Imperial College London

Department of Computing

Option Pricing with Linear Programming

by

Napat Rujeerapaiboon

Supervisor: Dr. Daniel Kuhn

Second Marker: Dr. Panos Parpas

Submitted in partial fulfilment of the requirements for the MSc Degree in Computing Science (Computational Management) of Imperial College London

September 2012

Abstract

Robust optimisation paradigm is a technique enabling problem modellers or decision makers to make an optimal decision under uncertainty. Unlike stochastic programming approach which relies on the knowledge of probability distributions of uncertain parameters, robust optimisation framework does not require such knowledge. Instead, it views each random parameter as a point moving freely in the prescribed uncertainty set. During the past few decades, robust optimisation has attracted a lot of attention from optimisation researchers as it seems to have techniques to transform an uncertain robust optimisation problem to an equivalent tractable uncertainty-free optimisation problem. In the sense of tractability, the robust optimisation approach is thus deemed superior to the stochastic programming approach, which is usually claimed to be intractable when the problem considered involves multiple decision stages. However, traditional robust optimisation approaches tend to neglect recourse possibilities, which in many cases makes the robust paradigm criticised for excessive conservatism. A recent technique to alleviate this shortcoming of the robust optimisation while preserving its tractability is to embed decision rules in the optimisation problem. Decision rules are used to characterise a decision variable as a function of previously observed information. It has been proven that some specific functional forms of decision rules, for example, a linear function or a piecewise linear function, still lead to tractable formulation of the optimisation problem under uncertainty. The employment of decision rules appears in both stochastic programming approach and robust optimisation approach.

In this project, we studied the robust optimisation framework and applied it to the option pricing problem. Option is a kind of contract; it is a financial instrument, and it has monetary value. Determination of a fair price of a given option is no trivial. Black-Scholes model is a very significant piece of work in this area. It provides a closed-form formula for pricing European options. The Black-Scholes model, however, relies on quite a lot of assumptions that usually do not hold in the real market. That being said, what is left behind the Black-Scholes model is the idea of finding a replicating portfolio which is dynamically rebalanced and matches the option payoff in every future scenario at the option's expiration date. If such a portfolio exists, then under the arbitrage-free assumption, the initial value of the portfolio and the price of the option must be equal. A perfect replicating portfolio nevertheless may not exist. Chen [15] then formulates the option pricing problem as a robust optimisation problem. The output of his model is a portfolio that matches most closely to the option payoff in the worst-case scenario allowed by a predefined uncertainty set. We make several contributions in this work. First, we provide an analysis of Chen's pricing model. Second, we propose a new way to formulate a robust option pricing problem which is in fact identical to Chen's model but ours is simpler and more intuitively understandable. Third, we prove that by using linear decision rules and piecewise linear decision rules the pricing model is at most as conservative as the original robust pricing model; the proof can obviously be applied to other applications apart from option pricing. Fourth, when an option considered is tied to multiple assets, we propose a method to use a factor model to provide input to the robust option pricing model, which makes the results significantly less conservative. Fifth, we develop a robust pricing model from both option writer's and option buyer's perspectives.

Keywords: Robust optimisation, Convex optimisation, Linear decision rule, Piecewise linear decision rule, Factor model, Option pricing.

Acknowledgements

I would like to express my sincere gratitude to Dr. Daniel Kuhn, my project supervisor, for all clarifications, discussions, and comments throughout this project. I could not imagine how hard it would be to carry out this project without his support in various topics; the great quality of this thesis was significantly contributed by him. My thanks also go to Angelos Georghiou, a Ph.D. student under Dr. Kuhn's supervision, for help and guidance he offered me during the project.

I would also like to thank Dr. Panos Parpas for agreeing to be my second marker and for his constructive comments.

Finally, I would like to thank my family and friends from my home country, especially my elder brother Pattra Rujeerapaiboon and my dear friend Sanprapha Vudhivorn, for being constantly supportive from a distance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction & Problem Statement

Financial options are tradable contracts tied to one or more assets. They enable investors to develop a strategy for speculative or hedging purposes in a way that the underlying asset(s) cannot offer. Many investors prefer, therefore, to invest in options rather than assets. In fact, sometimes the amount of money invested in the assets is lower than the amount of money invested in the corresponding options. As a result, a considerable amount of research effort has been put into investigating options. One of the fundamental questions is what the fair price of a given option should be. This branch of study is usually referred to as *financial option valuation.*

The Black-Scholes model (see Shah [40]) and the binomial options pricing model are two examples of the wide range of valuation techniques which have been suggested, but all are still imperfect. Some models rely on excessively restrictive fundamental assumptions, while others are computationally demanding. Black-Scholes formula is a closed-form option pricing equation invented by Black, Merton and Scholes. In their model, asset's price is considered a continuous process, and the price of an option is determined by replicating its payoff through a self-rebalancing portfolio. Despite the great success of the Black-Scholes model, it still appears to lack flexibility in pricing options with complicated payoff or those offering flexible exercising policy, even if the model's assumptions are all satisfied. So as to address this inflexibility, one can alternatively use binomial options pricing model, which is a discrete analogous of the Black-Scholes model. However, using the latter model, the problem size grows exponentially with the number of time periods, which limits scalability. Furthermore, even though at present a number of types of the options are being traded in the market, it may be the case that, in the future, even more complicated options will be introduced and, unfortunately, current models do not seem to be flexible enough to be applied to such novel developments.

Motivated by its tractability, robust linear optimisation model has attracted a great deal of interest from optimisation researchers (see, for example, Ben-Tal, Goryashko, Guslitzer and Nemirovski [4], Ben-Tal and Nemirovski [6], Bertsimas, Pachamanova and Sim [9], and Bertsimas and Sim [10]). Chen [15] has proposed a way to use this method to price options. A nice feature of this model is that, since the pricing problem is reduced to solving an optimisation problem, it does not require any assumption about the market. Moreover, previous pricing models typically output a single price for a given option whereas, in the robust pricing model, the investor is able to set a preferred level of risk-aversion by specifying an acceptable degree of uncertainty in, for instance, future price of the underlying asset. The robust pricing model is therefore capable of producing a family of prices corresponding to adjustments in this risk-aversion parameter.

The purpose of this project is to see how well Chen's model works on both well-understood examples, like European and American options, and more exotic types of option. In the first part of this report, an optimisation model for option pricing based on Chen's work is described and then derived. Chen's model is a robust optimisation problem with linear constraints and polyhedral uncertainty set, which makes the model simple and accurately solvable. In the second part, an alternative robust approach to options pricing based on various forms of decision rules is described. Decision rule techniques have recently been developed; they are specially designed to cater for the uncertainty that can arise in optimisation problems. It is often claimed that they can produce more accurate solutions because they are able to capture the dynamic behaviour of decision makers. In addition, possibilities of using the robust pricing model to price an option with more than one underlier are elaborated. Various extensions of the robust pricing model are also described. These include, for example, a super-replication robust pricing model for option writer and a sub-replication robust pricing model for option buyer. In the final part of this report, evaluation of the proposed pricing models is carried out from both theoretical point of view and market's point of view.

This project may be of importance to financial companies responsible for pricing options. It also confirms the significance of optimisation studies by providing significant real-world applications.

1.1 Structure and Contributions of the Thesis

In Chapter 2 and Chapter 3, we review and analyse research in this area. Specifically, we study various option pricing techniques in Chapter 2 including the Black-Scholes model, the binomial options pricing model, and the ϵ-arbitrage robust pricing model. Key idea of each pricing model is presented along with its advantages and disadvantages. In Chapter 3, we discuss modern theories and techniques in optimisation. We briefly present deterministic linear programming problem, the first class of optimisation problems investigated dating back to the era of World War II, and gradually introduce modern developments, especially techniques to handle uncertain data in optimisation problems. The ϵ-arbitrage robust pricing model, as an option pricing model which employs the optimisation technique called robust optimisation, is then thoroughly analysed in Chapter 4.

The main contributions of our work start from Chapter 5 in which we begin by identifying the shortcomings of the ϵ-arbitrage robust pricing model, and subsequently propose a new option pricing model, which is still based on robust optimisation. The new pricing model can be proven to be almost identical to the ϵ-arbitrage robust pricing model, but it is simpler and more intuitively understandable. We also introduce decision rules into this new robust pricing model; we provide some mathematical proofs in this regard to guarantee the performance of the linear decision rules and the piecewise linear decision rules. In Chapter 6, we show how to use the new robust pricing model when the option considered is tied to more than one asset. We discuss how difficult it is to restrict the movements of the underliers' prices in the future stages and propose a way to remedy the problem using a factor model. In Chapter 7, we create separate robust pricing models from both option writer's and option buyer's perspectives. They should be seen as an example of how to formulate pricing problems using different mindsets. Evaluation of the proposed robust pricing models is presented in Chapter 8. Chapter 8 also provides an analysis of the resulting pricing models as compared with the classic delta-hedging model. We end the project by providing a brief summary of what we have done and the possibilities of future research in this area in Chapter 9.

1.2 Notation

The following notations are used in the sequel:

- $A_{m\times n}$ means A is a real $m \times n$ matrix;
- $0_{m\times n}$ denotes an $m \times n$ zero matrix;
- $\mathbb{I}_n$ denotes an $n \times n$ square identity matrix;
- A^T denotes a transpose of matrix A;
- $\mathbb{R}$ and $\mathbb{R}_+$ denote a set of real numbers and a set of non-negative real numbers, respectively;

- $\mathbb{S}^n$ denotes a set of symmetric matrices in $\mathbb{R}^{n \times n}$;
- $\wedge$ denotes a logical and operator;
- $\tilde{\cdot}$ means that the input parameter $(\cdot)$, which can be either scalar, vector, or matrix, is subject to uncertainty;
- $\underline{\cdot}$ denotes an element-wise lower bound of the input parameter $(\cdot)$;
- $\bar{\cdot}$ denotes an element-wise upper bound of the input parameter $(\cdot)$;
- $\mathbb{E}(\cdot)$ denotes an expectation function;
- $\{\cdot\}_{i=\underline{I}}^{\bar{I}}$ denotes a collection of information $(\cdot)$ indexed by i ranging from $\underline{I}$ to $\bar{I}$;
- $(\cdot)^+$ denotes a positive part of the input parameter $(\cdot)$, i.e., $(\cdot)^+ = \max\{\cdot, 0\}$;
- $\phi(\cdot)$ denotes a standard normal distribution;
- $log(\cdot)$ denotes a natural logarithm function;
- $\mathrm{Tr}(\cdot)$ denotes a trace operator;
- $||\cdot||$ denotes a Euclidean norm function;
- $\mathrm{Prob}(\cdot)$ denotes probability of the input $(\cdot)$.

Chapter 2

Literature Review: Option Pricing

In this chapter and the next chapter, we review important academic concepts related to this project from various sources. In particular, we discuss in this chapter financial options and option pricing models. The next chapter in addition provides background knowledge about mathematical optimisation, both deterministic optimisation and optimisation under uncertainty. Some mathematical proofs of the closely-related concepts are provided in these two chapters. However, proofs of well-known theories may not be provided here since they can be found in other literatures.

2.1 Financial Options

In financial market, the term *asset* refers to any object available in the market whose price is exactly known at the present but liable to change in the future (see Higham [25]). Typical examples of the assets are stocks, bonds, and currencies. Financial option, or later on referred to as an option, is a kind of contract that is tied to one or more assets and involves two parties:

1. **Writer** is the party who issues the option. The writer has the responsibility to fulfil the contract if the option is exercised at a valid time by the holder.

2. **Holder** is the party who holds the option, i.e., bought it from the option writer. The option holder has a right to exercise the option, but he or she may as well not exercise the right depending on his or her preference and the market condition.

In typical option contracts, there are two basic components to be specified including:

1. **Strike price** is the price agreed today specifying the price of the underlying asset to be bought or sold at when the option is exercised at a valid future time.

2. **Expiration date** determines the period that the option can be exercised by its holder. Different types of options may have different ways to define such a period. For example, European options can be exercised only at the expiration date while American options can be exercised at any time before the specified expiration date.

Since the action taken on a particular asset is to sell or to buy, options can be divided into two categories depending on the right that the option holder has.

1. **Call option** is the term used to describe the option where its holder has the right to *buy* the underlying asset at the strike price from the option writer.

2. **Put option** is the term used to describe the option where its holder has the right to *sell* the underlying asset at the strike price to the option writer.

The simplest types of options in the market are European call option and European put option. Therefore, analysis of option usually starts with the European options first. As mentioned before, European option is easy to deal with because the option holder can exercise his or her right at only the prescribed time in the future, i.e., the expiration date. Payoff function of a European call option is given by $(S(T) - K)^+ = \max\{S(T) - K, 0\}$, where K denotes the agreed strike price and $S(T)$ (or S_T in discrete-time pricing models) denotes the price of the underlying asset at the expiration date T. From the payoff function it can be observed that the payoff can be zero, which corresponds to the intuition that if the future price of the asset at the expiry is below the strike price, any sensible holder will have no reason to exercise the option. Instead, the option holder can just buy the underlying asset from the market directly. However, in the case that the price of the underlying asset at the expiration date is larger than the strike price, the option holder had better opt to exercise his or her right. By exercising the right and selling the received asset in the market, the option holder can make an immediate profit of $S(T) - K$ with no remaining obligations. Similarly, payoff function of a European put option is given by $(K - S(T))^+ = \max\{K - S(T), 0\}$.

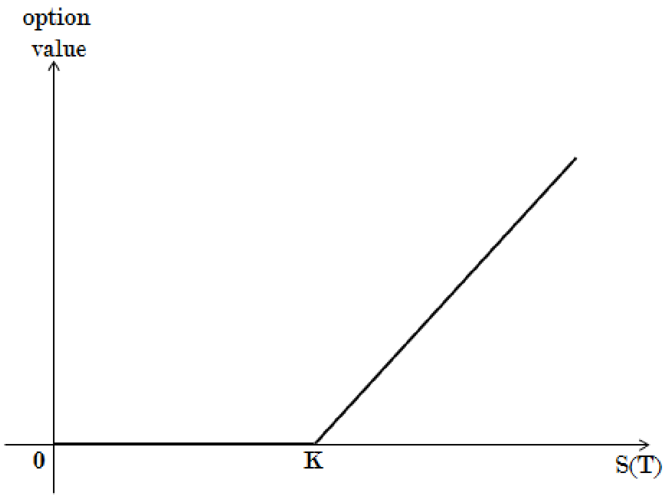

Figure 2.1: Payoff diagram for a European call option

Payoff diagram is a visualisation tool used for describing the option. For example, payoff diagram for a European call option is given by Figure 2.1.

Apart from European options, there are also other types of interesting options, for instance:

1. **American call (put) options** are similar to the European call (put) options. The only difference between them is that the holder of an American option can decide to exercise his or her right to buy the underlying asset from (sell it to) the option writer at any time before the expiration date.

2. **Asian call (put) options** refer to the options whose payoffs depend on the average price of the underlying asset rather than on the price of the asset at one specific time. Because of this characteristic of the Asian options, sometimes people call these options an *average option*. The payoff function of an Asian call option with strike price K and expiration date T and that of an Asian put option with the same strike price and expiration date are given by:

   (a) An Asian call option is described by the payoff function $\max\left\{\frac{1}{T}\int_0^T S(\tau)d\tau - K, 0\right\}$,

   (b) An Asian put option is described by the payoff function $\max\left\{K - \frac{1}{T}\int_0^T S(\tau)d\tau, 0\right\}$.

   Note that one can also define other Asian options by replacing the integral term with another average function, for example, arithmetic average

$$\frac{1}{T}\sum_{t=1}^{T} S_t \tag{2.1}$$

or geometric average

$$\left(\prod_{t=1}^{T} S_t\right)^{\frac{1}{T}}. \tag{2.2}$$

3. **Lookback call (put) options** refer to the options whose payoffs depend on either maximum price or minimum price of the underlying asset before the expiry. There are two subcategories of the lookback options: fixed strike lookback options and floating strike lookback options. We show below the payoff functions for different types of lookback options with strike price K and expiration date T.

   (a) A fixed strike lookback call option is described by the payoff function $\max\{S_{max} - K, 0\}$ where $S_{max} = \max_{t=1,2,\ldots T} S_t$.

   (b) A fixed strike lookback put option is described by the payoff function $\max\{K - S_{min}, 0\}$ where $S_{min} = \min_{t=1,2,\ldots T} S_t$.

   (c) A floating strike lookback call option is described by the payoff function $S_T - S_{min}$ where $S_{min} = \min_{t=1,2,\ldots T} S_t$.

   (d) A floating strike lookback put option is described by the payoff function $S_{max} - S_T$ where $S_{max} = \max_{t=1,2,\ldots T} S_t$.

   Comparing to the corresponding European option, a lookback option is usually considered more valuable since its payoff depend on the optimal price of the underlying asset rather than the price of the underlying asset at the expiration date. Hence, typically the price of the lookback option is higher than that of the corresponding European option.

The Asian Options and the lookback options are usually categorised as *European-style options* because of the similarity between them and the European options in terms of the exercising policy. The American options, on the other hand, provide the holder with a more flexible exercising policy. As a result, it is, from our point of view, more difficult to price American options.

The interesting question about the financial options is that what the **fair price** of a particular option should be. There have been many theories proposed to figure out the price of the option based on various assumptions. The most important assumption is called the *arbitrage-free assumption*. Roughly speaking, the arbitrage-free assumption or the no-arbitrage assumption states that there should be no opportunities for the investors to possibly gain money risklessly without investing his or her own money. It is a reasonable assumption for any financial markets although in reality the arbitrage opportunities do exist. This is because once the investors have noticed the presence of an arbitrage opportunity, all of them will try to seize the opportunity and henceforward it will vanish shortly after that.

Remark 2.1.1. *An arbitrage opportunity is mathematically defined as an investment opportunity that has zero probability of losing money and has a strictly positive probability of getting money. The arbitrage-free opportunity leads to the law of one price stating that the identical assets, i.e., those that yield the same cash flows, must have the same price. Otherwise, one could exploit the gap between the prices to produce the arbitrage opportunity by buying the cheaper one and selling it at, of course, a higher price.*

In this literature review, three models used for pricing options are discussed. The first one is called the *binomial options pricing model*, which directly applies the arbitrage-free assumption to the calculation of the option price. The second one is called the *Black-Scholes model*. The Black-Scholes formula is derived from Ito calculus whose details are not discussed in this report, but briefly Ito calculus can be thought of as an extension of the ordinary calculus specially developed for stochastic processes which are more suitable for describing objects which are subject to uncertainty as time progresses such as stock prices. The last one is called the *ϵ-arbitrage robust pricing model* (the robust pricing model), which basically applies the robust optimisation to solve the pricing problem. We are most interested in the last one since in this project we also aim to use optimisation techniques to solve the pricing problem as well.

2.2 Option Pricing Models

Previously, we have already presented about the definition of the financial option and some fundamental types of the options available in the market. In this section, we aim to focus more on the pricing problem, which is the problem of our interest. We begin by first presenting, from our point of view, the most basic option pricing model, namely the binomial options pricing model, and then we move to discuss the more complicated ones.

2.2.1 Binomial Options Pricing Model

We review the binomial options pricing model from Higham [25] and Luenberger [34]. The binomial options pricing model simply uses the binomial lattice for pricing the options. The model is capable of describing the asset price dynamics in discrete time. Although, in reality we prefer the continuous-time model to the discrete-time model, it is still possible to use the binomial lattice to achieve the time continuity by reducing the time gap between future stages. By that, one can consider the binomial options pricing model a decent model for describing the real price dynamics.

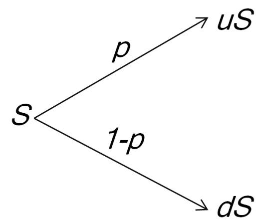

Figure 2.2: Component of the binomial lattice

The fundamental element in the binomial lattice model is shown in Figure 2.2. In the figure, the price either goes up by a factor u or goes down by a factor d. The probability of the price going up is denoted by a variable p. Combining multiple of this basic element together, one can construct a multi-period binomial lattice as shown in Figure 2.3 where $S_{i,j}$ is equal to $u^j d^{i-j} S_0$.

The problem now turns to be how to choose the model parameters, which consist of u, d, and p, appropriately. This process is sometimes called parameter tuning. Indeed, there is more than one way to pick the values for these parameters. Conventional choices of choosing these values are the ways that make the binomial lattice well approximate the geometric Brownian motion, which is one type of the stochastic processes that is widely used in science and engineering. Assuming that the asset price follows this motion, the asset price will be given by

$$S_t = S_0 e^{\nu t + \sigma z(t)}, \tag{2.3}$$

where for a given asset ν denotes the expected growth rate of its logarithm, σ denotes the volatility of the growth rate, and $z(t)$ is the standard Weiner process, it can be proven that the following ways of choosing parameters are reasonable.

- $p = \frac{1}{2} + \frac{1/2}{\sqrt{1 + (\sigma^2/(\nu^2 \Delta t))}}$, $u = e^{\sqrt{\sigma^2 \Delta t + (\nu \Delta t)^2}}$, and $d = e^{-\sqrt{\sigma^2 \Delta t + (\nu \Delta t)^2}}$

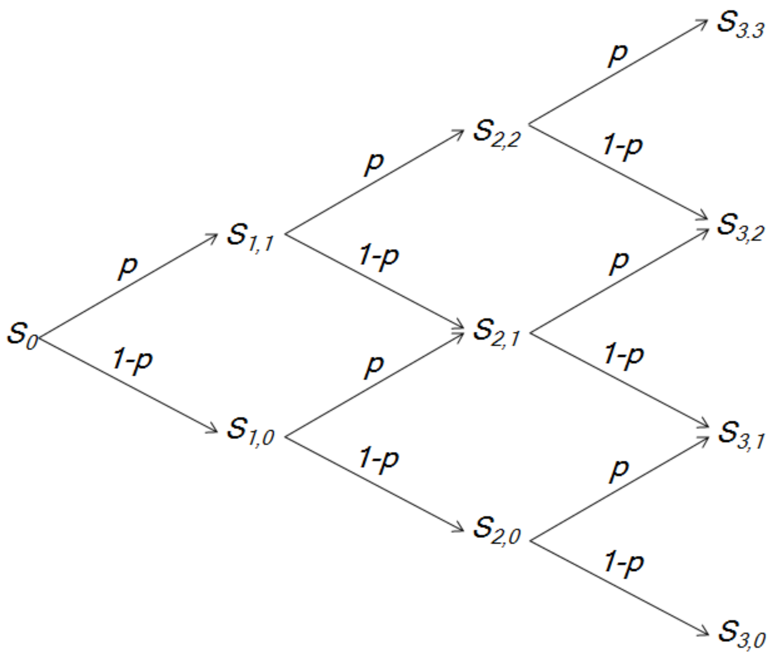

Figure 2.3: Multi-period binomial lattice

- $p = \dfrac{1}{2}$, $u = e^{\sigma\sqrt{\Delta t} + \nu\Delta t}$, and $d = e^{-\sigma\sqrt{\Delta t} + \nu\Delta t}$,

where Δt is the length of one time period.

Note that the typical values of ν and σ are 12% and 15% respectively (see Luenberger [34]). In our experiment, we will use these values for the simulation of our novel pricing approach.

Remark 2.2.1. *A standard Weiner process (or, alternatively, Brownian Motion) is a stochastic process $z(t)$ which is defined continuously on the time interval $[0, T]$ and has the following properties.*

1. *$z(0)$ is deterministic, and its value is zero.*
2. *$z(t_2) - z(t_1) \sim N(0, t_2 - t_1)$, for all $0 \le t_1 < t_2 \le T$, where $N(0, t_2 - t_1)$ is a normal distribution with mean equal to zero and variance equal to $t_2 - t_1$.*
3. *$z(t_2) - z(t_1)$ and $z(t_4) - z(t_3)$ are independent, for all $0 \le t_1 < t_2 \le t_3 < t_4 \le T$*

Consider the problem of finding a fair price of a given option. The idea is to find a replicating portfolio which has to match the payoff of the option of our interest in every future scenario. Typically, the replicating portfolio is constructed from a set of better-understood securities, for example, a stock and a risk-free asset, so that the exact price of the replicating portfolio can be readily calculated. Once such a portfolio has been obtained, using the arbitrage-free assumption, it can be concluded that the price of the option and the price of the replicating portfolio must be the same.

Remark 2.2.2. *Assets in the financial market can be classified as risk-free (or riskless) assets and risky assets. A risk-free asset has a deterministic return while a risky asset has a non-deterministic return. A typical example of risk-free assets is treasury bond and a typical example of risky assets is stock.*

Suppose that the risk-free rate is R. The lattice for the underlying asset can be constructed by tuning the parameters as described earlier, and the lattice for the risk-free asset is clearly easy to obtain. The bottom lattice in Figure 2.4 is for the financial option. The rightmost points in the lattice can be calculated as the payoff of the option can be uniquely determined from the realised

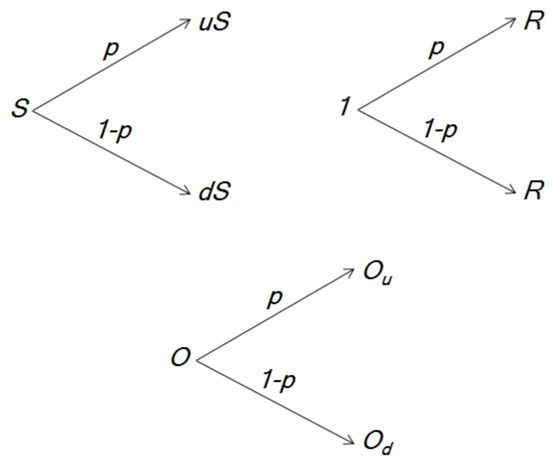

Figure 2.4: Finding replicating portfolio using binomial lattice

price of the underlying asset and the predefined strike price. One can find a replicating portfolio consisting of the underlying asset and the risk-free asset that perfectly matches the option payoff, i.e., the portfolio whose return is O_u and O_d in both scenarios, and conclude that the price of the option is the same as that of the portfolio. Alternatively, one can use the risk-neutral pricing formula below.

$$O = \frac{1}{R}\{qO_u + (1-q)O_d\}, \quad q = \frac{R-d}{u-d} \tag{2.4}$$

Proof.

Consider a portfolio P consisting of $\dfrac{O_u - O_d}{S(u-d)}$ units of the underlying asset and $\dfrac{uO_d - dO_u}{R(u-d)}$ units of the risk-free asset.

With probability p, the portfolio P will give a payoff of amount

$$\left(\frac{O_u - O_d}{S(u-d)}\right) uS + \left(\frac{uO_d - dO_u}{R(u-d)}\right) R = O_u$$

at the end of the period.

Similarly, with probability $1-p$, the portfolio P will give a payoff of amount

$$\left(\frac{O_u - O_d}{S(u-d)}\right) dS + \left(\frac{uO_d - dO_u}{R(u-d)}\right) R = O_d$$

at the end of the period.

As a result, this portfolio replicates the payoff of the option in both scenarios. Under the arbitrage-free assumption the fair price of the option must be equal to the initial price of the portfolio, which is

$$\left(\frac{O_u - O_d}{S(u-d)}\right) S + \left(\frac{uO_d - dO_u}{R(u-d)}\right) 1 = \frac{1}{R}\{qO_u + (1-q)O_d\}.$$

□

In this formula, the parameter q denotes the risk-neutral probability. Usually, q is different from the real probability p, and it can be observed that the probability p never appears in the calculation of the option price. The risk-neutral pricing formula is very convenient and can be easily extended for pricing the option with multiple periods of time. For instance, in order to fairly price a given European call option which has a strike price K and an expiration time T (quoted in years), the risk-neutral pricing formula can be used to derive a closed-form expression of the value of the European call option as follows supposing that the time T is divided into m periods and there is no risk-free rate, i.e., $R = 1$.

$$Price = \sum_{i=0}^{m} q^i (1-q)^{m-i} \binom{m}{i} \max\{u^i d^{m-i} S_0 - K, 0\} \tag{2.5}$$

The main advantage of using the binomial options pricing model is that this model can be used to price an option with complicated payoff function. However, as one might have expected, the limitation of this approach is that it is relatively slow as compared with other approaches, for example, the Black-Scholes model, which will be discussed next. This scalability issue is a result of the lattice size, i.e., the number of nodes in the lattice, which grows exponentially with the number of time periods. Therefore, it is possible but might not be a brilliant idea to use this model to price options in continuous time.

2.2.2 Black-Scholes Model

Again, in this model, the principal underlying theory is the arbitrage-free assumption. The idea behind this model is not much different from that of the binomial options pricing model; however, this model is capable of pricing the options under the continuity of time. Starting from assuming that the price of the underlying asset $S(t)$ is a continuous process which obeys the geometric Browning motion. Suggested by Ito's lemma, we have

$$dS = \mu S dt + \sigma S dz, \tag{2.6}$$

where z is the standard Weiner process and μ is equal to $\nu + \frac{1}{2}\sigma^2$. By applying the arbitrage-free argument, if the underlying asset pays no dividend, the closed-form formula for pricing a European call option ($V_c(\cdot,\cdot)$) and that for pricing a European put option ($V_p(\cdot,\cdot)$), knowing that $V_c(S,T) = (S-K)^+$ and $V_p(S,T) = (K-S)^+$, are shown to be

$$V_c(S,t) = S\phi(d_1) - Ke^{-r(T-t)}\phi(d_2), \tag{2.7}$$
$$V_p(S,t) = Ke^{-r(T-t)}\phi(-d_2) - S\phi(-d_1), \tag{2.8}$$

where
S is the price of the underlying asset;
$T - t$ is the time to maturity (T denotes the expiration date);
K is the strike price specified in the option contract;
r is the risk-free rate;
σ is the volatility of the asset return;
$\phi(\cdot)$ is standard normal distribution ($\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\eta^2/2}\, \mathrm{d}\eta$);

$$d_1 = \frac{log(S/K) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}};$$

$$d_2 = \frac{log(S/K) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} = d_1 - \sigma\sqrt{T-t}.$$

Given the closed-form expressions of the values of a European call option and a European put option, one might use the equation $\phi(-x) = 1 - \phi(x)$, which results from the symmetry of the bell-shaped probability density function of the normal distribution, and notice a put-call parity which relates the value of a European call option to the value of a European put option sharing the same strike price and expiration date.

$$V_p(S,t) - V_c(S,t) = Ke^{-r(T-t)}(\phi(-d_2) + \phi(d_2)) - S(\phi(-d_1) + \phi(d_1)) \tag{2.9}$$

In other words,

$$V_p(S,t) - V_c(S,t) + S = Ke^{-r(T-t)} \tag{2.10}$$

It is, in addition, possible to figure out the relationship between the Black-Scholes model and the binomial options pricing model. It has been shown that the output of the binomial options pricing model converges to the Black-Scholes formula as the length of one period of time becomes smaller ($\Delta t \to 0$). The Black-Scholes model has been used and proved successful in pricing financial options in the sense that, as shown by empirical studies, the Black-Scholes price is fairly close to the observed market price. The advantage of this model is that it is evidently faster than the binomial options pricing model. Matlab, a widely-used numerical computing environment, also provides a function for computing prices of the options using the Black-Scholes model together with a well-written on-line documentation accessible from *http://www.mathworks.co.uk*.

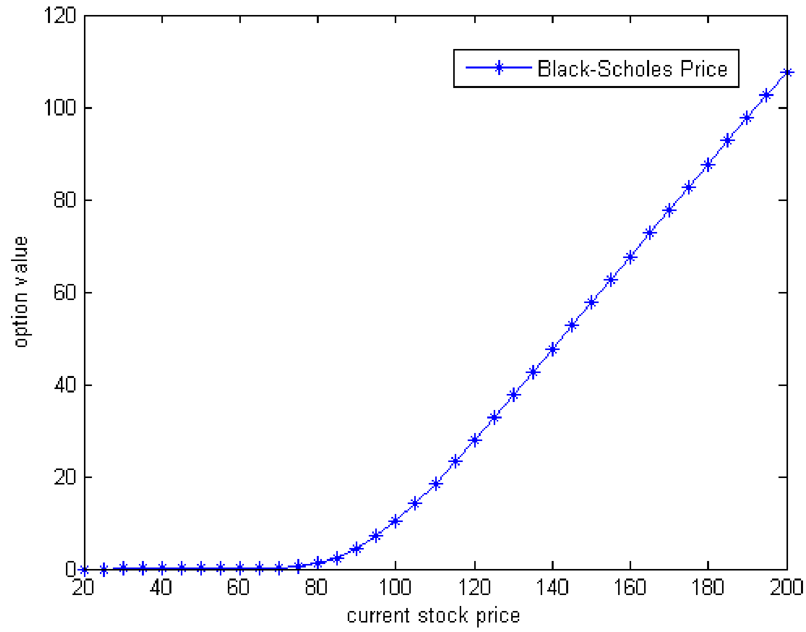

Figure 2.5: Illustration of the Black-Scholes model using Matlab

To illustrate the use of the Black-Scholes pricing model, we use a built-in function *blsprice* in Matlab to calculate the price of the option with a strike price equal to £100. In this example, we assume that the annualised risk-free rate is 10%, the price of the underlying asset today ranges from £20 to £200, and the expiration date of this option is one year. The result obtained is shown in Figure 2.5. Further reading about the Black-Scholes model and its extension to other options can be found in Hull [26], Musiela and Rutkowski [36], and Uğur [47].

On the other hand, even though it is possible to increase the flexibility of the Black-Scholes model, it appears to be unable to price options with complicated payoff functions, for example, arithmetic Asian options and American options. Fortunately, an American call option which is tied to a non-dividend paying asset is proved to be less beneficial when it is exercised early. In such cases, the value of the American call option has no difference from the value of the corresponding European call option (see Hull [26]). This is not true for American put options even if the option is linked to a non-dividend paying asset.

Although the studies of the option pricing seem to be carried out in a good direction, the first two models, namely the binomial options pricing model and the Black-Scholes model, sometimes do not work as well as expected. Both approaches aim to construct a perfect replicating portfolio, and to achieve that there are various conditions assumed to hold. For example, the typical variations of the binomial options pricing model and the Black-Scholes model usually assume that the price of the underlying asset of the option follows the geometric Brownian motion, and this is not always the case. One can empirically verify that this assumption, in fact, does not exactly hold in the market. There have been attempts to improve such models; however, the computational complexity of the model is inevitably increased. Especially in the Black-Scholes model, it is worth noting here that the elegant closed-form formula of the option value relies on many assumptions apart from that the price of the underlying asset follows the geometric Brownian motion. Some of them seem to be unrealistic in the real financial market. Examples of these assumptions are listed below.

1. There are no transaction costs.
2. The underlying asset can be arbitrarily divided.
3. Risk-free rate and the volatility of the underlying asset always remain unchanged.

Hence, another pricing approach was proposed.

2.2.3 ϵ-Arbitrage Robust Pricing Model

The ϵ-arbitrage robust pricing model uses the optimisation technique called robust optimisation to find a replicating portfolio and then price the option. Briefly, robust optimisation is an approach used to model optimisation problems which are subject to uncertainty, similarly to the stochastic programming approach. We review the ϵ-arbitrage robust pricing model from Chen [15].

To illustrate the major difference between the stochastic programming approach and the robust optimisation approach, let $f(x, \tilde{D})$ be the function to be optimised, where x denotes the decision variables and $\tilde{D}$ denotes the input data which can be subject to uncertainty. In this example, let further assume that we want to minimise the objective function f. The stochastic programming approach is expectation-based, which means it tries to optimise the expected value of the objective function.

$$\text{minimise}_x \ \mathbb{E}(x, \tilde{D}) \tag{2.11}$$

The robust optimisation approach, on the other hand, uses the idea of the worst-case analysis which can be described by

$$\text{minimise}_x \text{maximise}_{\tilde{D} \in U} \ f(x, \tilde{D}). \tag{2.12}$$

Although, it is application-dependent to choose the model for representing the optimisation problem, it is worth noting here that the robust optimisation is proved computationally tractable for a number of certain classes of convex optimisation problems. We provide in-depth review of optimisation models in Chapter 3.

The first step in this approach is to construct a robust optimisation model which corresponds to the pricing problem. To do so, let first suppose that $P(\tilde{S}, K)$ is the payoff function of the option with a strike price K. The random variable $\tilde{S}$ denotes a collection of the future prices of the underlying asset, which in each step can either increase or decrease. For a European call option which expires at time T, as an example, $P(\tilde{S}, K) = \max\{S_T - K, 0\}$.

In order to find a replicating portfolio, which in this approach does not have to perfectly match the payoff of the option, the objective function to be optimised is the difference in payoff between the option and the portfolio and is written as

$$|P(\tilde{S}, K) - W_T|, \tag{2.13}$$

where W_T is the wealth level of the portfolio at the expiration date of the option. If such a difference is zero, then a perfect hedging strategy exists. Otherwise, this term would be called an error or an arbitrage. The symbol ϵ is used to represent this difference. Using the robust optimisation technique, the objective of the ϵ-arbitrage robust pricing approach is to minimise the worst-case error ϵ. Once the optimal solution of the associated robust optimisation problem has been found, the final portfolio wealth W_T is expected to closely match the payoff of the option, and it would be reasonable to set the price of the option to the current value of this replicating portfolio.

More specifically, the robust optimisation problem for pricing the option is given by

ϵ-Arbitrage Robust Pricing Model

$$\begin{aligned}
&\text{minimise}_{\{x_t^S\}_{t=0}^T, \{x_t^B\}_{t=0}^T, \{y_t\}_{t=0}^{T-1}} \text{maximise}_{\{\tilde{r}_t^S\}_{t=0}^{T-1} \in U} \quad |P(\tilde{S}, K) - W_T| \\
&\text{subject to} \\
&\quad W_T = x_T^S + x_T^B \\
&\quad x_t^S = (1 + \tilde{r}_{t-1}^S)(x_{t-1}^S + y_{t-1}), \quad \forall t = 1, 2, ..., T \\
&\quad x_t^B = (1 + r_{t-1}^B)(x_{t-1}^B - y_{t-1}), \quad \forall t = 1, 2, ..., T,
\end{aligned} \tag{2.14}$$

where
x_t^S is the amount of money invested in the underlying asset at time t;
x_t^B is the amount of money invested in the risk-free asset at time t;
y_t is the amount of money moved from the risk-free asset to the underlying asset at time t (negative quantity means moving the money from the underlying asset to the risk-free asset instead);
$\tilde{r}_t^S$ is the return of the underlying asset during the period $[t, t+1]$ (the symbol $\sim$ is used to emphasise that this parameter is subject to uncertainty);
r_t^B is the return of the risk-free asset during the period $[t, t+1]$.

