUY	-0.0549165014	0.0125264419	0.0	0.0485727709
SHANGHAI LANSHENG	-0.0248771116	0.0204280991	0.8	0.1151561391
SHANGHAI LUJIAZUI	-0.0365153422	0.0115019002	0.0	0.0975192221
SHANGHAI MALING AQUARIUS	-0.0289890257	0.0174251502	0.6	0.0154985716
SHANGHAI MATERIAL TRADING	-0.0196952579	0.0190927679	1.6	0.0301507538
SHANGHAI MET	-0.1140493493	0.0422260623	2.0	0.0341607925
SHANGHAI NEW HUANG	-0.0091457832	0.0073684764	0.0	0.0505879671
SHANGHAI NEW WORLD	0.0708891663	0.0192590699	2.0	0.0774834571
SHANGHAI ORIENTAL PEARL	0.1305405571	0.0220843161	3.0	0.0540524454
SHANGHAI PHARMACEUTICAL	-0.0366287948	0.0097441011	0.0	0.0493241479
SHANGHAI POTEVIO	0.0329699187	0.0097783232	0.0	0.0470596546
SHANGHAI PUDONG	0.0950525938	0.0246082248	2.0	0.0934241945
SHANGHAI QIANGSHENG	0.0727442182	0.0155038455	2.0	0.0215291461
SHANGHAI SANMAO	-0.0150887626	0.0121794952	0.0	0.0355735977
SHANGHAI SHENDA	0.0058601560	0.0118991584	1.0	-0.1072834451
SHANGHAI SHENHUA	-0.0581844371	0.0197273855	1.0	0.0597521593
SHANGHAI SHENTONG	0.0950190606	0.0266508486	1.0	0.0056928014
SHANGHAI SHIMAO	-0.0773926957	0.0216022420	0.8	0.0624269796

Table 13: Data for the year 2006.

Company	Predicted return	Predicted risk	Reputation	Exact return
SHANGHAI TIANCHEN	-0.0708216672	0.0224048976	1.0	0.0589403440
SHANGHAI TONGDA	-0.0319173016	0.0203424160	1.0	0.0227346835
SHANGHAI TONGJI	-0.0131292431	0.0084359441	0.0	0.0142962297
SHANGHAI TUNNEL	-0.0105834208	0.0182357755	0.8	0.0495605350
SHANGHAI WANYE	0.0577323115	0.0185122543	1.0	0.0534647520
SHANGHAI WHITECAT	-0.1076064280	0.0257027351	2.0	0.0406619324
SHANGHAI WORLDBEST	-0.0109806491	0.0038436129	0.0	0.0469935885
SHANGHAI XINGYE	0.0263747400	0.0109722105	0.0	0.0096714005
SHANGHAI YAOHUA	-0.0626044473	0.0199479405	1.6	0.0344114249
SHANGHAI YIMIN	0.0367163423	0.0251428556	0.8	0.0450442134
SHANGHAI YUYUAN	0.0145495687	0.0132906492	1.0	0.0759007316
SHANGHAI ZHANGJIANG	-0.0941419317	0.0204742837	1.6	0.0694955321
SHANGHAI ZI JIANG	-0.0431602506	0.0085834720	0.6	0.0181346662
SHANXI COKING	0.0484943745	0.0426204217	0.8	0.0462482026
SHANXI LANHUA SCI-TECH	0.1157419852	0.0230683508	1.8	0.0733502195
SHANXI TOP ENERGY	0.0348536725	0.0350846023	2.0	0.0201344959
SHANXI XINGHUACUN	0.1184042050	0.0196670529	1.6	0.1472068408
SHEN MA INDUSTRY	0.0232619910	0.0091342147	0.0	0.0381608153
SHENERGY	0.1386494544	0.0283855367	1.6	0.0273779526
SHENJI KUNMING	0.0054462283	0.0078315203	0.0	0.0661136547
SHENYANG JINBEI	-0.0555022662	0.0294013357	1.0	0.0638986619
SICHUAN CHANGHONG	0.0622403458	0.0247995340	2.0	0.0338996021
SICHUAN CHUANTOU	-0.0387070379	0.0125688843	0.0	0.0396970200
SICHUAN GOLDEN SUMMIT	-0.0632850873	0.0086520847	0.0	0.0361418285
SICHUAN HEJIA	-0.1418943786	0.0190302748	0.6	0.0326539765
SICHUAN MINGXING	0.0021706978	0.0213893822	1.0	0.0345134618
SICHUAN MINJIANG	0.0109784567	0.0123640898	0.8	0.0320974330
SICHUAN SWELLFUN	-0.0026811893	0.0337692929	0.8	0.1259718008
SILVERTIE HOLDING	-0.0759133386	0.0155364492	1.0	0.0220257228
SINO CONSTRUCTION	-0.1083864362	0.0150916924	0.0	0.0107906158
SINOPEC SHANGHAI	0.1069573097	0.0306755759	0.8	0.0334979934
SINOPEC YIZHENG	0.0090325945	0.0212765167	0.8	0.0310843870
SINOTEX INVESTMENT	-0.0182679854	0.0071117349	0.0	0.0519906714
SONGLIAO AUTOMOTIVE	0.1157673205	0.0447517552	0.8	0.0325824785
SOUTHWEST	0.0014743899	0.0060334184	0.0	0.0361871942
STAR LAKE BIOSCIENCE	-0.0917903826	0.0194421584	1.0	0.0184872046
SUNNY LOAN TOP	-0.0537725650	0.0130708966	1.0	0.0433947609
SUNTEK TECHNOLOGY	-0.0764201300	0.0217445247	0.0	0.0070122156
SUZHOU NEW DISTRICT HI-TECH	0.0573781677	0.0147541011	0.8	0.0352177920
SVA ELECTRON	-0.0373559478	0.0302265888	1.0	0.0103094518
SVA INFORMATION	-0.0305441729	0.0137355607	0.0	0.0144984214
TAIYUAN HEAVY INDUSTRY	-0.0456675326	0.0349180938	0.6	0.0848245753
TIANJIN ENVIRONMENTAL	-0.0170808652	0.0173575825	0.8	0.0311995587
TIANJIN HI-TECH	-0.0217887135	0.0071352914	0.0	0.0902519622
TIANJIN PORT	0.0521158357	0.0201756916	4.0	0.0514240978
TIANJIN QUANYE BAZAAR	0.0041247273	0.0041110785	0.0	0.0709443747
TIBET RHODIOLA	0.0727289597	0.0250490323	0.6	0.0750725628
TIBET TOURISM	-0.0047232991	0.0326048840	1.6	0.0347944638
TONGHUA DONGBAO	0.1876908094	0.0504174832	0.0	0.1156260931
TOPSUN SCIENCE AND TECH	0.0249237509	0.0088953227	0.0	0.0423906582
TSINGHUA TONGFANG	-0.0026385401	0.0167631434	0.6	0.0387494325
TSINGTAO BREWERY	0.0287527098	0.0162931058	2.4	0.0605139880

Table 13: Data for the year 2006.

Company	Predicted return	Predicted risk	Reputation	Exact return
TUOPAI YEAST LIQUOR	0.0938866248	0.0187685128	0.0	0.0623320591
WEIFANG BEIDA	-0.0037632864	0.0171095903	0.0	0.0223428617
WINOWNER	-0.0201152770	0.0294619633	0.0	0.0186295909
WINSAN	-0.0893495339	0.0139879740	0.0	0.0713456125
WUHAN EAST LAKE HIGH TECH	-0.0230628822	0.0144954384	0.8	0.0293320918
WUHAN HANSHANG	-0.0488185071	0.0261206355	1.0	0.0458076683
WUHAN HUMANWELL HI-TECH	0.0487528337	0.0166111988	0.0	0.0471279174
WUHAN LINUO SOLAR ENERGY	0.0566349400	0.0287361523	1.0	0.0005549734
WUHAN SANZHEN	0.0097803106	0.0083807937	0.0	0.0315190726
WUHAN STEEL	0.0690725412	0.0221780543	1.2	0.0830705229
WUHAN XIANGLONG	-0.0136920915	0.0121209614	0.0	0.0509872546
WUXI TAIJI	-0.0572312742	0.0098941266	0.0	0.0599094197
XIAMEN C & D	0.0199707350	0.0167771849	0.6	0.0733766890
XIAMEN ENGINEERING	-0.1139003004	0.0242597380	0.8	0.0856037661
SHANGHAI FEILO	0.0386743314	0.0146975401	0.8	0.0293536855
XIAMEN INTERNATIONAL TRADE	0.0238122489	0.0150496252	1.6	0.0414435857
XIAMEN KING LONG MOTOR	0.0461336830	0.0122397521	1.0	0.1056456627
XIAMEN ELECTRONICS	0.0156037609	0.0067605171	0.0	0.0332078905
XIAMEN PROSOLAR	-0.0415699858	0.0092912331	0.0	0.0309683179
XI'AN SEASTAR	0.0060456608	0.0073156812	0.0	0.0308871632
XINING SPECIAL STEEL	0.0364811456	0.0206188662	0.6	0.0594988593
XINJIANG FRIENDSHIP	-0.0478488360	0.0098449798	0.0	0.0299459802
XINJIANG JOINWORLD	-0.0069761412	0.0074774750	0.0	0.1335925634
XINJIANG TALIMU	0.0697786100	0.0208880839	0.6	0.0310232415
XINJIANG TIANYE	0.0818644804	0.0232493028	1.2	0.0686750012
XINJIANG YILITE	0.0295837722	0.0071830276	0.0	0.0742774654
XINYU IRON & STEEL	-0.1299432496	0.0289134190	1.6	0.0386695610
Y.U.D. YANGTZE RIVER	-0.1335958022	0.0226138256	1.2	0.0244699995
YANTAI HUALIAN	0.0559631158	0.0380671062	1.0	0.0473810967
YANTAI XINCHAO	0.0138759347	0.0106021639	0.0	0.0589173470
YANZHOU COAL MINING	0.0355035447	0.0195513571	1.2	0.0367273419
YIBIN PAPER	-0.0310671597	0.0153789751	1.0	0.0374566005
YOUNGOR	0.0472460256	0.0105128967	1.6	0.0857188652
YUNNAN BOWIN TECH	0.0062459197	0.0069525418	0.0	0.0682643850
YUNNAN FREETRADE	-0.1507609822	0.0257555805	0.8	0.0622830729
YUNNAN MALONG	-0.0890798597	0.0177638519	0.0	0.0135328741
YUNNAN YUNTIANHUA	0.0513347835	0.0208527321	1.2	0.0709974245
YUNNAN YUNWEI	-0.0300766863	0.0121317072	0.8	0.0628534226
ZHE JIANG DONG RI	-0.0182989784	0.0121303505	0.0	0.0370270898
ZHEJIANG CHINA LIGHT	-0.0418709467	0.0135708006	0.8	0.0849419616
ZHEJIANG FURUN	-0.0137951640	0.0114708965	0.0	0.0384207223
ZHEJIANG GUYUELONGSHAN	-0.0497166465	0.0230124600	1.2	0.1077823882
ZHEJIANG HOLLEY TECH	-0.0757946250	0.0146723361	0.6	0.0269717933
ZHEJIANG HSD INDUSTRIAL	-0.0198700151	0.0120326780	0.0	0.0278895306
ZHEJIANG JUHUA	0.1034726200	0.0185821193	0.6	0.0279656090
ZHEJIANG MEDICINE	0.0385424760	0.0302788210	1.2	0.0193571551
ZHEJIANG ORIENT	-0.1189474569	0.0181246203	0.0	0.0403069341
ZHEJIANG QIANJIANG	0.0458345434	0.0241320346	1.6	-0.0076257635
ZHEJIANG SHENGHUA BIOK	-0.0300877568	0.0254343216	0.6	0.0229352035
ZHEJIANG XINHU VENTURE	-0.0144444240	0.0169263448	1.0	0.1089667349
ZHENGZHOU COAL & ELECTRIC	0.0578134616	0.0219822268	0.0	0.0289822035
ZHENGZHOU YUTONG BUS	0.0460616154	0.0092009629	2.4	0.1056121710

Table 13: Data for the year 2006.

Company	Predicted return	Predicted risk	Reputation	Exact return
ZHONGCHU DEVELOPMENT STOCK	0.0607750786	0.0211744132	0.6	0.0714561084
ZHONGLU	0.0446341223	0.0369160678	1.6	0.0110147881

Table 14: Data for the year 2007.