After solving the optimisation problem (2.14), the price of the option would thus be set to the initial price of the portfolio, i.e., the total amount of money initially invested in both the underlying asset of the option and the risk-free asset ($x_0^S + x_0^B$). Since the pricing model uses the concept of the robust optimisation approach, sometimes we refer to it in short as a *robust pricing model*.

In the pricing model (2.14), the uncertain parameters are the future rates of return of the (risky) underlying asset, i.e., $\{\tilde{r}_t^S\}_{t=0}^{T-1}$. One important aspect in the robust optimisation is how one should define the uncertainty set U. Using this pricing model, the investor is free to design the uncertainty set accordingly to his or her preference. However, the investor should be aware that the design of the uncertainty set results in the complexity of the robust counterpart of the pricing problem to some extent. The author of this work also proposed a way to design the uncertainty set by setting the boundaries on the single-period returns $\{\tilde{r}_t^S\}_{t=0}^{T-1}$:

$$|\tilde{r}_t^S - \mu| \leq \Gamma_t \sigma, \quad \forall t, \tag{2.15}$$

which is equivalent to

$$\mu - \Gamma_t \sigma \leq \tilde{r}_t^S \leq \mu + \Gamma_t \sigma, \quad \forall t, \tag{2.16}$$

where μ and σ denote mean and standard deviation of the random return $\tilde{r}_t^S$, respectively, and Γ_t is a predefined risk-aversion parameter.

Moreover, the author also created another set of boundaries on the cumulative returns $\{\tilde{R}_t^S\}_{t=1}^T$, where

$$\tilde{R}_t^S = \prod_{i=0}^{t-1}(1 + \tilde{r}_i^S). \tag{2.17}$$

This set of boundaries comes as a result of the central limit theorem (CLT), a famous theory in probability and statistics. As a result, we begin by briefly explaining the central limit theorem and then explain the boundaries proposed by Chen [15].

Theorem 2.2.1. (Central limit theorem) *Given a set of independent n random variables X_1, X_2, ..., X_n drawn from the identical distribution with the expected value of μ and the variance of σ^2,*

$$\lim_{n\to+\infty} \frac{\frac{1}{n}\sum_{i=1}^{N} X_i - \mu}{\sigma/\sqrt{n}}$$

approaches the standard normal distribution.

A rule of thumb when using the central limit theorem is that it is often claimed that a set of more than thirty independent and identically distributed random variables is sufficient for producing an approximate normally distributed sample mean, regardless of the distribution of the original random variables.

Suppose that μ_{log} and σ_{log} are mean and standard deviation of $\log(1 + \tilde{r}_t^S)$, respectively. Note that these items can be estimated empirically. Using the central limit theorem, the author suggested that

$$\frac{\frac{1}{t}\sum_{i=0}^{t-1}\log(1 + \tilde{r}_i^S) - \mu_{log}}{\sigma_{log}/\sqrt{t}} \sim N(0,1), \tag{2.18}$$

$$\frac{\frac{1}{t}\log\tilde{R}_t^S - \mu_{log}}{\sigma_{log}/\sqrt{t}} = \frac{\frac{1}{t}\log\prod_{i=0}^{t-1}(1 + \tilde{r}_i^S) - \mu_{log}}{\sigma_{log}/\sqrt{t}} \sim N(0,1). \tag{2.19}$$

Consequently, he suggested a set of boundaries on the cumulative returns as

$$\left|\frac{\frac{1}{t}\log\tilde{R}_t^S - \mu_{log}}{\sigma_{log}/\sqrt{t}}\right| \le \Gamma, \quad \forall t. \tag{2.20}$$

These boundaries can also be rewritten as

$$e^{t\mu_{log} - \Gamma\sqrt{t}\sigma_{log}} \le \tilde{R}_t^S \le e^{t\mu_{log} + \Gamma\sqrt{t}\sigma_{log}}, \quad \forall t. \tag{2.21}$$

Two major advantages of this pricing model can be seen from the problem formulation. The first one is that this approach can be used to price the options with complicated payoff functions with ease by the substitution of the payoff function $P(\tilde{S}, K)$ in the objective function. The second advantage of this approach is that this model does not need assumptions about asset price dynamics and market conditions. Recall that the Black-Scholes model heavily relies on that the price of the underlying asset follows the geometric Brownian motion, and in the binomial options pricing model the geometric Brownian motion is often assumed before tuning the model parameters. In this respect, the ϵ-arbitrage robust pricing model outperforms the other two models. Furthermore, while the binomial options pricing model and the Black-Scholes model output only a single so-called fair price for a given option, the ϵ-arbitrage robust pricing model can output a family of prices by adjusting the model parameters: Γ and $\{\Gamma_t\}_{t=1}^T$. The values of Γ and $\{\Gamma_t\}_{t=1}^T$ reflect the degree of risk-aversion of the investor.

2.3 Conclusions

To sum up, in this chapter, we provide background knowledge of financial options together with successful predecessor models for options pricing. All of the discussed pricing models rely on the arbitrage-free assumption, which is a crucial assumption leading to market equilibrium. When asset price follows the geometric Brownian motion, the Black-Scholes model and the binomial options pricing model arguably output the fairest option price. In real market, it is however hardly ever the case that the asset price exactly follows the geometric Brownian motion. Additionally, it is also difficult to simulate market conditions, for example, transaction costs and short-selling prohibition, using the Black-Scholes model or the binomial options pricing model. Both of the pricing models therefore lack ability to capture the behaviour of the real market. The ϵ-arbitrage robust pricing approach, on the other hand, models the market conditions via a set of constraints. Therefore, it can be deemed more flexible. Unlike the Black-Scholes model and the binomial options pricing model, the ϵ-arbitrage robust pricing model does not adopt the notion of the arbitrage-free pricing directly. Instead, it allows violation of the arbitrage-free assumption, but the arbitrage error should be minimised. The pricing model is then reduced to an optimisation problem. Optimisation techniques are then discussed in the following chapter before we thoroughly analyse the ϵ-arbitrage robust pricing model in Chapter 4.

Chapter 3

Literature Review: Mathematical Optimisation

Optimisation is an important area in mathematics and computer science. It refers to a process of selecting the best (optimal) choice from the pool of alternatives. Optimisation, therefore, takes a major role in almost all applications that require decision making. In addition to mathematics and computer science, applications of the optimisation theories can also be found in innumerable areas, for example, finance, energy, and engineering. For example, an investor in the financial market may want to find a way to allocate his or her budget in different assets in order to maximise the expected return at the end of the investment horizon. This example is referred to as portfolio optimisation problem. In this example, the decision to be made can be a fraction of his or her budget to be invested in each of the assets. This confirms the crucial importance of the optimisation studies. Mathematically speaking, the decision in the real world applications is a set of decision variables (x) in the optimisation model. The other type of the system variables whose values are not controlled by the decision maker is called the uncontrollable factor (D), for example, the assets' returns in the example of portfolio optimisation. A general formulation of the optimisation problem, assuming that we want to minimise the objective function f, is shown below.

General Optimisation Problem

$$
\begin{gathered}
\text{minimise}_x \; f(x, D) \\
\text{subject to} \\
g_i(x, D) \leq 0, \quad \forall i = 1, 2, ..., I \\
h_j(x, D) = 0, \quad \forall j = 1, 2, ..., J
\end{gathered}
\tag{3.1}
$$

Typically, the optimisation model, i.e., the mathematical model that represents the optimisation problem, consists of two important parts. The first part is the objective function (f), which is the function to be optimised (minimised or maximised). The other part of the optimisation model is a set of constraints which is a set of rules that the system variables, i.e., the decision variables and the uncontrollable factors, are expected to obey. A constraint can be either an inequality constraint (g_i) or an equality constraint (h_j). The optimisation problem is said to be infeasible if there is no decision x that can satisfy all of the constraints. Otherwise, the optimisation problem is said to be feasible.

Assume for the moment that the there is no uncontrollable factor D or the value of D is deterministic and exactly known, the optimisation problem can be reduced to

$$
\begin{gathered}
\text{minimise}_x \; f(x) \\
\text{subject to} \\
g_i(x) \leq 0, \quad \forall i = 1, 2, ..., I \\
h_j(x) = 0, \quad \forall j = 1, 2, ..., J.
\end{gathered}
$$

(3.2)

In particular, x can be a vector of an arbitrary number of decisions, i.e., $x \in \mathbb{R}^n$. In that case, f, g_i, and h_j become real-valued functions taking $x \in \mathbb{R}^n$ as input.

$$f: \quad \mathbb{R}^n \to \mathbb{R} \tag{3.3}$$
$$g_i: \quad \mathbb{R}^n \to \mathbb{R} \tag{3.4}$$
$$h_j: \quad \mathbb{R}^n \to \mathbb{R} \tag{3.5}$$

Note that there are many other ways to formulate the optimisation problems. For instance, it is possible to remove the equality constraint h_j without reducing the capabilities of the model as the constraint $h_j(x, D) = 0$ is equivalent to a pair of two inequality constraints: $h_j(x, D) \leq 0$ and $-h_j(x, D) \leq 0$.

3.1 Convex Optimisation

In this section, we review background knowledge of convex optimisation. First of all, we review all definitions needed and provide some examples. We then close this section by giving a general formulation of the convex optimisation problems.

Definition 3.1.1. (Convex set) *A set C is said to be convex if the line segment connecting two arbitrary points x_1 and x_2 from C is wholly contained in C, i.e.,*

$$\lambda x_1 + (1-\lambda)x_2 \in C, \quad \forall \lambda \in [0, 1]. \tag{3.6}$$

Example 3.1.1. *A circle $C = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$ is convex.*

Proof.
Let (x_1, y_1) and (x_2, y_2) be any two points in C. We have

$$\begin{aligned}
&(\lambda x_1 + (1-\lambda)x_2)^2 + (\lambda y_1 + (1-\lambda)y_2)^2 \\
&\qquad = \lambda^2(x_1^2 + y_1^2) + (1-\lambda)^2(x_2^2 + y_2^2) + 2\lambda(1-\lambda)(x_1x_2 + y_1y_2) \\
&\qquad \leq \lambda^2(x_1^2 + y_1^2) + (1-\lambda)^2(x_2^2 + y_2^2) + 2\lambda(1-\lambda)\sqrt{x_1^2 + y_1^2}\sqrt{x_2^2 + y_2^2} \\
&\qquad = \lambda^2 + (1-\lambda)^2 + 2\lambda(1-\lambda) \\
&\qquad = 1,
\end{aligned}$$

for any $\lambda \in [0, 1]$. This implies that the point $\lambda(x_1, y_1) + (1-\lambda)(x_2, y_2)$ is also in C; therefore, C is convex.

□

Definition 3.1.2. (Convex function) *A function $f: \mathbb{R}^n \to \mathbb{R}$ is said to be convex if its domain is a convex set and the following holds for any a and b taken from its domain and for any $\lambda \in [0, 1]$.*

$$f(\lambda a + (1-\lambda)b) \leq \lambda f(a) + (1-\lambda)f(b) \tag{3.7}$$

If the inequality strictly holds whenever $a \neq b$ and $\lambda \in (0, 1)$, f is said to be strictly convex.

Example 3.1.2. *A function $f(x) = x^2 - 1$ is convex on $\mathbb{R}$*

Proof.
For any x_1 and x_2 in $\mathbb{R}$, we have

$$\begin{aligned}
f(\lambda x_1 + (1-\lambda x_2)) &= (\lambda x_1 + (1-\lambda)x_2)^2 - 1 \\
&= \lambda^2 x_1^2 + 2\lambda(1-\lambda)x_1x_2 + (1-\lambda)^2 x_2^2 - 1 \\
&= \lambda x_1^2 + (1-\lambda)x_2^2 - \lambda(1-\lambda)\left(x_1^2 - 2x_1x_2 + x_2^2\right) - 1 \\
&= \lambda x_1^2 + (1-\lambda)x_2^2 - \lambda(1-\lambda)(x_1 - x_2)^2 - 1 \\
&\leq \lambda x_1^2 + (1-\lambda)x_2^2 - 1 \\
&= \lambda f(x_1) + (1-\lambda)f(x_2),
\end{aligned}$$

for any $\lambda \in [0, 1]$. The function f is, thus, convex.

□

Example 3.1.3. *A linear function $f : \mathbb{R}^n \to \mathbb{R}$ given by $f(x) = c^T x$, where c is a fixed vector in $\mathbb{R}^n$, is convex.*

Proof.
For any x_1, x_2 in $\mathbb{R}^n$ and for any $\lambda \in [0, 1]$, we have

$$\begin{aligned} f(\lambda x_1 + (1-\lambda)x_2) &= c^T(\lambda x_1 + (1-\lambda)x_2) \\ &= \lambda c^T x_1 + (1-\lambda)c^T x_2 \\ &= \lambda f(x_1) + (1-\lambda) f(x_2). \end{aligned}$$

Hence, f is convex.

□

Definition 3.1.3. (Affine function) *A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is an affine function if it can be written as*

$$f(x) = Ax + b, \tag{3.8}$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.

A convex optimisation problem is an optimisation problem in a form of (3.2) with additional restrictions:

- The objective function f must be a convex function;
- The inequality constraint functions $\{g_i\}_{i=1}^{I}$ must also be convex;
- The equality constraint functions must be affine, i.e., $h_j(x) = a_j^T x + b_j$, where $a_j \in \mathbb{R}^n$ and $b_j \in \mathbb{R}$.

Example 3.1.4. *The following optimisation problem is convex*

$$\begin{gathered} \text{minimise}_x \; x \\ \text{subject to} \\ x^2 \leq 1 \end{gathered} \tag{3.9}$$

Proof.
The objective function $f(x) = x$ is linear and therefore convex, and the constraint $g(x) = x^2 - 1 \leq 0$ is also convex. Thus, this is a convex optimisation problem.

□

3.2 Conic Optimisation

In this section, we study conic optimisation which entails an investigation of a set of thriving classes of convex optimisation problems. We provide necessary definitions and examples as well as the explanation of the structure of the conic optimisation problems.

Definition 3.2.1. (Cone) *A set C is said to be a cone if*

$$\theta x \in C, \quad \forall x \in C, \; \forall \theta \geq 0. \tag{3.10}$$

Definition 3.2.2. (Convex cone) *A set C is a convex cone if it is convex and it is a cone.*

Definition 3.2.3. (Polynomial norm) *A polynomial norm* $||\cdot||_p$ *is a function from* $\mathbb{R}^n$ *to* $\mathbb{R}$ *defined as*

$$||x||_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}. \tag{3.11}$$

A Euclidean norm is a polynomial norm with $p = 2$*. In this thesis, both* $||\cdot||_2$ *and* $||\cdot||$ *refer to the Euclidean norm.*

Definition 3.2.4. (Positive (semi)definiteness) *A square matrix* $Q \in \mathbb{R}^{n\times n}$ *is said to be positive semidefinite* ($Q \succeq 0$) *if*

$$d^T Qd \geq 0, \quad \forall d \in \mathbb{R}^n. \tag{3.12}$$

If the inequality strictly holds for every $d \neq 0$*,* Q *is then said to be positive definite* ($Q \succ 0$)*.*

Definition 3.2.5. (Proper cone) *A cone* K *is a proper cone if it satisfies all of the followings.*

- K *is convex.*
- K *is closed.*
- K *is a pointed cone, i.e., if* $x \in K$ *and* $-x \in K$*, then* $x = 0$*.*
- K *has a non-empty interior.*

Example 3.2.1. *We show below a list of some important cones that appear frequently in the optimisation studies.*

- *Non-negative orthant:*

$$\mathbb{R}^n_+ = \{x \in \mathbb{R}^n \mid x_i \geq 0, \quad \forall i = 1, 2, ..., n\} \tag{3.13}$$

- *Second-order cone:*

$$\zeta_{2,n+1} = \{(x, t) \in \mathbb{R}^{n+1} \mid ||x||_2 \leq t\} \tag{3.14}$$

- *Positive semidefinite cone:*

$$\mathbb{S}^n_+ = \{X \in \mathbb{S}^n \mid X \succeq 0\} \tag{3.15}$$

In the optimisation studies, we definitely have to deal with a lot of inequalities. In the most general setting, we have

$$a \leq b \Longleftrightarrow b - a \in \mathbb{R}_+. \tag{3.16}$$

The axiom above provides a way to determine the total order of a given subset of $\mathbb{R}$, i.e., two real numbers can always be compared and the bigger one can thus be determined. Analogously, if a and b are chosen from $\mathbb{R}^n$, we have

$$a \leq b \Longleftrightarrow b - a \in \mathbb{R}^n_+. \tag{3.17}$$

However, only partial ordering can be attained through this vector inequality. The example below is cited to illustrate why total ordering cannot be achieved.

Example 3.2.2.

- $\begin{bmatrix}1\\1\end{bmatrix} \leq \begin{bmatrix}3\\2\end{bmatrix}$
- $\begin{bmatrix}1\\4\end{bmatrix} \nleq \begin{bmatrix}3\\2\end{bmatrix}$ and $\begin{bmatrix}3\\2\end{bmatrix} \nleq \begin{bmatrix}1\\4\end{bmatrix}$

The notion of inequality can be further generalised using a proper cone K. This can be done by introducing a new operator $\preceq_K$ and defining it as

$$A \preceq_K B \Longleftrightarrow B - A \in K. \tag{3.18}$$

Example 3.2.3. *Instead of saying that* $[1 \quad 1]^T \leq [3 \quad 2]^T$*, we can equivalently say that* $[1 \quad 1]^T \preceq_{\mathbb{R}^2} [3 \quad 2]^T$*. In the very common cases, we usually drop the cone's description though. This means it should be understood that if* a *and* b *are vectors in* $\mathbb{R}^n$*,* $a \leq b$ *means* $b - a$ *is a non-negative vector, and if* A *and* B *are symmetric matrices in* $\mathbb{R}^{n\times n}$*,* $A \preceq B$ *means that* $B - A$ *is positive semidefinite.*

The following is a list of the fundamental properties of the generalised inequality.

- Reflexive: $x \preceq_K x$.
- Antisymmetric: if $x \preceq_K y$ and $y \preceq_K x$, then $x = y$.
- Transitive: if $x \preceq_K y$ and $y \preceq_K z$, then $x \preceq_K z$.
- Preserved under addition: if $x \preceq_K y$ and $u \preceq_K v$, then $x + u \preceq_K y + v$.
- Preserved under non-negative scaling: if $x \preceq_K y$ and $\alpha \geq 0$, then $\alpha x \preceq_K \alpha y$.

Conic optimisation is a vast subset of convex optimisation. A conic optimisation problem or a cone program typically has a linear objective function and is written in a form below.

Conic Optimisation Problem

$$\begin{gathered} \text{minimise}_x \; c^T x \\ \text{subject to} \\ Fx + g \preceq_K 0 \\ Ax = b \end{gathered} \tag{3.19}$$

3.2.1 Linear Programming Problems

Deterministic linear programming problems are linear programs which are not subject to uncertainty. Any form of the optimisation problem (3.2) where the objective function f, the inequality constraints g_i, and the equality constraints h_j are affine functions is called a linear program. However, it would be useful and convenient to introduce the standard form of linear programs, which is

Linear Programming Problem

$$\begin{gathered} \text{minimise}_x \; c^T x \\ \text{subject to} \\ Ax = b \\ x \geq 0. \end{gathered} \tag{3.20}$$

In this formulation, the decision variable x is a vector consisting of n real-valued entries. The cost coefficient c is also a vector of n real-valued numbers as well. The coefficient matrix A is a real matrix of dimension $m \times n$, and the vector b is a collection of m real numbers. Therefore, the matrix equation $Ax = b$ is, in fact, a collection of m equality constraints.

There are algorithms proposed to solve linear programming problems. The simplex algorithm was first published by Dantzig in 1947 (for more details about this algorithm, see Dantzig, Orden and Wolfe [17]). Broad and deep explanation of linear programming is gathered in Dantzig and Thapa [18]. Apart from the simplex algorithm, the interior point method is also proved efficient and beneficial for solving many classes of the optimisation problems including the linear programming problems (see Mehrotra [35]). These two algorithms are commonly used to solve linear programming problems, and typically both of them are already made available in standard solvers, for example, LINPROG in Matlab and CPLEX.

There have been a lot of studies about linear optimization; however, the theory that is most useful in this work is the *duality in linear programming.*

Definition 3.2.6. (Dual problem) *A dual problem of the primal linear programming problem (3.20) is another linear programming problem defined as*

Dual Linear Programming Problem

$$\begin{gathered} \text{maximise}_y \; b^T y \\ \text{subject to} \\ A^T y \leq c. \end{gathered} \tag{3.21}$$

After having established a pair of primal and dual problems, the next definition that we would like to introduce is *duality gap.*

Definition 3.2.7. (Duality gap) *Duality gap is defined as a difference between the optimal objective value of the primal problem (p^*) and that of the dual problem (d^*).*

There are three important properties regarding the relationship between the primal and the dual problems: the unboundedness property, the weak duality property, and the strong duality property. Explanation of this topic can be found in many literatures, for example, Boyd and Vandenberghe [12] and Bradley, Hax and Magnanti [14].

Theorem 3.2.1. (Unboundedness property) *If the primal (dual) problem is unbounded, i.e., the primal (dual) problem has an unbounded solution, then the dual (primal) problem is infeasible.*

Theorem 3.2.2. (Weak duality property) *The duality gap $p^* - d^*$ is always non-negative.*

Theorem 3.2.3. (Strong duality property) *If the primal (dual) problem has a finite optimal solution, then the dual (primal) problem also has a finite optimal solution and both solutions coincide, i.e., the duality gap is equal to zero.*

Note that the strong duality property becomes very useful when we want to solve a robust counterpart of the robust linear optimisation problem with a polyhedral uncertainty set. This will be discussed in details later. Duality theory is also available for many other classes of optimisation problems. In fact, it is proven that there generally is a dual problem for every conic optimisation problem.

3.2.2 Quadratic Constrained Quadratic Programming Problems

Quadratic constrained quadratic programming problems (QCQP) are problems whose objective function and constraints are quadratic functions of the decision variables. They are usually written in the following form.

Quadratic Constrained Quadratic Programming Problem

$$\begin{gathered} \text{minimise}_x \; \frac{1}{2} x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} \\ Ax = b \\ \frac{1}{2} x^T P_i x + q_i^T x + r_i \leq 0, \quad \forall i = 1, 2, ..., I, \end{gathered}$$

(3.22)

where $x \in \mathbb{R}^n$ is the decision variables and $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $P_i \in \mathbb{R}^{n \times n}$, $q_i \in \mathbb{R}^n$, and $r_i \in \mathbb{R}$ are input to the program.

3.2.3 Second-Order Cone Programming Problems

Similarly to the quadratic constrained quadratic programming problems, second-order cone programming problems are those of the following form.

Second-Order Cone Programming Problem

$$\begin{gathered}
\text{minimise}_x \; c_0^T x \\
\text{subject to} \\
A_0 x = b_0 \\
\|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad \forall i = 1, 2, ..., I,
\end{gathered} \tag{3.23}$$

where $x \in \mathbb{R}^n$ is the decision variables and the matrices $A_i \in \mathbb{R}^{n_i \times n}$, the vectors $b_i \in \mathbb{R}^{n_i}, c_i \in \mathbb{R}^n$, and $d_i \in \mathbb{R}$ are input to the program. The constraints of the program of this type can be rewritten using an inequality generalised by the second-order cone

$$\begin{pmatrix} A_i x + b_i \\ c_i^T x + d_i \end{pmatrix} \in \zeta_{2, n_i + 1}, \tag{3.24}$$

and thus they are called the second-order constraints.

Analogously to the linear programming problem, there is an elegant explicit formula for the dual second-order cone programming problem. Under certain conditions, weak and strong duality properties hold for a pair of primal and dual second-order cone programming problems. Without loss of generality, the constraint $A_0 x = b_0$ can always be omitted as a result of the equivalence between $A_0 x = b_0$ and $\begin{pmatrix} A_0 x - b_0 \\ 0 \end{pmatrix} \in \zeta_{2, n_0 + 1}$. Following from the simplification, a pair of primal and dual second-order cone programming problems is given by

(Primal)

$$\begin{gathered}
\text{minimise}_x \; c_0^T x \\
\text{subject to} \\
\|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad \forall i = 1, 2, ..., I
\end{gathered}$$

(Dual)

$$\begin{gathered}
\text{maximise}_{\{z_i, w_i\}_{i=1}^I} \; -\sum_{i=1}^{I} \left(b_i^T z_i + d_i w_i \right) \\
\text{subject to} \\
\sum_{i=1}^{I} \left(A_i^T z_i + c_i w_i \right) = c_0 \\
\|z_i\| \leq w_i, \quad \forall i = 1, 2, ..., I,
\end{gathered} \tag{3.25}$$

where $z_i \in \mathbb{R}^{n_i}$ and $w_i \in \mathbb{R}$ are dual variables, i.e., decision variables of the dual problem (see Lobo, Vandenberghe, Boyd, and Lebret [30]). Provided that the strong duality holds, an uncertainty-affected robust linear constraint with ellipsoidal uncertainty set can be formulated as a deterministic second-order cone programming problem.

3.2.4 Semidefinite Programming Problems

The problems of the following form are called semidefinite programming problems.

Semidefinite Programming Problem

$$\begin{gathered} \text{minimise}_x \ c^T x \\ \text{subject to} \\ Ax = b \\ \left(\sum_{i=1}^{n} x_i F_i\right) + G \preceq 0, \end{gathered} \tag{3.26}$$

where $x \in \mathbb{R}^n$ is the decision variables and $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $G, F_i \in \mathbb{S}^k$ are input to the program.

Semidefinite programming problem is often regarded as a generalised model encapsulating many other types of deterministic optimisation problems. For example, if G and $\{F_i\}_{i=1}^n$ are all diagonal matrices, the whole optimisation problem reduces to a simple linear programming problem.

Modern standard solvers usually have capabilities to solve semidefinite programs with acceptable amount of resource. In order to use the solvers, typically a semidefinite program has to be transformed to the following standard form.

$$\begin{gathered} \text{minimise}_X \ \text{Tr}(CX) \\ \text{subject to} \\ \text{Tr}(A_i X) = b_i, \quad \forall i = 1, 2, ..., I \\ X \succeq 0, \end{gathered} \tag{3.27}$$

where $X \in \mathbb{S}^n$ contains the decision variables, and $C, A_i \in \mathbb{S}^n$ and $b_i \in \mathbb{R}$ are input to the program.

3.3 Optimisation under Uncertainty

Indeed, if there are no uncontrollable factors or all of the controllable factors are deterministic, the optimisation problem would become a lot easier to solve. Unfortunately, that is rarely the case since almost all of the real world applications are subject to uncertainty.

Therefore, the assumption about not having an uncontrollable factor (D) or the uncontrollable factor being deterministic appears to be too restrictive. In reality, there are a number of system variables whose values are not under the control of the decision makers. For example, the portfolio optimisation problem is subject to uncertainty because the assets' returns are usually non-deterministic. It is seen from numerous applications that ignoring these uncertain factors can severely affect the optimisation problems in terms of both optimality and feasibility of the generated solution. In the utmost case, the generated solution can be meaningless if it is not implementable.

In order to take into account such uncertainty, the amended optimisation model can be rewritten as follows.

General Optimisation Problem under Uncertainty

$$\begin{gathered} \text{minimise}_x \ f(x, \tilde{D}) \\ \text{subject to} \\ g_i(x, \tilde{D}) \leq 0, \quad \forall i = 1, 2, ..., I \\ h_j(x, \tilde{D}) = 0, \quad \forall j = 1, 2, ..., J, \end{gathered}$$

(3.28)

where $\sim$ is used to emphasise the uncertain-affected input to the program.

As one might have guessed, even though the above optimisation model does consider the uncertainty that can arise in the optimisation problems, it is not a rigorous mathematical model since the objective function takes as input uncertain parameters and therefore is not well-defined. The model is presented in order to give some intuition about the idea of how to model the optimisation problems which are not immunised against uncertainty. In the following sections, we review three techniques used to deal with the uncertainty in the optimisation problems. All of them view an uncertain input in different ways. Note that none of them is clearly better than the others. It is application-dependent to choose what the appropriate approach to be used is.

3.3.1 Stochastic Programming Approach

As mentioned before that uncertainty factors can arise in the optimisation problems and that the deterministic optimisation approach is not specially designed to deal with this, other optimisation models with the capabilities to tackle the uncertainty have been proposed. The model that was first proposed in this regard is stochastic programming approach. Here, we review concept of the stochastic programming approach from Shapiro, Dentcheva and Ruszczyński [41] and Shapiro and Philpott [43].

In the stochastic programming approach, each unknown parameter is viewed as a random variable following a certain probability distribution. The purpose of the stochastic programming approach is to determine the optimal solution based on the expected value of the nominal objective function. In other words, the stochastic programming approach aims to find the solution that is optimal on average. Typically, the objective function of the stochastic programming problem is written as

$$\text{minimise}_x \ \mathbb{E}(x, \tilde{D}). \tag{3.29}$$

Normally, there are two types of decisions for the decision makers to make: here-and-now decisions and wait-and-see decisions (or recourse actions). The here-and-now decisions involve the decisions to be made before the unknown factors become materialised while the wait-and-see decisions are the decisions to be made depending upon future realisations of the unknown parameters. According to the types of decisions, stochastic programs can be categorized into two major groups which are known as: stochastic programming problems with recourse actions and chance-constrained problems.

In the stochastic programs with recourse actions, the decision maker first makes here-and-now decisions and then recursively waits and observes the new information in order to make subsequent decisions accordingly to the history of the previous observations. That this model allows the decision maker to defer making a decision until knowing more about the problem is realistic and desirable because in reality it is unlikely that the decision maker has to make all of the decisions hurriedly with an inadequate amount of knowledge he or she currently has. The stochastic programs with recourse actions can involve any arbitrary number of stages of time; the two-stage stochastic programming problem with recourse actions is the simplest variation of the problems of this type. In this variation, the decision maker has to make two separate decisions: first-stage decision and second-stage decision (recourse action), which is the decision to be made after observing some values. The example below shows how to formulate the two-stage stochastic programming problem with recourse actions assuming that there are finitely many number of future scenarios K, and both objective function and constraints are linear.

Two-Stage Stochastic Problem

$$\begin{gathered}
\text{minimise}_x \; c^T x + \sum_{k=1}^{K} p_k q_k^T y_k \\
\text{subject to} \\
Ax = b \\
T_k x + W_k y_k = h_k, \quad \forall k = 1, 2, ..., K \\
x \geq 0, y_k \geq 0, \quad \forall k = 1, 2, ..., K
\end{gathered} \tag{3.30}$$

In the model (3.30), x is the first-stage decision, and y is the second stage decision which can be either y_1, y_2, ... , or y_k depending on the future scenario which is determined by the unknown $\{\xi_k\}_{k=1}^{k=K}$, $\xi_k = (q_k, h_k, T_k, W_k)$. The probability of a scenario denoted by ξ_k going to happen is given by p_k. The term $c^T x$ is the cost incurred by the first-stage decision, and the summation term is the expected optimal cost from the second stage decision.

On the contrary, traditional chance-constrained problems involve only here-and-now decisions; therefore, the decision x has to be made without the knowledge of the realisation of the non-deterministic input $\tilde{D}$. In this type of stochastic programs, the objective function is deterministic while the feasibility of the problem is described by the chance constraints, where p_o and p_c in the following example are predefined probabilities. The values of p_o and p_c are the significance levels, which are the probabilities that the constraints have to be satisfied. In the example below, we use an integrated chance-constraint; however, it is also possible to define the chance constraints separately for every individual constraint $a_i x \leq b_i$ where a_i is the i^{th} row of the matrix A and b_i is the i^{th} element of the vector b.

Example 3.3.1. *The chance-constrained program corresponding to*

$$\begin{gathered}
\text{minimise}_x \; c^T x \\
\text{subject to} \\
Ax \leq b,
\end{gathered} \tag{3.31}$$

which is a linear program under uncertainty, is given by

Chance-Constrained Problem

$$\begin{gathered}
\text{minimise}_{x,\tau} \; \tau \\
\text{subject to} \\
\text{Prob}\left[c^T x \leq \tau\right] \geq p_o \\
\text{Prob}\left[Ax \leq b\right] \geq p_c.
\end{gathered} \tag{3.32}$$

It can be observed that both variations of the stochastic programs rely on the probability-based computation (the expectation of the objective function and the chance constraints); consequently, one shortcoming of the stochastic programming approach is that the actual probability distributions of the unknown parameters have to be known. Unfortunately, that is rarely the case. Moreover, even if we have the data of the actual or the assumed distributions, it is still tractably challenging to solve the resulting model.

3.3.2 Robust Optimisation Approach

Another optimisation model that is capable of dealing with the uncertainty that arises in the optimisation problems is the robust optimisation model. In fact, the robust optimisation problem is a specialisation of the chance-constrained problem where the significance levels are all set to one. This means the generated solution of the robust optimisation model has to strictly satisfy all of the constraints regardless of the realisation of the unknown parameters in the problem, and this characteristic of the generated solutions is termed *robustness*.

To grasp the main concept of the robust optimisation approach, consider the following linear program

$$\begin{gathered}\text{minimise}_x\ c^T x\\ \text{subject to}\\ Ax \leq b,\end{gathered} \tag{3.33}$$

where $x \in \mathbb{R}^n$ is a vector of decision variables and $c \in \mathbb{R}^n$ is a fixed cost vector. We further assume that the matrix $A \in \mathbb{R}^{m\times n}$ and the vector $b \in \mathbb{R}^m$ are uncertain. The assumption that c is fixed and subject to no uncertainty is not restrictive as we can always reformulate the optimisation problem in such a way that makes the objective function uncertainty-free (see Bertsimas and Sim [10]). If we know that all of the possible values of A and b can be described by an uncertainty set U, the robust optimisation approach guarantees that the generated solution x^* always complies with the constraint $Ax \leq b$ even if we do not know the realisation of A and b. That is the *robust counterpart* of this uncertain linear program with the prescribed uncertainty set U is given by

$$\begin{gathered}\text{minimise}_x\ c^T x\\ \text{subject to}\\ Ax \leq b, \quad \forall [A, b] \in U.\end{gathered} \tag{3.34}$$

Typically, the uncertainty set U is an infinite set, for example, $U = \{[A, b] | \underline{A} \leq A \leq \bar{A},\ \underline{b} \leq b \leq \bar{b}\}$. By simply enumerating all possibilities in U, the resulting optimisation problem will become semi-infinite because of the infinite number of constraints it has, and this seems to be intractable. A vast number of robust optimisation problems, however, have an equivalent tractable formulation, which can be obtained by using, for example, conic duality, e.g., the duality of linear programs and the duality of second-order cone programs. Problem modellers who opt to use the robust optimisation approach should also bear in mind that the shape of the uncertainty set U can very well affect the tractability of the robust problem.

Suppose that the cost vector c is no longer assumed to be uncertainty-immunised. The robust optimisation approach will then associate the given uncertain linear program with the following optimisation problem.

$$\text{minimise}_x\ \{\text{maximise}_{\xi=[A,b,c]\in U}\ c^T x : Ax \leq b, \quad \forall \xi \in U\} \tag{3.35}$$

The result implies that the robustness of the generated solution is equivalent to the optimality in the worst-case scenario allowed by U.

That the robust optimisation approach aims to find the solution that is optimal in the worst-case scenario might make the problem modeller or the decision maker look needlessly pessimistic. However, in the critical applications where constraint violation by no mean can be tolerated, the robust optimisation approach is evidently more of an appropriate choice as compared to the stochastic programming approach. Specifically, an investor who is very risk-averse may consider adopting the robust optimisation approach rather than the stochastic programming approach when deciding on his or her budget allocation.

Lately, the robust optimisation approach has been drawing a lot of attention from researchers in this field because of its tractability. We suggest reading from Ben-Tal, Goryashko, Guslitzer and Nemirovski [4], Ben-Tal and Nemirovski [6], Bertsimas, Pachamanova, and Sim [9], and Li and Floudas [29] for developing a more sophisticated understanding of the robust optimisation approach. A comprehensive review of robust optimisation can also be read from Ben-Tal, El Ghaoui and Nemirovski [3]. Also, application of the robust optimisation to portfolio optimisation problem can be reviewed from Ben-Tal, Margalit and Nemirovski [5] and Bertsimas and Pachamanova [8] as well as its application to the option pricing problem from Chen [15]. Furthermore, we refer the history enthusiast to Bandi and Bertsimas [2] for historical development of robust optimisation and its connection with probability theory and stochastic programming.

3.3.3 Decision Rule Approach

In this part, we review the concept of decision rule approach, another way to deal with uncertain optimisation problems. To begin with, we first emphasise that in dynamic optimisation problems, it would be desirable to model the decision variables as a function of previously observed information. This leads us to become aware of the major shortcoming of both stochastic programming approach with recourse actions and robust optimisation approach because of the following reasons.

- Tractability of the stochastic programming approach with recourse actions is severely affected by the number of decision stages. Even though the two-stage stochastic programming problems with recourse actions appear to be computationally tractable with Monte-Carlo sampling techniques (see Birge and Louveaux [11] and Ruszczyński and Shapiro [39]), to date there are still no tractable manners to deal with multi-stage recourse problems (see Shapiro and Nemirovski [42]).
- Traditional robust optimisation approach typically neglects possibilities of recourse actions. The generated robust solution is thus more susceptible to unnecessary conservatism than the true optimal solution. In other words, classical robust optimisation methodology employs the constant decision rules, i.e., values of the decision variables have to be decided here and now as a constant, and thus they are completely independent of the information not yet available.

The decision rule approach is another paradigm for decision making under uncertainty. Unlike other approximation methodologies, for example, Monte-Carlo sampling techniques, the decision rule approach enables the problem modeller to approximately solve the optimisation problems by selecting the analytical functional form of the recourse actions. Using some particular forms of recourse actions, the modeller obtains a tractable formulation of the optimisation problem of his or her interest while recourse actions are also taken into account.

The decision rule optimisation approach is relatively new. Studies in this area were pioneered by Ben-Tal, Goryashko, Guslitzer and Nemirovski [4] in 2004 when they introduced linear decision rules, approximation of the recourse action as a linear function of the uncertain parameters, in the context of robust optimisation approach, which were called adjustable robust solution in their work. Shortly afterwards, linear decision rules were proved applicable and useful as well in the stochastic programming paradigm (see Shapiro and Nemirovski [42]). However, the question about the loss of optimality when using linear decision rules remained, and subsequently it was addressed in Kuhn, Wiesemann and Georghiou [27] as they proposed a way to formulate the upper bound and the lower bound problems. The gap between both problems is a measure of how much confidence we should have in the obtained solution. The main advantage of using linear decision rules is that the resulting optimisation problem can usually be formulated as a tractable conic program. We refer the reader to Georghiou, Wiesemann and Kuhn [24] for an extensive and comprehensive review of the linear decision rules.

Despite its tractability and capability to consider recourse actions, in many cases, linear decision rules yield insufficiently accurate results. To address this issue, Chen, Sim, Sun and Zhang [16] proposed another approach called segregated linear decision rules which approximate recourse actions via piecewise linear decision rules. The idea was generalised by Georghiou, Wiesemann and Kuhn [23] showing that there exists a correspondence between non-linear decision rules and linear decision rules in the lifted space. The correspondence between both types of the decision rules implicitly implies the tractability of the piecewise linear decision rule approach. It is also proven that the piecewise linear decision rules are no less accurate than the linear decision rules in the sense that they provide a tighter upper and lower bounds to the true optimal solution. Apart from linear decision rules and piecewise linear decision rules, there are also other variations of decision rules, for example, polynomial decision rules which can be reviewed from Bampou and Kuhn [1].

3.4 Related Theories

In this section, we collect the optimisation theories which are strongly related to this thesis. The following theorem can be used to formulate a robust linear optimisation problem with polyhedral uncertainty set as a single deterministic linear programming problem, which is proved to be computationally tractable using simplex algorithm or interior-point method.

Theorem 3.4.1. (Pachamanova [37]) *Given an uncertain matrix $\tilde{A} \in \mathbb{R}^{m \times n}$, if a polyhedral uncertainty set P^A is a non-empty set given by $\{vec(\tilde{A}) \mid G \cdot vec(\tilde{A}) \leq d\}$ for some matrix $G \in \mathbb{R}^{l \times (m \times n)}$ and some vector $d \in \mathbb{R}^l$, then a given $\hat{x} \in \mathbb{R}^n$ satisfies the constraint $\tilde{a}_i \hat{x} \leq b_i$ for all $\tilde{A} \in P^A$ if and only if there exists a vector $p^i \in \mathbb{R}^l$ such that*

$$\begin{gathered}(p^i)^T d \leq b_i \\ (p^i)^T G = \hat{x}_i^T \\ p^i \geq 0,\end{gathered}$$

where $vec(\tilde{A})$ is a vector equivalent of matrix $\tilde{A}$ constructed by stacking the rows of matrix $\tilde{A}$ on top of one another, $\tilde{a}_i$ is the i^{th} row of the matrix $\tilde{A}$, and $\hat{x}_i$ is a vector in $\mathbb{R}^{(m \times n)}$ defined by the one containing $\hat{x}$ in entries $(i-1)n+1$ through $(i)(n)$ and zero elsewhere.

Proof.
Consider the following pair of primal and dual problems.

(Primal)

$$\begin{gathered}\text{minimise}_{p^i}\ (p^i)^T d \\ \text{subject to} \\ (p^i)^T G = \hat{x}_i^T \\ p^i \geq 0\end{gathered}$$

(Dual)

$$\begin{gathered}\text{maximise}_{\tilde{A}}\ \tilde{a}_i \hat{x} \\ \text{subject to} \\ G \cdot vec(\tilde{A}) \leq d\end{gathered}$$

($\Longrightarrow$) Suppose that there exists a vector $\hat{p}^i$ which is feasible in the primal problem and $(\hat{p}^i)^T d \leq b_i$. Equivalently, the primal problem is feasible, and since it is assumed that P^A is a non-empty set, it follows that the dual problem is feasible. By the unboundedness property of duality, it can be concluded that both primal and dual problems are feasible and bounded. By the strong duality property, the optimal objective function values of both problems must be equal.

$$\text{maximise}_{\tilde{A}}\ \tilde{a}_i \hat{x} = \text{minimise}_{p^i}\ (p^i)^T d \leq (\hat{p}^i)^T d \leq b_i$$

Hence, $\tilde{a}_i \hat{x} \leq b_i$ for all $\tilde{A} \in P^A$.

($\Longleftarrow$) Suppose that $\hat{x}_i$ satisfies the constraint $\tilde{a}_i \hat{x} \leq b_i$ for all $\tilde{A} \in P^A$. Then, the optimal objective function value of the dual problem, i.e., $\text{maximise}_{\tilde{A}}\ \tilde{a}_i \hat{x}$, is less than or equal to b_i, and therefore the dual problem is feasible and bounded. By the strong duality property, the optimal objective function values of the primal and the dual problems must coincide, and the proof of this direction thus completes.

□

Instead of using a polyhedral uncertainty set, sometimes it seems more appropriate, especially in a statistical sense, to use an ellipsoidal uncertainty set or an uncertainty set which can be written as an intersection of multiple ellipsoids. Similarly to Theorem 3.4.1, which uses duality in linear programming to formulate a deterministic version of a robust linear constraint associated with a polyhedral uncertainty set, duality in second-order cone programming provides a method for transforming a robust linear optimisation problem with discussed types of uncertainty set into a deterministic second-order cone programming problem, which still offers a good degree of scalability.

Last but not least, the final part of this section shows that even if the robust optimisation problem is not linear, it is still possible to determine a tractable deterministic version of the problem. Specifically, if the constraint is quadratic in the uncertain parameters x, the corresponding (exact or approximate) deterministic optimisation problem is then a semidefinite program.

Theorem 3.4.2. (S-lemma) *Given two symmetric matrices $W \in \mathbb{S}^n$ and $S \in \mathbb{S}^n$ where the inequality $x^T W x \geq 0$ is strictly feasible, i.e., there exists a vector $\hat{x} \in \mathbb{R}^n$ that $\hat{x}^T W \hat{x} > 0$, then the statement*

$$x^T W x \geq 0 \quad \Longrightarrow \quad x^T S x \geq 0$$

is true if and only if there exists a non-negative scalar λ such that $S \succeq \lambda W$.

Proposition 3.4.1. (Approximate S-lemma) *Given $S \in \mathbb{S}^n$ and $\{W_i\}_{i=1}^I \in \mathbb{S}^n$, the statement*

$$x^T W_i x \geq 0,\ \forall i = 1, 2, ..., I \quad \Longrightarrow \quad x^T S x \geq 0$$

holds when there exists a vector $\lambda \in \mathbb{R}_+^I$ such that $S - \sum_{i=1}^{I} \lambda_i W_i \succeq 0$.

Proofs of the theorem and the proposition are beyond the scope of this project, and therefore they are not discussed here. However, they can be found in many other literatures, for instance, Kuhn, Wiesemann and Georghiou [27] and Pólik and Terlaky [38]. It is, however, worth noting here that resulting from the S-lemma the reverse of the proposition 3.4.1 also holds when $I = 1$. The following proposition is another subtle case where the approximate S-lemma also becomes exact, i.e., the reverse statement holds.

Proposition 3.4.2. (Wiesemann, Kuhn and Rustem [49]) *For a set Ξ defined by*

$$\Xi = \{\xi \in \mathbb{R}^q \mid \xi^T O_l \xi + \xi^T o_l + \omega_l \geq 0, \quad \forall l = 1, 2, ..., L\}, \tag{3.36}$$

where $O_l \in \mathbb{S}^q$, $O_l \preceq 0$, $o_l \in \mathbb{R}^q$, and $\omega \in \mathbb{R}$, and for a matrix $S \in \mathbb{S}^q$, a vector $s \in \mathbb{R}^q$, and $\sigma \in \mathbb{R}$, if $S \succeq 0$, the following statements are equivalent.

1. $\exists \lambda \in \mathbb{R}_+^l$, $\begin{bmatrix} \sigma & \frac{1}{2}s^T \\ \frac{1}{2}s & S \end{bmatrix} - \sum_{l=1}^{L} \lambda_l \begin{bmatrix} \omega_l & \frac{1}{2}o_l^T \\ \frac{1}{2}o_l & O_l \end{bmatrix} \succeq 0.$
2. $\xi^T S \xi + \xi^T s + \sigma \geq 0,\ \forall \xi \in \Xi.$

3.5 Conclusions

In a few words, we discuss deterministic optimisation and optimisation under uncertainty in this chapter. Deterministic optimisation models, such as deterministic linear programs, usually cannot deal with the uncertain optimisation problems very well. However, they are still of great importance because they are efficiently solvable by the standard optimisation solvers. In financial applications, most optimisation problems are subject to uncertainty because there are typical uncertain factors, for example, assets' returns. The ϵ-arbitrage robust pricing model uses the robust optimisation approach to model the option pricing problem. The robust optimisation approach is one of the two competing techniques to deal with uncertainty in the optimisation problems. Therefore, we sometimes refer to the ϵ-arbitrage pricing model as a robust pricing model. The robust pricing model is essentially a robust linear optimisation problem. Its crude structure seems to be intractable because the uncertainty set associated with the robust pricing model contains infinitely many elements. Several techniques are then given to transform the robust optimisation problem into its deterministic equivalent or deterministic approximation. These techniques are the duality in linear programming, the duality in second-order cone programming, and the (approximate) S-lemma. For example, we can use the duality in linear programming to find a deterministic linear program corresponding to the robust linear optimisation problem whose uncertainty set is a polyhedron. This will be clearer when we investigate the ϵ-arbitrage robust pricing model in Chapter 4.

Chapter 4

Valuation of Single-Underlier Options

In this chapter, we focus on the valuation of fundamental European-style options: European options, Asian options, and lookback options, and fundamental American-style options: American call options and American put options. We mathematically derive the models for pricing these options based on the idea of the ϵ-arbitrage robust pricing model.

We begin by introducing an uncertainty model used for describing returns of the underlying asset in the future stages. This step is important in formulating the robust optimisation model as discussed in the previous chapter. We, then, investigate the valuation of a European call option, which is the simplest type of financial options and is well-understood, followed by the valuation of the other aforementioned types of financial options.

Note that main ideas in this chapter are accredited to Chen [15]. The resulting linear programs for pricing options are slightly different from the ones suggested by Chen in his original proposal though. To the best of our knowledge, this might be a matter of problem formulation. Furthermore, Chen developed his pricing model and tested the performance of the model by identifying the similarities between the output prices and the observed market prices. In this work, our interest is, however, also from the theoretical outlook. To this end, we assume that the asset price follows the geometric Brownian motion. This assumption is not necessary. The robust pricing model and the Black-Scholes model, if available, are then compared.

4.1 Uncertainty Model

As already mentioned in Chapter 2, Chen [15] proposed a way to construct a set of boundaries on both single-period returns (2.16) and cumulative returns (2.21). For simplicity, let $\underline{r}_t^s$ and $\bar{r}_t^s$ be the lower bound and the upper bound of the single-period return of the underlying asset at time t, respectively. Similarly, let $\underline{R}_t^s$ and $\bar{R}_t^s$ be the lower bound and the upper bound of the cumulative return of the underlying asset at time t, respectively. Therefore, the boundaries can be represented as follows.

Uncertainty Constraints

$$\begin{aligned}
-\tilde{R}_t^s &\leq -\underline{R}_t^s, & \forall t = 1, 2, ..., T \\
\tilde{R}_t^s &\leq \bar{R}_t^s, & \forall t = 1, 2, ..., T \\
-\tilde{R}_t^s &\leq -\tilde{R}_{t-1}^s(1 + \underline{r}_{t-1}^s), & \forall t = 1, 2, ..., T \\
\tilde{R}_t^s &\leq \tilde{R}_{t-1}^s(1 + \bar{r}_{t-1}^s), & \forall t = 1, 2, ..., T
\end{aligned} \tag{4.1}$$

For the sake of convenience, the values of $\tilde{R}_0^s$ and R_0^B can be, without loss of generality, set to one. If the returns $\{\tilde{R}_t^s\}_{t=1}^T$ satisfy the constraints (4.1), we say that $\{\tilde{R}_t^s\}_{t=1}^T$ lies in the uncertainty set U.

All of the constraints can be further put together into a single matrix inequality in the form of $G \cdot vec(\tilde{A}) \leq d$, which will make it easier to determine a deterministic equivalent of the robust pricing optimisation problem using Theorem 3.4.1.

$$\begin{bmatrix} -1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 0 & \dots & 0 \\ 0 & 0 & -1 & \dots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \dots & -1 \\ \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \dots & 1 \\ \\ -1 & 0 & 0 & \dots & 0 \\ 1+\underline{r}_1^s & -1 & 0 & \dots & 0 \\ 0 & 1+\underline{r}_2^s & -1 & \dots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \dots & -1 \\ \\ 1 & 0 & 0 & \dots & 0 \\ -1-\bar{r}_1^s & 1 & 0 & \dots & 0 \\ 0 & -1-\bar{r}_2^s & 1 & \dots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} \tilde{R}_1^s \\ \tilde{R}_2^s \\ \vdots \\ \tilde{R}_T^s \end{bmatrix} \leq \begin{bmatrix} -\underline{R}_1^s \\ -\underline{R}_2^s \\ -\underline{R}_3^s \\ \vdots \\ -\underline{R}_T^s \\ \\ \bar{R}_1^s \\ \bar{R}_2^s \\ \bar{R}_3^s \\ \vdots \\ \bar{R}_T^s \\ \\ -1-\underline{r}_0^s \\ 0 \\ 0 \\ \vdots \\ 0 \\ \\ 1+\bar{r}_0^s \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4.2}$$

4.2 ϵ-Arbitrage Model Simplification

The ϵ-arbitrage robust pricing model (2.14) can be further simplified by introducing three series of new variables defined below.

$$\alpha_t^S = \frac{x_t^S}{\tilde{R}_t^S} \tag{4.3}$$

$$\alpha_t^B = \frac{x_t^B}{R_t^B} \tag{4.4}$$

$$\beta_t = \frac{y_t}{\tilde{R}_t^S} \tag{4.5}$$

The following steps demonstrate how to transform the constraints in the pricing model (2.14) into a linear recursive form.

$$
\begin{aligned}
& x_t^S = (1+\tilde{r}_{t-1}^S)(x_{t-1}^S + y_{t-1}), && \forall t = 1,2,...,T \\
\Longrightarrow\quad & x_t^S = \frac{\tilde{R}_t^S}{\tilde{R}_{t-1}^S}(x_{t-1}^S + y_{t-1}), && \forall t = 1,2,...,T \\
\Longrightarrow\quad & \frac{x_t^S}{\tilde{R}_t^S} = \frac{x_{t-1}^S}{\tilde{R}_{t-1}^S} + \frac{y_{t-1}}{\tilde{R}_{t-1}^S}, && \forall t = 1,2,...,T \\
\Longrightarrow\quad & \alpha_t^S = \alpha_{t-1}^S + \beta_{t-1}, && \forall t = 1,2,...,T
\end{aligned}
\tag{4.6}
$$

$$
\begin{aligned}
& x_t^B = (1+r_{t-1}^B)(x_{t-1}^B - y_{t-1}), && \forall t = 1,2,...,T \\
\Longrightarrow\quad & x_t^B = \frac{R_t^B}{R_{t-1}^B}(x_{t-1}^B - y_{t-1}), && \forall t = 1,2,...,T \\
\Longrightarrow\quad & \frac{x_t^B}{R_t^B} = \frac{x_{t-1}^B}{R_{t-1}^B} - \frac{y_{t-1}}{R_{t-1}^B}, && \forall t = 1,2,...,T \\
\Longrightarrow\quad & \alpha_t^B = \alpha_{t-1}^B - \beta_{t-1}\frac{\tilde{R}_{t-1}^S}{R_{t-1}^B} && \forall t = 1,2,...,T
\end{aligned}
\tag{4.7}
$$

Using the equations (4.6) and (4.7), one can achieve the following.

$$
\alpha_T^S = \alpha_0^S + \sum_{t=0}^{T-1} \beta_t \tag{4.8}
$$

$$
\alpha_T^B = \alpha_0^B - \sum_{t=0}^{T-1} \beta_t \frac{\tilde{R}_t^S}{R_t^B} \tag{4.9}
$$

$$
W_T = x_T^S + x_T^B = \alpha_T^S \tilde{R}_T^S + \alpha_T^B R_T^B \tag{4.10}
$$

The ϵ-arbitrage robust pricing model (2.14) can then be reduced to

Simplified ϵ-Arbitrage Robust Pricing Model

$$
\begin{gathered}
\text{minimise}_{\alpha_0^S, \alpha_0^B, \{\beta_t\}_{t=0}^{T-1}} \text{maximise}_{\{\tilde{R}_t^S\}_{t=1}^{T} \in U} \\
\left| P(\tilde{S}, K) - \left(\alpha_0^S + \sum_{t=0}^{T-1} \beta_t\right) \tilde{R}_T^S - \left(\alpha_0^B - \sum_{t=0}^{T-1} \beta_t \frac{\tilde{R}_t^S}{R_t^B}\right) R_T^B \right|.
\end{gathered}
\tag{4.11}
$$

4.3 ϵ-Arbitrage Model Derivation for European-Style Options

In this section, we apply the idea of the ϵ-arbitrage robust pricing model to valuate European-style options. Three types of options are considered here: European options, Asian options, and fixed strike lookback options.

4.3.1 European Options

Using the simplified ϵ-arbitrage robust pricing model (4.11) and substituting the function

$$\left(\tilde{S}_T - K\right)^+ = \left(S_0 \tilde{R}_T^S - K\right)^+ \tag{4.12}$$

for the payoff function $P(\tilde{S}, K)$, the model for pricing a European call option is thus given by

European Call Option Pricing: ϵ-Arbitrage Formulation

$$\text{minimise}_{\alpha_0^S, \alpha_0^B, \{\beta_t\}_{t=0}^{T-1}} \text{maximise}_{\{\tilde{R}_t^S\}_{t=1}^T \in U}$$
$$\left| \left(S_0 \tilde{R}_T^S - K\right)^+ - \left(\alpha_0^S + \sum_{t=0}^{T-1} \beta_t\right) \tilde{R}_T^S - \left(\alpha_0^B - \sum_{t=0}^{T-1} \beta_t \frac{\tilde{R}_t^S}{R_t^B}\right) R_T^B \right|. \tag{4.13}$$

The pricing model (4.13) can be further transformed into the following optimisation problem considering the two possibilities of the payoff function and the two possibilities of the absolute value function. Observe that the resulting pricing model below has a set of constraints which are linear in the uncertain parameters $\{\tilde{R}_t^S\}_{t=1}^T$.

$$\begin{aligned}
&\text{minimise}_{\alpha_0^S, \alpha_0^B, \{\beta_t\}_{t=0}^{T-1}, \epsilon} \quad \epsilon \\
&\text{subject to} \\
&\left(S_0 \tilde{R}_T^S - K\right) - \left(\alpha_0^S + \sum_{t=0}^{T-1} \beta_t\right) \tilde{R}_T^S - \left(\alpha_0^B - \sum_{t=0}^{T-1} \beta_t \frac{\tilde{R}_t^S}{R_t^B}\right) R_T^B \le \epsilon, \quad \forall \{\tilde{R}_t^S\}_{t=1}^T \in U, \tilde{R}_T^S \ge \frac{K}{S_0} \\
&-\left(S_0 \tilde{R}_T^S - K\right) + \left(\alpha_0^S + \sum_{t=0}^{T-1} \beta_t\right) \tilde{R}_T^S + \left(\alpha_0^B - \sum_{t=0}^{T-1} \beta_t \frac{\tilde{R}_t^S}{R_t^B}\right) R_T^B \le \epsilon, \quad \forall \{\tilde{R}_t^S\}_{t=1}^T \in U, \tilde{R}_T^S \ge \frac{K}{S_0} \\
&-\left(\alpha_0^S + \sum_{t=0}^{T-1} \beta_t\right) \tilde{R}_T^S - \left(\alpha_0^B - \sum_{t=0}^{T-1} \beta_t \frac{\tilde{R}_t^S}{R_t^B}\right) R_T^B \le \epsilon, \quad \forall \{\tilde{R}_t^S\}_{t=1}^T \in U, \tilde{R}_T^S \le \frac{K}{S_0} \\
&\left(\alpha_0^S + \sum_{t=0}^{T-1} \beta_t\right) \tilde{R}_T^S + \left(\alpha_0^B - \sum_{t=0}^{T-1} \beta_t \frac{\tilde{R}_t^S}{R_t^B}\right) R_T^B \le \epsilon, \quad \forall \{\tilde{R}_t^S\}_{t=1}^T \in U, \tilde{R}_T^S \le \frac{K}{S_0}
\end{aligned} \tag{4.14}$$

Theorem 3.4.1 can be used to transform each uncertainty-affected constraint in the European call option pricing model (4.14) to a set of deterministic linear constraints. To do so, we have to formulate a matrix inequality for describing the uncertain factors. For example, consider the first constraint, we can combine a condition $\forall \{\tilde{R}_t^S\}_{t=1}^T \in U$, which is already written in a form of matrix inequality (4.2), and $\tilde{R}_T^S \ge \frac{K}{S_0}$, and then introduce dual variables appropriately. The result of this transformation is shown here.

$$
\begin{bmatrix}
-1 & 0 & 0 & \dots & 0 \\
0 & -1 & 0 & \dots & 0 \\
0 & 0 & -1 & \dots & 0 \\
 & & \vdots & & \\
0 & 0 & 0 & \dots & -1 \\
1 & 0 & 0 & \dots & 0 \\
0 & 1 & 0 & \dots & 0 \\
0 & 0 & 1 & \dots & 0 \\
 & & \vdots & & \\
0 & 0 & 0 & \dots & 1 \\
-1 & 0 & 0 & \dots & 0 \\
1+\underline{r}^s_1 & -1 & 0 & \dots & 0 \\
0 & 1+\underline{r}^s_2 & -1 & \dots & 0 \\
 & & \vdots & & \\
0 & 0 & 0 & \dots & -1 \\
1 & 0 & 0 & \dots & 0 \\
-1-\bar{r}^s_1 & 1 & 0 & \dots & 0 \\
0 & -1-\bar{r}^s_2 & 1 & \dots & 0 \\
 & & \vdots & & \\
0 & 0 & 0 & \dots & 1 \\
0 & 0 & 0 & \dots & -1
\end{bmatrix}
\begin{bmatrix}
\tilde{R}^s_1 \\ \tilde{R}^s_2 \\ \vdots \\ \tilde{R}^s_T
\end{bmatrix}
\leq
\begin{bmatrix}
-\underline{R}^s_1 \\ -\underline{R}^s_2 \\ -\underline{R}^s_3 \\ \vdots \\ -\underline{R}^s_T \\
\bar{R}^s_1 \\ \bar{R}^s_2 \\ \bar{R}^s_3 \\ \vdots \\ \bar{R}^s_T \\
-1-\underline{r}^s_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\
1+\bar{r}^s_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\
-K/S_0
\end{bmatrix}
\begin{array}{l}
\to -p_{1,1} \\ \to -p_{1,2} \\ \to -p_{1,3} \\ \vdots \\ \to -p_{1,T} \\
\to +q_{1,1} \\ \to +q_{1,2} \\ \to +q_{1,3} \\ \vdots \\ \to +q_{1,T} \\
\to -m_{1,1} \\ \to -m_{1,2} \\ \to -m_{1,3} \\ \vdots \\ \to -m_{1,T} \\
\to +n_{1,1} \\ \to +n_{1,2} \\ \to +n_{1,3} \\ \vdots \\ \to +n_{1,T} \\
\to -z_1
\end{array}
\tag{4.15}
$$

The variables $\{p_{1,i}\}_{i=1}^T, \{q_{1,i}\}_{i=1}^T, \{m_{1,i}\}_{i=1}^T, \{n_{1,i}\}_{i=1}^T$, and z_1 in (4.15) are dual variables associated with the uncertainty set of the first constraint in the European call option pricing model (4.14). Repeating the same procedure for the remaining constraints and applying Theorem 3.4.1, we obtain a deterministic linear programming problem displayed in the following page.

It can be seen that the size of the linear program used for pricing a European call option grows linearly with the number of time periods T. The same idea can also be used to price European put options as well since the payoff function of European put option

$$\left(K - \tilde{S}_T\right)^+ = \left(K - S_0\tilde{R}^S_T\right)^+ \tag{4.16}$$

is not much analytically different from European call option. By replacing $P(\tilde{S}, K)$ in the simplified ϵ-arbitrage robust pricing model (4.11) with the payoff function of the European put option, a corresponding deterministic linear program will be of the same size as the linear program used for pricing a European call option.

In the rest of this section, only call options are considered because the essence of the model derivation for a put option is the same as that for a call option. By using the same idea, the pricing model for a particular put option can be readily constructed.

European Call Option Pricing: Linear Equivalent Formulation

minimise$_{\alpha_0^S,\alpha_0^B,\{\beta_t\}_{t=0}^{T-1},\epsilon,\{p_{c,t},q_{c,t},m_{c,t},n_{c,t},z_c\}_{c=1,t=1}^{c=4,t=T}}$ ϵ

subject to

case I

$$\left(\sum_{t=1}^{T} p_{1,t}\underline{R}_t^S\right) + \left(\sum_{t=1}^{T} q_{1,t}\bar{R}_t^S\right) + \left(1+\underline{r}_0^S\right) m_{1,1} + \left(1+\bar{r}_0^S\right) n_{1,1} + \frac{z_1 K}{S_0} \le \epsilon + K + \alpha_0^B R_T^B - \beta_0 R_T^B$$

$$p_{1,t} + q_{1,t} + m_{1,t} + n_{1,t} - \left(1+\underline{r}_t^S\right) m_{1,t+1} - \left(1+\bar{r}_t^S\right) n_{1,t+1} = \beta t \frac{R_T^B}{R_t^B}, \quad \forall t = 1,2,...,T-1$$

$$p_{1,T} + q_{1,T} + m_{1,T} + n_{1,T} + z_1 = S_0 - \alpha_0^S - \left(\sum_{t=0}^{T-1} \beta_t\right)$$

$$p_{1,t} \le 0,\ q_{1,t} \ge 0,\ m_{1,t} \le 0,\ n_{1,t} \ge 0, \quad \forall t = 1,2,...,T, \qquad z_1 \le 0$$

case II

$$\left(\sum_{t=1}^{T} p_{2,t}\underline{R}_t^S\right) + \left(\sum_{t=1}^{T} q_{2,t}\bar{R}_t^S\right) + \left(1+\underline{r}_0^S\right) m_{2,1} + \left(1+\bar{r}_0^S\right) n_{2,1} + \frac{z_2 K}{S_0} \le \epsilon - K - \alpha_0^B R_T^B + \beta_0 R_T^B$$

$$p_{2,t} + q_{2,t} + m_{2,t} + n_{2,t} - \left(1+\underline{r}_t^S\right) m_{2,t+1} - \left(1+\bar{r}_t^S\right) n_{2,t+1} = -\beta t \frac{R_T^B}{R_t^B}, \quad \forall t = 1,2,...,T-1$$

$$p_{2,T} + q_{2,T} + m_{2,T} + n_{2,T} + z_2 = -S_0 + \alpha_0^S + \left(\sum_{t=0}^{T-1} \beta_t\right)$$

$$p_{2,t} \le 0,\ q_{2,t} \ge 0,\ m_{2,t} \le 0,\ n_{2,t} \ge 0, \quad \forall t = 1,2,...,T, \qquad z_2 \le 0$$

case III

$$\left(\sum_{t=1}^{T} p_{3,t}\underline{R}_t^S\right) + \left(\sum_{t=1}^{T} q_{3,t}\bar{R}_t^S\right) + \left(1+\underline{r}_0^S\right) m_{3,1} + \left(1+\bar{r}_0^S\right) n_{3,1} + \frac{z_3 K}{S_0} \le \epsilon + \alpha_0^B R_T^B - \beta_0 R_T^B$$

$$p_{3,t} + q_{3,t} + m_{3,t} + n_{3,t} - \left(1+\underline{r}_t^S\right) m_{3,t+1} - \left(1+\bar{r}_t^S\right) n_{3,t+1} = \beta t \frac{R_T^B}{R_t^B}, \quad \forall t = 1,2,...,T-1$$

$$p_{3,T} + q_{3,T} + m_{3,T} + n_{3,T} + z_3 = -\alpha_0^S - \left(\sum_{t=0}^{T-1} \beta_t\right)$$

$$p_{3,t} \le 0,\ q_{3,t} \ge 0,\ m_{3,t} \le 0,\ n_{3,t} \ge 0, \quad \forall t = 1,2,...,T, \qquad z_3 \ge 0$$

case IV

$$\left(\sum_{t=1}^{T} p_{4,t}\underline{R}_t^S\right) + \left(\sum_{t=1}^{T} q_{4,t}\bar{R}_t^S\right) + \left(1+\underline{r}_0^S\right) m_{4,1} + \left(1+\bar{r}_0^S\right) n_{4,1} + \frac{z_4 K}{S_0} \le \epsilon - \alpha_0^B R_T^B + \beta_0 R_T^B$$

$$p_{4,t} + q_{4,t} + m_{4,t} + n_{4,t} - \left(1+\underline{r}_t^S\right) m_{4,t+1} - \left(1+\bar{r}_t^S\right) n_{4,t+1} = -\beta t \frac{R_T^B}{R_t^B}, \quad \forall t = 1,2,...,T-1$$

$$p_{4,T} + q_{4,T} + m_{4,T} + n_{4,T} + z_4 = \alpha_0^S + \left(\sum_{t=0}^{T-1} \beta_t\right)$$

$$p_{4,t} \le 0,\ q_{4,t} \ge 0,\ m_{4,t} \le 0,\ n_{4,t} \ge 0, \quad \forall t = 1,2,...,T, \qquad z_4 \ge 0$$

4.3.2 Asian Options

In this part, we consider an Asian call option or an average call option which is defined in terms of arithmetic average. One can derive a pricing model for arithmetic Asian call options by substituting the payoff function

$$\left(\frac{1}{T}\sum_{t=1}^{T}\tilde{S}_t - K\right)^+ = \left(\frac{S_0}{T}\sum_{t=1}^{T}\tilde{R}_t^S - K\right)^+ \tag{4.17}$$

for $P(\tilde{S}, K)$ in the simplified ϵ-arbitrage robust pricing model (4.11).

Asian Call Option Pricing: ϵ-Arbitrage Formulation

$$\begin{gathered}\text{minimise}_{\alpha_0^S,\alpha_0^B,\{\beta_t\}_{t=0}^{T-1}}\text{maximise}_{\{\tilde{R}_t^S\}_{t=1}^T\in U}\\ \left|\left(\frac{S_0}{T}\sum_{t=1}^{T}\tilde{R}_t^S - K\right)^+ - \left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}_T^S - \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}_t^S}{R_t^B}\right)R_T^B\right|\end{gathered} \tag{4.18}$$

Again by considering the two possibilities of the payoff function and the two possibilities of the absolute value function, one can obtain a pricing model below whose constraints are linear in the uncertain parameters $\{\tilde{R}_t^S\}_{t=1}^T$. Its linear deterministic equivalent is shown in the next page.

$$\begin{aligned}
&\text{minimise}_{\alpha_0^S,\alpha_0^B,\{\beta_t\}_{t=0}^{T-1},\epsilon} \quad \epsilon\\
&\text{subject to}\\
&\left(\frac{S_0}{T}\sum_{t=1}^{T}\tilde{R}_t^S - K\right) - \left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}_T^S - \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}_t^S}{R_t^B}\right)R_T^B \le \epsilon,\\
&\qquad\qquad \forall\{\tilde{R}_t^S\}_{t=1}^T \in U,\ \frac{\sum_{t=1}^{T}\tilde{R}_t^S}{T} \ge \frac{K}{S_0}\\
&-\left(\frac{S_0}{T}\sum_{t=1}^{T}\tilde{R}_t^S - K\right) + \left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}_T^S + \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}_t^S}{R_t^B}\right)R_T^B \le \epsilon,\\
&\qquad\qquad \forall\{\tilde{R}_t^S\}_{t=1}^T \in U,\ \frac{\sum_{t=1}^{T}\tilde{R}_t^S}{T} \ge \frac{K}{S_0}\\
&-\left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}_T^S - \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}_t^S}{R_t^B}\right)R_T^B \le \epsilon,\\
&\qquad\qquad \forall\{\tilde{R}_t^S\}_{t=1}^T \in U,\ \frac{\sum_{t=1}^{T}\tilde{R}_t^S}{T} \le \frac{K}{S_0}\\
&\left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}_T^S + \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}_t^S}{R_t^B}\right)R_T^B \le \epsilon,\\
&\qquad\qquad \forall\{\tilde{R}_t^S\}_{t=1}^T \in U,\ \frac{\sum_{t=1}^{T}\tilde{R}_t^S}{T} \le \frac{K}{S_0}
\end{aligned} \tag{4.19}$$

Asian Call Option Pricing: Linear Equivalent Formulation

minimise$_{\alpha_0^S,\alpha_0^B,\{\beta_t\}_{t=0}^{T-1},\epsilon,\{p_{c,t},q_{c,t},m_{c,t},n_{c,t},z_c\}_{c=1,t=1}^{c=4,t=T}}$ $\quad \epsilon$

subject to

case I

$$\left(\sum_{t=1}^{T} p_{1,t}\underline{R}_t^S\right) + \left(\sum_{t=1}^{T} q_{1,t}\bar{R}_t^S\right) + \left(1+\underline{r}_0^S\right)m_{1,1} + \left(1+\bar{r}_0^S\right)n_{1,1} + \frac{z_1 K}{S_0} \leq \epsilon + K + \alpha_0^B R_T^B - \beta_0 R_T^B$$

$$p_{1,t} + q_{1,t} + m_{1,t} + n_{1,t} - \left(1+\underline{r}_t^S\right)m_{1,t+1} - \left(1+\bar{r}_t^S\right)n_{1,t+1} + \frac{z_1}{T} = \beta t\frac{R_T^B}{R_t^B} + \frac{S_0}{T}, \quad \forall t = 1,2,...,T-1$$

$$p_{1,T} + q_{1,T} + m_{1,T} + n_{1,T} + \frac{z_1}{T} = \frac{S_0}{T} - \alpha_0^S - \left(\sum_{t=0}^{T-1}\beta_t\right)$$

$$p_{1,t} \leq 0,\ q_{1,t} \geq 0,\ m_{1,t} \leq 0,\ n_{1,t} \geq 0, \quad \forall t = 1,2,...,T, \qquad z_1 \leq 0$$

case II

$$\left(\sum_{t=1}^{T} p_{2,t}\underline{R}_t^S\right) + \left(\sum_{t=1}^{T} q_{2,t}\bar{R}_t^S\right) + \left(1+\underline{r}_0^S\right)m_{2,1} + \left(1+\bar{r}_0^S\right)n_{2,1} + \frac{z_2 K}{S_0} \leq \epsilon - K - \alpha_0^B R_T^B + \beta_0 R_T^B$$

$$p_{2,t} + q_{2,t} + m_{2,t} + n_{2,t} - \left(1+\underline{r}_t^S\right)m_{2,t+1} - \left(1+\bar{r}_t^S\right)n_{2,t+1} + \frac{z_2}{T} = -\beta t\frac{R_T^B}{R_t^B} - \frac{S_0}{T}, \quad \forall t = 1,2,...,T-1$$

$$p_{2,T} + q_{2,T} + m_{2,T} + n_{2,T} + \frac{z_2}{T} = -\frac{S_0}{T} + \alpha_0^S + \left(\sum_{t=0}^{T-1}\beta_t\right)$$

$$p_{2,t} \leq 0,\ q_{2,t} \geq 0,\ m_{2,t} \leq 0,\ n_{2,t} \geq 0, \quad \forall t = 1,2,...,T, \qquad z_2 \leq 0$$

case III

$$\left(\sum_{t=1}^{T} p_{3,t}\underline{R}_t^S\right) + \left(\sum_{t=1}^{T} q_{3,t}\bar{R}_t^S\right) + \left(1+\underline{r}_0^S\right)m_{3,1} + \left(1+\bar{r}_0^S\right)n_{3,1} + \frac{z_3 K}{S_0} \leq \epsilon + \alpha_0^B R_T^B - \beta_0 R_T^B$$

$$p_{3,t} + q_{3,t} + m_{3,t} + n_{3,t} - \left(1+\underline{r}_t^S\right)m_{3,t+1} - \left(1+\bar{r}_t^S\right)n_{3,t+1} + \frac{z_3}{T} = \beta t\frac{R_T^B}{R_t^B}, \quad \forall t = 1,2,...,T-1$$

$$p_{3,T} + q_{3,T} + m_{3,T} + n_{3,T} + \frac{z_3}{T} = -\alpha_0^S - \left(\sum_{t=0}^{T-1}\beta_t\right)$$

$$p_{3,t} \leq 0,\ q_{3,t} \geq 0,\ m_{3,t} \leq 0,\ n_{3,t} \geq 0, \quad \forall t = 1,2,...,T, \qquad z_3 \geq 0$$

case IV

$$\left(\sum_{t=1}^{T} p_{4,t}\underline{R}_t^S\right) + \left(\sum_{t=1}^{T} q_{4,t}\bar{R}_t^S\right) + \left(1+\underline{r}_0^S\right)m_{4,1} + \left(1+\bar{r}_0^S\right)n_{4,1} + \frac{z_4 K}{S_0} \leq \epsilon - \alpha_0^B R_T^B + \beta_0 R_T^B$$

$$p_{4,t} + q_{4,t} + m_{4,t} + n_{4,t} - \left(1+\underline{r}_t^S\right)m_{4,t+1} - \left(1+\bar{r}_t^S\right)n_{4,t+1} + \frac{z_4}{T} = -\beta t\frac{R_T^B}{R_t^B}, \quad \forall t = 1,2,...,T-1$$

$$p_{4,T} + q_{4,T} + m_{4,T} + n_{4,T} + \frac{z_4}{T} = \alpha_0^S + \left(\sum_{t=0}^{T-1}\beta_t\right)$$

$$p_{4,t} \leq 0,\ q_{4,t} \geq 0,\ m_{4,t} \leq 0,\ n_{4,t} \geq 0, \quad \forall t = 1,2,...,T, \qquad z_4 \geq 0$$

4.3.3 Lookback Options

Among a variety of lookback call options, we consider in this part fixed strike lookback call options. A fixed strike lookback call option is similar to a European call option in the sense that it has a fixed strike price K; however, rather than being dependent on the difference between the asset price at expiry and the strike price, payoff of a lookback call option is made based on the difference between the optimal asset price achieved during the option's lifetime and the strike price. Mathematically speaking, payoff of a fixed strike lookback call option is defined as

$$(S_{max} - K)^+ = \left(S_0 \tilde{R}^S_{max} - K\right)^+, \tag{4.20}$$

where $\tilde{R}^S_{max} = \max_{t=1,2,...,T} \tilde{R}^S_t$.