Company	Predicted return	Predicted risk	Reputation	Exact return
AEROSPACE COMMUNICATIONS	0.0181777162	0.0503826489	1.0	0.1183612174
ALONG TIBET	-0.0035849493	0.0470220084	1.6	0.1108724807
AMOI ELECTRONICS	0.0064782773	0.0510809004	2.4	0.0405337018
ANHUI GOLDEN SEED WINERY	0.0100671262	0.0545696245	0.6	0.0780945118
ANHUI HELI	0.0469605313	0.0568011599	4.0	0.0450441308
ANHUI QUANCHAI ENGINE	-0.0043188076	0.0347837096	1.2	0.0844040360
ANHUI WANWEI	0.0112518087	0.0407840384	0.8	0.1164980026
BAIDA	0.0143338906	0.0321734202	2.0	0.0476497194
BAOTOU HUAZI	0.0082010954	0.0343980184	0.6	0.0853467643
BAOTOU TOMORROW TECH	-0.0053791183	0.0220805833	0.8	0.0574684667
BEIJING C & W TECH	-0.0132686845	0.0386554918	2.0	0.0937401029
BEIJING DOUBLE-CRANE	0.0100488890	0.0721803347	3.2	0.1317911980
BEIJING TRADE	0.0090012706	0.0325877423	1.0	0.0743545837
BEIJING TIANTAN	0.0386843961	0.0714374872	1.2	0.0724669083
BEIJING TONGRENTANG	0.0099366824	0.0156011013	3.2	0.0612087375
BEIJING CONSTRUCTION	0.0231016036	0.0515956557	1.2	0.0804329925
BEIJING WANDONG	-0.0031430490	0.0283047129	0.8	0.0997887112
BEIJING WANGFUJING	0.0445514325	0.0666552276	3.0	0.0635563087
BEIJING XIDAN	0.0097057470	0.0390287394	1.0	0.1195107812
BEIQI FOTON MOTOR	0.0218518108	0.0637375190	2.4	0.1180053707
BEIREN PRINTING	-0.0027710410	0.0237968394	1.0	0.0875390649
CHANG CHUN EURASIA	0.0201560885	0.0408657197	2.0	0.0774716914
CHANGCHUN ECONOMIC	0.0005566012	0.0373744704	1.0	0.0619071913
CHANGCHUN AUTOMOBIL	-0.0032579884	0.0372206930	2.0	0.1015996509
CHANGCHUN YIDONG	0.0001233423	0.0320052661	0.6	0.0976458751
CHANGLIN	0.0160432102	0.0594125354	2.4	0.0840464176
CHENGDU	0.0199118044	0.0275353033	3.0	0.0664687993
CHENGDU DR. PENG TELECOM	0.0351888157	0.0847151815	1.0	0.1462689930
CHENGDU QIANFENG	0.0019429526	0.0418375091	2.0	0.1660367973
CHENGSHANG	0.0191605940	0.0500302319	1.0	0.0755217804
CHINA ANIMAL	0.0296655420	0.0624602988	1.2	0.0891590090
CHINA CYTS TOURS	0.0232812040	0.0518162102	1.6	0.0835363435
CHINA DONGFANGHONG	0.0240096840	0.0546457475	0.8	0.0684325848
CHINA ENTERPRISE	0.0247384405	0.0451245538	2.0	0.0826202109
CHINA FIBERGLASS	0.0222049939	0.0447538247	1.8	0.1206377983
CHINA FIRST PENCIL	0.0133903439	0.0369571730	1.0	0.0829510991
CHINA GEZHOUBA	0.0124726508	0.0401533932	1.6	0.1339621793
CHINA HI-TECH	-0.0028991148	0.0209339607	1.0	0.0994464312
CHINA JIALING	0.0052153011	0.0332653013	1.0	0.1222417037
CHINA RAILWAY	0.0245604876	0.0311343631	3.0	0.0624593534
CHINA SATCOM	0.0060168224	0.0448665770	1.0	0.0608627932
CHINA SHIPPING	0.0083239760	0.0292882476	1.0	0.1358244797
CHINA SPORTS	0.0158391280	0.0417320381	0.6	0.1952332090
CHINA SHIPBUILDING	0.0444503767	0.0581346664	1.8	0.1875877647
CHINA TELEVISION	0.0156937975	0.0405312421	2.4	0.0755359596
CHINA WORLD TRADE	0.0219649492	0.0446510209	0.8	0.0633632317
CHINA-KINWA HIGH TECH	0.0174178967	0.0386746138	1.6	0.1270835254
CHONGQING BREWERY	0.0332303534	0.0499316759	3.0	0.0385689027
CHONGQING DEPARTMENT STORE	0.0210995704	0.0481441201	2.0	0.0761110242
CHONGQING ROAD	-0.0001648892	0.0224387458	1.6	0.1125122884
CHONGQING TAIJI	-0.0102494791	0.0514863930	2.0	0.1048338162

Table 14: Data for the year 2007.

Company	Predicted return	Predicted risk	Reputation	Exact return
CHONGQING WATER AND ELECTRIC	-0.0054161372	0.0242730400	1.0	0.1043398252
CHONGQING WANLI	-0.0050347139	0.0485055461	1.0	0.0798373777
CHONGQING SWELL	0.0009237269	0.0336894806	0.6	0.0935917288
CITYCHAMP DARTONG	0.0215232422	0.0275730637	3.0	0.0974684955
CNTIC TRADING	0.0107242731	0.0283479869	1.6	0.1095168396
COFCO XINJIANG TUNHE	0.0147241308	0.0961432309	4.0	0.0808328523
CRED HOLDING	-0.0072449604	0.0802900937	3.0	0.1010046781
CSC NANJING	0.0131153960	0.0363442666	3.0	0.1301694535
CSSC JIANGNAN	0.0164781370	0.0469711477	0.8	0.1806501072
DALIAN DAXIAN	0.0144538693	0.0394524801	3.0	0.1310299579
DALIAN THERMAL POWER	-0.0037396954	0.0258182958	1.0	0.0715939605
DASHANG	0.0335600142	0.0347344107	3.0	0.0438556891
DATANG HUAYIN ELECTRIC	0.0000676505	0.0344024939	2.0	0.0880936424
DATANG TELECOM TECH	0.0108042458	0.0439011213	1.2	0.0330721904
DAZHONG TRANSPORTATION	0.0160275354	0.0347628967	1.0	0.1184410029
DONG FANG BOILER	0.0405382330	0.0291367264	5.0	0.1142794894
DONGAN HEIBAO	-0.0084727693	0.0325643113	0.8	0.0955668000
DONGFANG ELECTRIC	0.0377535002	0.0425460750	4.0	0.1044324277
DONGFENG AUTOMOBILE	0.0044583958	0.0351003023	2.0	0.0876356440
DONGFENG ELECTRONIC	-0.0146305983	0.0390691130	0.8	0.0980248740
DOUBLE COIN HOLDINGS	0.0006419096	0.0195800491	2.0	0.0552211366
EASTERN COMMUNICATIONS	0.0212916274	0.0761090541	0.8	0.0415702581
FOUNDER TECH	0.0008233599	0.0308876443	3.0	0.0952657956
FUJIAN CEMENT	0.0069979801	0.0412460312	2.0	0.1370546763
FUJIAN DONGBAI	0.0256386873	0.0584651847	1.0	0.0697387264
FUJIAN QINGSHAN	0.0026100333	0.0297979633	1.0	0.1050658316
FUYAO GLASS	0.0320354237	0.0543987672	2.0	0.0776554716
GANSU QILIANSHAN	0.0069445753	0.0407923429	0.8	0.1137964400
GANSU TRISTAR	0.0060071054	0.0231719435	2.0	0.0714607460
GANSU YASHENG	0.0151589824	0.0469582481	1.6	0.1195148537
GD POWER DEVELOPMENT	0.0088133061	0.0249056939	3.0	0.0911647854
GINWA ENTERPRISE	-0.0012102425	0.0340100156	0.8	0.0833772934
GRINM SEMICONDUCTORS	0.0190660766	0.0613195424	0.6	0.0680667125
GUANGDONG MEIYAN	0.0121787590	0.0399030524	2.0	0.0868017045
GUANGDONG SHENGYI	0.0265037920	0.0342417991	3.0	0.0571098036
GUANGZHOU DEVELOPMENT	0.0033156779	0.0227754129	1.6	0.0745222854
GUANGZHOU IRON AND STEEL	0.0021533469	0.0327218590	2.0	0.0924458700
GUANGZHOU PEARL RIVER	0.0067504529	0.0359143845	1.0	0.0984208694
GUANGZHOU SHIPYARD	0.0489794336	0.0773376720	2.0	0.1360063367
GUIZHOU CHANGZHENG	-0.0094813149	0.0158864535	0.8	0.1443575599
GUIZHOU LIYUAN	0.0201078906	0.0531115750	0.8	0.1235643544
GUODIAN NANJING	0.0152193566	0.0361086111	1.2	0.0595454025
HABIN GONG	-0.0048877164	0.0261401147	1.0	0.0943654937
HAINAN AIRLINES	0.0064551770	0.0364255513	1.0	0.1120649021
HAINAN YEDAO	0.0007349757	0.0378945732	1.0	0.1215621553
HANDAN IRON & STEEL	0.0119388310	0.0223663100	1.8	0.0543491719
HANG ZHOU IRON & STEEL	0.0026255297	0.0308878274	1.2	0.0950357802
HANGZHOU JIEBAI	0.0151281250	0.0374130409	2.0	0.0596477544
HANGZHOU TIAN-MU-SHAN	0.0035132687	0.0334944990	1.0	0.0634717804
HARBIN AIR CONDITIONING	0.0024844524	0.0243487671	3.0	0.1076665267
HARBIN DONGAN	0.0006872562	0.0243419466	1.2	0.0882861057
BAIDA	0.0221065203	0.0455488920	1.6	0.1505060158

Table 14: Data for the year 2007.

Company	Predicted return	Predicted risk	Reputation	Exact return
HARBIN HIGH-TECH	0.0010863105	0.0352848403	1.0	0.0819770396
HARBIN PHARMACEUTICAL	0.0293529982	0.0378745724	2.4	0.0681099214
HARBIN PHARMACEUTICAL (group)	0.0079618472	0.0397230418	2.0	0.0796095821
HEBEI WEIYUAN	0.0103813418	0.0407806072	2.0	0.0681733445
HEILNGJIANG HEIHUA	0.0048984875	0.0374218660	1.2	0.0819214861
HENAN ANCAI HI-TECH	-0.0200655696	0.0358036309	1.2	0.1015432675
HENAN HUANGHE	-0.0023350213	0.0283621470	1.2	0.0526097008
HENAN PHARMACEUTICAL	0.0022216585	0.0272570067	1.2	0.0391433114
HENAN ORIENTAL SILVER STAR	-0.0018711946	0.0306572759	1.6	0.0954177648
HENAN YINGE	0.0057441906	0.0170055692	2.0	0.1255723940
HISENSE ELECTRIC	0.0019974371	0.0167429951	1.6	0.0631222919
HIT SHOUCHUANG TECH	-0.0110869862	0.0278285766	1.0	0.1039385902
HUADIAN ENERGY	-0.0031361296	0.0253116584	2.0	0.0721814798
HUALIAN SUPERMARKET	0.0448784566	0.0847024422	1.0	0.1283502795
HUANGSHAN TOURISM	0.0230479413	0.0413625907	2.4	0.0904667528
HUAXIN CEMENT	0.0136519327	0.0289362228	2.0	0.1235276618
HUBEI EASTERN GOLD	0.0258692457	0.0910345147	2.0	0.0433690529
HUBEI MAILYARD	0.0018599373	0.0349775312	1.0	0.1807122814
HUBEI XINGFA	0.0085103347	0.0393721131	0.8	0.1063499239
HUNAN HAILI	-0.0059721918	0.0208570055	0.8	0.0951010974
HUNAN HUASHENG	0.0019231242	0.0435830898	0.6	0.0915249615
HUNAN JINJIAN	0.0096929065	0.0525038698	0.6	0.0995197513
INNER MONGOLIA BAOTOU	0.0221078528	0.0554954205	0.8	0.1298674583
INNER MONGOLIA JINYU	0.0131122346	0.0413866788	1.0	0.1032233484
INNER MONGOLIA MENGDIAN	0.0059755209	0.0359214981	2.4	0.0955960641
INNER MONGOLIA YILI	0.0354713074	0.0401499247	3.0	0.0087442244
INSIGMA TECH	0.0180287633	0.0460552315	1.6	0.0522058580
IRICO DISPLAY	0.0005040026	0.0371807091	2.0	0.0715636717
JIANGSU CHENGXING	0.0062699443	0.0197699203	2.4	0.1015627108
JIANGSU CHUNLAN	-0.0083547108	0.0243752034	0.8	0.0599353408
JIANGSU HOLLY	0.0031393245	0.0323853563	0.8	0.1621194389
JIANGSU HONGTU HIGH TECH	0.0104045523	0.0392853496	0.6	0.1115580998
JIANGSU SOPO	-0.0042179915	0.0375022178	0.8	0.1200400060
JIANGSU SUNSHINE	0.0128969353	0.0392821294	0.6	0.0848309872
JIANGSU WUZHONG	-0.0146568477	0.0295697796	2.4	0.0586786783
JIANGSU YONGDING	-0.0042464072	0.0205331334	0.8	0.0757472122
JIANGSU YUEDA	0.0002169839	0.0382565532	2.0	0.0661906357
JIANGSU ZONGYI	0.0075318260	0.0182201959	3.0	0.1158748290
JIANGXI CHANGJIU	-0.0095302303	0.0575484010	1.2	0.1043112746
JIANGXI JIANGZHONG	0.0214832431	0.0512163489	0.8	0.0885787688
JILIN FOREST	0.0194441161	0.0381102079	1.8	0.0735696013
JILIN YATAI	0.0402460891	0.0707490628	2.0	0.0984726027
JINGNENG PROPERTY	0.0028545907	0.0317820488	1.0	0.1207958357
JINZHOU PORT	0.0033679871	0.0352416678	2.0	0.0578739677
JONJEE HIGH & NEW TECH	0.0129317580	0.0449567211	1.0	0.0756219843
LANZHOU GREAT WALL	0.0063739957	0.0292620864	0.6	0.0913206170
LANZHOU MINBAI	0.0036816401	0.0394627711	1.0	0.0940022008
LAWTON DEVELOPMENT	-0.0060025668	0.0328378418	1.0	0.0934264373
LESHAN ELECTRIC	-0.0018184817	0.0306628241	1.0	0.1366758798
LIAONIN GUONENG	0.0072136241	0.0433061004	1.0	0.0411473111
LIAONING CHENG	0.0486448932	0.0860456036	3.0	0.1436044909
LIAONING HONGYANG	-0.0027187065	0.0318067817	2.0	0.0911926210

Table 14: Data for the year 2007.