Substituting this payoff function for $P(\tilde{S}, K)$ in the simplified ϵ-arbitrage robust pricing model (4.11), we obtain a robust pricing model for fixed strike lookback call options.

Fixed Strike Lookback Call Option Pricing: ϵ-Arbitrage Formulation

$$\text{minimise}_{\alpha_0^S, \alpha_0^B, \{\beta_t\}_{t=0}^{T-1}} \text{maximise}_{\{\tilde{R}_t^S\}_{t=1}^T \in U}$$
$$\left| \left(S_0 \tilde{R}^S_{max} - K\right)^+ - \left(\alpha_0^S + \sum_{t=0}^{T-1} \beta_t\right) \tilde{R}^S_T - \left(\alpha_0^B - \sum_{t=0}^{T-1} \beta_t \frac{\tilde{R}^S_t}{R^B_t}\right) R^B_T \right| \tag{4.21}$$

The following pricing model (4.22) is equivalent to the model (4.21). The equivalence can be observed by enumerating all possible values of $\tilde{R}^S_{max}$ and considering the possibilities of the payoff function and the absolute value function.

By applying Theorem 3.4.1, the robust pricing model (4.22) can be further transformed into a single deterministic linear programming problem. The size of the resulting linear program still grows polynomially with the number of time periods T, which highlights the tractability of the ϵ-arbitrage robust pricing approach.

Since the linear deterministic equivalent has a fairly large number of constraints and seems a little bit cumbersome, we divide a set of constraints into $4T$ subsets. Each subset is denoted by t.I, t.II, t.III, or t.IV, where t $\in \{1, 2, ..., T\}$. The number t indicates that the subset is associated with the case where $\tilde{R}^S_{max} = \tilde{R}^S_t$. Moreover, it is seen from the pricing model (4.22) that for each value of t, there are four associated constraints. They are denoted by t.I, t.II, t.III, and t.IV in the linear equivalent formulation.

$$\text{minimise}_{\alpha_0^S, \alpha_0^B, \{\beta_t\}_{t=0}^{T-1}, \epsilon} \quad \epsilon$$
$$\text{subject to}$$

If $\tilde{R}^S_{max} = \tilde{R}^S_1$, i.e., $\tilde{R}^S_1 \geq \tilde{R}^S_t, \quad \forall t \in \{1, 2, ..., T\}$:

$$\begin{aligned}
\left(S_0\tilde{R}^S_1 - K\right) - \left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}^S_T - \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}^S_t}{R^B_t}\right)R^B_T &\leq \epsilon, \quad \forall\{\tilde{R}^S_t\}_{t=1}^T \in U, \tilde{R}^S_1 \geq \frac{K}{S_0} \\
-\left(S_0\tilde{R}^S_1 - K\right) + \left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}^S_T + \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}^S_t}{R^B_t}\right)R^B_T &\leq \epsilon, \quad \forall\{\tilde{R}^S_t\}_{t=1}^T \in U, \tilde{R}^S_1 \geq \frac{K}{S_0} \\
-\left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}^S_T - \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}^S_t}{R^B_t}\right)R^B_T &\leq \epsilon, \quad \forall\{\tilde{R}^S_t\}_{t=1}^T \in U, \tilde{R}^S_1 \leq \frac{K}{S_0} \\
\left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}^S_T + \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}^S_t}{R^B_t}\right)R^B_T &\leq \epsilon, \quad \forall\{\tilde{R}^S_t\}_{t=1}^T \in U, \tilde{R}^S_1 \leq \frac{K}{S_0}
\end{aligned}$$

If $\tilde{R}^S_{max} = \tilde{R}^S_2$, i.e., $\tilde{R}^S_2 \geq \tilde{R}^S_t, \quad \forall t \in \{1, 2, ..., T\}$:

$$\begin{aligned}
\left(S_0\tilde{R}^S_2 - K\right) - \left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}^S_T - \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}^S_t}{R^B_t}\right)R^B_T &\leq \epsilon, \quad \forall\{\tilde{R}^S_t\}_{t=1}^T \in U, \tilde{R}^S_2 \geq \frac{K}{S_0} \\
-\left(S_0\tilde{R}^S_2 - K\right) + \left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}^S_T + \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}^S_t}{R^B_t}\right)R^B_T &\leq \epsilon, \quad \forall\{\tilde{R}^S_t\}_{t=1}^T \in U, \tilde{R}^S_2 \geq \frac{K}{S_0} \\
-\left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}^S_T - \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}^S_t}{R^B_t}\right)R^B_T &\leq \epsilon, \quad \forall\{\tilde{R}^S_t\}_{t=1}^T \in U, \tilde{R}^S_2 \leq \frac{K}{S_0} \\
\left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}^S_T + \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}^S_t}{R^B_t}\right)R^B_T &\leq \epsilon, \quad \forall\{\tilde{R}^S_t\}_{t=1}^T \in U, \tilde{R}^S_2 \leq \frac{K}{S_0}
\end{aligned}$$

$$\vdots$$

If $\tilde{R}^S_{max} = \tilde{R}^S_T$, i.e., $\tilde{R}^S_T \geq \tilde{R}^S_t, \quad \forall t \in \{1, 2, ..., T\}$:

$$\begin{aligned}
\left(S_0\tilde{R}^S_T - K\right) - \left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}^S_T - \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}^S_t}{R^B_t}\right)R^B_T &\leq \epsilon, \quad \forall\{\tilde{R}^S_t\}_{t=1}^T \in U, \tilde{R}^S_T \geq \frac{K}{S_0} \\
-\left(S_0\tilde{R}^S_T - K\right) + \left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}^S_T + \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}^S_t}{R^B_t}\right)R^B_T &\leq \epsilon, \quad \forall\{\tilde{R}^S_t\}_{t=1}^T \in U, \tilde{R}^S_T \geq \frac{K}{S_0} \\
-\left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}^S_T - \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}^S_t}{R^B_t}\right)R^B_T &\leq \epsilon, \quad \forall\{\tilde{R}^S_t\}_{t=1}^T \in U, \tilde{R}^S_T \leq \frac{K}{S_0} \\
\left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}^S_T + \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}^S_t}{R^B_t}\right)R^B_T &\leq \epsilon, \quad \forall\{\tilde{R}^S_t\}_{t=1}^T \in U, \tilde{R}^S_T \leq \frac{K}{S_0}
\end{aligned}$$

$$(4.22)$$

Fixed Strike Lookback Call Option Pricing: Linear Equivalent Formulation

minimise$_{\alpha_0^S,\alpha_0^B,\{\beta_t\}_{t=0}^{T-1},\epsilon,\{p_{t,c,i},q_{t,c,i},m_{t,c,i},n_{t,c,i},z_{t,c,i}\}_{t=1,c=1,i=1}^{t=T,c=4,i=T}}$ ϵ

subject to

(If $\tilde{R}_{max}^S = \tilde{R}_t^S$ where $t < T$)

case t.I

$$\left(\sum_{i=1}^{T} p_{t,1,i}\underline{R}_i^S\right) + \left(\sum_{i=1}^{T} q_{t,1,i}\bar{R}_i^S\right) + \left(1+\underline{r}_0^S\right) m_{t,1,1} + \left(1+\bar{r}_0^S\right) n_{t,1,1} + \frac{z_{t,1,t}K}{S_0} \le \epsilon + K + \alpha_0^B R_T^B - \beta_0 R_T^B$$

$$p_{t,1,i} + q_{t,1,i} + m_{t,1,i} + n_{t,1,i} - \left(1+\underline{r}_i^S\right) m_{t,1,i+1} - \left(1+\bar{r}_i^S\right) n_{t,1,i+1} - z_{t,1,i} = \beta i \frac{R_T^B}{R_i^B}, \quad \forall i \notin \{t,T\}$$

$$p_{t,1,t} + q_{t,1,t} + m_{t,1,t} + n_{t,1,t} - \left(1+\underline{r}_t^S\right) m_{t,1,t+1} - \left(1+\bar{r}_t^S\right) n_{t,1,t+1} + \left(\sum_{i=1}^{T} z_{t,1,i}\right) = \beta t \frac{R_T^B}{R_t^B} + S_0,$$

$$p_{t,1,T} + q_{t,1,T} + m_{t,1,T} + n_{t,1,T} - z_{t,1,T} = -\alpha_0^S - \left(\sum_{t=0}^{T-1} \beta_t\right)$$

$$p_{t,1,i} \le 0,\ q_{t,1,i} \ge 0,\ m_{t,1,i} \le 0,\ n_{t,1,i} \ge 0,\ z_{t,1,i} \le 0 \quad \forall t = 1,2,...,T-1,\ \forall i = 1,2,...,T$$

case t.II

$$\left(\sum_{i=1}^{T} p_{t,2,i}\underline{R}_i^S\right) + \left(\sum_{i=1}^{T} q_{t,2,i}\bar{R}_i^S\right) + \left(1+\underline{r}_0^S\right) m_{t,2,1} + \left(1+\bar{r}_0^S\right) n_{t,2,1} + \frac{z_{t,2,t}K}{S_0} \le \epsilon - K - \alpha_0^B R_T^B + \beta_0 R_T^B$$

$$p_{t,2,i} + q_{t,2,i} + m_{t,2,i} + n_{t,2,i} - \left(1+\underline{r}_i^S\right) m_{t,2,i+1} - \left(1+\bar{r}_i^S\right) n_{t,2,i+1} - z_{t,2,i} = -\beta i \frac{R_T^B}{R_i^B}, \quad \forall i \notin \{t,T\}$$

$$p_{t,2,t} + q_{t,2,t} + m_{t,2,t} + n_{t,2,t} - \left(1+\underline{r}_t^S\right) m_{t,2,t+1} - \left(1+\bar{r}_t^S\right) n_{t,2,t+1} + \left(\sum_{i=1}^{T} z_{t,2,i}\right) = -\beta t \frac{R_T^B}{R_t^B} - S_0,$$

$$p_{t,2,T} + q_{t,2,T} + m_{t,2,T} + n_{t,2,T} - z_{t,2,T} = \alpha_0^S + \left(\sum_{t=0}^{T-1} \beta_t\right)$$

$$p_{t,2,i} \le 0,\ q_{t,2,i} \ge 0,\ m_{t,2,i} \le 0,\ n_{t,2,i} \ge 0,\ z_{t,2,i} \le 0 \quad \forall t = 1,2,...,T-1,\ \forall i = 1,2,...,T$$

case t.III

$$\left(\sum_{i=1}^{T} p_{t,3,i}\underline{R}_i^S\right) + \left(\sum_{i=1}^{T} q_{t,3,i}\bar{R}_i^S\right) + \left(1+\underline{r}_0^S\right) m_{t,3,1} + \left(1+\bar{r}_0^S\right) n_{t,3,1} + \frac{z_{t,3,t}K}{S_0} \le \epsilon + \alpha_0^B R_T^B - \beta_0 R_T^B$$

$$p_{t,3,i} + q_{t,3,i} + m_{t,3,i} + n_{t,3,i} - \left(1+\underline{r}_i^S\right) m_{t,3,i+1} - \left(1+\bar{r}_i^S\right) n_{t,3,i+1} + z_{t,3,i} = \beta i \frac{R_T^B}{R_i^B}, \quad \forall i \notin \{t,T\}$$

$$p_{t,3,t} + q_{t,3,t} + m_{t,3,t} + n_{t,3,t} - \left(1+\underline{r}_t^S\right) m_{t,3,t+1} - \left(1+\bar{r}_t^S\right) n_{t,3,t+1} - \left(\sum_{i=1}^{T} z_{t,3,i}\right) + 2z_{t,3,t} = \beta t \frac{R_T^B}{R_t^B},$$

$$p_{t,3,T} + q_{t,3,T} + m_{t,3,T} + n_{t,3,T} + z_{t,3,T} = -\alpha_0^S - \left(\sum_{t=0}^{T-1} \beta_t\right)$$

$$p_{t,3,i} \le 0,\ q_{t,3,i} \ge 0,\ m_{t,3,i} \le 0,\ n_{t,3,i} \ge 0,\ z_{t,3,i} \ge 0 \quad \forall t = 1,2,...,T-1,\ \forall i = 1,2,...,T$$

case t.IV

$$\left(\sum_{i=1}^{T} p_{t,4,i}\underline{R}_i^S\right) + \left(\sum_{i=1}^{T} q_{t,4,i}\bar{R}_i^S\right) + \left(1+\underline{r}_0^S\right) m_{t,4,1} + \left(1+\bar{r}_0^S\right) n_{t,4,1} + \frac{z_{t,4,t}K}{S_0}$$
$$\leq \epsilon - \alpha_0^B R_T^B + \beta_0 R_T^B$$

$$p_{t,4,i} + q_{t,4,i} + m_{t,4,i} + n_{t,4,i} - \left(1+\underline{r}_i^S\right) m_{t,4,i+1} - \left(1+\bar{r}_i^S\right) n_{t,4,i+1} + z_{t,4,i} = -\beta i \frac{R_T^B}{R_i^B}, \quad \forall i \notin \{t, T\}$$

$$p_{t,4,t} + q_{t,4,t} + m_{t,4,t} + n_{t,4,t} - \left(1+\underline{r}_t^S\right) m_{t,4,t+1} - \left(1+\bar{r}_t^S\right) n_{t,4,t+1} - \left(\sum_{i=1}^{T} z_{t,4,i}\right) + 2z_{t,4,t} = -\beta t \frac{R_T^B}{R_t^B},$$

$$p_{t,4,T} + q_{t,4,T} + m_{t,4,T} + n_{t,4,T} + z_{t,4,T} = \alpha_0^S + \left(\sum_{t=0}^{T-1} \beta_t\right)$$

$$p_{t,4,i} \leq 0,\ q_{t,4,i} \geq 0,\ m_{t,4,i} \leq 0,\ n_{t,4,i} \geq 0,\ z_{t,4,i} \geq 0 \quad \forall t = 1,2,...,T-1,\ \forall i = 1,2,...,T$$

(If $\tilde{R}_{max}^S = \tilde{R}_T^S$)

case T.I

$$\left(\sum_{i=1}^{T} p_{T,1,i}\underline{R}_i^S\right) + \left(\sum_{i=1}^{T} q_{T,1,i}\bar{R}_i^S\right) + \left(1+\underline{r}_0^S\right) m_{T,1,1} + \left(1+\bar{r}_0^S\right) n_{T,1,1} + \frac{z_{T,1,T}K}{S_0}$$
$$\leq \epsilon + K + \alpha_0^B R_T^B - \beta_0 R_T^B$$

$$p_{T,1,i} + q_{T,1,i} + m_{T,1,i} + n_{T,1,i} - \left(1+\underline{r}_i^S\right) m_{T,1,i+1} - \left(1+\bar{r}_i^S\right) n_{T,1,i+1} - z_{T,1,i} = \beta i \frac{R_T^B}{R_i^B},$$
$$\forall i = 1,2,...,T-1$$

$$p_{T,1,T} + q_{T,1,T} + m_{T,1,T} + n_{T,1,T} + \left(\sum_{i=1}^{T} z_{T,1,i}\right) = S_0 - \alpha_0^S - \left(\sum_{t=0}^{T-1} \beta_t\right)$$

$$p_{T,1,i} \leq 0,\ q_{T,1,i} \geq 0,\ m_{T,1,i} \leq 0,\ n_{T,1,i} \geq 0,\ z_{T,1,i} \leq 0 \quad \forall i = 1,2,...,T$$

case T.II

$$\left(\sum_{i=1}^{T} p_{T,2,i}\underline{R}_i^S\right) + \left(\sum_{i=1}^{T} q_{T,2,i}\bar{R}_i^S\right) + \left(1+\underline{r}_0^S\right) m_{T,2,1} + \left(1+\bar{r}_0^S\right) n_{T,2,1} + \frac{z_{T,2,T}K}{S_0}$$
$$\leq \epsilon - K - \alpha_0^B R_T^B + \beta_0 R_T^B$$

$$p_{T,2,i} + q_{T,2,i} + m_{T,2,i} + n_{T,2,i} - \left(1+\underline{r}_i^S\right) m_{T,2,i+1} - \left(1+\bar{r}_i^S\right) n_{T,2,i+1} - z_{T,2,i} = -\beta i \frac{R_T^B}{R_i^B},$$
$$\forall i = 1,2,...,T-1$$

$$p_{T,2,T} + q_{T,2,T} + m_{T,2,T} + n_{T,2,T} + \left(\sum_{i=1}^{T} z_{T,2,i}\right) = -S_0 + \alpha_0^S + \left(\sum_{t=0}^{T-1} \beta_t\right)$$

$$p_{T,2,i} \leq 0,\ q_{T,2,i} \geq 0,\ m_{T,2,i} \leq 0,\ n_{T,2,i} \geq 0,\ z_{T,2,i} \leq 0 \quad \forall i = 1,2,...,T$$

case T.III

$$\left(\sum_{i=1}^{T} p_{T,3,i}\underline{R}_i^S\right) + \left(\sum_{i=1}^{T} q_{T,3,i}\bar{R}_i^S\right) + \left(1+\underline{r}_0^S\right) m_{T,3,1} + \left(1+\bar{r}_0^S\right) n_{T,3,1} + \frac{z_{T,3,T}K}{S_0}$$
$$\leq \epsilon + \alpha_0^B R_T^B - \beta_0 R_T^B$$

$$p_{T,3,i} + q_{T,3,i} + m_{T,3,i} + n_{T,3,i} - \left(1+\underline{r}_i^S\right) m_{T,3,i+1} - \left(1+\bar{r}_i^S\right) n_{T,3,i+1} + z_{T,3,i} = \beta i \frac{R_T^B}{R_i^B},$$
$$\forall i = 1, 2, ..., T-1$$

$$p_{T,3,T} + q_{T,3,T} + m_{T,3,T} + n_{T,1,T} - \left(\sum_{i=1}^{T-1} z_{T,3,i}\right) + z_{T,3,T} = -\alpha_0^S - \left(\sum_{t=0}^{T-1} \beta_t\right)$$

$$p_{T,3,i} \leq 0,\ q_{T,3,i} \geq 0,\ m_{T,3,i} \leq 0,\ n_{T,3,i} \geq 0,\ z_{T,3,i} \geq 0 \quad \forall i = 1, 2, ..., T$$

case T.IV

$$\left(\sum_{i=1}^{T} p_{T,4,i}\underline{R}_i^S\right) + \left(\sum_{i=1}^{T} q_{T,4,i}\bar{R}_i^S\right) + \left(1+\underline{r}_0^S\right) m_{T,4,1} + \left(1+\bar{r}_0^S\right) n_{T,4,1} + \frac{z_{T,4,T}K}{S_0}$$
$$\leq \epsilon - \alpha_0^B R_T^B + \beta_0 R_T^B$$

$$p_{T,4,i} + q_{T,4,i} + m_{T,4,i} + n_{T,4,i} - \left(1+\underline{r}_i^S\right) m_{T,4,i+1} - \left(1+\bar{r}_i^S\right) n_{T,4,i+1} + z_{T,4,i} = -\beta i \frac{R_T^B}{R_i^B},$$
$$\forall i = 1, 2, ..., T-1$$

$$p_{T,4,T} + q_{T,4,T} + m_{T,4,T} + n_{T,4,T} - \left(\sum_{i=1}^{T-1} z_{T,4,i}\right) + z_{T,4,T} = \alpha_0^S + \left(\sum_{t=0}^{T-1} \beta_t\right)$$

$$p_{T,4,i} \leq 0,\ q_{T,4,i} \geq 0,\ m_{T,4,i} \leq 0,\ n_{T,4,i} \geq 0,\ z_{T,4,i} \geq 0 \quad \forall i = 1, 2, ..., T$$

It is evident that by using the same procedure a deterministic linear program corresponding to the fixed strike lookback put option pricing problem can be constructed. Moreover, one can also verify that the ϵ-arbitrage robust pricing model is capable of pricing floating strike lookback options as well. However, there is a study that establishes a put-call parity relating the price of the floating strike lookback put (call) option to the price of the fixed strike lookback call (put) option. It therefore is not entirely necessary to develop another model for pricing floating strike lookback options. The result of this study can be found in Hull [26]. For those who are keen on studying mathematical details of this study, we suggest reading from Wong and Kwok [50].

4.4 ϵ-Arbitrage Model Derivation for American-Style Options

Having discussed the employment of the ϵ-arbitrage robust pricing model for various European-style options in the previous section, in this section we shift our interest to standard American options. American option is typically considered to be more complicated than the previously discussed options because of the flexibility its holder has in exercising his or her right specified in the option contract. Unlike the European-style options, we investigate how to price both American call option and American put option. First, we discuss the American call option and explain why there is no need for us to develop another pricing model for American call options assuming that the underlying asset does not pay dividend. However, the same argument cannot be applied to American put options, and that makes us derive a robust pricing model for American put options. Again, the content of this section still heavily relies on the work of Chen [15].

4.4.1 American Call Options

If we assume that the underlying asset of the option pays no dividend, it is claimed that the American call option and the corresponding European call option should be equally valued because it would be optimal to keep the American call option until the expiration date. Consequently, one can use the developed European call option pricing model to find a fair price of a given American call option.

4.4.2 American Put Options

American put option, however, does not always give optimal payoff to its holder, if he or she insists on keeping the option until the very end. American put option, therefore, deserves further analysis. The simplified ϵ-arbitrage robust pricing model (4.11) cannot be used directly in this case since it is not reasonable to measure the arbitrage error based on the wealth level of the portfolio at the expiration date of the option as the holder of the American put option himself can decide when to exercise the option. Therefore, in order to develop an optimisation model representing the American put option pricing problem, all possibilities of the exercise time need to be taken into account. Below, the variable τ is used to indicate the exercise time; it can adopt any value from $\{1, 2, ..., T\}$, where as usual T denotes the expiration date of the option.

$$\text{minimise}_{\alpha_0^S, \alpha_0^B, \{\beta_t\}_{t=0}^{T-1}} \text{maximise}_{\{\tilde{R}_t^S\}_{t=1}^{T} \in U} \text{maximise}_{\tau=1,2,\ldots,T}$$
$$\left| \left(K - S_0 \tilde{R}_\tau^S \right)^+ - W_\tau \right|, \tag{4.23}$$

where W_τ is the wealth level of the portfolio at time τ.

Using the equation $W_\tau = x_\tau^S + x_\tau^B = \alpha_\tau^S \tilde{R}_\tau^S + \alpha_\tau^B R_\tau^B$ together with the recurrent relations (4.6) and (4.7), the above model can be reformulated as

American Put Option Pricing: ϵ-Arbitrage Formulation

$$\text{minimise}_{\alpha_0^S, \alpha_0^B, \{\beta_t\}_{t=0}^{T-1}} \text{maximise}_{\{\tilde{R}_t^S\}_{t=1}^{T} \in U} \text{maximise}_{\tau=1,2,\ldots,T}$$
$$\left| \left(K - S_0 \tilde{R}_\tau^S \right)^+ - \left(\alpha_0^S + \sum_{t=0}^{\tau-1} \beta_t \right) \tilde{R}_\tau^S - \left(\alpha_0^B - \sum_{t=0}^{\tau-1} \beta_t \frac{\tilde{R}_t^S}{R_t^B} \right) R_\tau^B \right|. \tag{4.24}$$

The intuition of the American put option pricing model (4.24) is that the arbitrage error needs to be optimised. In this setting, the optimal portfolio is expected to match well the payoff of the option with respect to any exercise time τ. By introducing a new decision variable ϵ to denote the arbitrage error, the model can be again reformulated in such a way that the uncertain parameters $\{\tilde{R}_t^s\}_{t=1}^T$ are excluded from the objective function. The resulting model is given by

$$
\begin{gathered}
\text{minimise}_{\alpha_0^S, \alpha_0^B, \{\beta_t\}_{t=0}^{T-1}, \epsilon} \quad \epsilon \\
\text{subject to}
\end{gathered}
$$

For every $\tau \in \{1, 2, ..., T\}$:

$$
\begin{aligned}
\left(K - S_0 \tilde{R}_\tau^S\right) - \left(\alpha_0^S + \sum_{t=0}^{\tau-1} \beta_t\right) \tilde{R}_\tau^S - \left(\alpha_0^B - \sum_{t=0}^{\tau-1} \beta_t \frac{\tilde{R}_t^S}{R_t^B}\right) R_\tau^B &\leq \epsilon, \quad \forall \{\tilde{R}_t^S\}_{t=1}^T \in U, \tilde{R}_\tau^S \leq \frac{K}{S_0} \\
-\left(K - S_0 \tilde{R}_\tau^S\right) + \left(\alpha_0^S + \sum_{t=0}^{\tau-1} \beta_t\right) \tilde{R}_\tau^S + \left(\alpha_0^B - \sum_{t=0}^{\tau-1} \beta_t \frac{\tilde{R}_t^S}{R_t^B}\right) R_\tau^B &\leq \epsilon, \quad \forall \{\tilde{R}_t^S\}_{t=1}^T \in U, \tilde{R}_\tau^S \leq \frac{K}{S_0} \\
-\left(\alpha_0^S + \sum_{t=0}^{\tau-1} \beta_t\right) \tilde{R}_\tau^S - \left(\alpha_0^B - \sum_{t=0}^{\tau-1} \beta_t \frac{\tilde{R}_t^S}{R_t^B}\right) R_\tau^B &\leq \epsilon, \quad \forall \{\tilde{R}_t^S\}_{t=1}^T \in U, \tilde{R}_\tau^S \geq \frac{K}{S_0} \\
\left(\alpha_0^S + \sum_{t=0}^{\tau-1} \beta_t\right) \tilde{R}_\tau^S + \left(\alpha_0^B - \sum_{t=0}^{\tau-1} \beta_t \frac{\tilde{R}_t^S}{R_t^B}\right) R_\tau^B &\leq \epsilon, \quad \forall \{\tilde{R}_t^S\}_{t=1}^T \in U, \tilde{R}_\tau^S \geq \frac{K}{S_0}
\end{aligned}
\tag{4.25}
$$

Similarly to the lookback option pricing model, the size of its linear deterministic equivalent grows quadratically with the number of time periods T. To avoid confusion, we divide a set of linear constraints into $4T$ subsets denoted by τ.I, τ.II, τ.III, and τ.IV, where $\tau \in \{1, 2, ..., T\}$ is the exercise time.

The linear equivalent formulation for pricing American put options is shown in the next page. We note here that the constraints for the case $\tau = T$ are slightly different from the other cases.

American Put Option Pricing: Linear Equivalent Formulation

minimise$_{\alpha_0^S,\alpha_0^B,\{\beta_t\}_{t=0}^{T-1},\epsilon,\{p_{\tau,c,i},q_{\tau,c,i},m_{\tau,c,i},n_{\tau,c,i},z_{\tau,c}\}_{\tau=1,c=1,i=1}^{\tau=T,c=4,i=T}}\quad \epsilon$

subject to

case τ.I

$$\left(\sum_{i=1}^{T} p_{\tau,1,i}\underline{R}_i^S\right)+\left(\sum_{i=1}^{T} q_{\tau,1,i}\bar{R}_i^S\right)+\left(1+\underline{r}_0^S\right)m_{\tau,1,1}+\left(1+\bar{r}_0^S\right)n_{\tau,1,1}+\frac{z_{\tau,1}K}{S_0}$$
$$\leq \epsilon - K+\alpha_0^B R_\tau^B-\beta_0 R_\tau^B$$
$$p_{\tau,1,i}+q_{\tau,1,i}+m_{\tau,1,i}+n_{\tau,1,i}-\left(1+\underline{r}_i^S\right)m_{\tau,1,i+1}-\left(1+\bar{r}_i^S\right)n_{\tau,1,i+1}=\beta i\frac{R_\tau^B}{R_i^B},\quad \forall i,\ i\leq\tau-1$$
$$p_{\tau,1,\tau}+q_{\tau,1,\tau}+m_{\tau,1,\tau}+n_{\tau,1,\tau}-\left(1+\underline{r}_\tau^S\right)m_{\tau,1,\tau+1}-\left(1+\bar{r}_\tau^S\right)n_{\tau,1,\tau+1}+z_{\tau,1}=-S_0-\alpha_0^S-\left(\sum_{t=0}^{T-1}\beta_t\right)$$
$$p_{\tau,1,i}+q_{\tau,1,i}+m_{\tau,1,i}+n_{\tau,1,i}-\left(1+\underline{r}_i^S\right)m_{\tau,1,i+1}-\left(1+\bar{r}_i^S\right)n_{\tau,1,i+1}=0,\quad \forall i,\ \tau+1\leq i\leq T-1$$
$$p_{\tau,1,T}+q_{\tau,1,T}+m_{\tau,1,T}+n_{\tau,1,T}=0$$
$$p_{\tau,1,i}\leq 0,\ q_{\tau,1,i}\geq 0,\ m_{\tau,1,i}\leq 0,\ n_{\tau,1,i}\geq 0,\ z_{\tau,1}\geq 0\quad \forall\tau,i\in\{1,2,...,T\}$$

case τ.II

$$\left(\sum_{i=1}^{T} p_{\tau,2,i}\underline{R}_i^S\right)+\left(\sum_{i=1}^{T} q_{\tau,2,i}\bar{R}_i^S\right)+\left(1+\underline{r}_0^S\right)m_{\tau,2,1}+\left(1+\bar{r}_0^S\right)n_{\tau,2,1}+\frac{z_{\tau,2}K}{S_0}$$
$$\leq \epsilon + K-\alpha_0^B R_\tau^B+\beta_0 R_\tau^B$$
$$p_{\tau,2,i}+q_{\tau,2,i}+m_{\tau,2,i}+n_{\tau,2,i}-\left(1+\underline{r}_i^S\right)m_{\tau,2,i+1}-\left(1+\bar{r}_i^S\right)n_{\tau,2,i+1}=-\beta i\frac{R_\tau^B}{R_i^B},\quad \forall i,\ i\leq\tau-1$$
$$p_{\tau,2,\tau}+q_{\tau,2,\tau}+m_{\tau,2,\tau}+n_{\tau,2,\tau}-\left(1+\underline{r}_\tau^S\right)m_{\tau,2,\tau+1}-\left(1+\bar{r}_\tau^S\right)n_{\tau,2,\tau+1}+z_{\tau,2}=S_0+\alpha_0^S+\left(\sum_{t=0}^{T-1}\beta_t\right)$$
$$p_{\tau,2,i}+q_{\tau,2,i}+m_{\tau,2,i}+n_{\tau,2,i}-\left(1+\underline{r}_i^S\right)m_{\tau,2,i+1}-\left(1+\bar{r}_i^S\right)n_{\tau,2,i+1}=0,\quad \forall i,\ \tau+1\leq i\leq T-1$$
$$p_{\tau,2,T}+q_{\tau,2,T}+m_{\tau,2,T}+n_{\tau,2,T}=0$$
$$p_{\tau,2,i}\leq 0,\ q_{\tau,2,i}\geq 0,\ m_{\tau,2,i}\leq 0,\ n_{\tau,2,i}\geq 0,\ z_{\tau,2}\geq 0\quad \forall\tau,i\in\{1,2,...,T\}$$

case τ.III

$$\left(\sum_{i=1}^{T} p_{\tau,3,i}\underline{R}_i^S\right)+\left(\sum_{i=1}^{T} q_{\tau,3,i}\bar{R}_i^S\right)+\left(1+\underline{r}_0^S\right)m_{\tau,3,1}+\left(1+\bar{r}_0^S\right)n_{\tau,3,1}+\frac{z_{\tau,3}K}{S_0}$$
$$\leq \epsilon +\alpha_0^B R_\tau^B-\beta_0 R_\tau^B$$
$$p_{\tau,3,i}+q_{\tau,3,i}+m_{\tau,3,i}+n_{\tau,3,i}-\left(1+\underline{r}_i^S\right)m_{\tau,3,i+1}-\left(1+\bar{r}_i^S\right)n_{\tau,3,i+1}=\beta i\frac{R_\tau^B}{R_i^B},\quad \forall i,\ i\leq\tau-1$$
$$p_{\tau,3,\tau}+q_{\tau,3,\tau}+m_{\tau,3,\tau}+n_{\tau,3,\tau}-\left(1+\underline{r}_\tau^S\right)m_{\tau,3,\tau+1}-\left(1+\bar{r}_\tau^S\right)n_{\tau,3,\tau+1}+z_{\tau,3}=-\alpha_0^S-\left(\sum_{t=0}^{T-1}\beta_t\right)$$
$$p_{\tau,3,i}+q_{\tau,3,i}+m_{\tau,3,i}+n_{\tau,3,i}-\left(1+\underline{r}_i^S\right)m_{\tau,3,i+1}-\left(1+\bar{r}_i^S\right)n_{\tau,3,i+1}=0,\quad \forall i,\ \tau+1\leq i\leq T-1$$
$$p_{\tau,3,T}+q_{\tau,3,T}+m_{\tau,3,T}+n_{\tau,3,T}=0$$
$$p_{\tau,3,i}\leq 0,\ q_{\tau,3,i}\geq 0,\ m_{\tau,3,i}\leq 0,\ n_{\tau,3,i}\geq 0,\ z_{\tau,3}\leq 0\quad \forall\tau,i\in\{1,2,...,T\}$$

case τ.IV

$$\left(\sum_{i=1}^{T} p_{\tau,4,i}\underline{R}_i^S\right) + \left(\sum_{i=1}^{T} q_{\tau,4,i}\bar{R}_i^S\right) + \left(1+\underline{r}_0^S\right) m_{\tau,4,1} + \left(1+\bar{r}_0^S\right) n_{\tau,4,1} + \frac{z_{\tau,4}K}{S_0}$$
$$\leq \epsilon - \alpha_0^B R_\tau^B + \beta_0 R_\tau^B$$
$$p_{\tau,4,i} + q_{\tau,4,i} + m_{\tau,4,i} + n_{\tau,4,i} - \left(1+\underline{r}_i^S\right) m_{\tau,4,i+1} - \left(1+\bar{r}_i^S\right) n_{\tau,4,i+1} = -\beta i \frac{R_\tau^B}{R_i^B}, \quad \forall i,\ i \leq \tau - 1$$
$$p_{\tau,4,\tau} + q_{\tau,4,\tau} + m_{\tau,4,\tau} + n_{\tau,4,\tau} - \left(1+\underline{r}_\tau^S\right) m_{\tau,4,\tau+1} - \left(1+\bar{r}_\tau^S\right) n_{\tau,4,\tau+1} + z_{\tau,4} = \alpha_0^S + \left(\sum_{t=0}^{T-1} \beta_t\right)$$
$$p_{\tau,4,i} + q_{\tau,4,i} + m_{\tau,4,i} + n_{\tau,4,i} - \left(1+\underline{r}_i^S\right) m_{\tau,4,i+1} - \left(1+\bar{r}_i^S\right) n_{\tau,4,i+1} = 0, \quad \forall i,\ \tau+1 \leq i \leq T-1$$
$$p_{\tau,4,T} + q_{\tau,4,T} + m_{\tau,4,T} + n_{\tau,4,T} = 0$$
$$p_{\tau,4,i} \leq 0,\ q_{\tau,4,i} \geq 0,\ m_{\tau,4,i} \leq 0,\ n_{\tau,4,i} \geq 0,\ z_{\tau,4} \leq 0 \quad \forall \tau, i \in \{1,2,...,T\}$$

4.5 Conclusions

On the whole, we use the conceptual idea of the ϵ-arbitrage robust pricing model to develop different pricing models for different options. Both of the European-style options and the American-style options are considered in this chapter. The pricing model for a particular option is a robust linear program. We then use Theorem 3.4.1, which comes a result of the duality in linear programming, to determine a corresponding uncertainty-free linear program. We observe one nice property of the ϵ-arbitrage robust pricing model, that is, the arising deterministic linear programs scale in a polynomial (linear or quadratic) way with the number of discretisation time steps.

Chapter 5

The New Robust Pricing Model

During our work, we identified two shortcomings of the ϵ-arbitrage robust pricing model which include:

- There is no guarantee that the output price will be non-negative. We encounter some certain cases that the ϵ-arbitrage robust pricing model outputs negative price while the payoff of the option is strictly non-negative. This should not be allowed in any circumstances since the negativity of the option price would result in arbitrage opportunity.
- The ϵ-arbitrage robust pricing model, from our point of view, is excessively complicated because, in order to achieve a robust linear optimisation problem, Chen [15] introduced new variables: $\{\alpha_t^S\}_{t=0}^T$, $\{\alpha_t^B\}_{t=0}^T$, and $\{\beta_t\}_{t=0}^{T-1}$. This step seems unnecessary to us. Furthermore, defining pricing models based on these artificial variables seems to impede further analysis on the replicating portfolio.

The shortcomings of the ϵ-arbitrage robust pricing model encourage us to figure out a different way to formulate a pricing model using robust optimisation. The main purpose of this chapter is to propose a new robust pricing model. Then, we introduce decision rules to our pricing model expecting that they are able to improve the accuracy of the generated optimal solutions.