Company	Predicted return	Predicted risk	Reputation	Exact return
LIAONING TIMES	-0.0073534868	0.0281960459	1.0	0.1092429461
LINHAI	-0.0037186345	0.0457213944	1.6	0.0993358757
LONG MARCH	0.0286994623	0.0378356877	3.0	0.0502906015
LUCKYFILM	-0.0059059575	0.0335087685	1.2	0.0990381405
LUOYANG GLASS	0.0021410019	0.0367803706	0.8	0.0759331465
MAANSHAN IRON & STEEL	0.0254677636	0.0399522433	2.0	0.0708529756
MEIDU HOLDING	0.0005101066	0.0320561976	2.0	0.1226637949
MIANYANG GAOXIN	-0.0121941942	0.0349191858	1.0	0.1181237180
MINMETALS DEVELOPMENT	0.0096775032	0.0282650626	1.6	0.1626023110
NANJING CHEMICAL	0.0017632369	0.0335241020	2.0	0.1117011692
NANJING MEDICAL	0.0172952029	0.0469581454	0.8	0.0770637387
NANJING PANDA ELECTRONICS	-0.0024156165	0.0307788903	1.0	0.0595335769
NANJING XINGANG HIGH-TECH	0.0197488717	0.0507970411	1.0	0.1166329083
NANJING XINJIEKOU	0.0128974313	0.0264230388	3.0	0.0684874200
NANNING	0.0031695265	0.0326551148	1.0	0.0703185244
NANZHI	-0.0030301524	0.0262635786	0.6	0.0749475790
NEUSOFT	0.0293651978	0.0639720427	2.0	0.0583805248
NING XIA HENG LI	-0.0172037953	0.0371074162	0.6	0.0836074616
NINGBO FUBANG JINGYE	-0.0019583229	0.0307178378	3.0	0.0416136700
NINGBO FUDA	-0.0087909132	0.0211892216	2.0	0.0454099227
NINGBO MARINE	-0.0005722219	0.0224200064	0.8	0.1032278790
NINGBO SHANSHAN	0.0053862694	0.0292985880	2.0	0.0986183233
NINGBO UNITED	0.0142370708	0.0428754493	0.8	0.0409817409
NINGBO VEKEN ELITE	0.0000775394	0.0241167374	1.0	0.0899529330
NORTH CHINA	0.0042118900	0.0354003929	2.0	0.1378407493
NORTHEAST EXPRESSWAY	0.0019864385	0.0235064252	2.0	0.0658935455
ORIENT INCORPORATION	0.0262761107	0.0502066844	2.0	0.0898850563
PHENIX OPTICAL	0.0129542327	0.0427044701	1.6	0.0475880510
PHOENIX	0.0059965797	0.0675101691	2.0	0.0423829475
QINGDAO HAIER	0.0166752721	0.0459728228	1.0	0.0779172497
QINGHAI JINRUI MIMERAL	-0.0013949255	0.0441913585	2.4	0.0629968342
S & P PHARMACEUTICAL	-0.0036671181	0.0323719740	1.6	0.1079692316
SAIC MOTOR CORPORATION	0.0277789159	0.0584740845	1.6	0.1028844925
SGSB	0.0041744869	0.0310753055	1.0	0.0467177488
SH FRIENDSHIP	0.0074399339	0.0187313193	2.0	0.0929369790
SH JINJIANG HOTELS	0.0179854961	0.0329892098	3.0	0.0500300375
SH JINQIAO EXPORT	0.0231203457	0.0438242944	2.0	0.0558356024
SH POWER TRANSMISSION	0.0174606222	0.0315348855	2.0	0.1379752570
SHAANXI BROADCAST	0.0219731057	0.0584968747	2.0	0.0588236615
SHAANXI QINLING	-0.0056976456	0.0512779436	2.4	0.1091026900
SHAN DONG HEUNGKONG	0.0121846297	0.0332017002	1.6	0.1502768765
SHANDONG DACHENG	0.0015529593	0.0322647428	1.0	0.0970533667
SHANDONG JIANGQUAN	0.0006817219	0.0357767270	1.0	0.0991246433
SHANDONG JIUFA EDIBLE	-0.0073702595	0.0315586952	1.2	0.0575316621
SHANDONG LANGCHAO	-0.0102289054	0.0177311746	0.8	0.0402856449
SHANDONG LUBEI	-0.0034085440	0.0297120680	0.8	0.0577008439
SHANDONG LUKANG	0.0055299121	0.0401815904	1.0	0.0737366390
SHANDONG NANSHAN	0.0230550365	0.0530732614	2.0	0.0949095665
SHANDONG TYAN HOME	-0.0003727293	0.0498008482	1.0	0.0948150909
SHANDONG LUXIN HIGH-TECH	-0.0054789472	0.0312256384	2.0	0.1002884122
SHANG HAI YA TONG	0.0009473781	0.0240891603	2.0	0.1045158031
SHANGHAI 3F NEW MATERIALS	0.0113650110	0.0222912728	3.0	0.0212374836

Table 14: Data for the year 2007.

Company	Predicted return	Predicted risk	Reputation	Exact return
SHANGHAI AEROSPACE	0.0262980983	0.0480991143	1.2	0.0333831426
SHANGHAI AJ	0.0121011036	0.0493391258	2.0	0.1034641844
SHANGHAI BAOSIGHT	0.0052354020	0.0201283057	1.6	0.0939843925
SHANGHAI BELLING	-0.0056980931	0.0190406668	0.6	0.0408498633
SHANGHAI CHENGTOU	0.0043250865	0.0195058588	2.0	0.1118922554
SHANGHAI CHLOR-ALKALI	-0.0071711068	0.0196530236	1.0	0.0461081689
SHANGHAI CONSTRUCTION	0.0042262429	0.0211737793	0.6	0.0956150797
SHANGHAI DAZHONG	0.0057981419	0.0312528414	1.0	0.1700631878
SHANGHAI DINGLI TECH	-0.0073488091	0.0215190017	1.0	0.1263848836
SHANGHAI DRAGON	-0.0032378470	0.0293667077	1.0	0.0688366522
SHANGHAI DUOLUN	-0.0012718812	0.0328971265	1.0	0.0909998468
SHANGHAI EAST-CHINA COMPUTER	-0.0108554484	0.0225945198	0.8	0.0932107765
SHANGHAI ELECTRIC	0.0207649410	0.0486677633	2.0	0.1009583792
SHANGHAI ERFANGJI	-0.0078315450	0.0177774838	1.0	0.0877364106
SHANGHAI FEILO ACOUSTICS	-0.0048227848	0.0175197896	2.0	0.0669907645
SHANGHAI FEILO	-0.0067753684	0.0117505398	1.0	0.0845176997
SHANGHAI FIRST PROVISIONS	0.0293553831	0.0186979200	5.0	0.0583469384
SHANGHAI FOSUN	0.0109949849	0.0195904899	2.4	0.1000437997
SHANGHAI FUDAN	-0.0066372285	0.0229500093	2.0	0.0720340889
SHANGHAI HAIBO	0.0017263471	0.0131469942	2.0	0.0886800969
SHANGHAI HAINIAO	-0.0104043220	0.0307280114	2.0	0.0861641805
SHANGHAI HAIXIN	0.0026400399	0.0255535107	1.0	0.1124805071
SHANGHAI HIGHLY	-0.0112020970	0.0259789910	2.0	0.0699018601
SHANGHAI HONGSHENG TECH	0.0033779775	0.0170959297	2.0	0.0433533301
SHANGHAI HUITONG	0.0107460285	0.0408807261	1.0	0.0655978615
SHANGHAI PHARMACEUTICAL	0.0096819002	0.0251975550	2.0	0.0867431019
SHANGHAI AIRPORT	0.0267970154	0.0205131432	2.4	0.0590776024
SHANGHAI JIABAO	0.0040954821	0.0319213302	1.0	0.0840195698
SHANGHAI JIAO DA NAN	-0.0026054639	0.0198487552	2.0	0.0390793742
SHANGHAI JIAO YUN	0.0088814226	0.0261795500	2.0	0.0895670273
SHANGHAI JIELONG	0.0102067785	0.0510754616	1.6	0.0530355802
SHANGHAI JIN JIANG	0.0067815078	0.0239263625	3.0	0.0629406185
SHANGHAI JINFENG	0.0227837487	0.0541126251	1.0	0.0529669070
SHANGHAI JINLING	-0.0003772836	0.0193610969	2.0	0.0952309112
SHANGHAI JOIN BUY	-0.0000974787	0.0342333239	1.0	0.0788500934
SHANGHAI LANSHENG	0.0265230666	0.0627535593	2.0	0.0412509334
SHANGHAI LUJIAZUI	0.0219800377	0.0522001520	1.0	0.0487450841
SHANGHAI MALING AQUARIUS	-0.0022519582	0.0209697175	1.6	0.0628793072
SHANGHAI MATERIAL TRADING	0.0019988740	0.0269008512	3.0	0.0316717115
SHANGHAI MET	-0.0098890711	0.0489644085	2.0	0.0983232901
SHANGHAI NEW HUANG	0.0049443434	0.0314400796	1.0	0.1330022092
SHANGHAI NEW WORLD	0.0289358607	0.0375460101	3.0	0.0390966791
SHANGHAI ORIENTAL PEARL	0.0220154678	0.0278307319	4.0	0.0397542654
SHANGHAI PHARMACEUTICAL	0.0067632613	0.0302710795	0.8	0.0767195904
SHANGHAI POTEVIO	0.0033464162	0.0309781195	1.0	0.0320509144
SHANGHAI PUDONG	0.0322783762	0.0473529321	3.0	0.0789001251
SHANGHAI QIANGSHENG	0.0036785803	0.0181867101	3.0	0.0875269475
SHANGHAI SANMAO	-0.0054425798	0.0289360883	1.0	0.1187407545
SHANGHAI SHENDA	-0.0473075313	0.0414712377	0.0	0.0827792909
SHANGHAI SHENHUA	0.0078798400	0.0401788255	2.0	0.0622365254
SHANGHAI SHENTONG	-0.0032644948	0.0251639753	2.0	0.0776307019
SHANGHAI SHIMAO	0.0127068194	0.0397342372	2.0	0.1247326092

Table 14: Data for the year 2007.

Company	Predicted return	Predicted risk	Reputation	Exact return
SHANGHAI TIANCHEN	-0.0009883468	0.0428137143	1.0	0.0781031609
SHANGHAI TONGDA	0.0013471583	0.0231785931	2.0	0.0638210019
SHANGHAI TONGJI	-0.0079867215	0.0160829836	1.0	0.0809720403
SHANGHAI TUNNEL	0.0039532940	0.0323911144	1.0	0.0976344383
SHANGHAI WANYE	0.0028207256	0.0392391776	2.0	0.1072564838
SHANGHAI WHITECAT	0.0061104762	0.0295001811	2.0	0.0456538292
SHANGHAI WORLDBEST	0.0003438346	0.0319195651	0.8	0.0801028203
SHANGHAI XINGYE	-0.0190772095	0.0225869383	1.0	0.1159151411
SHANGHAI YAOHUA	0.0015131791	0.0297835861	3.0	0.1005876638
SHANGHAI YIMIN	0.0147481093	0.0325460320	2.0	0.0597010636
SHANGHAI YUYUAN	0.0190745622	0.0407355883	2.0	0.1114975293
SHANGHAI ZHANGJIANG	0.0202359072	0.0383347862	2.4	0.0828642794
SHANGHAI ZI JIANG	-0.0005688117	0.0151200573	1.2	0.1154108599
SHANXI COKING	0.0092922824	0.0494242914	1.6	0.0823591022
SHANXI LANHUA SCI-TECH	0.0302771776	0.0341707189	2.4	0.0756272660
SHANXI TOP ENERGY	0.0105660310	0.0338970033	3.0	0.1263270798
SHANXI XINGHUACUN	0.0513576252	0.0678449981	3.0	0.0154624047
SHEN MA INDUSTRY	0.0044907118	0.0240306706	1.0	0.0997939063
SHENERGY	0.0091689776	0.0281982418	3.0	0.0776083743
SHENJI KUNMING	0.0073948902	0.0401179560	1.0	0.1561240822
SHENYANG JINBEI	0.0042112571	0.0495973960	2.0	0.0962400288
SICHUAN CHANGHONG	0.0021270694	0.0315041280	3.0	0.0620918683
SICHUAN CHUANTOU	0.0010761132	0.0286795064	1.0	0.1810142531
SICHUAN GOLDEN SUMMIT	-0.0027763399	0.0275900891	1.0	0.1468869138
SICHUAN HEJIA	-0.0046173860	0.0302019434	1.6	0.1066552128
SICHUAN MINGXING	0.0007633672	0.0313825586	2.0	0.0763843466
SICHUAN MINJIANG	0.0047979549	0.0222154006	2.0	0.0970756214
SICHUAN SWELLFUN	0.0319992233	0.0720007491	2.0	0.0666697838
SILVERTIE HOLDING	-0.0034698977	0.0229801209	2.0	0.1491072114
SINO CONSTRUCTION	-0.0091964943	0.0189714497	1.0	0.1257651450
SINOPEC SHANGHAI	0.0151187547	0.0311128552	2.0	0.0885003504
SINOPEC YIZHENG	0.0013853662	0.0293597439	2.0	0.0879241688
SINOTEX INVESTMENT	0.0031074190	0.0335978738	0.8	0.1290502843
SONGLIAO AUTOMOTIVE	-0.0101599276	0.0534534804	2.0	0.1043680347
SOUTHWEST	-0.0012572122	0.0261797202	1.0	0.0976260930
STAR LAKE BIOSCIENCE	-0.0035644714	0.0229115313	2.0	0.0963239507
SUNNY LOAN TOP	0.0008247801	0.0290016723	1.0	0.1740853069
SUNTEK TECHNOLOGY	-0.0180424672	0.0264153672	1.0	0.1471643234
SUZHOU NEW DISTRICT HI-TECH	0.0053531613	0.0222838697	1.6	0.0851484144
SVA ELECTRON	-0.0083872555	0.0325540001	2.0	0.0847350679
SVA INFORMATION	-0.0097835050	0.0213387911	1.0	0.0900606931
TAIYUAN HEAVY INDUSTRY	0.0223417142	0.0543040683	1.2	0.1303216784
TIANJIN ENVIRONMENTAL	0.0010485766	0.0252685869	2.0	0.0816750834
TIANJIN HI-TECH	0.0214886429	0.0471425703	1.0	0.0685005698
TIANJIN PORT	0.0220491752	0.0266976341	4.0	0.0984720626
TIANJIN QUANYE BAZAAR	0.0107564040	0.0411039668	1.0	0.1028755530
TIBET RHODIOLA	0.0141561288	0.0485391809	1.2	0.0121882080
TIBET TOURISM	0.0013288851	0.0397380057	2.4	0.0584424584
TONGHUA DONGBAO	0.0066088978	0.0899646814	1.0	0.0411852610
TOPSUN SCIENCE AND TECH	0.0011011894	0.0294088113	0.8	0.1312640148
TSINGHUA TONGFANG	0.0035974796	0.0281624007	1.6	0.1285927877
TSINGTAO BREWERY	0.0228110613	0.0274322383	4.0	0.0908853889

Table 14: Data for the year 2007.