In the ϵ-arbitrage robust pricing model, the proposed price for a given option is the initial wealth level (W_0) of the portfolio which optimally matches the option payoff. To address the first shortcoming, we can just simply add a constraint

$$W_0 = x_0^S + x_0^B = \alpha_0^S + \alpha_0^B \geq 0 \tag{5.1}$$

to the simplified ϵ-arbitrage pricing model (4.11) to ascertain the non-negativity of the output price. The amended model is, thus, given by

$$\text{minimise}_{\alpha_0^S,\alpha_0^B,\{\beta_t\}_{t=0}^{T-1}} \text{maximise}_{\{\tilde{R}_t^S\}_{t=1}^T \in U}$$
$$\left| P(\tilde{S},K) - \left(\alpha_0^S + \sum_{t=0}^{T-1}\beta_t\right)\tilde{R}_T^S - \left(\alpha_0^B - \sum_{t=0}^{T-1}\beta_t\frac{\tilde{R}_t^S}{R_t^B}\right)R_T^B \right|$$
$$\text{subject to}$$
$$\alpha_0^S + \alpha_0^B \geq 0.$$

Regarding the second shortcoming, we reformulate the simplified ϵ-arbitrage pricing model (4.11) in a way that enables the modellers to develop more sophisticated understanding of the decision variables. Transforming the model by replacing $\tilde{R}_t^S$ with $\dfrac{\tilde{S}_t}{S_0}$ and by replacing R_t^B with $\dfrac{B_t}{B_0}$ where $\tilde{S}_t$ and B_t are the price of the underlying asset of the option and the price of the risk-free asset at time t, respectively, we yield the following optimisation problem.

$$
\begin{gathered}
\text{minimise}_{\alpha_0^S,\alpha_0^B,\{\beta_t\}_{t=0}^{T-1}} \text{maximise}_{\{\tilde{R}_t^S\}_{t=1}^{T}\in U} \\
\left| P(\tilde{S},K) - \left(\frac{\alpha_0^S + \sum\limits_{t=0}^{T-1} \beta_t}{S_0} \right) \tilde{S}_T - \left(\frac{\alpha_0^B - \beta_0}{B_0} - \sum_{t=1}^{T-1} \frac{\beta_t \tilde{S}_t}{S_0 B_t} \right) B_T \right| \\
\text{subject to} \\
\alpha_0^S + \alpha_0^B \geq 0
\end{gathered} \tag{5.2}
$$

Moreover, we introduce the following set of new variables. This is to construct a new robust pricing model whose variables are more meaningful.

$$
\begin{aligned}
n_0^S &= \frac{\alpha_0^S + \beta_0}{S_0} \\
n_0^B &= \frac{\alpha_0^B - \beta_0}{B_0} \\
u_t &= \frac{\beta_t}{S_0}, \quad \forall t = 1, 2, ..., T-1
\end{aligned} \tag{5.3}
$$

Hence, we can rewrite the new pricing model as

Amended ϵ-Arbitrage Robust Pricing Model

$$
\begin{gathered}
\text{minimise}_{n_0^S,n_0^B,\{u_t\}_{t=1}^{T-1}} \ \text{maximise}_{\{S_t\}_{t=1}^{T}\in U} \\
\left| P(\tilde{S},K) - \left(n_0^S + \sum_{t=1}^{T-1} u_t \right) \tilde{S}_T - \left(n_0^B - \sum_{t=1}^{T-1} \frac{u_t \tilde{S}_t}{B_t} \right) B_T \right| \\
\text{subject to} \\
n_0^S S_0 + n_0^B B_0 \geq 0.
\end{gathered} \tag{5.4}
$$

5.1 Constant Decision Rule Pricing Model

It can be observed that the amended ϵ-arbitrage robust pricing model (5.4) corresponds to the following optimisation problem.

Constant Decision Rule Pricing Model

$$
\begin{aligned}
&\text{minimise}_{\{n_t^S\}_{t=0}^{T},\{n_t^B\}_{t=0}^{T},\{u_t\}_{t=1}^{T},\{v_t\}_{t=1}^{T}} \text{maximise}_{\{\tilde{S}_t\}_{t=1}^{T}\in U} \quad |P(\tilde{S},K) - W_T| \\
&\text{subject to} \\
&\quad W_T = n_T^S \tilde{S}_T + n_T^B B_T \\
&\quad n_t^S = n_{t-1}^S + u_t, \qquad \forall t = 1, 2, ..., T \\
&\quad n_t^B = n_{t-1}^B + v_t, \qquad \forall t = 1, 2, ..., T \\
&\quad u_t \tilde{S}_t + v_t B_t = 0, \qquad \forall t = 1, 2, ..., T \\
&\quad n_0^S S_0 + n_0^B B_0 \geq 0
\end{aligned}
$$

(5.5)

In this proposed robust pricing model, the variable n_t^S can be thought of as the number of the option's underlying assets being held in the portfolio at time t. Similarly, the variable n_t^B is the number of the risk-free assets being held in the portfolio at the same time. The variables u_t and v_t denote the number of the underlying assets and the number of the risk-free assets bought at time t, respectively. As before, W_t is the wealth level of the portfolio at time t, and its value is equal to $n_t^S \tilde{S}_t + n_t^B B_t$. With only a slight abuse of notation, we denote by U the uncertainty set containing the admissible underlying asset prices.

We call this pricing model a *constant decision rule pricing model* because the decision variables $\{u_t\}_{t=1}^T$ are modelled as a set of constants. By replacing $P(\tilde{S}, K)$ with the payoff function of a given European-style option, one can obtain a corresponding constant decision rule pricing model as follows.

European Call Option Pricing: CON-ECO

$$\text{minimise}_{n_0^S, n_0^B, \{u_t\}_{t=1}^{T-1}} \ \text{maximise}_{\{S_t\}_{t=1}^T \in U}$$
$$\left| \left(\tilde{S}_T - K\right)^+ - \left(n_0^S + \sum_{t=1}^{T-1} u_t\right) \tilde{S}_T - \left(n_0^B - \sum_{t=1}^{T-1} \frac{u_t \tilde{S}_t}{B_t}\right) B_T \right|$$
$$\text{subject to}$$
$$n_0^S S_0 + n_0^B B_0 \geq 0$$

(5.6)

Asian Call Option Pricing: CON-ACO

$$\text{minimise}_{n_0^S, n_0^B, \{u_t\}_{t=1}^{T-1}} \ \text{maximise}_{\{S_t\}_{t=1}^T \in U}$$
$$\left| \left(\frac{1}{T}\sum_{t=1}^{T} \tilde{S}_t - K\right)^+ - \left(n_0^S + \sum_{t=1}^{T-1} u_t\right) \tilde{S}_T - \left(n_0^B - \sum_{t=1}^{T-1} \frac{u_t \tilde{S}_t}{B_t}\right) B_T \right|$$
$$\text{subject to}$$
$$n_0^S S_0 + n_0^B B_0 \geq 0$$

(5.7)

Fixed Strike Lookback Call Option Pricing: CON-LCO

$$\text{minimise}_{n_0^S, n_0^B, \{u_t\}_{t=1}^{T-1}} \ \text{maximise}_{\{S_t\}_{t=1}^T \in U}$$
$$\left| \left(\max_{t=1,2,..,T} \tilde{S}_t - K\right)^+ - \left(n_0^S + \sum_{t=1}^{T-1} u_t\right) \tilde{S}_T - \left(n_0^B - \sum_{t=1}^{T-1} \frac{u_t \tilde{S}_t}{B_t}\right) B_T \right|$$
$$\text{subject to}$$
$$n_0^S S_0 + n_0^B B_0 \geq 0$$

(5.8)

A slight difference is expected in the case of American put option because of the flexibility in its exercising policy. Enumerating all possible exercise times $\tau = 1, 2, .., T$, a constant decision rule pricing model for American put options is thus given by

American Put Option Pricing: CON-APO

$$\begin{gathered}\text{minimise}_{n_0^S, n_0^B, \{u_t\}_{t=1}^{T-1}} \text{ maximise}_{\{S_t\}_{t=1}^{T} \in U} \text{ maximise}_{\tau=1,2,\ldots,T} \\ \left| \left(K - \tilde{S}_\tau\right)^+ - \left(n_0^S + \sum_{t=1}^{\tau-1} u_t\right) \tilde{S}_\tau - \left(n_0^B - \sum_{t=1}^{\tau-1} \frac{u_t \tilde{S}_t}{B_t}\right) B_\tau \right| \\ \text{subject to} \\ n_0^S S_0 + n_0^B B_0 \geq 0. \end{gathered} \tag{5.9}$$

For each pricing model, its linear deterministic equivalent can be obtained in two different ways. The direct approach is done by formulating the uncertainty set in a form of matrix inequality and subsequently applying Theorem 3.4.1.

$$\begin{bmatrix} -1 & 0 & 0 & \ldots & 0 \\ 0 & -1 & 0 & \ldots & 0 \\ 0 & 0 & -1 & \ldots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \ldots & -1 \\ \\ 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \ldots & 1 \\ \\ -1 & 0 & 0 & \ldots & 0 \\ 1+\underline{r}_1^s & -1 & 0 & \ldots & 0 \\ 0 & 1+\underline{r}_2^s & -1 & \ldots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \ldots & -1 \\ \\ 1 & 0 & 0 & \ldots & 0 \\ -1-\bar{r}_1^s & 1 & 0 & \ldots & 0 \\ 0 & -1-\bar{r}_2^s & 1 & \ldots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix} \begin{bmatrix} \tilde{S}_1 \\ \tilde{S}_2 \\ \vdots \\ \tilde{S}_T \end{bmatrix} \leq \begin{bmatrix} -S_0\underline{R}_1^s \\ -S_0\underline{R}_2^s \\ -S_0\underline{R}_3^s \\ \vdots \\ -S_0\underline{R}_T^s \\ \\ S_0\bar{R}_1^s \\ S_0\bar{R}_2^s \\ S_0\bar{R}_3^s \\ \vdots \\ S_0\bar{R}_T^s \\ \\ -S_0(1+\underline{r}_0^s) \\ 0 \\ 0 \\ \vdots \\ 0 \\ \\ S_0(1+\bar{r}_0^s) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5.10}$$

Alternatively, one could use a shortcut which is to consider the corresponding linear program derived in Chapter 4 when we discuss the original ϵ-arbitrage robust pricing model and then:

1. Replace $\alpha_0^S + \beta_0$ with $S_0 n_0^S$;
2. Replace $\alpha_0^B - \beta_0$ with $B_0 n_0^B$;
3. Replace β_t with $S_0 u_t$;
4. Replace R_t^B with $\dfrac{B_t}{B_0}$;
5. Divide the right hand side of the constraints by S_0;
6. Add a constraint to ensure the positivity of the proposed price, i.e., $n_0^S S_0 + n_0^B B_0 \geq 0$.

Moreover, the value of B_0 can be, without loss of generality, set to one. As an exemplar, the following linear program corresponds to the European call option pricing problem.

European Call Option Pricing: Linear Equivalent Formulation

minimise$_{n_0^S, n_0^B, \{u_t\}_{t=1}^{T-1}, \epsilon, \{p_{c,t}, q_{c,t}, m_{c,t}, n_{c,t}, z_c\}_{c=1,t=1}^{c=4,t=T}} \quad \epsilon$
subject to

$n_0^S S_0 + n_0^B \geq 0$

case I

$$\left(\sum_{t=1}^{T} p_{1,t} \underline{R}_t^S\right) + \left(\sum_{t=1}^{T} q_{1,t} \bar{R}_t^S\right) + \left(1 + \underline{r}_0^S\right) m_{1,1} + \left(1 + \bar{r}_0^S\right) n_{1,1} + \frac{z_1 K}{S_0} \leq \frac{1}{S_0} \left(\epsilon + K + n_0^B B_T\right)$$

$$p_{1,t} + q_{1,t} + m_{1,t} + n_{1,t} - \left(1 + \underline{r}_t^S\right) m_{1,t+1} - \left(1 + \bar{r}_t^S\right) n_{1,t+1} = u_t \frac{B_T}{B_t}, \quad \forall t = 1, 2, ..., T-1$$

$$p_{1,T} + q_{1,T} + m_{1,T} + n_{1,T} + z_1 = 1 - n_0^S - \left(\sum_{t=1}^{T-1} u_t\right)$$

$$p_{1,t} \leq 0, \; q_{1,t} \geq 0, \; m_{1,t} \leq 0, \; n_{1,t} \geq 0, \quad \forall t = 1, 2, ..., T, \qquad z_1 \leq 0$$

case II

$$\left(\sum_{t=1}^{T} p_{2,t} \underline{R}_t^S\right) + \left(\sum_{t=1}^{T} q_{2,t} \bar{R}_t^S\right) + \left(1 + \underline{r}_0^S\right) m_{2,1} + \left(1 + \bar{r}_0^S\right) n_{2,1} + \frac{z_2 K}{S_0} \leq \frac{1}{S_0} \left(\epsilon - K - n_0^B B_T\right)$$

$$p_{2,t} + q_{2,t} + m_{2,t} + n_{2,t} - \left(1 + \underline{r}_t^S\right) m_{2,t+1} - \left(1 + \bar{r}_t^S\right) n_{2,t+1} = -u_t \frac{B_T}{B_t}, \quad \forall t = 1, 2, ..., T-1$$

$$p_{2,T} + q_{2,T} + m_{2,T} + n_{2,T} + z_2 = -1 + n_0^S + \left(\sum_{t=1}^{T-1} u_t\right)$$

$$p_{2,t} \leq 0, \; q_{2,t} \geq 0, \; m_{2,t} \leq 0, \; n_{2,t} \geq 0, \quad \forall t = 1, 2, ..., T, \qquad z_2 \leq 0$$

case III

$$\left(\sum_{t=1}^{T} p_{3,t} \underline{R}_t^S\right) + \left(\sum_{t=1}^{T} q_{3,t} \bar{R}_t^S\right) + \left(1 + \underline{r}_0^S\right) m_{3,1} + \left(1 + \bar{r}_0^S\right) n_{3,1} + \frac{z_3 K}{S_0} \leq \frac{1}{S_0} \left(\epsilon + n_0^B B_T\right)$$

$$p_{3,t} + q_{3,t} + m_{3,t} + n_{3,t} - \left(1 + \underline{r}_t^S\right) m_{3,t+1} - \left(1 + \bar{r}_t^S\right) n_{3,t+1} = u_t \frac{B_T}{B_t}, \quad \forall t = 1, 2, ..., T-1$$

$$p_{3,T} + q_{3,T} + m_{3,T} + n_{3,T} + z_3 = -n_0^S - \left(\sum_{t=1}^{T-1} u_t\right)$$

$$p_{3,t} \leq 0, \; q_{3,t} \geq 0, \; m_{3,t} \leq 0, \; n_{3,t} \geq 0, \quad \forall t = 1, 2, ..., T, \qquad z_3 \geq 0$$

case IV

$$\left(\sum_{t=1}^{T} p_{4,t} \underline{R}_t^S\right) + \left(\sum_{t=1}^{T} q_{4,t} \bar{R}_t^S\right) + \left(1 + \underline{r}_0^S\right) m_{4,1} + \left(1 + \bar{r}_0^S\right) n_{4,1} + \frac{z_4 K}{S_0} \leq \frac{1}{S_0} \left(\epsilon - n_0^B B_T\right)$$

$$p_{4,t} + q_{4,t} + m_{4,t} + n_{4,t} - \left(1 + \underline{r}_t^S\right) m_{4,t+1} - \left(1 + \bar{r}_t^S\right) n_{4,t+1} = -u_t \frac{B_T}{B_t}, \quad \forall t = 1, 2, ..., T-1$$

$$p_{4,T} + q_{4,T} + m_{4,T} + n_{4,T} + z_4 = n_0^S + \left(\sum_{t=1}^{T-1} u_t\right)$$

$$p_{4,t} \leq 0, \; q_{4,t} \geq 0, \; m_{4,t} \leq 0, \; n_{4,t} \geq 0, \quad \forall t = 1, 2, ..., T, \qquad z_4 \geq 0$$

5.2 Linear Decision Rule Pricing Model

After identifying the shortcomings of the ϵ-arbitrage robust pricing model and proposing a new robust pricing model as an amended version of it, in this section, we consider another possibility to improve the pricing model. Both of the previously discussed robust pricing models, namely the ϵ-arbitrage robust pricing model and the constant decision rule pricing model, employ the robust optimisation approach. Using traditional robust optimisation approach, the model consists of only here-and-now decisions; thus, the pricing models seem to be unrealistic and subject to over conservatism from the optimisation perspective.

In the constant decision rule pricing model, we have a series of decision variables $\{u_t\}_{t=1}^T$ representing the number of the underlying assets to be bought at time t. Instead of modelling a decision variable u_t as a here-and-now decision which is completely independent of the information having been observed so far, i.e., $\{\tilde{S}_i\}_{i=1}^t$, it can be modelled as a function of such information.

$$u_t = f_{u_t}\left(\tilde{S}_1, \tilde{S}_2, ..., \tilde{S}_t\right) \tag{5.11}$$

Note that a decision variable should not be expressed as a function of the information not yet known to reflect the inability of the decision maker to foresee the future. If f_{u_t} is a constant function, then u_t does not depend on any information about the asset prices. Thus, the pricing model remains unchanged. The optimal value of the objective function, namely the arbitrage error, is expected to be more accurate (less conservative and closer to the true optimal objective value) by increasing the expressiveness of the function f_{u_t}. For example, if f_{u_t} is defined as an affine function of its arguments, we have

$$u_t = u_{t,0} + u_{t,1}\tilde{S}_1 + u_{t,2}\tilde{S}_2 + ... + u_{t,t}\tilde{S}_t. \tag{5.12}$$

The functions $\{f_{u_t}\}_{t=1}^T$ are usually referred to as decision rules or policies. However, the cost of specifying decision rules in the optimisation problems is inevitable. To preserve the tractability of the nominal robust optimisation problem, linear decision rules are employed in this section, and the comparison between the linear decision rule pricing model and the constant decision rule pricing model is presented afterwards in Chapter 8. As an example, below we show how to employ linear decision rules in the proposed robust pricing model when the option considered is a European call option. For the other types of options: Asian options, lookback options, and American options, the procedure below can also be applied.

By introducing the decision rules to the new robust pricing model for European call options (CON-ECO), we achieve

European Call Option Pricing: The Decision Rule Formulation

$$\begin{gathered}
\text{minimise}_{n_0^S, n_0^B, \{u_t(\{\tilde{S}_i\}_{i=1}^t)\}_{t=1}^{T-1}} \ \text{maximise}_{\{S_t\}_{t=1}^T \in U} \\
\left| \left(\tilde{S}_T - K\right)^+ - \left(n_0^S + \sum_{t=1}^{T-1} u_t\left(\{\tilde{S}_i\}_{i=1}^t\right)\right)\tilde{S}_T - \left(n_0^B - \sum_{t=1}^{T-1} \frac{u_t\left(\{\tilde{S}_i\}_{i=1}^t\right)\tilde{S}_t}{B_t}\right) B_T \right| \\
\text{subject to} \\
n_0^S S_0 + n_0^B B_0 \geq 0.
\end{gathered} \tag{5.13}$$

To make the resulting optimisation model more elegant and easier to deal with, we first introduce a vector ξ_t containing the information previously observed up to time t. Moreover, let ξ denote ξ_T.

$$\xi_t = \begin{bmatrix} 1 \\ \tilde{S}_1 \\ \tilde{S}_2 \\ \vdots \\ \tilde{S}_t \end{bmatrix}_{(t+1)\times 1} \tag{5.14}$$

$$\xi = \xi_T = \begin{bmatrix} 1 \\ \tilde{S}_1 \\ \tilde{S}_2 \\ \vdots \\ \tilde{S}_T \end{bmatrix}_{(T+1)\times 1} \tag{5.15}$$

Note that there is always a scalar, i.e., one, added at the top of the information vector. This simple trick is very useful especially when applying the linear decision rules to the pricing model. Furthermore, a linear relationship between ξ_t and ξ can be observed. That is

$$\xi_t = M_t \xi, \tag{5.16}$$

where M_t is a $(t+1) \times (T+1)$ matrix defined as an aggregation of an identity matrix $\mathbb{I}_{t+1}$ and a zero matrix $0_{(t+1)\times(T-t)}$.

$$M_t = \left[\mathbb{I}_{t+1} | 0_{(t+1)\times(T-t)}\right] = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & & & & \vdots & \vdots & & & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \end{bmatrix}_{(t+1)\times(T+1)} \tag{5.17}$$

Let e_t be a t^{th} canonical basis vector of appropriate dimension, i.e., a vector comprised of $T+1$ entries containing one at the t^{th} position and zero everywhere else. A relation between $\tilde{S}_t$ and ξ can therefore be represented as

$$\tilde{S}_t = e_{t+1}^T \xi_T = e_{t+1}^T \xi. \tag{5.18}$$

Lastly, we introduce a set of constants $\{r_t\}_{t=1}^T$ defined as

$$r_t = \frac{B_T}{B_t}. \tag{5.19}$$

The new robust pricing model for European call options can then be equivalently rewritten as follows.

$$\begin{gathered} \text{minimise}_{n_0^S, n_0^B, \{u_t(\xi_t)\}_{t=1}^{T-1}} \ \text{maximise}_{\{S_t\}_{t=1}^T \in U} \\ \left| \left(e_{T+1}^T \xi - K\right)^+ - \left(n_0^S + \sum_{t=1}^{T-1} u_t(\xi_t)\right) e_{T+1}^T \xi - \left(n_0^B B_T - \sum_{t=1}^{T-1} u_t(\xi_t)\, e_{t+1}^T \xi r_t\right) \right| \\ \text{subject to} \\ n_0^S S_0 + n_0^B B_0 \geq 0 \end{gathered} \tag{5.20}$$

If the decision $u_t(\xi_t)$ is modelled as a linear function of ξ_t, the expression (5.12) for every $t = 1, 2, ..., T$ is expected. Since the first element of the vector ξ_t is restricted to be one, we can express these linear decision rules as

$$u_t(\xi_t) = \begin{bmatrix} u_{t,0} \\ u_{t,1} \\ \vdots \\ u_{t,t} \end{bmatrix}^T \xi_t. \tag{5.21}$$

The implication of this is that there is a vector $U_t \in \mathbb{R}^{t+1}$ through which the linear dependence is exhibited.

$$u_t(\xi_t) = U_t^T \xi_t = U_t^T M_t \xi \tag{5.22}$$

With these linear decision rules, the European call option pricing model (5.20) can be rewritten as,

$$\begin{gathered} \text{minimise}_{n_0^S, n_0^B, \{U_t\}_{t=1}^{T-1}} \ \text{maximise}_{\xi \in \Xi} \\ \left| \left(e_{T+1}^T \xi - K\right)^+ - \left(n_0^S + \sum_{t=1}^{T-1} U_t^T M_t \xi\right) e_{T+1}^T \xi - \left(n_0^B B_T - \sum_{t=1}^{T-1} U_t^T M_t \xi e_{t+1}^T \xi r_t\right) \right| \\ \text{subject to} \\ n_0^S S_0 + n_0^B B_0 \geq 0, \end{gathered} \tag{5.23}$$

where Ξ is an uncertainty set describing acceptable values of ξ. It is, in fact, very similar to the uncertainty set U in the constant decision rule pricing model.

The pricing model below is obtained by rearranging terms in the previous one. Notice that the objective function is quadratic in the uncertain parameter ξ.

European Call Option Pricing: LIN-ECO

$$\begin{gathered} \text{minimise}_{n_0^S, n_0^B, \{U_t\}_{t=1}^{T-1}} \ \text{maximise}_{\xi \in \Xi} \\ \left| \left(\xi^T e_{T+1} - K\right)^+ + \xi^T \left(\sum_{t=1}^{T-1} \left(e_{t+1} U_t^T M_t r_t - e_{T+1} U_t^T M_t\right) \right) \xi + \xi^T \left(-e_{T+1} n_0^S\right) + \left(-n_0^B B_T\right) \right| \\ \text{subject to} \\ n_0^S S_0 + n_0^B B_0 \geq 0 \end{gathered} \tag{5.24}$$

In this pricing model, as mentioned earlier, the uncertainty set Ξ is used to describe the possibilities of ξ. Recall that in the constant decision rule pricing model the uncertainty set U is a polyhedron obtained from the upper bound and the lower bound of the price of the underlier as well as the upper bound and the lower bound of the single-period return at time $t = 1, 2, ..., T$. This idea is still adopted in this pricing model; however, according to our definition of ξ, there is, in addition, one further constraint to be enforced, that is, $\xi^T e_1 = 1$.

To be more specific, we define the uncertainty set Ξ as a set of vectors that comply to a list of greater-than-or-equal-to constraints as follows.

$$
\begin{aligned}
\Xi = \{\xi \in \mathbb{R}^{T+1} \mid\ & \xi^T e_1 \geq 1 \quad \wedge \\
& \xi^T e_2 = \tilde{S}_1 \geq S_0 \underline{R}_1^S \quad \wedge \\
& \xi^T e_3 = \tilde{S}_2 \geq S_0 \underline{R}_2^S \quad \wedge \\
& \vdots \\
& \xi^T e_{T+1} = \tilde{S}_T \geq S_0 \underline{R}_T^S \quad \wedge \\
& \\
& \xi^T (-e_1) \geq -1 \quad \wedge \\
& \xi^T (-e_2) = -\tilde{S}_1 \geq -S_0 \bar{R}_1^S \quad \wedge \\
& \xi^T (-e_3) = -\tilde{S}_2 \geq -S_0 \bar{R}_2^S \quad \wedge \\
& \vdots \\
& \xi^T (-e_{T+1}) = -\tilde{S}_T \geq -S_0 \bar{R}_T^S \quad \wedge \\
& \\
& \xi^T e_2 = \tilde{S}_1 \geq S_0 \left(1 + \underline{r}_0^S\right) \quad \wedge \\
& \xi^T \left(e_3 - \left(1 + \underline{r}_1^S\right) e_2\right) = \tilde{S}_2 - \left(1 + \underline{r}_1^S\right) \tilde{S}_1 \geq 0 \quad \wedge \\
& \xi^T \left(e_4 - \left(1 + \underline{r}_2^S\right) e_3\right) = \tilde{S}_3 - \left(1 + \underline{r}_2^S\right) \tilde{S}_2 \geq 0 \quad \wedge \\
& \vdots \\
& \xi^T \left(e_{T+1} - \left(1 + \underline{r}_{T-1}^S\right) e_T\right) = \tilde{S_T} - \left(1 + \underline{r}_{T-1}^S\right) \tilde{S}_{T-1} \geq 0 \quad \wedge \\
& \\
& \xi^T (-e_2) = -\tilde{S}_1 \geq -S_0 \left(1 + \bar{r}_0^S\right) \quad \wedge \\
& \xi^T \left(-e_3 + \left(1 + \bar{r}_1^S\right) e_2\right) = -\tilde{S}_2 + \left(1 + \bar{r}_1^S\right) \tilde{S}_1 \geq 0 \quad \wedge \\
& \xi^T \left(-e_4 + \left(1 + \bar{r}_2^S\right) e_3\right) = -\tilde{S}_3 + \left(1 + \bar{r}_2^S\right) \tilde{S}_2 \geq 0 \quad \wedge \\
& \vdots \\
& \xi^T \left(-e_{T+1} + \left(1 + \bar{r}_{T-1}^S\right) e_T\right) = -\tilde{S_T} + \left(1 + \bar{r}_{T-1}^S\right) \tilde{S}_{T-1} \geq 0 \\
\}&
\end{aligned}
\tag{5.25}
$$

Although the objective function and the uncertainty set are explicitly defined, the whole optimisation problem still does not seem to be solvable by standard solvers because it is a semi-infinite optimisation problem. We therefore need to determine its deterministic version. To begin with, we introduce the following propositions.

Proposition 5.2.1. *For any ξ defined in (5.15) and any matrix $A \in \mathbb{R}^{(T+1)\times(T+1)}$, $\xi^T A \xi = \xi^T \left(\frac{1}{2}A + \frac{1}{2}A^T\right)\xi$*

Proof.

The proposition follows from the fact that $\xi^T A \xi = \left(\xi^T A \xi\right)^T = \xi^T A^T \xi$.

□

Proposition 5.2.2. *For any ξ defined in (5.15), any matrix $A \in \mathbb{R}^{(T+1)\times(T+1)}$, any vector $b \in \mathbb{R}^{(T+1)}$, and any $c \in \mathbb{R}$,*

1. $\left(\frac{1}{2}A + \frac{1}{2}A^T + \frac{1}{2}e_1 b^T + \frac{1}{2}b e_1^T + c e_1 e_1^T\right)$ *is symmetric.*

2. $\xi^T A\xi + \xi^T b + c = \xi^T \left(\frac{1}{2}A + \frac{1}{2}A^T + \frac{1}{2}e_1 b^T + \frac{1}{2}be_1^T + ce_1e_1^T\right)\xi.$

Proof.

part 1:

$$\left(\frac{1}{2}A + \frac{1}{2}A^T + \frac{1}{2}e_1 b^T + \frac{1}{2}be_1^T + ce_1e_1^T\right)^T = \left(\frac{1}{2}A^T + \frac{1}{2}A + \frac{1}{2}be_1^T + \frac{1}{2}e_1 b^T + ce_1e_1^T\right)$$

Hence,

$$\left(\frac{1}{2}A + \frac{1}{2}A^T + \frac{1}{2}e_1 b^T + \frac{1}{2}be_1^T + ce_1e_1^T\right)^T = \left(\frac{1}{2}A + \frac{1}{2}A^T + \frac{1}{2}e_1 b^T + \frac{1}{2}be_1^T + ce_1e_1^T\right)$$

part 2:

$$\xi^T A\xi + \xi^T b + c = \xi^T A\xi + b^T\xi + c$$

Therefore,

$$\begin{aligned} \xi^T A\xi + \xi^T b + c &= \xi^T A\xi + \xi^T e_1 b^T\xi + \xi^T ce_1e_1^T\xi \\ &= \xi^T\left(A + e_1b^T + ce_1e_1^T\right)\xi \end{aligned}$$

By applying Proposition 5.2.1 to the right hand side of the equation, the proof thus completes.

□

Introducing a new variable ϵ to denote the objective function, the European call option pricing model (5.24) can again be rewritten as

$$\begin{gathered} \text{minimise}_{n_0^S, n_0^B, \{U_t\}_{t=1}^{T-1}, \epsilon} \ \text{maximise}_{\xi\in\Xi} \quad \epsilon \\ \text{subject to} \end{gathered}$$

$$\begin{aligned} \xi^T\left(-\sum_{t=1}^{T-1}\left(e_{t+1}U_t^T M_t r_t - e_{T+1}U_t^T M_t\right)\right)\xi + \xi^T\left(e_{T+1}n_0^S - e_{T+1}\right) + \left(\epsilon + K + n_0^B B_T\right) \geq 0,& \\ \forall\xi\in\Xi,\ \xi^T e_{T+1} - K \geq 0& \\ \xi^T\left(\sum_{t=1}^{T-1}\left(e_{t+1}U_t^T M_t r_t - e_{T+1}U_t^T M_t\right)\right)\xi + \xi^T\left(-e_{T+1}n_0^S + e_{T+1}\right) + \left(\epsilon - K - n_0^B B_T\right) \geq 0,& \\ \forall\xi\in\Xi,\ \xi^T e_{T+1} - K \geq 0& \\ \xi^T\left(-\sum_{t=1}^{T-1}\left(e_{t+1}U_t^T M_t r_t - e_{T+1}U_t^T M_t\right)\right)\xi + \xi^T\left(e_{T+1}n_0^S\right) + \left(\epsilon + n_0^B B_T\right) \geq 0,& \\ \forall\xi\in\Xi,\ -\xi^T e_{T+1} + K \geq 0& \\ \xi^T\left(\sum_{t=1}^{T-1}\left(e_{t+1}U_t^T M_t r_t - e_{T+1}U_t^T M_t\right)\right)\xi + \xi^T\left(-e_{T+1}n_0^S\right) + \left(\epsilon - n_0^B B_T\right) \geq 0,& \\ \forall\xi\in\Xi,\ -\xi^T e_{T+1} + K \geq 0.& \end{aligned} \tag{5.26}$$

There are four series of constraints in the resulting optimisation problem. For each one, the constraint is written as a quadratic constraint in ξ, and its associated uncertainty set is defined as a polyhedron. Using Proposition 5.2.2, each of these quadratic constraints can be rewritten in a form of $\xi^T F\xi \geq 0$ while a polyhedron can also be rewritten as $\{\xi \in \mathbb{R}^{T+1} \mid \xi^T G_i\xi \geq 0,\ \forall i = 1, 2, ..., I\}$

satisfying that F and $\{G_i\}_{i=1}^I$ are symmetric matrices.

Roughly speaking, Proposition 5.2.2 enables us to reformulate the European call option pricing model as a robust optimisation problem with quadratic constraints, each of which is associated with an uncertainty set which is an intersection of finitely many ellipsoids.

$$
\begin{gathered}
\text{minimise}_{n_0^S, n_0^B, \{U_t\}_{t=1}^{T-1}, \epsilon} \ \text{maximise}_{\xi \in \Xi} \quad \epsilon \\
\text{subject to} \\
\begin{array}{l}
\xi^T F^{(1)} \xi \geq 0, \quad \forall \xi,\ \xi^T G_i^{(1)} \xi \geq 0,\ \forall i \\
\xi^T F^{(2)} \xi \geq 0, \quad \forall \xi,\ \xi^T G_i^{(2)} \xi \geq 0,\ \forall i \\
\xi^T F^{(3)} \xi \geq 0, \quad \forall \xi,\ \xi^T G_i^{(3)} \xi \geq 0,\ \forall i \\
\xi^T F^{(4)} \xi \geq 0, \quad \forall \xi,\ \xi^T G_i^{(4)} \xi \geq 0,\ \forall i
\end{array}
\end{gathered}
\tag{5.27}
$$

Using the approximate S-lemma (Proposition 3.4.1), we obtain a set of approximate deterministic constraints.

$$
F^{(j)} - \sum_{i=1}^{I} \lambda_i^{(j)} G_i^{(j)} \succeq 0,\ \lambda_i^{(j)} \geq 0, \quad j = 1, 2, 3, 4 \tag{5.28}
$$

Hence, the approximation of the deterministic equivalent of this pricing model is a semidefinite program. It is important to note here that the obtained semidefinite program seems unlikely to be an exact deterministic equivalent because using the approximate S-lemma this way can guarantee only one direction is true, i.e.,

$$
\exists \lambda_i^{(j)} \geq 0,\ F^{(j)} - \sum_{i=1}^{I} \lambda_i^{(j)} G_i^{(j)} \succeq 0 \quad \Longrightarrow \quad \xi^T F^{(j)} \xi \geq 0, \quad \forall \xi,\ \xi^T G_i^{(j)} \xi \geq 0,\ \forall i,
$$

and unfortunately the reverse direction does not always hold.

Despite not being exact, a semidefinite program obtained from the approximate S-lemma appears to perform well in many cases. It is therefore natural for us to investigate how good this approximation is in the context of option pricing. Below, we prove that in terms of optimality the linear decision rule pricing model is at least as good as the constant decision rule pricing model, i.e., the linear decision rule approach is able to identify the portfolio that matches better the payoff of the option.

Proposition 5.2.3. *A semidefinite program corresponding to the linear decision rule pricing model is at most as conservative as a deterministic equivalent of the constant decision rule pricing model.*

Proof.
A constant decision rule can be viewed as a specialization of a linear decision rule as it can be written as follows.

$$
u_t^{(c)}(\xi_t) = U_t^{(c)T} \xi_t = \begin{bmatrix} u_{t,0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \xi_t \tag{5.29}
$$

By replacing the linear decision rules U_t in (5.26) with the constant decision rules $U_t^{(c)}$, the arising optimisation model contains no quadratic terms in asset prices ($\tilde{S}_i \tilde{S}_j,\ 1 \leq i, j \leq T$). This result together with the symmetry of $F^{(j)}$ lead to quite a simple structure of the matrices $F^{(j)}$.

$$F^{(j)} = \begin{bmatrix} \diamond & \diamond & \diamond & \dots & \diamond \\ \diamond & 0 & 0 & \dots & 0 \\ \diamond & 0 & 0 & \dots & 0 \\ \vdots & \vdots & & & \vdots \\ \diamond & 0 & 0 & \dots & 0 \end{bmatrix}_{(T+1)\times(T+1)}, \tag{5.30}$$

where $\diamond$ represents the value that can be either zero or non-zero.

Similarly, matrices $G_i^{(j)}$, which define the boundary of the uncertainty set containing acceptable values of $\xi = \begin{bmatrix} 1 \ \tilde{S}_1 \ \tilde{S}_2 \ \dots \ \tilde{S}_T \end{bmatrix}^T$, have an analogous structure

$$G_i^{(j)} = \begin{bmatrix} \diamond & \diamond & \diamond & \dots & \diamond \\ \diamond & 0 & 0 & \dots & 0 \\ \diamond & 0 & 0 & \dots & 0 \\ \vdots & \vdots & & & \vdots \\ \diamond & 0 & 0 & \dots & 0 \end{bmatrix}_{(T+1)\times(T+1)} \tag{5.31}$$

because the uncertainty set in this setting is a polyhedron, and a polyhedron cannot contain a non-zero quadratic term.

In other words, we can rewrite matrices $F^{(j)}$ and $G_i^{(j)}$ as

$$F^{(j)} = \begin{bmatrix} \mu^{(j)} & \frac{1}{2}\left(f^{(j)}\right)^T \\ \frac{1}{2}f^{(j)} & F'^{(j)} \end{bmatrix}, \quad G_i^{(j)} = \begin{bmatrix} \nu_i^{(j)} & \frac{1}{2}\left(g_i^{(j)}\right)^T \\ \frac{1}{2}g_i^{(j)} & G_i'^{(j)} \end{bmatrix}, \tag{5.32}$$

where $F'^{(j)}$ and $G_i'^{(j)}$ are zero matrices. Notice that zero matrices are both positive semidefinite and negative semidefinite.

It can be concluded from Proposition 3.4.2 that $F^{(j)} - \sum_{i=1}^{I} \lambda_i^{(j)} G_i^{(j)} \succeq 0$ is equivalent to $\xi'^T F'^{(j)} \xi' + \xi'^T f^{(j)} + \mu^{(j)} \geq 0, \quad \forall \xi' \in \{\xi' \in \mathbb{R}^T \mid \xi'^T G_i'^{(j)} \xi' + \xi'^T g_i^{(j)} + \nu_i^{(j)} \geq 0, \quad \forall i = 1, 2, ..., I\}$

In other words,

$$F^{(j)} - \sum_{i=1}^{I} \lambda_i^{(j)} G_i^{(j)} \succeq 0 \iff \begin{bmatrix} 1 \\ \xi' \end{bmatrix}^T F^{(j)} \begin{bmatrix} 1 \\ \xi' \end{bmatrix} \geq 0, \quad \forall \xi', \ \begin{bmatrix} 1 \\ \xi' \end{bmatrix}^T G_i^{(j)} \begin{bmatrix} 1 \\ \xi' \end{bmatrix} \geq 0, \ \forall i = 1, 2, ..., I$$

The obtained result thus implies that if the constant decision rules were used, the corresponding semidefinite program would be an exact deterministic equivalent, not just an approximation, of the pricing model under uncertainty.

As a constant decision rule is a specialisation of a linear decision rule, the generated optimal solution from the semidefinite program corresponding to the constant decision rule pricing model must be feasible in the semidefinite program corresponding to the linear decision rule pricing model. Note that, by the argument of the S-lemma, this generated solution must also remain feasible in the pricing model (5.26) as well. Hence, in terms of the optimality of the generated solution, it is not possible for the constant decision rule pricing model to outperform the linear decision rule pricing model.

$\square$

5.3 Piecewise Linear Decision Rule Pricing Model

In this section, instead of employing linear decision rules, we formulate another pricing model using piecewise linear decision rules. The motivation for this is that even though linear decision rules usually perform better than constant decision rules in the sense that the observed information is taken into account, still they are usually subject to insufficient accuracy. The piecewise linear decision rule approach was proposed to address this issue using the idea of solving the optimisation problems under uncertainty in the lifted space, which gives decision variables extra flexibility and yet preserves the tractability of the original linear decision rule approach. To be more specific, in the piecewise linear decision rule pricing model, we define the decision variables $\{u_t\}_{t=1}^T$ as

$$\begin{aligned} u_t = u_{t,0} &+ u_{t,1}\min(\tilde{S}_1, z_1) + u_{t,2}\max(\tilde{S}_1 - z_1, 0) \\ &+ u_{t,3}\min(\tilde{S}_2, z_2) + u_{t,4}\max(\tilde{S}_2 - z_2, 0) \\ &\vdots \\ &+ u_{t,2t-1}\min(\tilde{S}_t, z_t) + u_{t,2t}\max(\tilde{S}_t - z_t, 0), \end{aligned} \tag{5.33}$$

where $\{z_t\}_{t=1}^T$ are predefined breakpoints along the axis $\tilde{S}_t$.

Because of the fact that $\min(\tilde{S}_t, z_t) + \max(\tilde{S}_t - z_t, 0) = S_t$, the linear decision rule pricing model can be thought of as a specialisation of the piecewise linear decision rule pricing model where $u_{t,2i-1} = u_{t,2i}$, $\forall i = 1, 2, ..., t$. It can be thus concluded that the piecewise linear decision rule pricing model is richer than the linear decision rule pricing model, and therefore it is richer than the constant decision rule pricing model.

Furthermore, it is worth noting here that the piecewise linear decision rule pricing model is considered to be subject to uncertainty in a lifted space which consists of $2T$ uncertain parameters: $\{\min(\tilde{S}_t, z_t)\}_{t=1}^T$ and $\{\max(\tilde{S}_t - z_t, 0)\}_{t=1}^T$. To avoid future confusion, in the sequel, we use the term *original space* to refer to the space where the uncertainty set Ξ describing the possibilities of the scenario $\xi \in \mathbb{R}^{T+1}$ in the linear decision rule pricing model belongs to. The uncertainty set Ξ is a polyhedron and thus can be expressed by a matrix inequality $W\xi \geq h$, where W and h are a matrix and a vector of appropriate dimensions respectively. A scenario ξ' in the piecewise linear decision rule pricing model is described by an uncertainty set Ξ' in the *lifted space*. As an example, we demonstrate the procedure of how to employ piecewise linear decision rules in the proposed European call option pricing model.

Recall that by introducing the decision rules to the new robust pricing model for the European call options (see (5.13)), we have

European Call Option Pricing: The Decision Rule Formulation

$$\begin{gathered} \text{minimise}_{n_0^S, n_0^B, \{u_t(\{\tilde{S}_i\}_{i=1}^t)\}_{t=1}^{T-1}} \ \text{maximise}_{\{S_t\}_{t=1}^T \in U} \\ \left| \left(\tilde{S}_T - K\right)^+ - \left(n_0^S + \sum_{t=1}^{T-1} u_t\left(\{\tilde{S}_i\}_{i=1}^t\right)\right)\tilde{S}_T - \left(n_0^B - \sum_{t=1}^{T-1} \frac{u_t\left(\{\tilde{S}_i\}_{i=1}^t\right)\tilde{S}_t}{B_t}\right)B_T\right| \\ \text{subject to} \\ n_0^S S_0 + n_0^B B_0 \geq 0. \end{gathered} \tag{5.34}$$

Let ξ'_t again denote the information previously observed up to time t in the lifted space and ξ' denote ξ'_T. (The definition of ξ_t denoting the previously observed information in the linear decision rule pricing model still remains unchanged.)

$$\xi'_t = \begin{bmatrix} 1 \\ \min(\tilde{S}_1, z_1) \\ \max(\tilde{S}_1 - z_1, 0) \\ \min(\tilde{S}_2, z_2) \\ \max(\tilde{S}_2 - z_2, 0) \\ \vdots \\ \min(\tilde{S}_t, z_t) \\ \max(\tilde{S}_t - z_t, 0) \end{bmatrix}_{(2t+1)\times 1} \tag{5.35}$$

$$\xi' = \xi'_T = \begin{bmatrix} 1 \\ \min(\tilde{S}_1, z_1) \\ \max(\tilde{S}_1 - z_1, 0) \\ \min(\tilde{S}_2, z_2) \\ \max(\tilde{S}_2 - z_2, 0) \\ \vdots \\ \min(\tilde{S}_T, z_T) \\ \max(\tilde{S}_T - z_T, 0) \end{bmatrix}_{(2T+1)\times 1} \tag{5.36}$$

Similarly to the linear decision rule pricing model, a linear relationship between ξ'_t and ξ' can be observed. That is

$$\xi'_t = M'_t \xi', \tag{5.37}$$

where M'_t is a $(2t+1)\times(2T+1)$ matrix defined as an aggregation of an identity matrix $\mathbb{I}_{2t+1}$ and a zero matrix $0_{(2t+1)\times(2T-2t)}$.

$$M'_t = \left[\mathbb{I}_{2t+1} | 0_{(2t+1)\times(2T-2t)}\right] = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & & & & \vdots & \vdots & & & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \end{bmatrix}_{(2t+1)\times(2T+1)} \tag{5.38}$$

Let e_t be a t^{th} canonical basis vector of appropriate dimension (which should be clear from the context), i.e., a vector containing one at the t^{th} position and zero elsewhere. A relation between $\tilde{S}_t$ and ξ' can then be written as follows.

$$\tilde{S}_t = (e_{2t}^T + e_{2t+1}^T)\xi'_T = (e_{2t}^T + e_{2t+1}^T)\xi' \tag{5.39}$$

Lastly, we reuse the set of constants $\{r_t\}_{t=1}^T$ and define them in the same way we do in the previous section.

$$r_t = \frac{B_T}{B_t} \tag{5.40}$$

The European call pricing model (5.34) can then be equivalently rewritten as follows.

$$\begin{gathered}
\text{minimise}_{n_0^S, n_0^B, \{u_t(\xi_t')\}_{t=1}^{T-1}} \ \text{maximise}_{\{S_t\}_{t=1}^{T} \in U} \\
\left| \left((e_{2T}^T + e_{2T+1}^T)\xi' - K\right)^+ - \left(n_0^S + \sum_{t=1}^{T-1} u_t\left(\xi_t'\right)\right)(e_{2T}^T + e_{2T+1}^T)\xi' - \right. \\
\left. \left(n_0^B B_T - \sum_{t=1}^{T-1} u_t\left(\xi_t'\right)(e_{2t}^T + e_{2t+1}^T)\xi' r_t\right) \right| \\
\text{subject to} \\
n_0^S S_0 + n_0^B B_0 \geq 0
\end{gathered} \tag{5.41}$$

For each t, if the decision variable $u_t\left(\xi_t'\right)$ is modelled as a piecewise linear function of the history of observation ξ_t', then it can be written as

$$u_t(\xi_t') = \begin{bmatrix} u_{t,0} \\ u_{t,1} \\ \vdots \\ u_{t,2t} \end{bmatrix}^T \xi_t'. \tag{5.42}$$

The implication of this is that there is a vector U_t comprised of $2t+1$ elements through which the linear dependence is exhibited.

$$u_t(\xi_t') = U_t^T \xi_t' = U_t^T M_t' \xi' \tag{5.43}$$

These piecewise linear decision rules lead to the following formulation of the European call option pricing problem.

$$\begin{gathered}
\text{minimise}_{n_0^S, n_0^B, \{U_t\}_{t=1}^{T-1}} \ \text{maximise}_{\xi' \in \Xi'} \\
\left| \left((e_{2T}^T + e_{2T+1}^T)\xi' - K\right)^+ - \left(n_0^S + \sum_{t=1}^{T-1} U_t^T M_t' \xi'\right)(e_{2T}^T + e_{2T+1}^T)\xi' - \right. \\
\left. \left(n_0^B B_T - \sum_{t=1}^{T-1} U_t^T M_t' \xi' (e_{2t}^T + e_{2t+1}^T)\xi' r_t\right) \right| \\
\text{subject to} \\
n_0^S S_0 + n_0^B B_0 \geq 0,
\end{gathered} \tag{5.44}$$

which can be rewritten as

European Call Option Pricing: PIE-ECO

$$\begin{gathered}
\text{minimise}_{n_0^S, n_0^B, \{U_t\}_{t=1}^{T-1}} \ \text{maximise}_{\xi' \in \Xi'} \\
\left| \left(\xi'^T(e_{2T} + e_{2T+1}) - K\right)^+ + \xi'^T \left(\sum_{t=1}^{T-1} \left((e_{2t} + e_{2t+1}) U_t^T M_t' r_t - (e_{2T} + e_{2T+1}) U_t^T M_t'\right)\right) \xi' \right. \\
\left. + \xi'^T \left(-(e_{2T} + e_{2T+1}) n_0^S\right) + \left(-n_0^B B_T\right) \right| \\
\text{subject to} \\
n_0^S S_0 + n_0^B B_0 \geq 0.
\end{gathered}$$

(5.45)

Notice that the objective function is again quadratic in the uncertain parameter ξ'.

In this pricing model, the possibilities of ξ' are described by the uncertainty set Ξ'. Recall that, in the constant decision rule pricing model and the linear decision rule pricing model, U and Ξ are polyhedrons obtained from the upper bound and the lower bound of the price of the option's underlier as well as the upper bound and the lower bound of its single-period return at time $t = 1, 2, ..., T$. This idea is still adopted in the piecewise linear decision rule pricing model, but it cannot be used directly since the uncertainty set Ξ' describes the acceptable values of ξ', not $\xi = [1\ \tilde{S}_1\ \tilde{S}_2 \ldots \tilde{S}_T]^T$. Unfortunately, to the best of our knowledge, the exact tractable representation of Ξ' cannot be easily obtained. Georghiou, Wiesemann and Kuhn [23], however, show that it is still possible to determine $\hat{\Xi}'$, a tractable approximation of Ξ'. This approximation is obtained by first identifying as small as possible a box $l \leq \xi \leq u$ containing the uncertainty set $\Xi = \{\xi \in \mathbb{R}^{T+1} \mid W\xi \geq h\}$, which is a polyhedron, inside. Hence, the uncertainty set Ξ can be rewritten as $\Xi = \{\xi \in \mathbb{R}^{T+1} \mid W\xi \geq h \ \wedge\ l \leq \xi \leq u\}$. The outer approximation $\hat{\Xi}'$ of the convex hull of Ξ' is thus an intersection of the polyhedron $W\xi \geq h$ in the lifted space and the convex hull of the box $l \leq \xi \leq u$ in the lifted space. In this model, $\hat{\Xi}'$ can be readily constructed as such a box is already available in the form of the lower bound ($\underline{S}_t = S_0\underline{R}_t^S$) and the upper bound ($\bar{S}_t = S_0\bar{R}_t^S$) of the underlier's price.

$$
\begin{aligned}
\hat{\Xi}' = \{\xi' \in \mathbb{R}^{2T+1} \mid\ & \text{(constraints on the lower and the upper bounds of the underlier's prices,}\\
& \text{i.e., the outer box)}\\
& \begin{bmatrix}
\dfrac{z_t}{z_t - S_0\underline{R}_t^S} & \dfrac{-1}{z_t - S_0\underline{R}_t^S} & 0 \\[2ex]
\dfrac{-S_0\underline{R}_t^S}{z_t - S_0\underline{R}_t^S} & \dfrac{1}{z_t - S_0\underline{R}_t^S} & \dfrac{-1}{S_0\bar{R}_t^S - z_t} \\[2ex]
0 & 0 & \dfrac{1}{S_0\bar{R}_t^S - z_t}
\end{bmatrix}
\begin{bmatrix} 1 \\ \xi'^T e_{2t} \\ \xi'^T e_{2t+1} \end{bmatrix} \geq 0,\\
& \quad \forall t = 1, 2, ..., T \quad \wedge\\[2ex]
& \text{(constraints on the lower bounds of the single-period returns)}\\
& \xi'^T(e_2 + e_3) = \tilde{S}_1 \geq S_0(1 + \underline{r}_0^S) \quad \wedge\\
& \xi'^T\left(e_{2t} + e_{2t+1} - \left(1 + \underline{r}_{t-1}^S\right)(e_{2t-2} + e_{2t-1})\right)\\
& \qquad = \tilde{S}_t - \left(1 + \underline{r}_{t-1}^S\right)\tilde{S}_{t-1} \geq 0\ , \quad \forall t = 2, 3, ..., T \quad \wedge\\[2ex]
& \text{(constraints on the upper bounds of the single-period returns)}\\
& \xi'^T(e_2 + e_3) = \tilde{S}_1 \leq S_0(1 + \bar{r}_0^S) \quad \wedge\\
& \xi'^T\left(e_{2t} + e_{2t+1} - \left(1 + \bar{r}_{t-1}^S\right)(e_{2t-2} + e_{2t-1})\right)\\
& \qquad = \tilde{S}_t - \left(1 + \bar{r}_{t-1}^S\right)\tilde{S}_{t-1} \leq 0\ , \quad \forall t = 2, 3, ..., T \quad \wedge\\[2ex]
& \text{(constraint on the first element of } \xi')\\
& \xi'^T e_1 = 1\\
& \}
\end{aligned}
$$

(5.46)

Definition 5.3.1. (Retraction operator) *A retraction operator R is a function mapping a scenario $\xi' \in \mathbb{R}^{2T+1}$ in the lifted space to a corresponding scenario $\xi \in \mathbb{R}^{T+1}$ in the original space.*

$$R : \mathbb{R}^{2T+1} \longrightarrow \mathbb{R}^{T}, \quad R(\xi') = \xi.$$

In our setting, R is linear since

$$\xi'^T(e_{2t} + e_{2t+1}) = \min(\tilde{S}_t, z_t) + \max(\tilde{S}_t - z_t, 0) = \tilde{S}_t \tag{5.47}$$

In other words, we have

$$R(\xi') = R\xi' = \xi. \tag{5.48}$$

Specifically, R is a $(T+1) \times (2T+1)$ matrix with the following description.

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & \dots & 0 & 0 \\ \vdots & & & & & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 1 \end{bmatrix} \tag{5.49}$$

The operator R is used a lot in further analysis on the piecewise linear decision rule pricing model. Another component that we deem useful is a spanned linear decision rule pricing model defined as follows.

Definition 5.3.2. (Spanned linear decision rule pricing model) *A spanned linear decision rule pricing model is defined as the original linear decision rule pricing model with every ξ (in both constraints and uncertainty sets) replaced by $R\xi'$.*

The spanned linear decision rule pricing model can be considered to be in a lifted space because the uncertain parameter is denoted by ξ'. The proposition below relates the spanned linear decision rule pricing model to the original linear decision rule pricing model.

Proposition 5.3.1. *The spanned linear decision rule pricing model and the linear decision rule pricing model attain the same optimal solution.*

Proof.

Each of the constraints in the original linear decision rule pricing model is quadratic in the uncertain parameter ξ. Consequently, we can write each constraint in the following manner where j is the index of the constraint in the model.

$$\xi^T F^{(j)} \xi \geq 0, \quad \forall \xi \in \Xi^{(j)}$$

In this formulation, we can think of $F^{(j)}$ as a decision matrix because it is comprised of decision variables: $\{U_t\}_{t=1}^{T-1}, n_0^S,\ n_0^B$, and ϵ, in a specific pattern. According to our definition, a set of constraints in the spanned linear decision rule pricing model is given by

$$(R\xi')^T F^{(j)} (R\xi') \geq 0, \quad \forall \xi',\ R\xi' \in \Xi^{(j)}.$$

In order to prove that both pricing models yield the same optimal solution, it is sufficient to show that both problems have the same feasible set. The common technique for verifying the equality of two sets is to show that each of them is a subset of the other.

($\Longrightarrow$) Suppose that $\check{F}$ satisfies the condition

$$\xi^T \check{F} \xi \geq 0, \quad \forall \xi \in \Xi^{(j)}.$$

Thus, for any ξ' such that $R\xi' \in \Xi^{(j)}$, we have

$$(R\xi')^T \check{F} (R\xi') \geq 0,$$

which implies the feasibility of $\check{F}$ in the spanned linear decision rule pricing model.

($\Longleftarrow$) In the reverse direction, suppose that $\check{F}$ satisfies the condition

$$(R\xi')^T \check{F}(R\xi') \geq 0, \quad \forall \xi',\ R\xi' \in \Xi^{(j)}.$$

For any $\xi \in \Xi^{(j)} \subset \mathbb{R}^{T+1}$, there exists a vector $\xi' \in \mathbb{R}^{2T+1}$ such that $R\xi' = \xi$. For example, we can assign the value of the i^{th} position of ξ to the $(2i-1)^{th}$ position of ξ', $i = 1, 2, ..., T+1$, and assign zero to every other entry in ξ'. Hence,

$$\xi^T \check{F} \xi = (R\xi')^T \check{F}(R\xi') \geq 0,$$

which implies the feasibility of $\check{F}$ in the linear decision rule pricing model.

□

In order to evaluate the performance of the piecewise linear decision rule pricing model, we introduce two more definitions below to describe a desirable characteristic that the predefined breakpoints $\{z_t\}_{t=1}^T$ should have.

Definition 5.3.3. (Proper breakpoint) *For an uncertain parameter $\upsilon \in \mathbb{R}$ taking value in $[\underline{\upsilon}, \bar{\upsilon}]$, a breakpoint z_υ for υ is said to be a proper breakpoint if $z_\upsilon \in [\underline{\upsilon}, \bar{\upsilon}]$.*

Definition 5.3.4. (Proper piecewise linear decision rule pricing model) *A piecewise linear decision rule pricing model is said to be proper if every of its breakpoints is proper.*

The proposition below confirms that, by using the piecewise linear decision rules, the resulting pricing model is expected to perform better than the linear decision rule pricing model does.

Proposition 5.3.2. *The proper piecewise linear decision rule pricing model is at most as conservative as the linear decision rule pricing model.*

Proof.
The outline of this proof is to show that the uncertainty set $\hat{\Xi}'$ in the proper piecewise linear decision rule pricing model is at least as restrictive as the uncertainty set in the spanned linear decision rule pricing model.

$$\xi' \in \hat{\Xi}' \Longrightarrow R\xi' \in \Xi$$

Following the claim, since a piecewise linear decision rule is a generalisation of a linear decision rule, any feasible solution in the spanned linear decision rule pricing model has to also be feasible in the piecewise linear decision rule pricing model. We can thus conclude that the piecewise linear decision rule pricing model provides a tighter upper bound to the true optimal objective function value than that produced by the spanned linear decision rule pricing model, which is identical to the one obtained from the original linear decision rule pricing model.

Recall that there are four categories of boundary conditions in the linear decision rule pricing model: the lower bound of the asset price ($\tilde{S}_t \geq S_0 \underline{R}_t^S$), the upper bound of the asset price ($\tilde{S}_t \leq S_0 \bar{R}_t^S$), the lower bound of the single-period return ($\tilde{S}_t \geq \tilde{S}_{t-1}(1 + \underline{r}_{t-1}^S)$), and the upper bound of the single-period return ($\tilde{S}_t \leq \tilde{S}_{t-1}(1 + \bar{r}_{t-1}^S)$). Therefore, we need to show that a scenario ξ in the original space which is mapped from any admissible ξ' in $\hat{\Xi}'$ using the retraction operator satisfies all of these constraints.

Since we explicitly include in the description of $\hat{\Xi}'$ the constraints on the lower bound and the upper bound of the single-period return at time $t = 1, 2, ..., T$, we only have to prove that the description of $\hat{\Xi}'$ (see (5.46)) implies the lower bound and the upper bound of the price of the underlying asset as well.

Consider a constraint

$$\begin{bmatrix} \dfrac{z_t}{z_t - S_0\underline{R}_t^S} & \dfrac{-1}{z_t - S_0\underline{R}_t^S} & 0 \\ \dfrac{-S_0\underline{R}_t^S}{z_t - S_0\underline{R}_t^S} & \dfrac{1}{z_t - S_0\underline{R}_t^S} & \dfrac{-1}{S_0\bar{R}_t^S - z_t} \\ 0 & 0 & \dfrac{1}{S_0\bar{R}_t^S - z_t} \end{bmatrix} \begin{bmatrix} 1 \\ \xi'^T e_{2t} \\ \xi'^T e_{2t+1} \end{bmatrix} \geq 0$$

in the $[\xi'^T e_{2t},\ \xi'^T e_{2t+1}]$ plane. The constraint implies a closed triangle determined by three edges. The vertices of this triangle are given by

$$\begin{aligned} \left(\xi'^T e_{2t},\ \xi'^T e_{2t+1}\right) &= (S_0\underline{R}_t^S, 0), \\ \left(\xi'^T e_{2t},\ \xi'^T e_{2t+1}\right) &= (z_t, 0), \\ \left(\xi'^T e_{2t},\ \xi'^T e_{2t+1}\right) &= (z_t, S_0\bar{R}_t^S - z_t). \end{aligned}$$

Any arbitrary point in this triangle can be written as a convex combination of these three vertices.

$$\begin{pmatrix} \xi'^T e_{2t} \\ \xi'^T e_{2t+1} \end{pmatrix} = \alpha_1 \begin{pmatrix} S_0\underline{R}_t^S \\ 0 \end{pmatrix} + \alpha_2 \begin{pmatrix} z_t \\ 0 \end{pmatrix} + \alpha_3 \begin{pmatrix} z_t \\ S_0\bar{R}_t^S - z_t \end{pmatrix}, \qquad \alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}_+, \alpha_2 + \alpha_2 + \alpha_3 = 1$$

That implies the acceptable range of $\xi'^T(e_{2t} + e_{2t+1})$.

$$\begin{aligned} \xi'^T(e_{2t} + e_{2t+1}) &= \xi'^T e_{2t} + \xi'^T e_{2t+1} \\ &= \alpha_1 S_0\underline{R}_t^S + \alpha_2 z_t + \alpha_3 z_t + \alpha_3 \left(S_0\bar{R}_t^S - z_t\right) \\ &= \alpha_1 S_0\underline{R}_t^S + \alpha_2 z_t + \alpha_3 S_0\bar{R}_t^S \\ &\in [\min\left(S_0\underline{R}_t^S, z_t, S_0\bar{R}_t^S\right), \max\left(S_0\underline{R}_t^S, z_t, S_0\bar{R}_t^S\right)] \\ &= [S_0\underline{R}_t^S, S_0\bar{R}_t^S], \end{aligned}$$

where the last equation holds for any proper breakpoint z_t. Therefore, $\hat{\Xi}'$ also determines the bounds of the asset price. The uncertainty set $\hat{\Xi}'$ in the lifted space is then at least as restrictive as Ξ, and the proof thus completes.

□

It is observed that this representation of $\hat{\Xi}'$ is a polyhedron. Therefore, it can be treated as an intersection of finitely many ellipsoids. The piecewise linear decision rule pricing model itself seems to be intractable because there are infinitely many elements in the uncertainty set; however, we can again apply the approximate S-lemma (Proposition 3.4.1) so as to obtain the deterministic version of the pricing model, which is a semidefinite program. However, as mentioned earlier, the approximate S-lemma generally results in merely approximate optimisation problem. A comparison between the deterministic semidefinite program corresponding to the piecewise linear decision rule pricing model and that corresponding to the linear decision rule pricing model should therefore be carried out.

Proposition 5.3.3. *A deterministic semidefinite program corresponding to the proper piecewise linear decision rule pricing model is at most as conservative as a deterministic semidefinite program corresponding to the linear decision rule pricing model.*

Proof.
Recall that there are four series of constraints (indexed by j = 1,2,3,4)

$$\xi^T F^{(j)} \xi \geq 0, \quad \forall \xi,\ \xi^T G_i^{(j)} \xi \geq 0,\ \forall i = 1, 2, ..., I$$

in the linear decision rule pricing model, each of which is approximated by

$$F^{(j)} - \sum_{i=1}^{I} \lambda_i^{(j)} G_i^{(j)} \succeq 0,\ \lambda_i^{(j)} \geq 0,$$

where F^j is a matrix made of decision variables: the initial number of the underlying assets and the initial number of the risk-free assets in the portfolio as well as the adjustments to be made to these figures in subsequent periods. Let the optimal solution to this semidefinite program be denoted by $\left(\{F^{*(j)}\}_{j=1}^4, \{\{\lambda_i^{*(j)}\}_{i=1}^I\}_{j=1}^4\right)$. Therefore,

$$F^{*(j)} - \sum_{i=1}^{I} \lambda_i^{*(j)} G_i^{(j)} \succeq 0,\ \lambda_i^{*(j)} \geq 0.$$

From the positive semidefiniteness of $F^{*(j)} - \sum_{i=1}^{I} \lambda_i^{*(j)} G_i^{(j)}$, we have

$$d^T \left(F^{*(j)} - \sum_{i=1}^{I} \lambda_i^{*(j)} G_i^{(j)} \right) d \geq 0, \quad \forall d \in \mathbb{R}^{T+1}.$$

Let R be the retraction operator as in (5.49). The following holds.

$$(Rd)^T \left(F^{*(j)} - \sum_{i=1}^{I} \lambda_i^{*(j)} G_i^{(j)} \right) (Rd) \geq 0, \quad \forall d \in \mathbb{R}^{2T+1}$$

The above statement implies that

$$R^T F^{*(j)} R - \sum_{i=1}^{I} \lambda_i^{*(j)} R^T G_i^{(j)} R = R^T \left(F^{*(j)} - \sum_{i=1}^{I} \lambda_i^{*(j)} G_i^{(j)} \right) R \succeq 0.$$

The result corresponds to the feasibility of this solution in the semidefinite program approximating the spanned linear decision rule pricing model because the inequality $\xi^T F^{*(j)} \xi \geq 0$ in the original space can be transformed to an inequality $(R\xi')^T F^{*(j)} (R\xi') = \xi'^T (R^T F^{*(j)} R) \xi' \geq 0$ in the lifted space. The same argument goes for the uncertainty set, which is given by $\{\xi \in \mathbb{R}^{T+1} \mid \xi^T G_i^{(j)} \xi \geq 0,\ \forall i = 1, 2, ..., I\}$. However, according to the description of the piecewise linear decision rule pricing model, we do not directly map the uncertainty set from the original space to the lifted space. Specifically, we transform a collection of the constraints on the upper bound and the lower bound of the asset price to its convex hull in the lifted space. Consider part of the summation term that contributes to the upper bound ($\tilde{S}_t \leq \bar{S}_t = S_0 \bar{R}_t^S$) and the lower bound of the underlier's price ($\tilde{S}_t \geq \underline{S}_t = S_0 \underline{R}_t^S$). Let say it is

$$R^T \left(\lambda_{t_1}^{*(j)} \begin{bmatrix} -\underline{S}_t & 0 & \dots & 0 & \frac{1}{2} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \frac{1}{2} & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix} + \lambda_{t_2}^{*(j)} \begin{bmatrix} \bar{S}_t & 0 & \dots & 0 & -\frac{1}{2} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ -\frac{1}{2} & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \right) R.$$

Note that the non-zero elements appear only at positions 1×1, $(t+1) \times 1$, and $1 \times (t+1)$. Rearranging terms in the formula above, we get the resulting $(2T+1) \times (2T+1)$ matrix given by

$$
\begin{bmatrix}
-\lambda_{t_1}^{*(j)}\underline{S}_t+\lambda_{t_2}^{*(j)}\bar{S}_t & 0 & \dots & 0 & \dfrac{\lambda_{t_1}^{*(j)}-\lambda_{t_2}^{*(j)}}{2} & \dfrac{\lambda_{t_1}^{*(j)}-\lambda_{t_2}^{*(j)}}{2} & 0 & \dots & 0\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\dfrac{\lambda_{t_1}^{*(j)}-\lambda_{t_2}^{*(j)}}{2} & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\dfrac{\lambda_{t_1}^{*(j)}-\lambda_{t_2}^{*(j)}}{2} & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0
\end{bmatrix},
$$

where the non-zero elements appear only at positions 1×1, $2t\times 1$, $(2t+1)\times 1$, $1\times 2t$, and $1\times(2t+1)$. This matrix can be rewritten as

$$
\lambda_{t_2}^{*(j)}(\bar{S}_t-\underline{S}_t)
\begin{bmatrix}
\dfrac{z_t}{z_t-\underline{S}_t} & 0 & \dots & 0 & \dfrac{-1}{2(z_t-\underline{S}_t)} & 0 & 0 & \dots & 0\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\dfrac{-1}{2(z_t-\underline{S}_t)} & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0
\end{bmatrix}
$$

$$
+\lambda_{t_1}^{*(j)}(\bar{S}_t-\underline{S}_t)
\begin{bmatrix}
0 & 0 & \dots & 0 & 0 & \dfrac{1}{2(\bar{S}_t-z_t)} & 0 & \dots & 0\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\dfrac{1}{2(\bar{S}_t-z_t)} & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0
\end{bmatrix}
$$

$$
+\left(\lambda_{t_2}^{*(j)}(\bar{S}_t-z_t)+\lambda_{t_1}^{*(j)}(z_t-\underline{S}_t)\right)
\begin{bmatrix}
\dfrac{-\underline{S}_t}{z_t-\underline{S}_t} & 0 & \dots & 0 & \dfrac{1}{2(z_t-\underline{S}_t)} & \dfrac{-1}{2(\bar{S}_t-z_t)} & 0 & \dots & 0\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\dfrac{1}{2(z_t-\underline{S}_t)} & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\dfrac{-1}{2(\bar{S}_t-z_t)} & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0\\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots\\
0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0
\end{bmatrix}.
$$

The latter three matrices are representative of the outer box in the description of $\hat{\Xi}'$ (see (5.46)).

Moreover, all of the three coefficients are non-negative provided that the breakpoint z_t is proper. Therefore, we successfully map the optimal solution of the semidefinite program corresponding to the linear decision rule pricing model to a feasible solution of the semidefinite program corresponding to the proper piecewise linear decision rule pricing model. The proof thus completes.

$\square$

5.4 Conclusions

In this chapter, we propose a new way to formulate the robust optimisation model representing the option pricing problem. The proposed model, i.e., the constant decision rule pricing model, can be considered as an amended version of the original ϵ-arbitrage robust pricing model. We also introduce the linear decision rules and the piecewise linear decision rules to the new robust pricing model. As an example, we show how to use these forms of decision rules when the option considered is a European call option. Such decision rules, in addition, can be applied to the Asian call option pricing model, the fixed strike lookback call option pricing model, and the American put option pricing model as well. We also develop theories that guarantee that, in terms of the optimality of the generated solutions, the piecewise linear decision rule pricing model performs at least as good as the linear decision rule pricing model, and the linear decision rule pricing model performs as good as the constant decision rule pricing model, if not better.

Chapter 6

Valuation of Multiple-Underlier Options

In the previous chapters, we consider only options with single underlying asset. In the real market, there are, in addition, options whose payoffs depend on multiple underliers. In this chapter, we show that the robust pricing model can also be extended to price such options. Analogously to the previously discussed pricing models, the extended pricing model outputs a portfolio consisting of basic securities which are the underliers of the option and the risk-free asset. The price of the option should thus be set to today's value of the portfolio because the final wealth (W_T) of the output portfolio matches most closely to the option payoff in the worst-case sense.

Suppose that the payoff of the option depends on the realised prices of M securities denoted by $1, 2, ..., M$. The price of the security m at time t is subject to uncertainty and denoted by a random variable $\tilde{S}_t^m$. For the sake of convenience, we also denote the risk-free asset by the number zero. A replicating portfolio is characterised by the number of each asset in the portfolio at different times. Specifically, a portfolio is determined by n_t^m $(t = 0, 1, 2, ..., T,\ m = 0, 1, 2, ..., M)$ which represents the number of assets m held in the portfolio at time t. Let uncertain parameter $\tilde{S}$ be a collection of assets' prices $\{\tilde{S}_t^m\}_{t=1,m=1}^{t=T,m=M}$. As before, P denotes the option payoff, which depends on the information $\tilde{S}$ and the strike price K. Hence, we can construct a general robust option pricing model for multiple-underlier options as follows.

$$
\begin{aligned}
&\text{minimise}_{\{n_t^m\}_{t=0,m=0}^{t=T,m=M},\{u_t^m\}_{t=1,m=0}^{t=T,m=M}}\ \text{maximise}_{\{\tilde{S}_t^m\}_{t=1,m=1}^{t=T,m=M}\in U} \quad \left|P(\tilde{S},K) - W_T\right| \\
&\text{subject to} \\
&\quad W_T = n_T^0 B_T + \sum_{m=1}^{M} n_T^m \tilde{S}_T^m \\
&\quad n_t^m = n_{t-1}^m + u_t^m, \qquad \forall t = 1, 2, ..., T, \quad \forall m = 0, 1, ..., M \\
&\quad u_t^0 B_t + \sum_{m=1}^{M} u_t^m \tilde{S}_t^m = 0, \qquad \forall t = 1, 2, ..., T \\
&\quad n_0^0 B_0 + \sum_{m=1}^{M} n_0^m S_0^m \geq 0,
\end{aligned}
\tag{6.1}
$$

where U is an uncertainty set describing the uncertain parameters $\{\tilde{S}_t^m\}_{t=1,m=1}^{t=T,m=M}$. The second and the third constraints account for the rebalancing activities which can be done before the expiration date of the option. Solving the pricing model, the proposed price for the option is

$$
n_0^0 B_0 + \sum_{m=1}^{M} n_0^m S_0^m, \tag{6.2}
$$

which is the initial wealth of the portfolio. The last constraint in the model is needed in order to prevent the model from outputting negative prices, which would lead to an arbitrage opportunity provided that the option payoff can never be negative.

From the rebalancing equations, i.e., the second and the third constraints in the pricing model, it is observed that

$$n_T^m = n_0^m + \sum_{t=1}^{T} u_t^m, \quad \forall m = 1, 2, ..., M, \tag{6.3}$$

$$u_t^0 = \frac{-\sum\limits_{m=1}^{M} u_t^m \tilde{S}_t^m}{B_t}, \quad \forall t = 1, 2, ..., T. \tag{6.4}$$

These results lead to a simplified version of the pricing model below.

$$\begin{gathered}
\text{minimise}_{\{n_0^m\}_{m=0}^{m=M}, \{u_t^m\}_{t=1,m=1}^{t=T-1,m=M}} \text{maximise}_{\{\tilde{S}_t^m\}_{t=1,m=1}^{t=T,m=M} \in U} \\
\left| P(\tilde{S}, K) - \left(n_0^0 - \sum_{t=1}^{T-1} \frac{\sum\limits_{m=1}^{M} u_t^m \tilde{S}_t^m}{B_t} \right) B_T - \sum_{m=1}^{M} \left(n_0^m + \sum_{t=1}^{T-1} u_t^m \right) \tilde{S}_T^m \right| \\
\text{subject to} \\
n_0^0 B_0 + \sum_{m=1}^{M} n_0^m S_0^m \geq 0
\end{gathered} \tag{6.5}$$

For notational convenience, we introduce new series of variables as follows. These variables are extensively used in the model derivation henceforward.

- $H_0 = n_0^0 B_T$
- For $1 \leq m \leq M$ and $1 \leq t \leq T-1$, $\quad H_t^m = -\dfrac{u_t^m B_T}{Bt}$
- For $1 \leq m \leq M$, $\quad H_T^m = n_0^m + \sum\limits_{t=1}^{T-1} u_t^m$

6.1 Basket Options

Specifically, in this chapter, the option of our interest is a (European) basket option. A basket option is very similar to a European option whose payoff is determined by the realised value of its underlier. Payoff of a basket option is determined by the realised value of the basket of the underlying assets. Mathematically speaking, the payoff of the basket call option is given by

$$P(\tilde{S}, K) = \left(\sum_{m=1}^{M} w_m \tilde{S}_T^m - K \right)^+, \tag{6.6}$$

where the weights $\{w_m\}_{m=1}^{M}$ and the strike price K are agreed today in the option contract. Conventionally, these weights sum to one, but this is not a necessary condition though. The robust pricing model for basket call options is given by

Basket Call Option Pricing: CON-BKT

$$\text{minimise}_{\{n_0^m\}_{m=0}^{m=M},\{u_t^m\}_{t=1,m=1}^{t=T-1,m=M}} \text{maximise}_{\{\tilde{S}_t^m\}_{t=1,m=1}^{t=T,m=M}\in U}$$
$$\left|\left(\sum_{m=1}^{M} w_m \tilde{S}_T^m - K\right)^+ - \left(H_0 + \sum_{m=1}^{M}\sum_{t=1}^{T} H_t^m \tilde{S}_t^m\right)\right|$$
$$\text{subject to}$$
$$n_0^0 B_0 + \sum_{m=1}^{M} n_0^m S_0^m \geq 0. \tag{6.7}$$

The pricing model can be further rewritten as

$$\text{minimise}_{\{n_0^m\}_{m=0}^{m=M},\{u_t^m\}_{t=1,m=1}^{t=T-1,m=M},\epsilon}\ \epsilon$$
$$\text{subject to}$$
$$\begin{aligned}
\left(\sum_{m=1}^{M} w_m \tilde{S}_T^m - K\right) - \left(H_0 + \sum_{m=1}^{M}\sum_{t=1}^{T} H_t^m \tilde{S}_t^m\right) &\leq \epsilon,\ \forall\{\tilde{S}_t^m\}_{t=1,m=1}^{t=T,m=M} \in U, \sum_{m=1}^{M} w_m \tilde{S}_T^m \geq K\\
-\left(\sum_{m=1}^{M} w_m \tilde{S}_T^m - K\right) + \left(H_0 + \sum_{m=1}^{M}\sum_{t=1}^{T} H_t^m \tilde{S}_t^m\right) &\leq \epsilon,\ \forall\{\tilde{S}_t^m\}_{t=1,m=1}^{t=T,m=M} \in U, \sum_{m=1}^{M} w_m \tilde{S}_T^m \geq K\\
-\left(H_0 + \sum_{m=1}^{M}\sum_{t=1}^{T} H_t^m \tilde{S}_t^m\right) &\leq \epsilon,\ \forall\{\tilde{S}_t^m\}_{t=1,m=1}^{t=T,m=M} \in U, \sum_{m=1}^{M} w_m \tilde{S}_T^m \leq K\\
+\left(H_0 + \sum_{m=1}^{M}\sum_{t=1}^{T} H_t^m \tilde{S}_t^m\right) &\leq \epsilon,\ \forall\{\tilde{S}_t^m\}_{t=1,m=1}^{t=T,m=M} \in U, \sum_{m=1}^{M} w_m \tilde{S}_T^m \leq K\\
n_0^0 B_0 + \sum_{m=1}^{M} n_0^m S_0^m &\geq 0.
\end{aligned} \tag{6.8}$$

6.2 Uncertainty Model

Assuming that historical prices of each asset are available, the upper bounds and the lower bounds of the assets' prices and their single-period returns in subsequent periods can be estimated as before (see (2.16) and (2.21)). Moreover, to avoid over conservatism, it is desirable to impose another type of restriction on the movement of assets' prices as a whole.