Company	Predicted return	Predicted risk	Reputation	Exact return
TUOPAI YEAST LIQUOR	0.0096897036	0.0394436533	1.0	0.0924219128
WEIFANG BEIDA	-0.0143767668	0.0294313508	1.0	0.1265057727
WINOWNER	-0.0187538611	0.0386872303	1.0	0.1231762526
WINSAN	0.0044575223	0.0464727621	1.0	0.0624665487
WUHAN EAST LAKE HIGH TECH	-0.0055207364	0.0274327107	2.0	0.0808123158
WUHAN HANSHANG	0.0033206743	0.0387994793	2.0	0.0664409523
WUHAN HUMANWELL HI-TECH	0.0061120371	0.0288333819	1.0	0.0817013085
WUHAN LINUO SOLAR ENERGY	-0.0074269488	0.0289022126	2.0	0.0872508244
WUHAN SANZHEN	-0.0029255871	0.0246293354	0.8	0.0852972972
WUHAN STEEL	0.0363369255	0.0379549331	2.4	0.1015003469
WUHAN XIANGLONG	0.0036455924	0.0340823965	1.0	0.0559431223
WUXI TAIJI	0.0051399444	0.0383021223	1.0	0.1096993240
XIAMEN C & D	0.0189953358	0.0388450342	1.2	0.1029096757
XIAMEN ENGINEERING	0.0117792442	0.0506093798	0.8	0.1009152276
SHANGHAI FEILO	0.0064283751	0.0213938597	1.6	0.0770375338
XIAMEN INTERNATIONAL TRADE	0.0096065940	0.0264211618	3.0	0.1193093423
XIAMEN KING LONG MOTOR	0.0334443355	0.0506651613	2.0	0.0471115978
XIAMEN ELECTRONICS	0.0010040613	0.0224464310	0.8	0.0287595278
XIAMEN PROSOLAR	-0.0043500582	0.0255286498	0.6	0.1185770008
XI'AN SEASTAR	-0.0076233686	0.0266911716	0.6	0.1494483890
XINING SPECIAL STEEL	0.0113199265	0.0385699441	1.6	0.1511319224
XINJIANG FRIENDSHIP	-0.0055618682	0.0255305128	1.0	0.1158414435
XINJIANG JOINWORLD	0.0372996281	0.0657039696	0.8	0.0836934944
XINJIANG TALIMU	-0.0023745070	0.0310766253	1.2	0.0889163786
XINJIANG TIANYE	0.0216130374	0.0366186183	2.4	0.0858805185
XINJIANG YILITE	0.0175471170	0.0389566174	0.6	0.0858723824
XINYU IRON & STEEL	-0.0022207822	0.0376941618	1.6	0.1278570820
Y.U.D. YANGTZE RIVER	-0.0076246858	0.0290922500	1.6	0.0814601670
YANTAI HUALIAN	-0.0039129766	0.0517225010	2.0	0.1103135846
YANTAI XINCHAO	0.0067019095	0.0369730105	1.0	0.1010950421
YANZHOU COAL MINING	0.0122102773	0.0257588713	2.4	0.1005870204
YIBIN PAPER	-0.0029514381	0.0310281988	2.0	0.0664736660
YOUNGOR	0.0276331943	0.0399972662	3.0	0.1104539961
YUNNAN BOWIN TECH	0.0148770681	0.0367367115	1.0	0.1082874781
YUNNAN FREETRADE	0.0037465494	0.0468202379	1.6	0.0666452955
YUNNAN MALONG	-0.0042616088	0.0206484325	0.8	0.1108873785
YUNNAN YUNTIANHUA	0.0267536956	0.0346109569	2.4	0.1133276925
YUNNAN YUNWEI	0.0129952882	0.0358344820	1.6	0.1481124586
ZHE JIANG DONG RI	-0.0010829261	0.0285016827	0.8	0.0658140507
ZHEJIANG CHINA LIGHT	0.0167492152	0.0478362451	2.0	0.0778194574
ZHEJIANG FURUN	-0.0028438681	0.0293492845	0.8	0.0846752284
ZHEJIANG GUYUELONGSHAN	0.0293818382	0.0577557509	2.4	0.0424528682
ZHEJIANG HOLLEY TECH	-0.0064655719	0.0240858474	0.8	0.1168341785
ZHEJIANG HSD INDUSTRIAL	-0.0060452241	0.0254098238	1.0	0.1128253126
ZHEJIANG JUHUA	0.0074410212	0.0222641297	1.2	0.0808181437
ZHEJIANG MEDICINE	-0.0001980467	0.0314042343	2.4	0.1321507035
ZHEJIANG ORIENT	-0.0032307399	0.0336908803	1.0	0.1246721499
ZHEJIANG QIANJIANG	-0.0023235815	0.0162260780	2.0	0.0672186776
ZHEJIANG SHENGHUA BIOK	-0.0120240550	0.0318131380	0.6	0.0775491385
ZHEJIANG XINHU VENTURE	0.0250106450	0.0591622348	2.0	0.0944047926
ZHENGZHOU COAL & ELECTRIC	-0.0029026095	0.0227741050	0.8	0.1225076603
ZHENGZHOU YUTONG BUS	0.0380716179	0.0462050621	4.0	0.0860930248

Table 14: Data for the year 2007.

Company	Predicted return	Predicted risk	Reputation	Exact return
ZHONGCHU DEVELOPMENT STOCK	0.0230146554	0.0372516637	1.6	0.0654569265
ZHONGLU	0.0000378272	0.0359098262	3.0	0.0653801085

Appendix B. Code Used for Testing.