Since the previous single-period returns are known, the covariance matrix of single-period returns of the assets of our interest can be estimated by using, for instance, sample covariance matrix. Furthermore, the asset price in the first period, i.e., $\tilde{S}_1^m$, is linearly dependent on the single-period return since $\tilde{S}_1^m = S_0^m(1+\tilde{r}_0^m)$. It is therefore possible to include a correlation-based constraint in the uncertainty set U, which contains acceptable values of the uncertain parameters $\{\tilde{S}\}_{t=1,m=1}^{t=T,m=M}$ of the pricing model, using this covariance matrix Σ.

To sum up, the uncertainty set U in the basket option pricing model has five types of constraints:

- Lower bounds of the assets' prices: $\tilde{S}_t^m \geq \underline{S}_t^m, \quad \forall t = 1,2,...,T,\ \forall m = 1,2,...,M;$
- Upper bounds of the assets' prices: $\tilde{S}_t^m \leq \bar{S}_t^m, \quad \forall t = 1,2,...,T,\ \forall m = 1,2,...,M;$

- Lower bounds of the single-period returns: $\tilde{S}_t^m \geq \tilde{S}_{t-1}^m(1+\underline{r}_{t-1}^m)$, $\forall t = 1,2,...,T$, $\forall m = 1,2,...,M$;
- Upper bounds of the single-period returns: $\tilde{S}_t^m \leq \tilde{S}_{t-1}^m(1+\bar{r}_{t-1}^m)$, $\forall t = 1,2,...,T$, $\forall m = 1,2,...,M$;
- Movement of the first-period assets' prices: $\left\|C(\tilde{R}_1 - \check{R}_1)\right\| \leq \delta$, where
  - $C = \Sigma^{-1/2}$ which can be calculated via Cholesky decomposition, i.e., $C^T C = \Sigma^{-1}$, provided that Σ is symmetric and positive definite;
  - $\tilde{R}_1 = \left[\frac{\tilde{S}_1^1}{S_0^1}\ \frac{\tilde{S}_1^2}{S_0^2} \cdots \frac{\tilde{S}_1^M}{S_0^M}\right]^T$;
  - $\check{R}_1$ is an expectation of $\tilde{R}_1$.

Notice that all of these conditions can be equivalently expressed via membership of the second-order cone $\zeta_{2,n} = \left\{ \begin{bmatrix} u \\ t \end{bmatrix} \,\middle|\, u \in \mathbb{R}^{n-1}, t \in \mathbb{R}, \|u\| \leq t \right\}$.

- $\left[\tilde{S}_t^m - \underline{S}_t^m\right] \in \zeta_{2,1}$, $\quad \forall t = 1,2,...,T,\ \forall m = 1,2,...,M.$
- $\left[-\tilde{S}_t^m + \bar{S}_t^m\right] \in \zeta_{2,1}$, $\quad \forall t = 1,2,...,T,\ \forall m = 1,2,...,M.$
- $\left[\tilde{S}_t^m - \tilde{S}_{t-1}^m(1+\underline{r}_{t-1}^m)\right] \in \zeta_{2,1}$, $\quad \forall t = 1,2,...,T,\ \forall m = 1,2,...,M.$
- $\left[-\tilde{S}_t^m + \tilde{S}_{t-1}^m(1+\bar{r}_{t-1}^m)\right] \in \zeta_{2,1}$, $\quad \forall t = 1,2,...,T,\ \forall m = 1,2,...,M.$
- $\begin{bmatrix} C(\tilde{R}_1 - \check{R}_1) \\ \delta \end{bmatrix} \in \zeta_{2,M+1}$.

6.3 Deterministic Equivalent Derivation

Consider the pricing model (6.8). Each constraint is linear in the uncertain parameters $\{\tilde{S}_t^m\}_{t=1,m=1}^{t=T,m=M}$. It can be rewritten as a sub-optimisation problem whose constraints are defined through the membership of the second-order cone. As an exemplar, consider the fourth constraint in the pricing model (6.8).

$$\left(H_0 + \sum_{m=1}^{M}\sum_{t=1}^{T} H_t^m \tilde{S}_t^m\right) \leq \epsilon, \quad \forall \{\tilde{S}_t^m\}_{t=1,m=1}^{t=T,m=M} \in U, \sum_{m=1}^{M} w_m \tilde{S}_T^m \leq K \tag{6.9}$$

This is equivalent to saying that

$$\left(\begin{array}{c} \text{minimise}_{\{\tilde{S}_t^m\}_{t=1,m=1}^{t=T,m=M}} \ -\sum_{m=1}^{M}\sum_{t=1}^{T} H_t^m \tilde{S}_t^m \\ \text{subject to} \\ \left[\tilde{S}_t^m - \underline{S}_t^m\right] \in \zeta_{2,1}, \quad \forall t = 1,2,...,T,\ \forall m = 1,2,...,M \\ \left[-\tilde{S}_t^m + \bar{S}_t^m\right] \in \zeta_{2,1}, \quad \forall t = 1,2,...,T,\ \forall m = 1,2,...,M \\ \left[\tilde{S}_t^m - \tilde{S}_{t-1}^m(1+\underline{r}_{t-1}^m)\right] \in \zeta_{2,1}, \quad \forall t = 1,2,...,T,\ \forall m = 1,2,...,M \\ \left[-\tilde{S}_t^m + \tilde{S}_{t-1}^m(1+\bar{r}_{t-1}^m)\right] \in \zeta_{2,1}, \quad \forall t = 1,2,...,T,\ \forall m = 1,2,...,M \\ \begin{bmatrix} C(\tilde{R}_1 - \check{R}_1) \\ \delta \end{bmatrix} \in \zeta_{2,M+1} \\ \left[-\sum_{m=1}^{M} w_m \tilde{S}_T^m + K\right] \in \zeta_{2,1} \end{array}\right) \geq H_0 - \epsilon. \tag{6.10}$$

We can then use the duality in second-order cone programming (the dual problem of the sub-optimisation problem (6.10) is a maximisation problem with no uncertain parameters $\{\tilde{S}_t^m\}_{t=1,m=1}^{t=T,m=M}$) and obtain its deterministic equivalent, which is a second-order cone programming problem.

6.4 Incorporation of Multi-Period Returns Information

Intuitively, it may seem that we could also include in the pricing model the restriction on the movement of assets' prices at every stage $t > 1$ in the same way we do with the first-stage assets' prices in order to prevent the pricing model from being overly conservative. However, there is a reason of us not doing so in the previous sections. In order to impose a restriction on the first-stage assets' prices, we use the knowledge of the covariance matrix of assets' returns, i.e., the sample covariance matrix of assets' single-period returns. Analogously, if we have precise knowledge about covariance matrix of τ-period returns Σ_τ, then it is also possible to impose a restriction on the movement of assets' prices at time τ; this restriction is given by

$$\left\| C_\tau(\tilde{R}_\tau - \check{R}_\tau) \right\| \leq \delta_\tau, \tag{6.11}$$

where

δ_τ is a predefined parameter whose value reflects the degree of risk-tolerance;

$\tilde{R}_\tau$ is a vector given by $\left[\frac{\tilde{S}^1_\tau}{S^1_0} \; \frac{\tilde{S}^2_\tau}{S^2_0} \cdots \frac{\tilde{S}^M_\tau}{S^M_0} \right]^T$;

$\check{R}_\tau$ is an expectation of $\tilde{R}_\tau$;

$C_\tau = \Sigma_\tau^{-1/2}$ which can be obtained via Cholesky decomposition, i.e., $C_\tau^T C_\tau = \Sigma_\tau^{-1}$.

In other words, the approach requires Σ_τ to be invertible. Hence, using sample covariance matrix would need at least historical data of length $M \times \tau$, approximately. It is hardly ever the case for anyone to have such tremendous amount of information on hand. Moreover, even if the sample covariance matrix is invertible, it can still be the case that the covariance matrix is poorly estimated. The problem of *few samples and many variables* does not arise in only finance applications but also in, for example, data compression (principal components analysis) and Bayesian statistics. A considerable amount of effort from many researchers has thus been put to address this issue for over a century. Various methods have been proposed to ensure some (application-dependent) desirable properties, for example, invertibility, sparsity, and positive definiteness, of the estimated covariance matrices. This branch of research area is typically referred to as covariance matrix estimator in literature (see, for instance, DeMiguel, Martín-Utrera and Nogales [19] and Ledoit and Wolf [28]).

The main focus of this section is to provide a conceptual idea of how to incorporate in the pricing model multi-period returns information when the sample covariance matrix does not work nicely, especially when the available historical data is not sufficient. In such a situation, one can use a factor model assuming the existence of linear relationships between the random variables of his or her interest denoted by $\{rv_i\}_{i=1}^V$ and a set of so-called *factors* $\{f_i\}_{i=1}^k$:

$$rv_i = a_i + b_{i,1}f_1 + b_{i,2}f_2 + \ldots + b_{i,k}f_k + e_i, \tag{6.12}$$

where the factors can be chosen freely by problem modeller. $\{a_i\}_{i=1}^V$ and $\{b_i\}_{i=1}^V$ are constants. $\{e_i\}_{i=1}^V$ are errors or residual terms; e_i itself is a random variable and its expectation is supposed to be zero. The error terms are usually assumed to be independent and uncorrelated with every factor. Given that the aforementioned conditions are satisfied, assuming for simplicity that a single-factor model ($k = 1, f_1 = f$) is employed, we then have a new way to estimate mean and covariance matrix as the equation (6.12) suggests

$$\begin{aligned} \mathbb{E}(rv_i) &= a_i + b_i\mathbb{E}(f), \\ \sigma_i^2 &= b_i^2\sigma_f^2 + \sigma_{e_i}^2, \\ \sigma_{ij} &= b_ib_j\sigma_f^2, \qquad i \neq j, \end{aligned} \tag{6.13}$$

where σ_{ij} (σ_i^2) is the covariance between random variables rv_i and rv_j (variance of rv_i). σ_f^2 and $\sigma_{e_i}^2$ denote variances of the factor and the error respectively. As a result, factor model offers a significant savings of the parameters to be estimated in the covariance matrix. In general, the parameters a_i and b_i are chosen so that the sum of squared errors is minimised. This concept is the same as that of the well-known linear regression model, which outputs a straight line that best fits the input data set.

Especially in the basket option pricing model, the random variables are assets' τ-period cumulative returns. One possible definition of the factor f is a τ-period cumulative return of the basket itself, where the basket value at time τ is given by

$$\sum_{m=1}^{M} w_m \tilde{S}_\tau^m. \tag{6.14}$$

For more literature about factor model, we recommend to the reader Fan, Fan and Lv [22] and the references therein.

6.5 Conclusions

Pricing options with multiple underliers is more challenging than pricing those with single underlier. Although, the formulation of the robust pricing model for multiple-underlier options is not much different from the single-underlier option robust pricing model, we have to be careful of how to design the uncertainty set. One naive way is to consider the movement of each asset price individually. Another approach is to also impose a restriction on the movement of the first-period assets' prices using the sample covariance matrix of single-period returns. Both approaches seem to work quite fine, but they are prone to excessive conservatism. Ideally, it is desirable to impose restrictions on the movements of assets' prices in every period. Traditionally, one can use inverse of the covariance matrix of τ-period cumulative returns to do so. However, relying solely on the sample covariance matrix estimator is not practical as we would need a ridiculously large amount of data to ensure invertibility of the sample covariance matrix. Hence, we propose a way to estimate covariance matrices using a factor model instead. Using the duality in second-order cone programming, the corresponding deterministic optimisation problem of the multiple-underlier option robust pricing model is a second-order cone program, which again can be efficiently solved.

Chapter 7

Super/Sub Robust Replication

Our previous option pricing models output a portfolio whose final wealth matches most closely to the option payoff in the worst-case sense. Subsequently, we set the initial wealth of such a portfolio as the price of the option in order to minimise the worst-case arbitrage error. This seems to be a good method for fairly pricing a given option. There is, however, no guarantee whether the final wealth of the portfolio will be greater than or lower than the option payoff. This leads to the other two categories of the pricing models: super-replication pricing model and sub-replication pricing model (see, for example, Edirisinghe, Naik and Uppal [20] and Vayanos, Wiesemann and Kuhn [48]). The super-replication pricing model considers only super-replicating portfolios while the sub-replication pricing model considers only sub-replicating portfolios. The super-replicating and the sub-replicating portfolios are defined as follows.

- Super-replicating portfolio is a portfolio which we are certain that its wealth is always greater than or equal to the option payoff at the expiration date of the option regardless of the realised values of the underlying securities.
- Sub-replicating portfolio is a portfolio which we are certain that its wealth is always less than or equal to the option payoff at the expiration date of the option regardless of the realised values of the underlying securities.

From the perspective of the option writer, a set of super-replicating portfolios is deemed to be a safe choice to be used for pricing the option as a result of his obligation to fulfil the option contract by paying the amount of the option payoff to the option holder in the case that the option holder opts to exercise the option. Super-replicating portfolios create a boundary reassuring that the amount of money the option writer has to pay is no greater than this level, and therefore it is reasonable to price the option by determining the minimal today's price of such portfolios to avoid overpricing the option. Evidently, by the reverse argument, one can conclude that from the option holder's point of view the sub-replicating portfolios should be used to price the option.

7.1 Super-Replication Robust Pricing Model

Using the idea of super-replication, we formulate a super-replication robust pricing model for an option with arbitrary payoff function $P(\tilde{S}, K)$ below.

$$
\begin{aligned}
&\text{minimise}_{\{n_t^s\}_{t=0}^T, \{n_t^B\}_{t=0}^T, \{u_t\}_{t=1}^T, \{v_t\}_{t=1}^T} \; n_0^S S_0 + n_0^B B_0 \\
&\text{subject to} \\
&\quad W_T = n_T^S \tilde{S}_T + n_T^B B_T \\
&\quad n_t^S = n_{t-1}^S + u_t, \qquad \forall t = 1, 2, ..., T \\
&\quad n_t^B = n_{t-1}^B + v_t, \qquad \forall t = 1, 2, ..., T \\
&\quad u_t \tilde{S}_t + v_t B_t = 0, \qquad \forall t = 1, 2, ..., T \\
&\quad P(\tilde{S}, K) \le W_T, \qquad \forall \{\tilde{S}_t\}_{t=1}^T \in U
\end{aligned}
$$

(7.1)

As before, the decision variables n_t^S and n_t^B denote the number of the underlying assets and the number of the risk-free assets held in portfolio at time t, respectively. The initial price of the super-replicating portfolio, which we want to minimise, is given by $n_0^S S_0 + n_0^B B_0$ while the final wealth of the portfolio is denoted by W_T. Changes to be made to the portfolio at time $t = 1, 2, ..., T$ can be made through the decision variables u_t and v_t. The crucial part of this pricing model is in the last constraint where we limit the space of portfolios of our interest to those with final wealth greater than or equal to the option payoff with certainty, given that the movement of the asset price is described by the uncertainty set U.

The super-replication robust pricing model can be simplified by eliminating some of the decision variables by using the equality constraints. The resulting pricing model is presented below.

$$\text{minimise}_{n_0^S, n_0^B, \{u_t\}_{t=1}^{T-1}} \; n_0^S S_0 + n_0^B B_0$$

$$\text{subject to}$$

$$P(\tilde{S}, K) \leq \left(n_0^S + \sum_{t=1}^{T-1} u_t \right) \tilde{S}_T + \left(n_0^B - \sum_{t=1}^{T-1} \frac{u_t \tilde{S}_t}{B_t} \right) B_T, \quad \forall \{\tilde{S}_t\}_{t=1}^T \in U \tag{7.2}$$

7.2 Sub-Replication Robust Pricing Model

Similarly to the super-replication robust pricing model, a sub-replication robust pricing model can be formulated as

$$\begin{array}{ll} \multicolumn{2}{l}{\text{maximise}_{\{n_t^s\}_{t=0}^T, \{n_t^B\}_{t=0}^T, \{u_t\}_{t=1}^T, \{v_t\}_{t=1}^T} \; n_0^S S_0 + n_0^B B_0} \\ \multicolumn{2}{l}{\text{subject to}} \\ \quad W_T = n_T^S \tilde{S}_T + n_T^B B_T & \\ \quad n_t^S = n_{t-1}^S + u_t, & \forall t = 1, 2, ..., T \\ \quad n_t^B = n_{t-1}^B + v_t, & \forall t = 1, 2, ..., T \\ \quad u_t \tilde{S}_t + v_t B_t = 0, & \forall t = 1, 2, ..., T \\ \quad P(\tilde{S}, K) \geq W_T, & \forall \{\tilde{S}_t\}_{t=1}^T \in U, \end{array} \tag{7.3}$$

which again can be reduced to

$$\text{maximise}_{n_0^S, n_0^B, \{u_t\}_{t=1}^{T-1}} \; n_0^S S_0 + n_0^B B_0$$

$$\text{subject to}$$

$$P(\tilde{S}, K) \geq \left(n_0^S + \sum_{t=1}^{T-1} u_t \right) \tilde{S}_T + \left(n_0^B - \sum_{t=1}^{T-1} \frac{u_t \tilde{S}_t}{B_t} \right) B_T, \quad \forall \{\tilde{S}_t\}_{t=1}^T \in U. \tag{7.4}$$

Moreover, the super- and the sub-replication robust pricing models can also employ linear decision rules and piecewise linear decision rules using the method discussed in Chapter 5. In short, this chapter accentuates that there are different ways to formulate a robust pricing model. That the robust pricing model is adaptive and easy to construct is also a nice feature of this pricing approach, apart from its tractability and its flexibility.

Chapter 8

Numerical Results

The aim of this chapter is to provide an evaluation report of the previously discussed robust option pricing models. We divide this chapter into three major parts as follows.

- **Implementation details**: This section describes the details of the implementation of our robust pricing models. Tools and machine specifications are listed in this part.
- **Theoretical evaluation**: In this section, we explicitly assume that asset price follows the geometric Brownian motion. Based on the assumption, the Black-Scholes model is arguably the most reliable pricing model because the asset price is considered as a continuous random process, and a perfect hedge achieved by continually adjusting the replicating portfolio is obtainable in the case of European call option and lookback call option. We then evaluate the proposed robust pricing models by observing the closeness between the robust prices and the Black-Scholes prices. In the case of arithmetic Asian call option, there is no closed-form pricing formula, and thus we opt to use the approximate Black-Scholes model instead. American put option robust pricing model, basket call option robust pricing model, and super- and sub-replication robust pricing models are also evaluated in this section.
- **Comparison with market prices**: Comparison between the output prices from the proposed robust pricing models and the market prices is conducted. Three types of option are considered: American call option, American put option, and European index call option. We note here that for the American-style options, we consider in this section options tied to a non-dividend paying stock, and therefore the European call option pricing model can justifiably be used to price the American call options. The data used in this part was extracted from Yahoo! Finance and Google Finance on July 21, 2012.
- **Discussion of the results**: We end this chapter by providing an explanation for the figures obtained from the experiments.

8.1 Implementation Details

Every robust pricing model is implemented as a Matlab function. Optimisation problem solvers used in the experiment are LINPROG and SDPT3.

- **LINPROG** is a Matlab built-in function which is used to solve linear programs.
- **SDPT3** is an external Matlab software package for solving semidefinite programs. In our experiment, it is also used for solving second-order cone programs as well. Further information about this software can be reviewed from Toh, Todd and Tütüncü [44] and Toh, Tütüncü and Todd [45].

Apart from the solvers, our implementation relies on YALMIP, a Matlab toolbox offering a language for modelling optimisation problems. In simple cases, YALMIP itself is capable of formulating and solving the robust counterpart problems (see Löfberg [32]); however, we do not use this functionality as it is not very straightforward to do so with our pricing problems. For further reference regarding this topic, we recommend Löfberg [31].

All numerical experiments were conducted on a 2.30GHz, Intel Core i5-2410M CPU machine with 4GB of RAM.

8.2 Theoretical Evaluation

There are six parts in this section evaluating six different categories of the proposed pricing models. In this section, assets' prices are assumed to follow the geometric Brownian motion with parameters listed in Table 8.1, where the symbols S_0, μ, and σ denote current asset price, expected growth rate of the logarithmic price, and volatility of return of each asset, respectively.

Table 8.1: Asset parameters

	Asset I	Asset II
S_0	£100	£100
μ	12%	7%
σ	15%	45%

All options considered in this section have an expiration date one year from now. The one year's time is divided into twelve periods (T), each of which is one-month long (p). The annualised risk-free rate of return r_f is assumed to be fixed at 5%, and the strike price K varies between £60 and £140.

Table 8.2: Option parameters

	Option
p	1/12
T	12
r_f	5%
K	[£60, £140]

When considering options with single underlier, we use Asset I in Table 8.1 as a model asset. Asset II is used only when evaluating the basket option pricing model. In every experiment, all risk-aversion parameters are assigned to a single value c:

$$\Gamma = c, \tag{8.1}$$

$$\Gamma_t = c, \quad \forall t = 1, 2, ..., T, \tag{8.2}$$

$$\delta = c, \tag{8.3}$$

where the last one is used only when the basket option pricing model is evaluated.

8.2.1 European Call Option Pricing Model

We evaluate three European call option pricing models: the constant decision rule pricing model (CON-ECO), the linear decision rule pricing model (LIN-ECO), and the piecewise linear decision rule pricing model (PIE-ECO), assuming that the options from Table 8.2 are tied to the Asset I listed in Table 8.1. The results are presented in Table 8.3 and Table 8.4. In the sequel, the column name *Error* is short for minimum worst-case arbitrage error, i.e., the optimal value of the objective function ϵ. We also report the average time taken to price an option using each of the proposed pricing models in both tables. As an alternative, we also present the results graphically in Figure 8.1 and Figure 8.2.

Table 8.3: Evaluation of the European call option pricing models ($\Gamma = \Gamma_t = 1$)

K	**Black-Scholes Formula**	**CON-ECO**		**LIN-ECO**		**PIE-ECO**	
		Price	Error	Price	Error	Price	Error
60	42.926	42.920	0.000	42.920	0.000	42.920	0.000
70	33.426	33.407	0.000	33.407	0.000	33.407	0.000
80	24.078	23.894	0.000	23.894	0.000	23.894	0.000
90	15.467	14.380	0.000	14.380	0.000	14.380	0.000
100	8.592	5.727	1.349	5.727	1.349	5.727	1.349
110	4.076	0.938	4.006	0.938	4.006	0.938	4.006
120	1.660	0.000	4.015	0.000	4.015	0.000	4.015
130	0.590	0.000	0.488	0.000	0.488	0.000	0.488
140	0.186	0.000	0.000	0.000	0.000	0.000	0.000
	Average time taken (s)	0.0946		3.0682		5.0048	

Table 8.4: Evaluation of the European call option pricing models ($\Gamma = \Gamma_t = 2$)

K	**Black-Scholes Formula**	**CON-ECO**		**LIN-ECO**		**PIE-ECO**	
		Price	Error	Price	Error	Price	Error
60	42.926	42.920	0.000	42.920	0.000	42.920	0.000
70	33.426	33.407	0.000	33.407	0.000	33.407	0.000
80	24.078	23.894	0.000	23.894	0.000	23.894	0.000
90	15.467	15.814	2.931	15.814	2.931	15.814	2.931
100	8.592	9.656	6.261	9.656	6.261	9.656	6.261
110	4.076	4.883	8.134	4.883	8.134	4.883	8.134
120	1.660	1.495	8.550	1.495	8.550	1.495	8.550
130	0.590	0.000	7.675	0.000	7.675	0.000	7.675
140	0.186	0.000	5.191	0.000	5.191	0.000	5.191
	Average time taken (s)	0.1039		2.9569		4.5731	

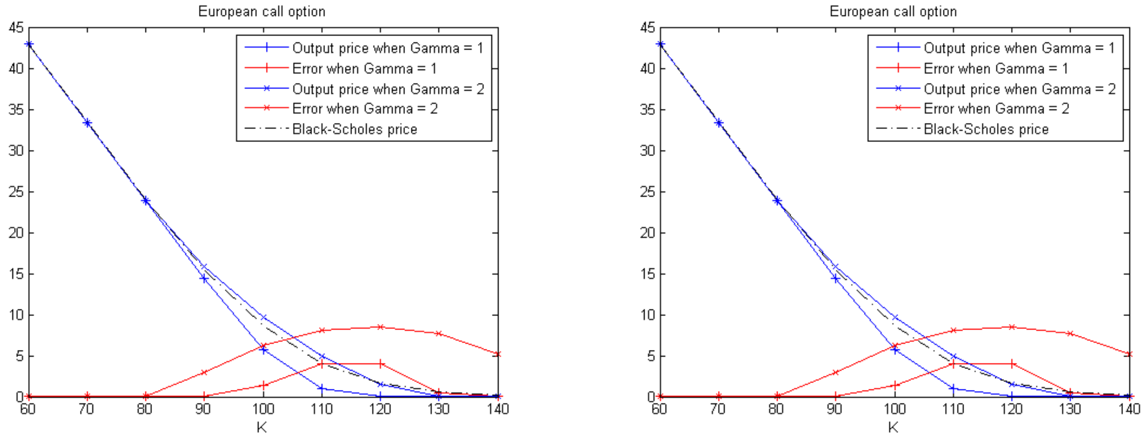

Figure 8.1: Comparison between CON-ECO(left)/LIN-ECO(right) and the Black-Scholes model

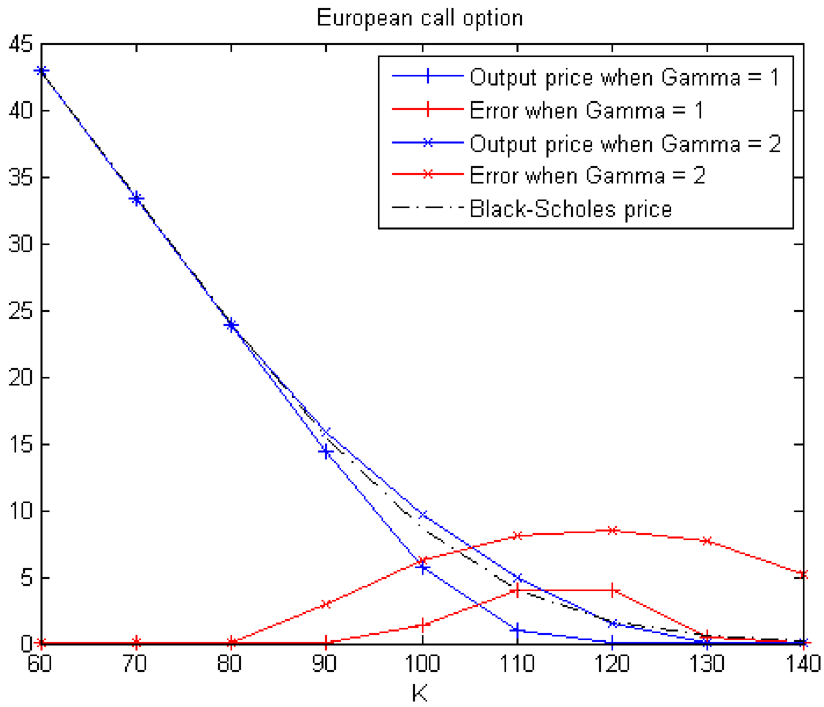

Figure 8.2: Comparison between PIE-ECO and the Black-Scholes model

8.2.2 Asian Call Option Pricing Model

The Asian call options of our interest are those with payoff functions defined using an arithmetic average function. Three pricing models are evaluated here including the constant decision rule pricing model (CON-ACO), the linear decision rule pricing model (LIN-ACO), and the piecewise linear decision rule pricing model (PIE-ACO). In the experiment, it is assumed that the options from Table 8.2 are tied to the Asset I listed in Table 8.1. The results are presented in Table 8.5 and Table 8.6 below. The numerical data is also displayed in Figure 8.3 and Figure 8.4.

Table 8.5: Evaluation of the Asian call option pricing models ($\Gamma = \Gamma_t = 1$)

K	Black-Scholes Formula[1]	CON-ACO		LIN-ACO		PIE-ACO	
		Price	Error	Price	Error	Price	Error
60	40.467	40.669	0.000	40.669	0.000	40.669	0.000
70	30.955	31.156	0.000	31.156	0.000	31.156	0.000
80	21.448	21.643	0.000	21.643	0.000	21.643	0.000
90	12.166	12.130	0.000	12.130	0.000	12.130	0.000
100	4.698	3.722	1.640	3.722	1.640	3.722	1.640
110	1.050	0.000	2.749	0.000	2.749	0.000	2.749
120	0.130	0.000	0.000	0.000	0.000	0.000	0.000
130	0.010	0.000	0.000	0.000	0.000	0.000	0.000
140	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Average time taken (s)	0.0608		2.9625		4.6281	

[1] To the best of our knowledge, unlike European call options, there is no analytical closed-form formula available for pricing Asian call options defined in terms of arithmetic averages even if the price of the underlying asset follows the geometric Brownian motion; however, there is an approximate formula suggested in Chapter 24 of Hull [26].

Table 8.6: Evaluation of the Asian call option pricing models ($\Gamma = \Gamma_t = 2$)

K	Black-Scholes Formula	CON-ACO		LIN-ACO		PIE-ACO	
		Price	Error	Price	Error	Price	Error
60	40.467	40.669	0.000	40.669	0.000	40.669	0.000
70	30.955	31.156	0.000	31.156	0.000	31.156	0.000
80	21.448	21.643	0.000	21.643	0.000	21.643	0.000
90	12.166	12.751	1.638	12.751	1.638	12.751	1.638
100	4.698	6.390	4.778	6.390	4.778	6.390	4.778
110	1.050	2.099	5.742	2.099	5.742	2.099	5.742
120	0.130	0.000	4.564	0.000	4.564	0.000	4.564
130	0.010	0.000	1.156	0.000	1.156	0.000	1.156
140	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Average time taken (s)	0.0828		3.0042		4.5493	

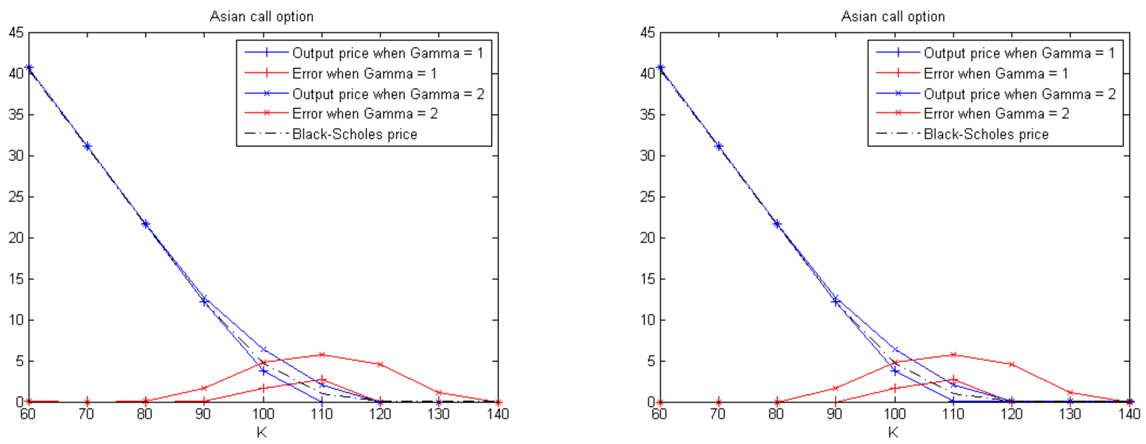

Figure 8.3: Comparison between CON-ACO(left)/LIN-ACO(right) and the Black-Scholes model

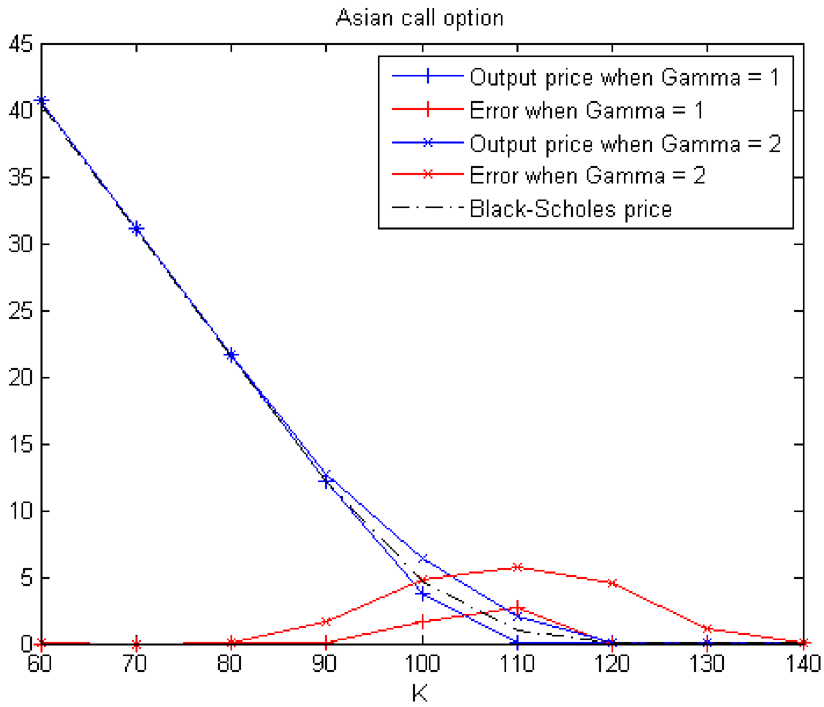

Figure 8.4: Comparison between PIE-ACO and the Black-Scholes model

8.2.3 Fixed Strike Lookback Call Option Pricing Model

We consider here only fixed strike lookback call options. Again, the pricing models implemented using constant decision rules (CON-LCO), linear decision rules (LIN-LCO), and piecewise linear decision rules (PIE-LCO) are evaluated. In the following, it is assumed that the options from Table 8.2 are tied to the Asset I listed in Table 8.1. The results from this experiment are shown in Table 8.7 and Table 8.8, which correspond to Figure 8.5 and Figure 8.6 respectively.

Table 8.7: Evaluation of the lookback call option pricing models ($\Gamma = \Gamma_t = 1$)

K	**Black-Scholes Formula**[2]	**CON-LCO**		**LIN-LCO**		**PIE-LCO**	
		Price	Error	Price	Error	Price	Error
60	52.937	48.370	6.707	48.370	6.707	48.370	6.707
70	43.425	38.857	6.707	38.857	6.707	38.857	6.707
80	33.912	29.343	6.707	29.343	6.707	29.343	6.707
90	24.400	19.830	6.707	19.830	6.707	19.830	6.711
100	14.888	10.317	6.707	10.317	6.707	10.317	6.707
110	7.233	4.037	6.539	4.037	6.539	3.969	6.602
120	2.999	0.000	4.015	0.000	4.015	0.000	4.017
130	1.079	0.000	0.488	0.001	0.489	0.096	0.554
140	0.344	0.000	0.000	0.000	0.000	0.000	0.000
	Average time taken (s)	13.2395		13.1471		30.2902	

[2] Similarly to the European call options, an exact formula for pricing lookback call options is available. It can be found in, for example, Chapter 24 of Hull [26].

Table 8.8: Evaluation of the lookback call option pricing models ($\Gamma = \Gamma_t = 2$)

K	**Black-Scholes Formula**	**CON-LCO**		**LIN-LCO**		**PIE-LCO**	
		Price	Error	Price	Error	Price	Error
60	52.937	65.332	28.903	56.051	15.084	56.051	15.085
70	43.425	46.538	15.084	46.538	15.084	46.538	15.084
80	33.912	37.025	15.084	37.025	15.084	37.025	15.085
90	24.400	27.511	15.084	27.511	15.084	27.511	15.084
100	14.888	17.998	15.084	17.998	15.084	17.998	15.084
110	7.233	9.899	14.459	9.899	14.459	9.899	14.459
120	2.999	6.325	11.712	6.325	11.712	6.179	12.047
130	1.079	1.257	8.287	1.256	8.287	1.239	8.288
140	0.344	0.000	5.191	0.000	5.191	0.000	5.194
	Average time taken (s)	13.9152		12.7267		29.2439	

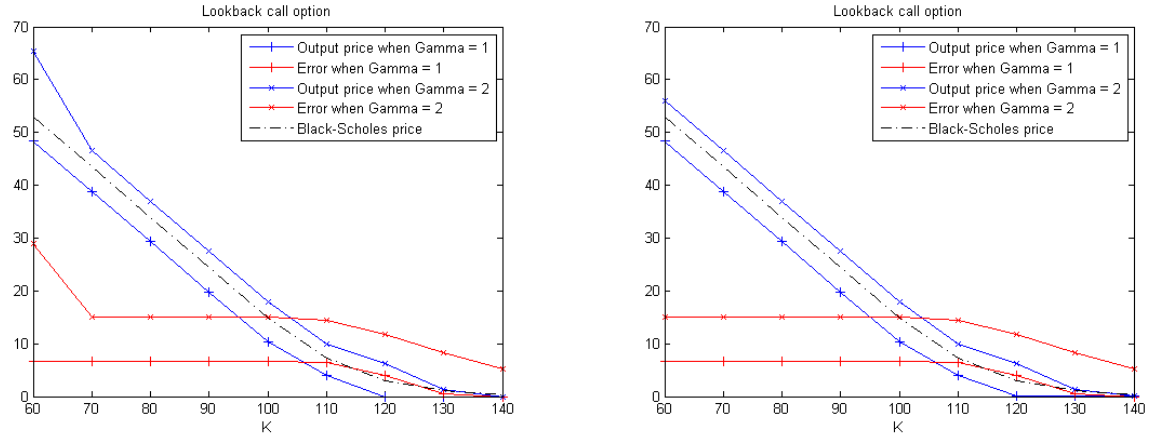

Figure 8.5: Comparison between CON-LCO(left)/LIN-LCO(right) and the Black-Scholes model

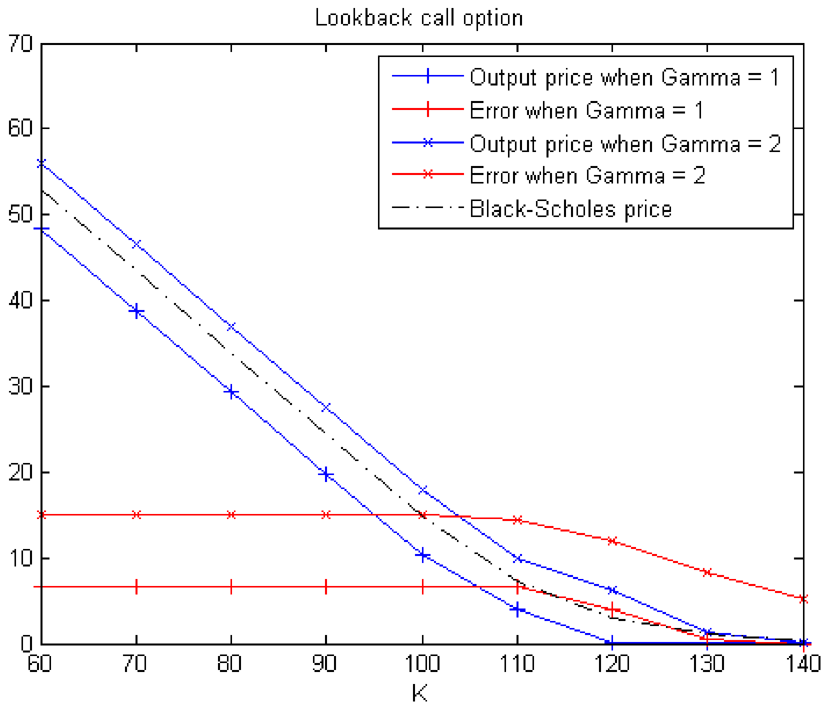

Figure 8.6: Comparison between PIE-LCO and the Black-Scholes model

8.2.4 American Put Option Pricing Model

We perform experiment to price American put options using our pricing models, which consist of the constant decision rule pricing model (CON-APO), the linear decision rule pricing model (LIN-APO), and the piecewise linear decision rule pricing model (PIE-APO). Assuming that the options from Table 8.2 are tied to the Asset I listed in Table 8.1, the obtained results are shown in Table 8.9 and Table 8.10. Alternatively, the reader can view the results from Figure 8.7 and Figure 8.8.