This appendix lists the code in C++ used for the selection of an optimal portfolio. The code also contains the functions that are not a part of the algorithm for portfolio selection, but were used for testing the performance of the proposed algorithm.

```
// Created by Tanja Magoc
// Date created: March 15, 2009
// INVESTMENT PORTFOLIO
/* The program uses utility-based multi-criteria decision making setting
   and fuzzy integration over intervals to determine an optimal asset for a client.
   The maximum number of assets in a database: 400.
   The maximum number of characteristics: 10.
   The maximum number of assets in a portfolio: 50.
*/

#include<iostream>
#include<fstream>
#include<string>
#include<cmath>
using namespace std;

bool readFile(ifstream& ifile, char* strFile);
int readingDataFromFile(ifstream& ifile, char* fileName, struct Asset listOfAssets[]);
void readingReturnsOfNextYearFromFile(ifstream& ifile, char* fileName,
     struct ReturnsOnly nextYearAssets[]);
void inputShapley(string criteria[], float shapley[][2], int numberCriteria);
float absoluteValue(float x);
float findMax(float particularCriterion[], int originalNumberAssets);
float findMin(float particularCriterion[], int originalNumberAssets);
float utilityForAsset_direct(int originalNumberAssets, float particularCriterion[], float min,
      float max, float valueOfParticularAsset);
float utilityForAsset_inverse(int originalNumberAssets, float particularCriterion[], float min,
      float max, float valueOfParticularAsset);
```

```
void choquet_calculation(float choquet[][2], float u[], float shapley[][2],
     float interaction[][10][2], int numberCriteria, int assetNumber);
int intervalComparison(float interval1[], float interval2[]);
void quickSortOnIntervalChoquet(float sortedChoquet[][3], int originalNumberAssets);
void createSelectedListOfAssets(struct NewAsset newList[], struct Asset listOfAssets[],
     float sortedChoquet[][3], int numberAssets);
float utilityForPortfolio_direct(int numberAssets, float particularCriterion[], float weights[],
      float min, float max);
float utilityForPortfolio_inverse(int numberAssets, float particularCriterion[], float weights[],
      float min, float max);
float penaltyForTotalMoney(float weight[], int numberAssets, int numberCriteria);
float penaltyForShortselling(float weight[], int numberAssets, int numberCriteria);
void fitness(struct NewAsset newList[], float weight[], float shapley[][2],
     float interaction[][10][2], int numberCriteria, int numberAssets, float fit[],
     float choquet[][2], float minValue[], float maxValue, float justReturns[],
     float justRisks[], float justReputations[]);
void calculateFitnessOfEachIndividual(float population[][50], float fitnesses[][3],
     int populationSize, int numberOfGenes, struct NewAsset newList[], float shapley[][2],
     float interaction[][10][2], int numberCriteria, float choquet[][2]);
void sortFitnesses(float populatoin[][50], float fitnesses[][2], int populationSize,
     int numberOfGenes);
void createIndividual(float individual[], int numberOfGenes, float lowerBound, float upperBound);
void createLowerExtremeIndividual(float individual[], int numberOfGenes, float lowerBound);
void createUpperExtremeIndividual(float individual[], int numberOfGenes, float upperBound);
void createEquallyWeightedExtremeIndividual(float individual[], int numberOfGenes);
void initialPopulation(float population[][50], int populationSize, int numberOfGenes,
     float lowerBound, float upperBound);
int poolForReproduction(float pool[][50], float population[][50], int populationSize,
    int numberOfGenes, int tenPercent, int fivePercent);
int selection(int sizeOfPool);
void crossover(float pool[][50], float population[][50], int populationSize, int numberOfGenes,
     int position1, int position2, int tenPercent, int fivePercent);
void crossoverWithExtremes(float population[][50], int numberOfGenes,
```

```
     int iterationsForExtremeIndividuals, int populationSize);
void crossoverWithEquallyWeightedExtreme(float population[][50], int numberOfGenes,
     int iterationsForExtremeIndividuals, int populationSize);
void mutation(int numberOfGenes, int individual, float pool[][50], float lowerBound,
     float upperBound);
void optimization(int numberOfGenes, float bestIndividual[], float bestFitness[],
     struct NewAsset newList[], float shapley[][2], float interaction[][10][2],
     int numberCriteria, float choquet[][2], struct ReturnsOnly nextYearReturns[]);

struct Asset /* Stores name (company) and data (return, risk, reputation) about an asset. */
{
    string nameOfAsset;
    float returnOfAsset;
    float riskOfAsset;
    float reputationOfAsset;
};

struct NewAsset /* Used for the assets selected for the second stage of the algorithm. */
{
    string nameOfAsset;
    float returnOfAsset;
    float riskOfAsset;
    float reputationOfAsset;
    int originalPosition; /* The position in the original list of assets. */
};

struct ReturnsOnly /* Used to hold accurate (not predicted) returns of assets.
                      Used for testing phase of the return of a portfolio. */
{
    string nameOfAsset;
    float returnOfAsset;
};
```

```
int main(void)
{
    string criteria[10]; /* The array of the names of criteria (i.e., return, risk, etc.). */
    Asset listOfAssets[400]; /* The array that stores all data about assets. */
    ReturnsOnly nextYearAssets[400]; /* The array that stores the next year's data about assets;
                                        used for testing purposes only. */
    NewAsset newList[50]; /* The array of data for assets selected for the second stage of
                             the algorithm. */
    bool status; /* Used to store the value of whether a file could be opened for reading
                    (the file that contains assets' data). */
    float shapley[10][2]; /* The array of the Shapley values entered by a user; the Shapley
                             values are entered as intervals: [i][0] holds the lower bound,
                             [i][1] holds the upper bound of the interval. */
    float interaction[10][10][2]; /* The array of interaction indices of order 2;
                                     [i][j][0] is the lower bound, [i][j][1] is the
                                     upper bound of the interval. */
    float u[10]; /* The array of the utility values of an asset: u[0] = the utility of return,
                    u[1] = risk, u[2] = reputation. */
    float choquet[400][2]; /* The array of the values of the Choquet integral of each asset
                              (in stage 1 of the algorithm); [i][0] is the lower bound,
                              [i][1] is the upper bound of the interval. */
    float sortedChoquet[400][3]; /* The array that keeps track of the assets when sorted
                                    based on the interval Choquet integral; the third field
                                    in the second dimension is used to keep track of the
                                    position of the asset in the original order of assets. */
    int numberCriteria; /* The total number of criteria. */
    int originalNumberAssets; /* The total number of assets in the market. */
    int numberAssets; /* The number of assets selected for the second stage of the algorithm. */
    float bestIndividual[50]; /* The distribution of wealth in the current best solution. */
    float bestFitness[2]; /* The current best fitness. It is an interval: bestFitness[0] holds
                             the lower bound, and bestFitness[1] holds the upper bound of
                             the interval. */
    float return1, return2, return3; /* Used for actual returns of the benchmark portfolios. */
```

```
    float riskAdjustedReturn[400][2]; /* The array holds the values of predicted risk-adjusted
                                         returns. It is used for testing purposes only.
                                         riskAjdustedReturn[0] holds the value of the return,
                                         riskAdjustedReturn[1] holds the position of the asset
                                         in the original list of assets. */
    float temp[2]; /* A temporary array. */
    float dataArrays[10][400]; /* Used to hold the returns (at [0][i]), the risks (at [1][i]),
                                  and the reputations (at [2][i]) of the assets. */
    float maxValue[10], minValue[10]; /* Used to hold the maximum and the minimum values of
                                         each criterion based on the used database. */

// Reading predicted data from a file.
   ifstream infile;
   status = readFile(infile, "c:\\Users\\Tanja\\Documents\\test1");

   if (!status)
   {
        cout << "File 1 could not be opened for reading \n";
        return 0;
    }
    else /* Create an array containing predicted data about assets; the array is of
            the struct type Asset. */
        originalNumberAssets = readingDataFromFile(infile, "c:\\Users\\Tanja\\Documents\\test1",
            listOfAssets);

// Reading accurate returns from a file.
// These values are used only for testing purposes, and are not needed in the actual investment
   portfolio algorithm.
    ifstream infile2;
    status = readFile(infile2, "c:\\Users\\Tanja\\Documents\\test1accurateData");

    if (!status)
    {
```

```
        cout << "File 2 could not be opened for reading \n";
        return 0;
    }
    else /* Create an array containing accurate returns of assets; the array is of
            the struct type ReturnsOnly. */
        readingReturnsOfNextYearFromFile(infile2, "c:\\Users\\Tanja\\Documents\\test1accurateData",
            nextYearAssets);

// Defining criteria.
    numberCriteria = 3;

    criteria[0] = "return";
    criteria[1] = "risk";
    criteria[2] = "reputation";

// Ask a user to input the Shapley values for each criterion. The Shapley values are entered
   as intervals in the range [0,1].
    inputShapley(criteria, shapley, numberCriteria);

// Interaction indices of order 2 are hard-coded by an expert. They are entered as intervals
   in the range [-1,1].
// interaction[i][j] is the interaction index of order 2 between the criteria i and j.
// interaction[i][j][0] holds the lower bound and interaction[i][j][1] hold the upper bound.
    interaction[0][1][0] = -0.9; /* return-risk */
    interaction[0][1][1] = -0.8;
    interaction[0][2][0] = 0.4; /* return-reputation */
    interaction[0][2][1] = 0.5;
    interaction[1][2][0] = -0.6; /* risk-reputation */
    interaction[1][2][1] = -0.5;

// Create separate temporary arrays for the return, the risk, and the reputation of each asset.
// They are used in calculation of the utility values of each criterion for each asset.
    for(int i = 0; i < originalNumberAssets; i++)
```

```
    {
        dataArrays[0][i] = listOfAssets[i].returnOfAsset; /* return */
        dataArrays[1][i] = listOfAssets[i].riskOfAsset; /* risk */
        dataArrays[2][i] = listOfAssets[i].reputationOfAsset; /* reputation */
    }

// Find the max and the min of each criterion based on the database values. They will be used
   to define utility functions.
    for(int i = 0; i < numberCriteria; i++)
    {
        maxValue[i] = findMax(dataArrays[i], originalNumberAssets);
        minValue[i] = findMin(dataArrays[i], originalNumberAssets);
    }

// Calculate the utility values of each criterion for each asset and the Choquet integral for
   each asset.
    for(int i = 0; i < originalNumberAssets; i++)
    {
        // Calculate the utility values of each criterion for the asset i.
        u[0] = utilityForAsset_direct(originalNumberAssets, dataArrays[0], minValue[0],
               maxValue[0], dataArrays[0][i]);
        u[1] = utilityForAsset_inverse(originalNumberAssets, dataArrays[1], minValue[1],
               maxValue[1], dataArrays[1][i]);
        u[2] = utilityForAsset_direct(originalNumberAssets, dataArrays[2], minValue[2],
               maxValue[2], dataArrays[2][i]);

        // Calculate the Choqiet integral for the asset i.
        choquet_calculation(choquet, u, shapley, interaction, numberCriteria, i);
    }

// Order the assets from the asset with the highest value of the Choquet integral to the asset
   with the lowest value of the Choquet integral.
// The comparisons are based on the degree of preference used for comparing intervals.
```

```
    for(int i = 0; i < originalNumberAssets; i++)
    /* Initialize the array that will be sorted. */
    {
        sortedChoquet[i][0] = choquet[i][0];
        sortedChoquet[i][1] = choquet[i][1];
        sortedChoquet[i][2] = i; /* Keep the positions of the assets in the original array. */
    }

    quickSortOnIntervalChoquet(sortedChoquet, originalNumberAssets);

// Select the number of assets to proceed to the second stage of the algorithm.
    do
    {
        cout << "Enter the number of assets to be included in a portfolio: ";
        cin >> numberAssets;
    }
    while ((numberAssets > originalNumberAssets) || (numberAssets <= 0));

// Create a new array containing only data of selected assets.
    createSelectedListOfAssets(newList, listOfAssets, sortedChoquet, numberAssets);

// Second stage of the algorithm: distribute weights among the selected assets.

// Initialize the best distribution and the outcome (the Choquet value) of the best
   distribution.
    for(int i = 0; i < numberAssets; i++)
        bestIndividual[i] = 0.0; /* 'bestIndividual' = The best distribution. */
    bestFitness[0] = 0.0; /* The lower bound of the Choquet integral of the best individual. */
    bestFitness[1] = 0.0; /* The upper bound of the Choquet integral of the best individual. */

// Optimization part.
    optimization(numberAssets, bestIndividual, bestFitness, newList, shapley, interaction,
```

```
    numberCriteria, choquet, nextYearAssets);

// Print out the optimal portfolio.
    cout << "The optimal portfolio \n";
    for(int i = 0; i < numberAssets; i++)
        cout << listOfAssets[newList[i].originalPosition].nameOfAsset << ": " <<
                bestIndividual[i] << "\n";
    cout << "\n";

// Calculate and display the actual returns of benchmark portfolios.
// Used only for testing purposes.
// 1. An equally weighted portfolio using all assets.
    return1 = 0;
    for(int i = 0; i < originalNumberAssets; i++)
        return1 += nextYearAssets[i].returnOfAsset;
    return1 = return1 / originalNumberAssets;
    cout << "The return of the benchmark 1: " << return1 << "\n";

// 2. An equally weighted portfolio using only the selected assets.
    return2 = 0;
    for(int i = 0; i < numberAssets; i++)
        return2 += nextYearAssets[newList[i].originalPosition].returnOfAsset;
    return2 = return2 / numberAssets;
    cout << "The return of the benchmark 2: " << return2 << "\n";

// 3. An equally weighted portfolio of the top N assets based on the expected risk-adjusted
      return strategy, where N is the number of assets selected to proceed to the second stage
      of the algorithm.
    return3 = 0;
    for(int i = 0; i < originalNumberAssets; i++)
    {
        /* Calculate the predicted risk-adjsuted return for each asset. */
        riskAdjustedReturn[i][0] = listOfAssets[i].returnOfAsset / listOfAssets[i].riskOfAsset;
```

```
        riskAdjustedReturn[i][1] = i;
    }
    for(int i = 0; i < (originalNumberAssets-1); i++) /* Sort predicted risk-adjusted returns
                                                         using bubble sort. */
        for(int j = i+1; j < originalNumberAssets; j++)
            if (riskAdjustedReturn[i][0] < riskAdjustedReturn[j][0])
            {
                temp[0] = riskAdjustedReturn[i][0];
                temp[1] = riskAdjustedReturn[i][1];
                riskAdjustedReturn[i][0] = riskAdjustedReturn[j][0];
                riskAdjustedReturn[i][1] = riskAdjustedReturn[j][1];
                riskAdjustedReturn[j][0] = temp[0];
                riskAdjustedReturn[j][1] = temp[1];
            }
    for(int i = 0; i < numberAssets; i++) /* Calculate the actual return. */
        return3 += nextYearAssets[(int)riskAdjustedReturn[i][1]].returnOfAsset;
    return3 = return3 / numberAssets;
    cout << "The return of the benchmark 3: " << return3 << "\n";

    return 0;
}

bool readFile (ifstream& ifile, char* strFile)
/* Checks if there exists a file 'strFile' to open it for reading data. */
/* The expected file contains predicted data about assets. */
/* Input: the file 'strFile' containing the data. */
/* Output: 'false' if the file can't be open; and 'true' if the file is open. */
/* Side output (if 'true): ifile contains the file 'strFile'. */
{
    ifile.open(strFile);
    if (ifile.fail())
        return false;
    else
```

```
        return true;
}

int readingDataFromFile(ifstream& ifile, char* fileName, struct Asset listOfAssets[])
/* Reads data from the file 'fileName' into the array 'listOfAssets'. */
/* Each element of the array contains the name, the return, the risk, and the reputation
   of a company. */
/* Inputs: 'ifile' containing the file;
           the name of the file. */
/* Output: the number of elements in the array, i.e., the number of assets in the market. */
/* Side output: the array 'listOfAssets' that contains predicted data for each asset. */
{
    int originalNumberAssets = 0;

    ifile >> listOfAssets[originalNumberAssets].nameOfAsset;
    if (listOfAssets[originalNumberAssets].nameOfAsset != "")
    {
        ifile >> listOfAssets[originalNumberAssets].returnOfAsset;
        ifile >> listOfAssets[originalNumberAssets].riskOfAsset;
        ifile >> listOfAssets[originalNumberAssets].reputationOfAsset;
        originalNumberAssets += 1; /* Increase the counter of assets. */
    }

    while(!ifile.fail())
    {
        ifile >> listOfAssets[originalNumberAssets].nameOfAsset;
        if (listOfAssets[originalNumberAssets].nameOfAsset != "")
        {
            ifile >> listOfAssets[originalNumberAssets].returnOfAsset;
            ifile >> listOfAssets[originalNumberAssets].riskOfAsset;
            ifile >> listOfAssets[originalNumberAssets].reputationOfAsset;
            originalNumberAssets += 1;
        }
```

```
    }

    ifile.close();
    return originalNumberAssets;
}

void readingReturnsOfNextYearFromFile(ifstream& ifile, char* fileName, struct ReturnsOnly
     nextYearAssets[])
/* Reads data from the file 'fileName' into the array 'nextYearAssets'. */
/* Each element of the array contains the name and the actual return of the company known only
   after the fact. */
/* Inputs: 'ifile' containing the file;
           the name of the file. */
/* Output: none.
/* Side output: the array 'nextYearAssets' that contains the actual return of each asset. */
{
    int counter = 0;

    ifile >> nextYearAssets[counter].nameOfAsset;
    if (nextYearAssets[counter].nameOfAsset != "")
    {
        ifile >> nextYearAssets[counter].returnOfAsset;
        counter += 1; /* Increase the counter of assets. */
    }

    while(!ifile.fail())
    {
        ifile >> nextYearAssets[counter].nameOfAsset;
        if (nextYearAssets[counter].nameOfAsset != "")
        {
            ifile >> nextYearAssets[counter].returnOfAsset;
            counter += 1;
        }
```

```
    }

    ifile.close();
}

void inputShapley(string criteria[], float shapley[][2], int numberCriteria)
/* User inputs Shapley values for each criterion. */
/* The values are entered as intervals on the range [0,1]. */
/* The same values are used for each asset. */
/* Inputs: the array containing all criteria;
           the number of criteria. */
/* Output: none. */
/* Side output: a 2-dimenstional array of Shapley values, where shapley[i][0] contains the lower
   bound and shapley[i][1] contains the upper bound of the Shapley value of the criterion i. */
{
    cout << "Enter the Shapley values for each criterion \n \n";
    for (int i = 0; i < numberCriteria; i++)
    {
        do
        {
            cout << "The lower bound of "<< criteria[i] << ": ";
            cin >> shapley[i][0];
        }
        while ((shapley[i][0] > 1) || (shapley[i][0] < 0));
        do
        {
            cout << "The upper bound of "<< criteria[i] << ": ";
            cin >> shapley[i][1];
        }
        while ((shapley[i][1] > 1) || (shapley[i][1] < 0) || (shapley[i][1] < shapley[i][0]));
    }
}
```

```
float absoluteValue(float x)
/* Calculates the absolute value of x. */
/* Input: a floating point number x. */
/* Output: the absolute value of x. */
/* Side output: none. */
{
    if (x >= 0) return x;
    else return (-x);
}

float findMax(float particularCriterion[], int originalNumberAssets)
/* Finds the maximum value of all the elements of the array 'particularCriterion'. */
/* Inputs: the array;
           the number of elements in the array. */
/* Output: the maximum value of the array. */
/* Side output: none. */
{
    float max;
    max = particularCriterion[0];
    for (int i = 1; i < originalNumberAssets; i++)
        if (particularCriterion[i] > max)
            max = particularCriterion[i];

    return max;
}

float findMin(float particularCriterion[], int originalNumberAssets)
/* Finds the minimum value of all the elements of the array 'particularCriterion'. */
/* Inputs: the array;
           the number of elements in the array. */
/* Output: the minimum value of the array. */
/* Side output: none. */
{
```

```
    float min;
    min = particularCriterion[0];
    for (int i = 1; i < originalNumberAssets; i++)
        if (particularCriterion[i] < min)
            min = particularCriterion[i];

    return min;
}

float utilityForAsset_direct(int originalNumberAssets, float particularCriterion[], float min,
      float max, float valueOfParticularAsset)
/* Calculates the utility value of the criterion 'particularCriterion' for the asset whose value
   of the criterion is given by 'valueOfParticularAsset'. */
/* It applies only to criteria where a higher value of the criterion has a higher utility. */
/* Inputs: the number of assets;
           the criterion in consideration 'particularCriterion';
           the min and the max of 'particularCriterion' in the database used;
           the value of the particular criterion for the particular asset being examined. */
/* Output: the utility value of the particular criterion for the given asset. */
/* Side output: none. */
{
    float value;

    /* The utility function maps the max into 1, the min into 0,
       and all the other values by a linear function to the interval (0,1). */
    value = (valueOfParticularAsset - min)/(max - min); /* 'value' is the utility value
                                                           of the criterion for the given asset. */
    return value;
}

float utilityForAsset_inverse(int originalNumberAssets, float particularCriterion[], float min,
      float max, float valueOfParticularAsset)
/* Calculates the utility value of the criterion 'particularCriterion' for the asset whose value
```

```
   of the criterion is given by 'valueOfParticularAsset'. */
/* It applies only to criteria where a higher value of the criterion has a lower utility. */
/* Inputs: the number of assets;
           the criterion in consideration 'particularCriterion';
           the min and the max of 'particularCriterion' in the database used;
           the value of the particular criterion for the particular asset being examined. */
/* Output: the utility value of the particular criterion for the given asset. */
/* Side output: none. */
{
    float value;

    /* The utility function maps the max into 0, the min into 1,
       and all the other values by a linear function to the interval (0,1). */
    value = (max - valueOfParticularAsset)/(max - min); /* 'value' is the utility value of
                                                          the criterion for the given asset. */
    return value;
}

void choquet_calculation(float choquet[][2], float u[], float shapley[][2], float
     interaction[][10][2], int numberCriteria, int assetNumber)
/* Calculates the Choquet integral w.r.t. a 2-additive measure of the asset given by its number
   'assetNumber' in the listOfAssets. */
/* The calculation is performed over the intervals, so the value of the Choquet integral is
   an interval: choquet[i][0] is the lower bound and choquet[i][1] is the upper bound of
   the Choquet integral of the asset i. */
/* Inputs: the utility values of each criterion for the given asset;
           the Shapley values of each criterion;
           the interaction indices of order 2 of each pair of criteria;
           the number of criteria;
           the order number of the asset in the array of assets. */
/* Output: none. */
/* Side output: updates the value of the Choquet integral for the given asset in the array
                'choquet'. */
```

```
{
    float lowerBound, upperBound; /* Intermediate values of a lower and an upper bound of
                        the Choquet integral while taking care of interaction indices. */
    float x; /* A temporary variable. */
    float sumLower, sumUpper; /* Intermediate values of a lower and an upper bound of
                     the Choquet integral while taking care of the Shapley values. */
    float lower, upper; /* Temporary variables. */

    lowerBound = 0.0;
    upperBound = 0.0;
    x = 0.0;

    /* Taking care of the interaction indices in the Choquet integral. */
    for(int i = 0; i < (numberCriteria-1); i++)
        for(int j = (i+1); j < numberCriteria; j++)
        {
            if (interaction[i][j][0] > 0) /* Complementary criteria. */
            {
                x = min(u[i], u[j]);
                lowerBound = lowerBound + x * interaction[i][j][0];
                upperBound = upperBound + x * interaction[i][j][1];
            }
            else /* Redundant criteria and independent criteria (interaction[][]=0 in the case
                    of independent criteria). */
            {
                x = max(u[i], u[j]);
                lowerBound = lowerBound + absoluteValue(interaction[i][j][1]) * x;
                /* The absolute value of the upper bound of a negative interaction index will
                   be greater than the absolute value of the lower bound, so the lower bound of
                   the Choquet integral is updated by a smaller increase which is the absolute
                   value of the dimension [1] (the upper bound).*/
                upperBound = upperBound + absoluteValue(interaction[i][j][0]) * x;
            }
```

```
    } /* The end of taking care of the interaction indices. */


/* Taking care of the Shapley values in the Choquet integral. */
for(int i = 0; i < numberCriteria; i++)
{
    sumLower = 0.0;
    sumUpper = 0.0;
    for(int j = 0; j < numberCriteria; j++)
        if (i < j)
        {
            if (interaction[i][j][0] > 0)
            {
                sumLower = sumLower + interaction[i][j][0];
                sumUpper = sumUpper + interaction[i][j][1];
            }
            else /* interaction[i][j][0] < 0. */
            {
                sumLower = sumLower + absoluteValue(interaction[i][j][1]);
                sumUpper = sumUpper + absoluteValue(interaction[i][j][0]);
            }
        } /* The end of i < j. */
        else /* interaction[i][j] for i > j is not explicitly defined, but we know that
                interaction[i][j] = interaction[j][i] */
            if (i > j)
            {
                if (interaction[j][i][0] > 0)
                {
                    sumLower = sumLower + absoluteValue(interaction[j][i][0]);
                    sumUpper = sumUpper + absoluteValue(interaction[j][i][1]);
                }
                else
                {
                    sumLower = sumLower + absoluteValue(interaction[j][i][1]);
```

```
                        sumUpper = sumUpper + absoluteValue(interaction[j][i][0]);
                    }
                } /* The end of i > j. */

        lower = sumUpper * (-0.5);
        upper = sumLower * (-0.5);

        lower = (shapley[i][0] + lower) * u[i];
        upper = (shapley[i][1] + upper) * u[i];
    } /* The end of taking care of the Shapley values. */

    lowerBound += lower; /* Add the interaction indices part and the Shapley values part. */
    upperBound += upper;

    choquet[assetNumber][0] = lowerBound; /* Update the array of the Choquet values with
                                             the value for the particular asset. */
    choquet[assetNumber][1] = upperBound;
}

int intervalComparison(float interval1[], float interval2[])
/* Compares 2 intervals using the degree of preference. */
/* An interval is represented by its lower bound 'interval[0]' and its upper bound
  'interval[1]'. */
/* Inputs: two intervals. */
/* Output: returns 1 if the first interval is better; returns 0 otherwise. */
/* Side output: none. */
{
    float degree; /* The degree of preference. */

    if ((interval1[1] >= interval2[1]) && (interval1[0] >= interval2[0]))
        degree = 1;
    else if ((interval1[1] > interval2[1]) && (interval1[0] <= interval2[0]))
            degree = (interval1[1] - interval2[1]) /
```

```
                    ((interval1[1] - interval2[1]) + (interval2[0] - interval1[0]));
            else /* The second interval has a higher (or equal) upper bound. */
            {
               degree = intervalComparison(interval2, interval1);
               degree = 1 - degree;
            }

    if (degree >= 0.5) return 1;
    else return 0;
}

void quickSortOnIntervalChoquet(float sortedChoquet[][3], int originalNumberAssets)
/* Sorts elements of the array of the Choquet values using the interval comparison. */
/* Quick sort algorithm is used. */
/* The array is sorted from higher values to lower values, so the best value is in
   the first position. */
/* Inputs: the array holding the Choquets integral values of each asset:
           the first two dimensions hold the lower and the upper bounds of the Choquet value,
           the third dimension holds the position of the asset in the original (unsorted) array;
           the number of assets. */
/* Output: none. */
/* Side output: the sorted array of the values of the Choquet integrals. */
{
    float beginningArray[400][3]; /* A temporary array to hold the elements in front of
                                     the pivot element. */
    float endArray[400][3]; /* A temporary array to hold the elements after the pivot. */
    int lengthBeginningArray = 0;
    int lengthEndArray = 0;
    int degree; /* The degree of preference, which is a result of the comparison of
                   two intervals. */

    if (originalNumberAssets > 1)
    {
```

```
for (int i = 0; i < (originalNumberAssets - 1); i++)
    {
        degree = intervalComparison(sortedChoquet[i],
                 sortedChoquet[originalNumberAssets-1]);
                 /* Compare the element to the pivot. */
        if (degree == 1) /* The element is better than the pivot; move it to the front
                            of the array. */
        {
            for (int j = 0; j < 3; j++)
                beginningArray[lengthBeginningArray][j] = sortedChoquet[i][j];
            lengthBeginningArray += 1;
        }
        else /* The element is worse than the pivot; move it to the end of the array.*/
        {
            for (int j = 0; j < 3; j++)
                endArray[lengthEndArray][j] = sortedChoquet[i][j];
            lengthEndArray += 1;
        }
    }

quickSortOnIntervalChoquet(beginningArray, lengthBeginningArray); /* Recursive calls. */
quickSortOnIntervalChoquet(endArray, lengthEndArray);

/* Combine the arrays in front of the pivot and after the pivot into the array
   'sortedChoquet'. */
for (int i = 0; i < (lengthBeginningArray); i++)
    for (int j = 0; j < 3; j++)
        sortedChoquet[i][j] = beginningArray[i][j];

for (int j = 0; j < originalNumberAssets; j++)
    sortedChoquet[lengthBeginningArray][j] = sortedChoquet[originalNumberAssets-1][j];

for (int i = 0; i < (lengthEndArray); i++)
```

```
            for (int j = 0; j < 3; j++)
                sortedChoquet[lengthBeginningArray + i+1][j] = endArray[i][j];
    }
}

void createSelectedListOfAssets(struct NewAsset newList[], struct Asset listOfAssets[],
     float sortedChoquet[][3], int numberAssets)
/* Creates an array containing only data of the assets selected for the second stage of
   the algorithm. */
/* Inputs: the array containing all data about all assets;
           the sorted array of the Choquet values;
           the number of assets selected for the second stage of the algorithm. */
/* Output: none. */
/* Side output: the array 'newList' containing all data of the selected assets. */
{
    int orderedAssetNumber;

    for(int i = 0; i < numberAssets; i++)
    {
        orderedAssetNumber = sortedChoquet[i][2]; /* The position of the i-th asset in
                                                     the original 'listOfAssets'. */
        newList[i].nameOfAsset = listOfAssets[orderedAssetNumber].nameOfAsset;
        newList[i].returnOfAsset = listOfAssets[orderedAssetNumber].returnOfAsset;
        newList[i].riskOfAsset = listOfAssets[orderedAssetNumber].riskOfAsset;
        newList[i].reputationOfAsset = listOfAssets[orderedAssetNumber].reputationOfAsset;
        newList[i].originalPosition = orderedAssetNumber;
    }
}

float utilityForPortfolio_direct(int numberAssets, float particularCriterion[], float weight[],
      float min, float max)
/* Calculates the utility value of the criterion 'particularCriterion' for the portfolio given
   by weights 'weight[]'. */
```

```
/* It applies only to criteria where a higher value of the criterion has a higher utility. */
/* Inputs: the number of assets;
           the array of values of the particular criterion for each asset;
           the array of weights (i.e., the distribution of wealth among the assets);
           the minimum value of the particular criterion in the given database;
           the maximum value of the particular criterion in the given database. */
/* Output: the utility value of the 'particularCriterion' of the given portfolio. */
/* Side output: none. */
{
    float utility = 0.0;

    for(int i = 0; i < numberAssets; i++)
        utility = utility + weight[i] * particularCriterion[i];
    utility = (utility - min) / (max - min);

    return utility;
}

float utilityForPortfolio_inverse(int numberAssets, float particularCriterion[], float weight[],
      float min, float max)
/* Calculates the utility value of the criterion 'particularCriterion' for the portfolio given
   by weights 'weight[]'. */
/* It applies only to criteria where a higher value of the criterion has a lower utility. */
/* Inputs: the number of assets;
           the array of values of the particular criterion for each asset;
           the array of weights (i.e., the distribution of wealth among the assets);
           the minimum value of the particular criterion in the given database;
           the maximum value of the particular criterion in the given database. */
/* Output: the utility value of the 'particularCriterion' of the given portfolio. */
/* Side output: none. */
{
    float utility = 0.0;
```

```
    for(int i = 0; i < numberAssets; i++)
        utility = utility + weight[i] * particularCriterion[i];
    utility = (max - utility) / (max - min);

    return utility;
}

float penaltyForTotalMoney(float weight[], int numberAssets, int numberCriteria)
/* Calculates the penalty for spending more than 100% or less than 50% of money.*/
/* Inputs: the array of weights (i.e., the distribution of wealth);
           the number of assets in the portfolio;
           the number of criteria. */
/* Output: the penalty for not using enough money or using too much money. */
/* Side output: none. */
{
    float sumOfWeights = 0;
    float value; /* Penalty. */

    for(int i = 0; i < numberAssets; i++)
        sumOfWeights += weight[i];

    if ((sumOfWeights >= 0.5) && (sumOfWeights <= 1.01))
        value = 0; /* The penalty is 0 if at least 50% of money is invested and no more
                      than all money is invested. */
    else
        value = (numberAssets * numberAssets) + (sumOfWeights - 1) * (sumOfWeights - 1);
        /* Otherwise the penalty is the maximum value obtainable by the Choquet integral,
           which is the numberCriteria squared, increased by the squared distance of
           the sumOfWeights from 1. */

    return value;
}
```

```
float penaltyForShortselling(float weight[], int numberAssets, int numberCriteria)
/* The penalty for short selling (borrowing money from one asset to buy another asset). */
/* Inputs: the array of weights (i.e., the distribution of wealth);
           the number of assets in the portfolio;
           the number of criteria. */
/* Output: the penalty for short selling. */
/* Side output: none. */
{
    float sumOfPenalties = 0;

    for(int i = 0; i < numberAssets; i++)
        if (weight[i] <= 0)
            sumOfPenalties += numberAssets; /* The penalty for short selling of an asset
                                               is equal to the number of criteria. */
                                            /* A higher penalaty is accumulated if there are
                                               more assets used for short selling. */

    return sumOfPenalties; /* The penalty is 0 if there is no short selling. */
}

void fitness(struct NewAsset newList[], float weight[], float shapley[][2],
     float interaction[][10][2], int numberCriteria, int numberAssets, float fit[],
     float choquet[][2], float minValue[], float maxValue[], float justReturns[],
     float justRisks[], float justReputations[])
/* Calculates the fitness of a portfolio. */
/* The fitness is expressed as an interval: fit[0] in the lower bound and fit[1] is the upper
   bound. */
/* Inputs: the array containing data of the assets in the portfolio;
           the weights (i.e., the allocation of wealth to the assets in the portfolio);
           the Shapley values;
           the interaction indices;
           the number of criteria;
           the number of assets in the portfolio;
```

```
        the 'choquet' array, which is used only to ease the use of the 'choquetCalculation'
            function to calculate the fitness of an individual. */
/* Output: none. */
/* Side output: the value of the fitness of a particular individual (i.e., the portfolio)
   given by 'fit[]'. */
{
    float utility[10]; /* utility[0] = the utility or the return of a portfolio; u[1] = risk;
                          u[2] = reputation. */
    float sum = 0;
    float penaltyTotalMoney, penaltyShortSelling;

    utility[0] = utilityForPortfolio_direct(numberAssets, justReturns, weight, minValue[0],
                 maxValue[0]); /* The utility of return. */
    utility[1] = utilityForPortfolio_inverse(numberAssets, justRisks, weight, minValue[1],
                 maxValue[1]); /* The utility of risk. */
    utility[2] = utilityForPortfolio_direct(numberAssets, justReputations, weight, minValue[2],
                 maxValue[3]); /* The utility of reputation. */

    penaltyTotalMoney = penaltyForTotalMoney(weight, numberAssets, numberCriteria);
    penaltyShortSelling = penaltyForShortselling(weight, numberAssets, numberCriteria);

    choquet_calculation(choquet, utility, shapley, interaction, numberCriteria,
                   (numberAssets+1)); /* (numberAssets+1) is used as a temporary variable to
                                  store the value of the Choquet integral of a portfolio. */
    for(int i = 0; i < 2; i++)
    {
        fit[i] = choquet[numberAssets + 1][i]; /* The Choquet integral value of the individual
                                              (i.e., portfolio). */
        fit[i] = fit[i] - penaltyTotalMoney - penaltyShortSelling; /* The fitness of the
                                                 individual (i.e., portfolio). */
    }
}
```

```
void calculateFitnessOfEachIndividual(float population[][50], float fitnesses[][2],
     int populationSize, int numberOfGenes, struct NewAsset newList[], float shapley[][2],
     float interaction[][10][2], int numberCriteria, float choquet[][2])
/* Creates the array 'fitnesses' that contains the fitness of each individual. The fitness is
   expressed as an interval: fitnesses[i][0] is the lower bound and fitnesses[i][1] is
   the upper bound of the fitness of the individual i. */
/* Inputs: the current population (i.e., portfolios);
           the number of individuals in the population;
           the number of genes (i.e., assets) in each individual;
           the array 'newList' containing the data about the assets in the portfolio;
           the Shapley values;
           the interaction indices of order 2;
           the number of criteria;
           the array of the Choquet values of each asset in the portfolio; */
/* Output: none. */
/* Side output: creates the array 'fitnesses' containing fitness of each individual. */
{
    float fit[2]; /* a temporary array. */
    float maxValue[10]; /* The maximum value of each criterion based on the used database. */
    float minValue[10]; /* The minimum value of each criterion based on the used database. */
    float dataArray[3][50]; /* The arrays holding only values of a particular criterion of the
                               assets in the portfolio: [0][i] holds the return, [1][i] holds
                               the risk, [2][i] holds the reputation of each asset. */

    fit[0] = 0;
    fit[1] = 0;
    for (int i = 0; i < numberOfGenes; i++)
    {
        // Create an array containing only values of a particular criterion for the assets
           in the portfolio.
        dataArray[0][i] = newList[i].returnOfAsset;
        dataArray[1][i] = newList[i].riskOfAsset;
        dataArray[2][i] = newList[i].reputationOfAsset;
```

```
    }

    // Find the maximum and the minimum value of each criterion of the assets in the portfolio.
       They will be used in the utility functions.
    for(int i = 0; i < 3; i++)
    {
        maxValue[i] = findMax(dataArray[i], numberOfGenes);
        minValue[i] = findMin(dataArray[i], numberOfGenes);
    }

    for(int i = 0; i < populationSize; i++)
    {
        // Calcuate the fitness of each individual in the population.
        fitness(newList, population[i], shapley, interaction, numberCriteria, numberOfGenes,
                fit, choquet, minValue, maxValue, dataArray[0], dataArray[1], dataArray[2]);
        // Update the array of fitnesses with the fitness of the i-th individual.
        fitnesses[i][0] = fit[0];
        fitnesses[i][1] = fit[1];
    }
}

void sortFitnesses(float population[][50], float fitnesses[][2], int populationSize,
     int numberOfGenes)
/* Sorts the array that contains the fitness of each individual. */
/* The degree of preference is used to compare intervals. */
/* The bubble sort algorithm is used. */
/* Inputs: the current population (i.e., the portfolios);
           the array containing the fitness of each individual;
           the size of the population;
           the number of genes (i.e., the number of assets in the portfolio. */
/* Output: none. */
/* Side output: the sorted array of fitnesses. */
{
```

```
    float temp[50];
    float tempFitness[2];
    int degree; /* The degree of preference. */

    for(int i = 0; i < (populationSize-1); i++)
        for(int j = (i+1); j < populationSize; j++)
        {
            degree = intervalComparison(fitnesses[i], fitnesses[j]);
            if (degree == 0) /* The second interval is better. */
            {
                /* Switch the intervals. */
                for(int k = 0; k < numberOfGenes; k++)
                {
                    temp[k] = population[i][k];
                    population[i][k] = population[j][k];
                    population[j][k] = temp[k];
                }
                tempFitness[0] = fitnesses[i][0];
                tempFitness[1] = fitnesses[i][1];
                fitnesses[i][0] = fitnesses[j][0];
                fitnesses[i][1] = fitnesses[j][1];
                fitnesses[j][0] = tempFitness[0];
                fitnesses[j][1] = tempFitness[1];
            }
        }
}

void createIndividual(float individual[], int numberOfGenes, float lowerBound, float upperBound)
/* Randomly creates an individual. */
/* The i-th dimension of the individual represents the weight (i.e., the allocation of wealth)
   assigned to the i-th asset. */
/* Inputs: the number of genes;
           the lower and the upper bound of the values a gene can take. */
```

```
/* Output: none. */
/* Side output: the distribution of wealth in the 'individual'. */
{
    float temp;

    for(int i = 0; i < numberOfGenes; i++)
    {
        temp = rand();

        // Scale the random value to the interval [0,1].
        temp = (temp * (upperBound-lowerBound)) / 32767;
        temp = temp + lowerBound;

        individual[i] = temp;
    }
}

void createLowerExtremeIndividual(float individual[], int numberOfGenes, float lowerBound)
/* Creates an individual whose all genes are equal to the lowest possible value that
   a gene can take. */
/* Inputs: the number of genes in an individual;
           the lower bound of the value a gene can take. */
/* Output: none. */
/* Side output: the individual whose all genes are equal to the lower bound. */
{
    for(int i = 0; i < numberOfGenes; i++)
        individual[i] = lowerBound;
}

void createUpperExtremeIndividual(float individual[], int numberOfGenes, float upperBound)
/* Creates an individual whose all genes are equal to the highest possible value that
   a gene can take. */
/* Inputs: the number of genes in an individual;
```

```
          the upper bound of the value a gene can take. */
/* Output: none. */
/* Side output: the individual whose all genes are equal to the upper bound. */
{
    for(int i = 0; i < numberOfGenes; i++)
        individual[i] = upperBound;
}

void createEquallyWeightedExtremeIndividual(float individual[], int numberOfGenes)
/* Creates an individual whose all genes are equal, and the sum of the genes is equal to 1. */
/* Inputs: the number of genes in an individual. */
/* Output: none. */
/* Side output: the individual whose all genes are all equal, and their sum is equal to 1. */
{
    for(int i = 0; i < numberOfGenes; i++)
        individual[i] = (1.0 / numberOfGenes);
}

void initialPopulation(float population[][50], int populationSize, int numberOfGenes,
     float lowerBound, float upperBound)
/* (Almost) randomly generates the initial population:
   the first three individuals are forced to be the three extreme individuals;
   all the other individuals are generated randomly. */
/* Inputs: the population size;
          the number of genes in an individual;
          the lower and the upper bounds of the value a gene can take. */
/* Output: none. */
/* Side output: the initial population 'population'. */
{
    createLowerExtremeIndividual(population[0], numberOfGenes, lowerBound);
    createUpperExtremeIndividual(population[1], numberOfGenes, upperBound);
    createEquallyWeightedExtremeIndividual(population[2], numberOfGenes);
    for(int i = 3; i < populationSize; i++) /* Randomly create the remaining individuals. */
```

```
        createIndividual(population[i], numberOfGenes, lowerBound, upperBound);
}


int poolForReproduction(float pool[][50], float population[][50], int populationSize,
    int numberOfGenes, int tenPercent, int fivePercent)
/* Increases the pool of individulas for reproduction by creating:
   five extra individuals for each of the top 10% individuals;
   three extra individuals for the next 5%;
   one extra individual for the next 5%. */
/* This way, the highest probability to be selected is given to top 10% of the population. */
/* Inputs: the 'population' array;
           the size of the population;
           the number of genes;
           the size of the 10% of the populatin;
           the size of the 5% of the population. */
/* Output: the size of the pool for reproduction. */
/* Side output: the array containing the individuals in the pool for reproduction. */
{
    int poolSize;

    poolSize = populationSize;

    for(int i = 0; i < populationSize; i++) /* Copy all the individuals from the population
                                               to the pool. */
        for(int j = 0; j < numberOfGenes; j++)
            pool[i][j] = population[i][j];

    for(int k = 0; k < 5; k++) /* Create 5 extra individuals for each individual in top 10%. */
        for(int i = 0; i < tenPercent; i++)
        {
            for(int j = 0; j < numberOfGenes; j++)
                pool[poolSize][j] = population[i][j];
            poolSize++;
```

```
        }

    for(int k = 0; k < 3; k++) /* Create 3 extra individuals for each individual in next 5%. */
        for(int i = 0; i < fivePercent; i++)
        {
            for(int j = 0; j < numberOfGenes; j++)
                pool[poolSize][j] = population[i+tenPercent][j];
            poolSize++;
        }

    for(int i = 0; i < fivePercent; i++) /* Create 1 extra individual for each individual
                                            in next 5%. */
    {
        for(int j = 0; j < numberOfGenes; j++)
            pool[poolSize][j] = population[i+tenPercent+fivePercent][j];
        poolSize++;
    }

    return poolSize;
}

int selection(int size)
/* Randomly selects an integer value in the interval [0, size-1]*/
/* Input: the size from which to select an integer. */
/* Output: the selected integer. */
/* Side output: none. */
{
    float temp;
    int position;

    temp = rand();
    temp = (temp/32767) * (size - 1);
    position=(int) temp;
```

```
    return position;
}


void crossover(float pool[][50], float population[][50], int populationSize, int numberOfGenes,
     int position1, int position2, int tenPercent, int fivePercent)
/* Performs the crossover step between two given individuals. */
/* Inputs: the array of individuals in the pool for reproduction;
           the array containing the individuals in the population;
           the population size;
           the number of genes;
           the positions of two individuals in the pool that are chosen for crossover;
           the number of individuals in the 10% and in the 5% of the population. */
/* Output: none. */
/* Side output: the population updated with new individuals generated by crossover. */
{
    int position;
    float temp;
    float tempIndividual1[50], tempIndividual2[50];

    for(int j = 0; j < numberOfGenes; j++)
    {
        temp = rand();
        temp = temp/32767; /* Select a random value in the interval [0,1]. */

        tempIndividual1[j] = temp * pool[position1][j] + (1-temp) * pool[position2][j];
        tempIndividual2[j] = temp * pool[position2][j] + (1-temp) * pool[position1][j];
    }

    // Update the population with the first individual generated.
    // Make sure that the position being updated is within the boundary of the population
       (not the pool).
    if (position1 < populationSize) position = position1;
```

```
    else
        if (position1 < (populationSize + tenPercent * 5))
           position = (position1 - populationSize) % tenPercent;
        else if (position1 < (populationSize + tenPercent * 5 + fivePercent * 3))
                position = (position1 - populationSize - tenPercent*5)%fivePercent + tenPercent;
             else position1 = (position1 - populationSize - tenPercent * 5 - fivePercent * 3) +
                                  tenPercent + fivePercent;

    for(int i = 0; i < numberOfGenes; i++)
        population[position][i] = tempIndividual1[i];

    // Update the population with the second individual generated.
    if (position2 < populationSize) position = position2;
    else
        if (position2 < (populationSize + tenPercent * 5))
            position = (position2 - populationSize) % tenPercent;
        else if (position2 < (populationSize + tenPercent * 5 + fivePercent * 3))
                 position = (position2 - populationSize - tenPercent*5)%fivePercent + tenPercent;
             else position = (position2 - populationSize - tenPercent * 5 - fivePercent * 3) +
                                 tenPercent + fivePercent;

    for(int i = 0; i < numberOfGenes; i++)
        population[position][i] = tempIndividual2[i];
}

void crossoverWithExtremes(float population[][50], int numberOfGenes,
     int iterationsForExtremeIndividuals, int populationSize)
/* Performs the crossover step of a randomly selected individual and the lower extreme
   individual. */
/* Inputs: the array of individuals in the population;
           the number of genes;
           the number of crossover iterations to perform with the lower extreme individual;
           the population size. */
```

```
/* Output: none. */
/* Side output: the population updated with new individuals generated by crossover. */
{
    int individual;
    int counter = 3; /* New individuals start at the position 3 in the population. */
    float temp;

    for(int i = 0; i < iterationsForExtremeIndividuals; i++)
    {
        individual = selection(populationSize); /* Generate a random individual for crossover. */
        for(int j = 0; j < numberOfGenes; j++)
        {
            temp = rand();
            temp = temp/32767; /* Scale the randomly chosen number to the interval [0,1]. */

            // Create two new individuals.
            population[counter][j] = temp*population[0][j]+(1-temp)*population[individual][j];
            population[counter+1][j] = temp*population[individual][j]+(1-temp)*population[0][j];
        }
        counter += 2;
    }
}

void crossoverWithEquallyWeightedExtreme(float population[][50], int numberOfGenes,
     int iterationsForExtremeIndividuals, int populationSize)
/* Performs the crossover step of a randomly selected individual and the equally weighted
   extreme individual. */
/* Inputs: the array of individuals in the population;
           the number of genes;
           the number of crossover iterations to perform with the extreme individual;
           the population size. */
/* Output: none. */
/* Side output: the population updated with new individuals generated by crossover. */
```

```
{
    int individual;
    int counter = 33; /* New individuals start at the position 33 in the population. */
    float temp;

    for(int i = 0; i < iterationsForExtremeIndividuals; i++)
    {
        individual = selection(populationSize); /* Generate a random individual for crossover. */
        for(int j = 0; j < numberOfGenes; j++)
        {
            temp = rand();
            temp = temp/32767; /* Scale the randomly chosen number to the interval [0,1]. */

            // Create two new individuals.
            population[counter][j] = temp*population[2][j]+(1-temp)*population[individual][j];
            population[counter+1][j] = temp*population[individual][j]+(1-temp)*population[2][j];
        }
        counter += 2;
    }
}

void mutation(int numberOfGenes, int individual, float population[][50], float lowerBound,
     float upperBound)
/* Performs the mutation step of a randomly selected individual. */
/* Inputs: the number of genes;
           the position in the population of the individual selected for mutation;
           the array of population;
           the lower and the upper bounds of the values that a gene can take. */
/* Output: none. */
/* Side output: the population updated with the mutated individual. */
{
    int genePosition;
    float temp;
```

```
    // Randomly select the gene to mutate.
    genePosition = selection(numberOfGenes);

    // Randomly generate the new value of the selected gene.
    temp = rand();
    temp = (temp * (upperBound - lowerBound)) / 32767;
    temp = temp + lowerBound;

    // Update the population with the mutated gene.
    population[individual][genePosition] = temp;
}

void optimization(int numberOfGenes, float bestIndividual[50], float bestFitness[2],
     struct NewAsset newList[], float shapley[][2], float interaction[][10][2],
     int numberCriteria, float choquet[][2], struct ReturnsOnly nextYearReturns[])
/* Calculates the optimal portfolio and the return of the portfolio. */
/* Improved genetic algorithm is used as the optimization technique. */
/* Inputs: the number of genes;
           the current best distribution 'bestIndividual';
           the current best fitness 'bestFitness';
           the array of data of the assets in the portfolio 'newList';
           the Shapley values;
           the interaction indices of order 2;
           the number of criteria;
           the number of assets in the portfolio;
           the array of the Choquet values of the assets;
           the array of the actual returns next year (used for testing purposes only). */
/* Output: none. */
/* Side output: the optimal portfolio;
                the return of the optimal portfolio (on average). */
{
    const int numberOfIterations = 1000; /* The number of iterations for selection, crossover,
```

```
                                        and mutation. */
const int populationSize = 500;
const int iterationsForExtremeIndividuals = 15; /* The extreme individuals are crossovered
                        with other individuals before regular selection for crossovers. */
const float lowerBound = -0.1; /* The lower bound of genes. Some short selling is allowed,
                                  but not much. */
const float upperBound = 0.4; /* The upper bound of genes. Not more than 40% of total wealth
                                 is allowed to be invested in an asset. */
const float mutationRate = 0.05;
float population[500][50]; /* The second dimension is the number of the assets in
                              a portfolio. */
float pool[950][50]; /* The population is increased to give a higher probability of
                        selection to the individuals with a higher fitness. */
float fitnesses[950][2]; /* The fitness of each individual in the pool expressed as
                            an interval. */
int sizeOfPool; /* The size of the pool for reproduction. */
float ten, five; /* Temporary variables. */
int tenPercent, fivePercent; /* The number of individuals in 10% and 5% of the population. */
int individual1, individual2; /* Temporary variables. */
int numberOfIndividualsToMutate; /* The number of individuals selected for mutation.
                                    It depends on the size of the population. */
int degree; /* The degree of preference when comparing intervals. */
int tempSRand; /* Used to generate random initial population. */
float tempReturns[50]; /* Used to hold the return of the portfolio produced by
                          an optimization. */
float returnOfPortfolio; /* The average of tempReturns. It is the return of the optimal
                            portfolio (on average). */
float sum = 0.0; /* Temporary variable. */
int testCases; /* The number of running genetic algorithm to obtain an optimal portfolio.
                  Used only for testing purposes to calculate the return of the optimal
                  portfolio on average. */

testCases = 15;
```

```
tempSRand = 1000;
for(int i = 0; i < testCases; i++)
    tempReturns[i] = 0; /* tempReturns[i] holds the actual return of each run of the
                           optimization algorithm. */

for(int k = 0; k < testCases; k++) /* Run the algorithm 'testCases' times to find the return
                                      on average. It is used for more accurate testing. */
{
    tempSRand += 175;
    srand(tempSRand); /* Generate a new seed for each run of random initialization of
                         the population. */

    // Initialization of the population.
    initialPopulation(population, populationSize, numberOfGenes, lowerBound, upperBound);

    crossoverWithExtremes(population, numberOfGenes, iterationsForExtremeIndividuals,
                          populationSize);
    crossoverWithEquallyWeightedExtreme(population, numberOfGenes,
                          iterationsForExtremeIndividuals, populationSize);

    for(int i = 0; i < populationSize; i++)
    {
        fitnesses[i][0] = 0;
        fitnesses[i][1] = 0;
    }

    calculateFitnessOfEachIndividual(population, fitnesses, populationSize, numberOfGenes,
                                     newList, shapley, interaction, numberCriteria, choquet);
    sortFitnesses(population, fitnesses, populationSize, numberOfGenes);

    // Calculate the fitness of the best individual and save the fitness and the genes of
       the best individual. The algorithm is using the elicist strategy.
    for(int i = 0; i < numberOfGenes; i++)
```

```
    bestIndividual[i] = population[0][i];
bestFitness[0] = fitnesses[0][0];
bestFitness[1] = fitnesses[0][1];

// Calculate the size of 10% and 5% of the population.
ten = 0.1 * populationSize;
tenPercent = (int) ten;
five = 0.05 * populationSize;
fivePercent = (int) five;

for(int iteration = 0; iteration < numberOfIterations; iteration++)
{
    sizeOfPool = poolForReproduction(pool, population, populationSize, numberOfGenes,
                 tenPercent, fivePercent);

    // Crossover.
    for(int i = 0; i < (populationSize/2); i++)
    {
        individual1 = selection(sizeOfPool);
        individual2 = selection(sizeOfPool);
        crossover(pool, population, populationSize, numberOfGenes, individual1,
                  individual2, tenPercent, fivePercent);
    }

    // Mutation.
    numberOfIndividualsToMutate = (int) (mutationRate * sizeOfPool);
    for(int i = 0; i < numberOfIndividualsToMutate; i++)
    {
        individual1 = selection(populationSize);
        mutation(numberOfGenes, individual1, population, lowerBound, upperBound);
    }

    // Calculate and sort new fitnesses.
```

```
    calculateFitnessOfEachIndividual(population, fitnesses, populationSize,
          numberOfGenes, newList, shapley, interaction, numberCriteria, choquet);
    sortFitnesses(population, fitnesses, populationSize, numberOfGenes);

    degree = intervalComparison(fitnesses[0], bestFitness);
    if (degree == 0) /* The previous 'bestFitness' is better than the best fitness of
           the new population, so the previous best is restored in the population. */
    {
        for(int i = (populationSize-1); i > 0; i--)
            for(int j = 0; j < numberOfGenes; j++)
                population[i][j] = population[i-1][j];
        for(int i = (populationSize-1); i > 0; i--)
            for(int j = 0; j < 3; j++)
                fitnesses[i][j] = fitnesses[i-1][j];

        for(int i = 0; i < numberOfGenes; i++)
            population[0][i] = bestIndividual[i];
        for(int i = 0; i < 2; i++)
            fitnesses[0][i] = bestFitness[i];
    }
    else /* Update the values of the current best individual and the best fitness. */
    {
        bestFitness[0] = fitnesses[0][0];
        bestFitness[1] = fitnesses[0][1];
        for(int i = 0; i < numberOfGenes; i++)
            bestIndividual[i] = population[0][i];
    }
} /* The end of the iterations of the genetic algorithm. */

for(int m = 0; m < numberOfGenes; m++) /* Calculate the actual return of the portfolio
                                           in the particular iteration. */
    tempReturns[k] +=
    bestIndividual[m]*nextYearReturns[newList[m].originalPosition].returnOfAsset;
```

```
    } /* The end of repetitions 'testCases' times. */

    for(int i = 0; i < testCases; i++)
        sum += tempReturns[i];
    returnOfPortfolio = sum / testCases; /* The accurate return of the portfolio (on average). */
    cout << "The return of the optimal portfolio: " << returnOfPortfolio << "\n";
}
```

Appendix C. Results of Testing.

The proposed algorithm was tested against three different benchmarks that were described in the section 7. Also, three different portfolios were created containing 10, 20, and 30 assets, respectively. Each of the portfolios was tested using three different sets of the Shapley values, representing three different behaviors of investors. The first set of the Shapley values represents an individual that equally cares about all three considered characteristics. The second set of the Shapley values represents an individual that considers the return and the risk of an asset equally important but does not care much about the reputation of a company. Finally, the third set of the Shapley values represents an investor for whom the return is the most important, the risk is much less important than the return, and the reputation has very little importance.

The table 15 shows the returns of a ten-assets portfolio for each of the three sets of Shapley values in the year 2005 as well as the returns of each benchmark portfolio in these cases. Similarly, the tables 16 and 17 show the same results for a 20-assets and a 30-assets portfolio, respectively. Finally, the tables 18 – 20 and 21 – 23 show the results of performing the same tests in the years 2006 and 2007, respectively.

Table 15: Test results for the year 2005 for a portfolio with 10 assets

Shapley values	Return	[0.7-0.9]	[0.7-0.9]	[0.8-0.9]
	Risk	[0.7-0.9]	[0.7-0.9]	[0.4-0.6]
	Reputation	[0.7-0.9]	[0.1-0.3]	[0.2-0.3]
Returns	Our portfolio	0.01295050	0.00063347	-0.00294310
	Benchmark 1	-0.01479010	-0.01479010	-0.01479010
	Benchmark 2	-0.00019587	-0.00778515	-0.00850107
	Benchmark 3	-0.00367453	-0.00367453	-0.00367453

Table 16: Test results for the year 2005 for a portfolio with 20 assets

Shapley values	Return	[0.7-0.9]	[0.7-0.9]	[0.8-0.9]
	Risk	[0.7-0.9]	[0.7-0.9]	[0.4-0.6]
	Reputation	[0.7-0.9]	[0.1-0.3]	[0.2-0.3]
Returns	Our portfolio	-0.00197000	0.00998300	0.01048000
	Benchmark 1	-0.01479010	-0.14790100	-0.01479010
	Benchmark 2	-0.00843885	-0.00815986	-0.00730711
	Benchmark 3	-0.00607422	-0.00607422	-0.00607422

Table 17: Test results for the year 2005 for a portfolio with 30 assets

Shapley values	Return	[0.7-0.9]	[0.7-0.9]	[0.8-0.9]
	Risk	[0.7-0.9]	[0.7-0.9]	[0.4-0.6]
	Reputation	[0.7-0.9]	[0.1-0.3]	[0.2-0.3]
Returns	Our portfolio	-0.00941000	0.00230300	0.00469200
	Benchmark 1	-0.01479010	-0.01479010	-0.01479010
	Benchmark 2	-0.00956979	-0.00864094	-0.00586865
	Benchmark 3	-0.00783090	-0.00783090	-0.00783090

Table 18: Test results for the year 2006 for a portfolio with 10 assets

Shapley values	Return	[0.7-0.9]	[0.7-0.9]	[0.8-0.9]
	Risk	[0.7-0.9]	[0.7-0.9]	[0.4-0.6]
	Reputation	[0.7-0.9]	[0.1-0.3]	[0.2-0.3]
Returns	Our portfolio	0.06931000	0.05366400	0.06665300
	Benchmark 1	0.05259360	0.05259360	0.05259360
	Benchmark 2	0.05254120	0.05835060	0.05835060
	Benchmark 3	0.06506960	0.06506960	0.06506960

Table 19: Test results for the year 2006 for a portfolio with 20 assets

Shapley values	Return	[0.7-0.9]	[0.7-0.9]	[0.8-0.9]
	Risk	[0.7-0.9]	[0.7-0.9]	[0.4-0.6]
	Reputation	[0.7-0.9]	[0.1-0.3]	[0.2-0.3]
Returns	Our portfolio	0.22275000	0.20569400	0.22718600
	Benchmark 1	0.05259360	0.05259360	0.05259360
	Benchmark 2	0.05810980	0.05438740	0.05438740
	Benchmark 3	0.06068870	0.06068870	0.06068870

Table 20: Test results for the year 2006 for a portfolio with 30 assets

Shapley values	Return	[0.7-0.9]	[0.7-0.9]	[0.8-0.9]
	Risk	[0.7-0.9]	[0.7-0.9]	[0.4-0.6]
	Reputation	[0.7-0.9]	[0.1-0.3]	[0.2-0.3]
Returns	Our portfolio	0.33893500	0.32463600	0.32458100
	Benchmark 1	0.05259360	0.05259360	0.05259360
	Benchmark 2	0.05674090	0.05300470	0.05300470
	Benchmark 3	0.05715990	0.05715990	0.05715990

Table 21: Test results for the year 2007 for a portfolio with 10 assets

Shapley values	Return	[0.7-0.9]	[0.7-0.9]	[0.8-0.9]
	Risk	[0.7-0.9]	[0.7-0.9]	[0.4-0.6]
	Reputation	[0.7-0.9]	[0.1-0.3]	[0.2-0.3]
Returns	Our portfolio	0.10011000	0.08723500	0.10318300
	Benchmark 1	0.09012640	0.09012640	0.09012640
	Benchmark 2	0.08621640	0.08036320	0.08761540
	Benchmark 3	0.07017940	0.07017940	0.07017940

Table 22: Test results for the year 2007 for a portfolio with 20 assets

Shapley values	Return	[0.7-0.9]	[0.7-0.9]	[0.8-0.9]
	Risk	[0.7-0.9]	[0.7-0.9]	[0.4-0.6]
	Reputation	[0.7-0.9]	[0.1-0.3]	[0.2-0.3]
Returns	Our portfolio	0.28962700	0.32878500	0.33978100
	Benchmark 1	0.09012640	0.09012640	0.09012640
	Benchmark 2	0.07546530	0.08761260	0.08761260
	Benchmark 3	0.07756360	0.07756360	0.07756360

Table 23: Test results for the year 2007 for a portfolio with 30 assets

Shapley values	Return	[0.7-0.9]	[0.7-0.9]	[0.8-0.9]
	Risk	[0.7-0.9]	[0.7-0.9]	[0.4-0.6]
	Reputation	[0.7-0.9]	[0.1-0.3]	[0.2-0.3]
Returns	Our portfolio	0.50538900	0.52052500	0.50036200
	Benchmark 1	0.09012640	0.09012640	0.09012640
	Benchmark 2	0.08137220	0.08106470	0.08170970
	Benchmark 3	0.07180490	0.07180490	0.07180490

Curriculum Vita

Tanja Magoč was born in Novi Sad, Serbia, where she graduated from a high school majoring in mathematics and computer science in May of 1996. She entered The University of Texas at El Paso in the fall of 1996 on a tennis scholarship, and earned bachelor's degree in Mathematics with a minor in Computer Science in December 2000.

In spring of 2001, Tanja Magoč entered the Graduate School at The University of Texas at El Paso, and earned master's degree in mathematics in July 2002. She earned both B.S. and M.S. degrees receiving the Outstanding Student in the Department of Mathematical Sciences award. During her graduate studies, she worked as a Teaching Assistant in the Department of Mathematical Sciences, where upon graduation, she worked for five years as a Lecturer and the last three years also as the Director of Modular Precalculus Program.

In Fall 2006, while still working for another year as a lecturer, Tanja Magoč returned to the Graduate School at The University of Texas at El Paso to obtain a Ph.D. in Computer Science. While working on her doctorate degree, she worked as a Teaching Assistant and, in the last semester, as an Instructor in the Department of Computer Science.

Tanja Magoč attended several international research conferences where she presented her work. In collaboration with several faculty members, she published several journal papers and a book chapter in Springer published book on Foundations on Computational Intelligence.

Permanent address: Kace Dejanovic 26
21000 Novi Sad
Serbia
Europe

This dissertation was typed by Tanja Magoč.