Table 8.9: Evaluation of the American put option pricing models ($\Gamma = \Gamma_t = 1$)

K	Binomial Model[3]	CON-APO		LIN-APO		PIE-APO	
		Price	Error	Price	Error	Price	Error
60	0.000	0.000	0.000	0.000	0.000	0.000	0.000
70	0.010	0.000	0.000	0.000	0.000	0.000	0.000
80	0.225	0.000	0.000	0.000	0.000	0.000	0.000
90	1.259	0.000	0.000	0.000	0.000	0.000	0.000
100	4.331	1.416	2.221	1.416	2.221	1.416	2.221
110	10.577	6.819	4.892	6.819	4.892	6.819	4.892
120	20.000	15.746	4.465	15.749	4.468	15.746	4.465
130	30.000	26.500	2.972	26.500	2.972	26.500	2.972
140	40.000	36.231	3.201	36.231	3.201	36.231	3.201
Average time taken (s)		1.6555		9.4339		17.8699	

[3] To the best of our knowledge, there is no tractable analytical formula for pricing American put options due to the flexibility the option holder has in exercising the option. We then compare the output prices from our models with the prices from the binomial options pricing model instead.

Table 8.10: Evaluation of the American put option pricing models ($\Gamma = \Gamma_t = 2$)

K	Binomial Model	CON-APO		LIN-APO		PIE-APO	
		Price	Error	Price	Error	Price	Error
60	0.000	0.000	0.000	0.000	0.000	0.000	0.000
70	0.010	0.000	0.000	0.000	0.000	0.000	0.000
80	0.225	0.000	0.000	0.000	0.000	0.000	0.000
90	1.259	1.433	2.931	1.442	2.951	1.433	2.931
100	4.331	4.788	6.261	4.788	6.261	4.788	6.261
110	10.577	9.567	8.166	9.567	8.166	9.567	8.166
120	20.000	15.971	8.723	15.971	8.723	15.971	8.723
130	30.000	24.245	7.861	24.245	7.861	24.245	7.861
140	40.000	34.823	5.447	34.824	5.447	34.824	5.447
Average time taken (s)		2.0532		9.3561		18.4796	

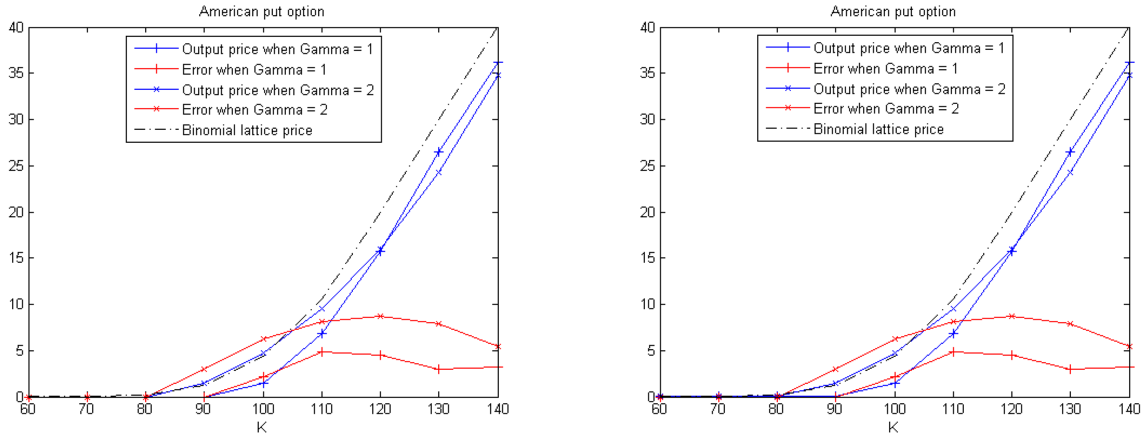

Figure 8.7: Comparison between CON-APO(left)/LIN-APO(right) and the binomial options pricing model

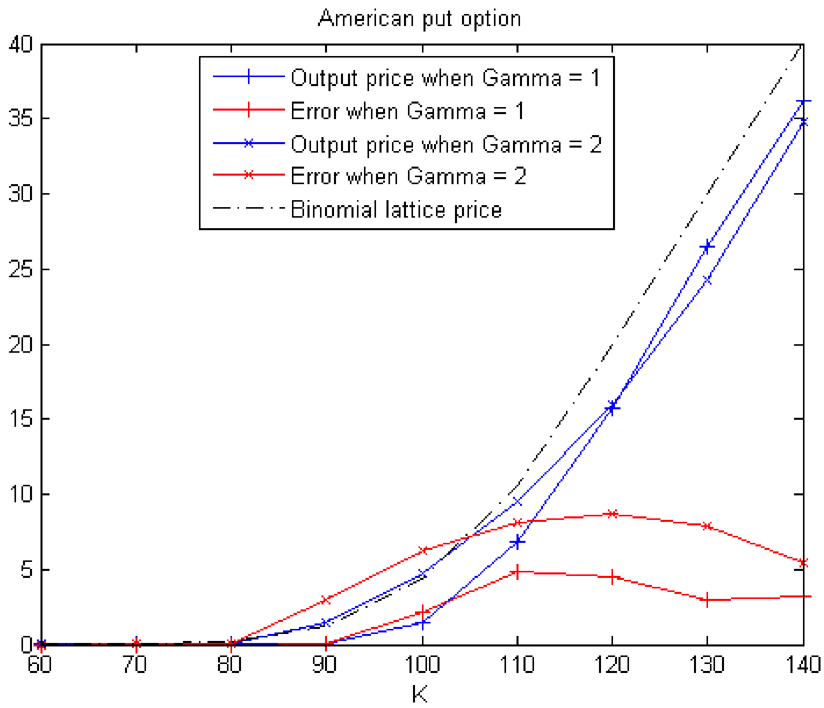

Figure 8.8: Comparison between PIE-APO and the binomial options pricing model

8.2.5 Basket Call Option Pricing Model

In order to demonstrate that the idea of the proposed robust option pricing model can be applied to options linked to multiple underliers as well as those tied to only one underlier, we develop a robust pricing model for basket call options using constant decision rule approach (CON-BKT) and then evaluate it. In the experiment, we consider equal weighted basket options with details listed in Table 8.2, and it is assumed that they are linked to two underlying assets: Asset I and Asset II from Table 8.1. We further assume that the only reliable source of information of the movements across assets' prices is a sample covariance matrix of single-period returns. Such information is then incorporated in the pricing model to prevent the pricing model from being overly conservative. It is assumed that the correlation between assets' single-period returns is $\rho = 0.5$. The results of the experiment are shown in Table 8.11 and Figure 8.9.

Table 8.11: Evaluation of the basket call option pricing model

K	**CON-BKT** ($\Gamma = \Gamma_t = \delta = 1$)		**CON-BKT** ($\Gamma = \Gamma_t = \delta = 2$)	
	Price	Error	Price	Error
60	42.920	0.000	42.920	0.000
70	33.407	0.000	34.843	3.074
80	23.894	0.000	28.103	7.282
90	15.902	3.246	22.021	10.798
100	9.706	6.409	16.598	13.621
110	4.933	8.077	11.834	15.753
120	1.582	8.250	7.728	17.191
130	0.000	7.031	4.281	17.937
140	0.000	4.219	1.493	17.992

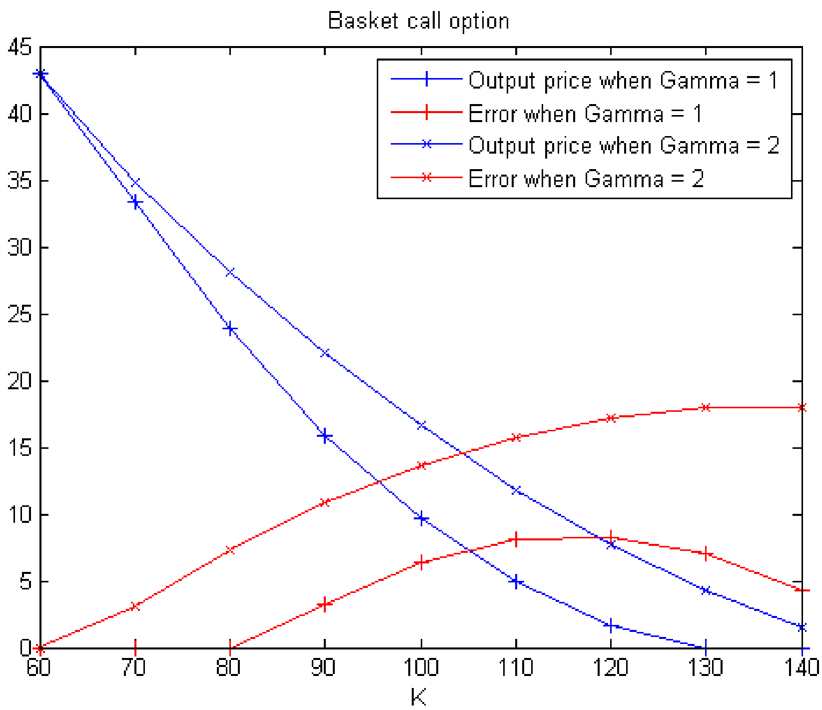

Figure 8.9: Basket call option pricing model

Alternatively, it is also possible to carry out the experiment on the basket call option robust pricing model in a different way by fixing the strike price K and running different values for the current assets' prices S_0 instead of fixing the current assets' prices while varying the strike price.

Figure 8.10 displays the output prices when we fix the strike price at £100, model the price of each of the underlying assets as a value from the range [£60, £140], and set all risk-aversion parameters to 0.5. Figure 8.11 presents the corresponding minimum worst-case arbitrage errors.

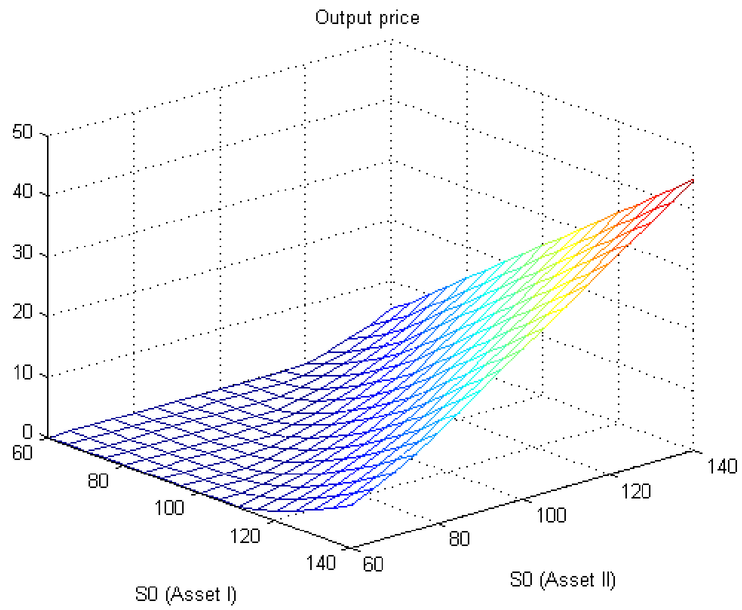

Figure 8.10: Output prices from the basket call option robust pricing model when strike price is fixed

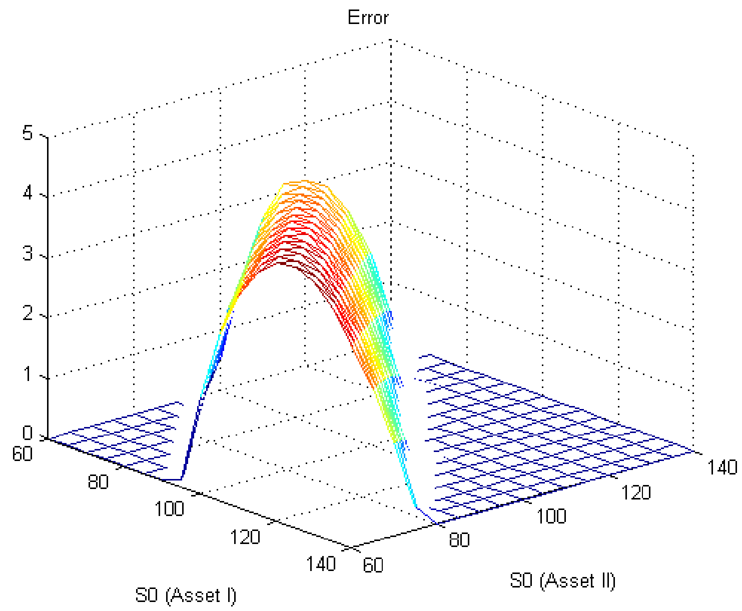

Figure 8.11: Minimum worst-case arbitrage errors from the basket call option robust pricing model when strike price is fixed

8.2.6 Super- and Sub-Replication Pricing Models

In this part, we implement the super-replication (SUP-ECO) and the sub-replication (SUB-ECO) robust pricing models for European call options. The model is implemented using piecewise linear decision rules expecting that it should output the portfolio matching most closely to the option payoff as compared with using constant decision rules and linear decision rules. As usual, we consider the options described by the parameters given in Table 8.2 which are assumed to be linked to the Asset I from Table 8.1. Results from the experiment are illustrated in Table 8.12 and can be viewed from Figure 8.12.

Table 8.12: Evaluation of the European call option super- and sub-replication robust pricing models

K	**Black-Scholes Formula**	**SUP-ECO**		**SUB-ECO**	
		$\Gamma = \Gamma_t = 1$	$\Gamma = \Gamma_t = 2$	$\Gamma = \Gamma_t = 1$	$\Gamma = \Gamma_t = 2$
60	42.926	42.920	42.920	42.920	42.920
70	33.426	33.407	33.407	33.407	33.407
80	24.078	23.894	23.894	23.894	23.894
90	15.467	14.380	18.602	14.380	14.381
100	8.592	7.010	15.611	4.867	4.867
110	4.076	4.749	12.621	0.000	0.000
120	1.660	2.487	9.630	0.000	0.000
130	0.590	0.225	6.639	0.000	0.000
140	0.186	0.000	3.648	0.000	0.000

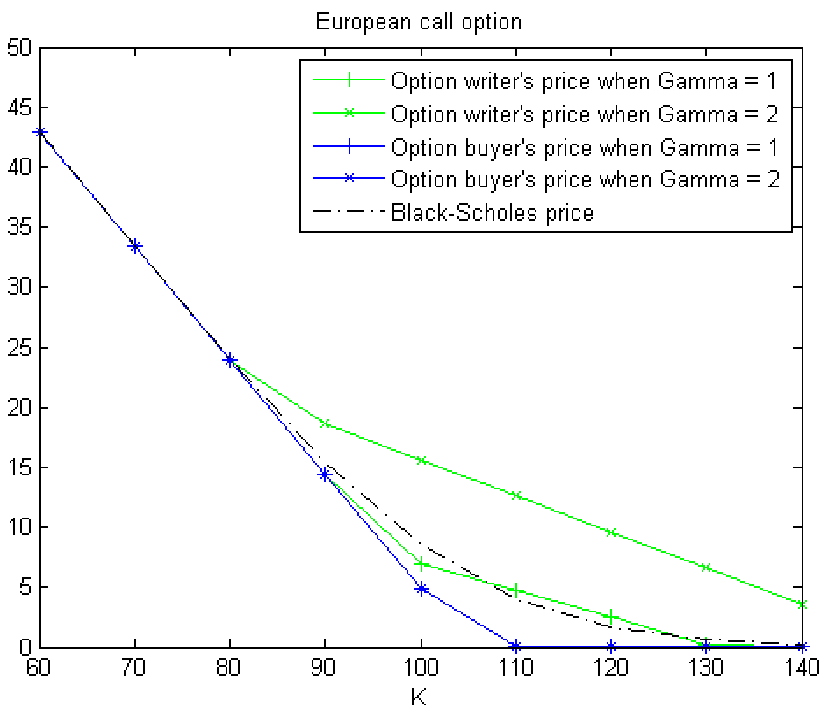

Figure 8.12: Comparison between SUP-ECO/SUB-ECO and the Black-Scholes model

We also use this part in order to show that the proposed robust pricing model is flexible enough to price various other options. For example, we can develop a robust pricing model to price a butterfly spread option whose payoff is given by

$$P = (S_T - 90)^+ - 2\,(S_T - 100)^+ + (S_T - 110)^+ . \tag{8.4}$$

The butterfly spread option pricing model is evaluated assuming that the underlying asset is the Asset I from Table 8.1 with current price S_0 ranging from £60 to £140 instead of a fixed value of £100. All risk-aversion parameters are set to 0.5. We present the experiment result in Figure 8.13.

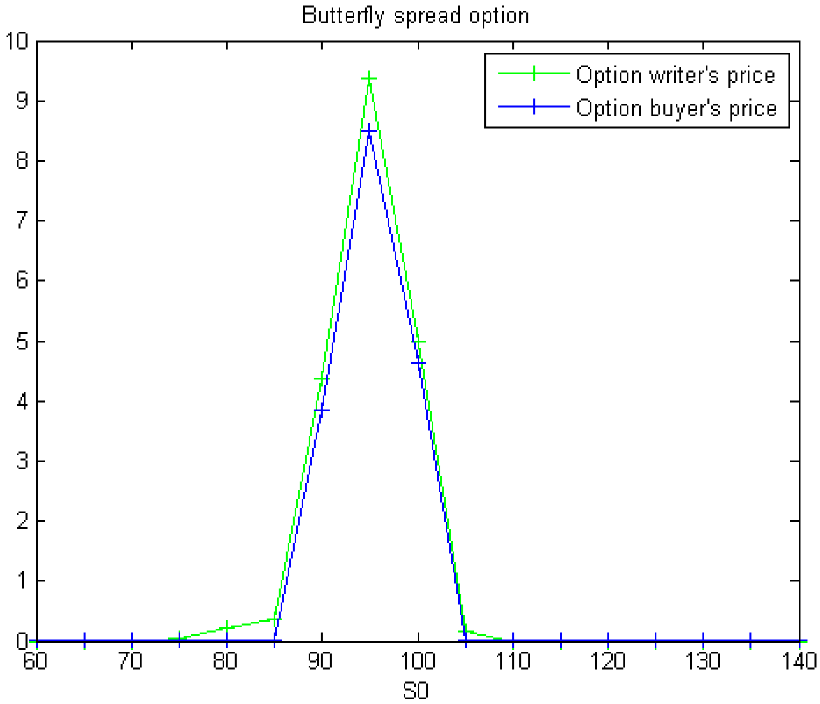

Figure 8.13: Butterfly spread option super- and sub-replication robust pricing models

8.3 Comparison with Market Prices

In this section, we present the numerical comparisons between the output prices of our pricing models and the observed market prices. We divide this section into two parts: comparison with market stock options and comparison with market index options. The data used in the experiment was collected from Yahoo! Finance and Google Finance on July 21, 2012. We note here that we use one-year LIBOR (London Interbank Offered Rate) rate as a reference risk-free rate in all experiments.

8.3.1 Comparison with Market Stock Options

We consider American-style options with RIMM (Research In Motion Limited) as an underlying stock. RIMM is a non-dividend paying stock, and therefore we can use the European call option pricing model to price the call options. American put options are of course priced by the American put option pricing model. The initial stock price is \$6.78. We use the pricing models developed based on the piecewise linear decision rules to evaluate the closeness between the robust pricing paradigm and the observed market prices. Evaluation of the options with strike prices between 4 and 12 which expire on October 20, 2012 (T = 13 weeks) is reported in Table 8.13 and Table 8.14. Note that we use one year historical data for parameter estimation.

Consider first a list of American call options linked to the stock RIMM and value them by using the piecewise linear decision rule pricing model (PIE-ECO). In Table 8.13, we present the obtained results. The underlined entries in Table 8.13 (and also Table 8.14) are the ones closest to the corresponding observed market prices.

Table 8.13: Comparison between the European call option pricing model and the observed market prices

K	**Market Price**	**PIE-ECO**				
		$\Gamma = \Gamma_t = 1.0$	$\Gamma = \Gamma_t = 1.5$	$\Gamma = \Gamma_t = 2.0$	$\Gamma = \Gamma_t = 2.5$	$\Gamma = \Gamma_t = 3.0$
4	3.10	2.527	2.570	2.707	2.874	<u>3.050</u>
5	2.02	1.267	1.573	1.845	<u>2.095</u>	2.327
6	1.29	0.337	0.793	<u>1.141</u>	<u>1.438</u>	1.702
7	0.80	0.000	0.230	0.595	<u>0.903</u>	1.174
8	0.48	0.000	0.000	0.208	<u>0.490</u>	0.744
9	0.27	0.000	0.000	0.000	<u>0.199</u>	0.411
10	0.17	0.000	0.000	0.000	0.030	<u>0.176</u>
11	0.11	0.000	0.000	0.000	0.000	<u>0.039</u>
12	0.09	0.000	0.000	0.000	0.000	<u>0.000</u>

For American put options, they can also be priced by the piecewise linear decision rule pricing model (PIE-APO). The results are given in Table 8.14.

Table 8.14: Comparison between the American put option pricing model and the observed market prices

K	**Market Price**	**PIE-APO**				
		$\Gamma = \Gamma_t = 1.0$	$\Gamma = \Gamma_t = 1.5$	$\Gamma = \Gamma_t = 2.0$	$\Gamma = \Gamma_t = 2.5$	$\Gamma = \Gamma_t = 3.0$
4	0.14	0.000	0.000	0.000	<u>0.084</u>	0.259
5	0.29	0.000	0.000	0.061	<u>0.302</u>	0.534
6	0.59	0.000	0.097	0.356	<u>0.642</u>	0.906
7	1.08	0.310	0.536	0.808	<u>1.104</u>	1.375
8	1.73	1.209	1.209	1.417	<u>1.689</u>	1.943
9	2.40	2.207	2.207	2.207	<u>2.395</u>	2.607
10	3.25	3.206	3.206	3.206	<u>3.224</u>	3.370
11	4.39	4.204	4.204	4.204	4.204	<u>4.229</u>
12	5.00	5.203	5.203	5.203	5.203	<u>5.190</u>

Alternatively, the numbers appearing in Table 8.13 and Table 8.14 are also presented graphically in Figure 8.14 and Figure 8.15 respectively.

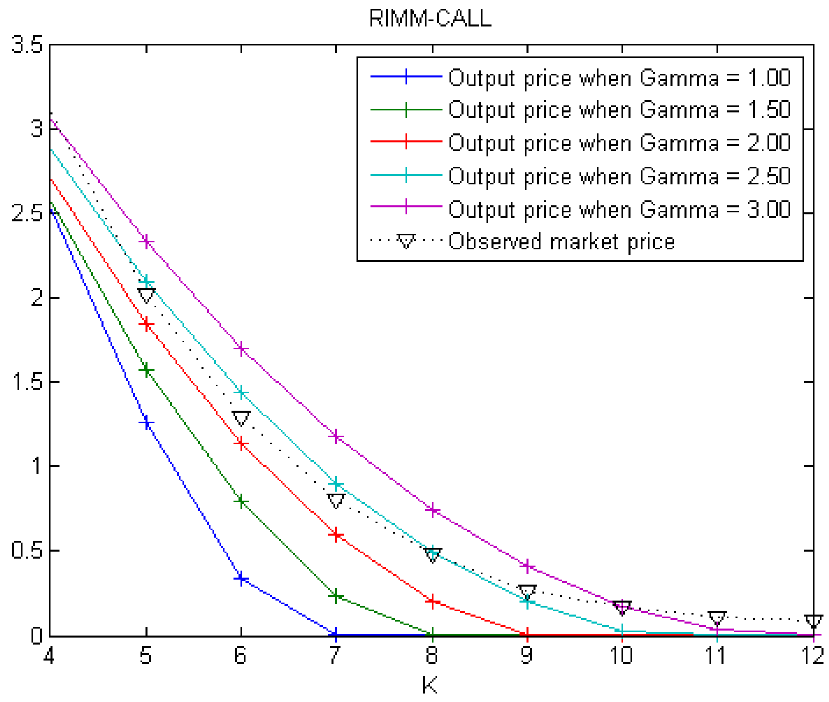

Figure 8.14: RIMM call options priced by PIE-ECO

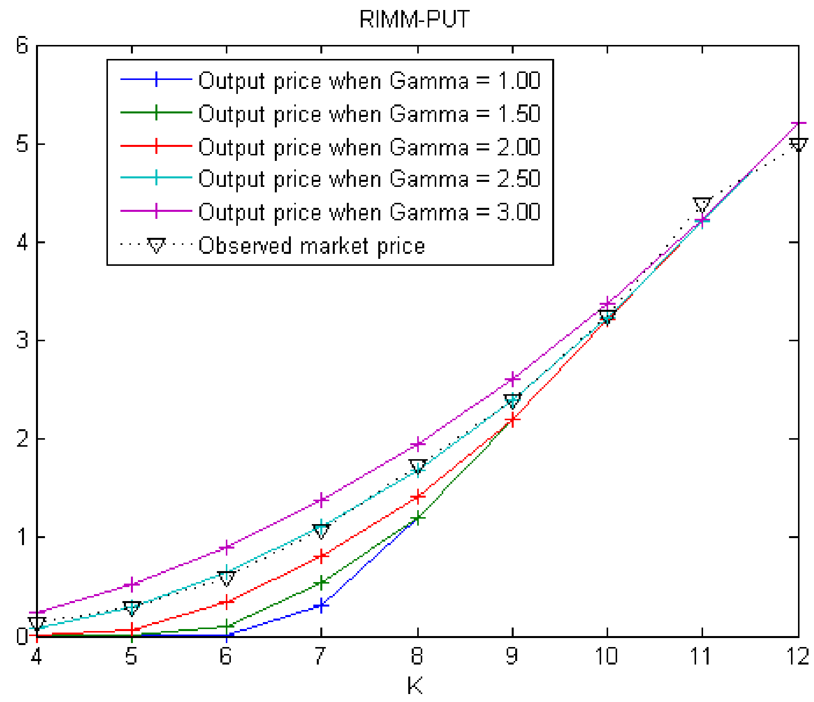

Figure 8.15: RIMM put options priced by PIE-APO

8.3.2 Comparison with Market Index Options

In this part, we consider index call options. Their payoffs are determined by an index called 1/100 Dow Jones Industrial Average (DJIA). The value of DJIA is a sum of the values of 30 large stocks divided by a common divisor called Dow Divisor, so it is an equal weighted index. Dow Divisor is periodically updated in order to keep the index value consistent after affected by certain events, for example, stock splits or dividend payouts. The current value of the 1/100 DJIA is \$128.23 as of the date of the experiment.

Two experiments are performed to compare the basket call option pricing models and the observed market prices. The first experiment relies on the use of the sample covariance matrix of stocks' single-period returns to impose a restriction on the first-stage prices of the index's components. As described before that using sample covariance matrices does not generally perform well when we want to impose other restrictions on the movement of stocks' prices at other stages apart from the first stage ($t = 1$), the second experiment is conducted by using a single-factor model for covariance matrix estimation rather than using a sample covariance matrix. The factor chosen to estimate a covariance matrix of τ-period cumulative returns is τ-period cumulative return of the index itself. The covariance matrix of τ-period cumulative returns is used to impose a restriction on the movement of stocks' prices at time τ. We denote by CON-BKT-SAMPLE the basket call option pricing model which takes as input the sample covariance matrix of single-period returns, and denote by CON-BKT-FACTOR the basket call option pricing model which takes as input the covariance matrices of τ-period cumulative returns, $1 \leq \tau \leq T$, estimated by the factor model.

Valuation of the options with strike prices between \$120 and \$140 that expire on December 22, 2012 (T = 22 weeks) performed using the first (CON-BKT-SAMPLE) and the second (CON-BKT-FACTOR) approaches is reported in Table 8.15 and Table 8.16, respectively. The underlined entries are again the ones closest to the corresponding observed market prices. We also present the experiment results in Figure 8.16 and Figure 8.17 as well. In this experiment, three year historical data is used for parameter estimation.

Table 8.15: Comparison between the basket call option pricing model (sample covariance matrix approach) and the observed market prices

K	Market Price	CON-BKT-SAMPLE				
		$\Gamma = \Gamma_t = 1.0$	$\Gamma = \Gamma_t = 1.5$	$\Gamma = \Gamma_t = 2.0$	$\Gamma = \Gamma_t = 2.5$	$\Gamma = \Gamma_t = 3.0$
120	11.400	8.950	<u>11.081</u>	13.224	15.359	17.474
122	9.000	7.545	<u>9.814</u>	12.026	14.203	16.346
124	6.780	<u>6.245</u>	8.616	10.880	13.088	15.252
126	5.830	<u>5.050</u>	7.488	9.786	12.014	14.192
128	4.700	<u>3.959</u>	6.430	8.744	10.981	13.166
130	3.650	<u>2.974</u>	5.441	7.753	9.990	12.174
132	2.650	<u>2.093</u>	4.521	6.814	9.039	11.215
134	2.000	<u>1.317</u>	3.672	5.928	8.130	10.291
136	1.710	<u>0.647</u>	2.892	5.092	7.261	9.400
138	1.170	0.081	<u>2.181</u>	4.309	6.434	8.543
140	0.600	<u>0.000</u>	1.540	3.578	5.648	7.720

Table 8.16: Comparison between the basket call option pricing model (factor model approach) and the observed market prices

K	**Market Price**	**CON-BKT-FACTOR**				
		$\Gamma = \Gamma_t = 1.0$	$\Gamma = \Gamma_t = 1.5$	$\Gamma = \Gamma_t = 2.0$	$\Gamma = \Gamma_t = 2.5$	$\Gamma = \Gamma_t = 3.0$
120	11.400	8.770	8.770	9.375	10.483	<u>11.563</u>
122	9.000	6.779	6.779	7.996	<u>9.177</u>	10.306
124	6.780	4.788	5.369	<u>6.715</u>	7.950	9.115
126	5.830	2.797	4.126	<u>5.532</u>	6.803	7.992
128	4.700	1.268	3.014	<u>4.448</u>	5.736	6.935
130	3.650	0.291	2.030	<u>3.462</u>	4.748	5.946
132	2.650	0.000	1.177	<u>2.574</u>	3.839	5.024
134	2.000	0.000	0.453	<u>1.784</u>	3.011	4.170
136	1.710	0.000	0.000	1.093	<u>2.262</u>	3.382
138	1.170	0.000	0.000	0.500	<u>1.592</u>	2.661
140	0.600	0.000	0.000	0.005	<u>1.002</u>	2.008

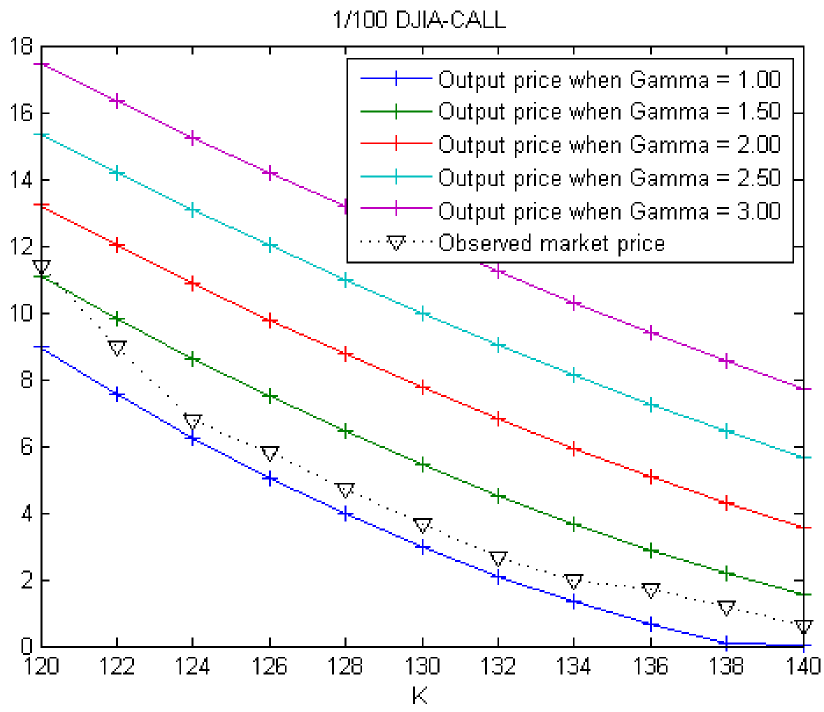

Figure 8.16: 1/100 DJIA call options priced by CON-BKT-SAMPLE

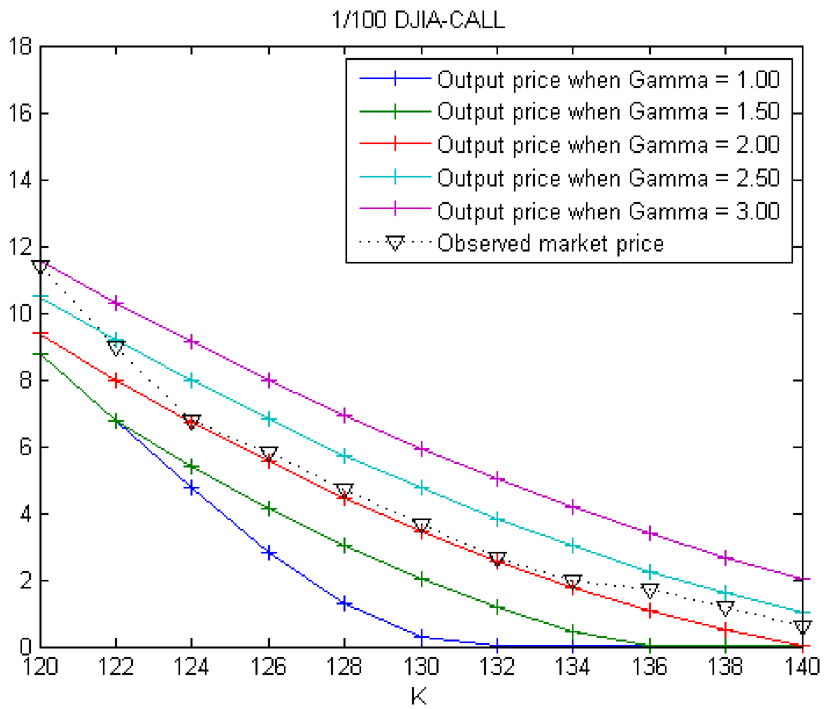

Figure 8.17: 1/100 DIJA call options priced by CON-BKT-FACTOR

8.4 Discussion of the Results

In this section, we analyse the figures obtained from our experiments. Despite being proved to be superior (or not inferior, to be exact) in terms of optimality to the constant decision rule pricing model, the piecewise linear decision rule pricing model and the linear decision rule pricing model exhibit no noticeable improvement as they appear to be unable to reduce the minimum worst-case arbitrage error ϵ^*, i.e., the difference between the final wealth level of the output portfolio and the option payoff. It is even seen in some cases that the constant decision rule pricing model outputs a slightly smaller ϵ^* as compared with the linear decision rule pricing model and the piecewise linear decision rule pricing model. This is possible because, as noted by Tütüncü, Toh and Todd [46], the SDPT3 solver has numerical difficulties solving some optimisation problems. We have also tried to reduce the value of ϵ^* by increasing the flexibility of the employed decision rules by using multiple-breakpoint piecewise linear decision rules and non-axial piecewise linear decision rules; however, the optimal objective value still remains roughly the same[4].

That the constant decision rule pricing model, the linear decision rule pricing model, and the piecewise linear decision rule pricing model seem to output similar replicating portfolios suggests that the robust pricing paradigm proposes a dynamic hedging strategy which is radically different from the classic delta-hedging strategy. This is because, using the robust pricing model, the knowledge of asset's prices in subsequent periods appears to have no significant effect to the portfolio adjustment, i.e., recourse possibilities are of no use, while the delta-hedging strategy heavily relies on the knowledge of such. Specifically, consider a European call option with single underlier and assume that its price follows the geometric Brownian motion. Delta-hedging strategy suggests continuously rebalancing a portfolio in such a way that the number of underliers held in portfolio at time t is updated constantly and given by

[4]Explanation of this part is not included in the thesis as it is merely an extension of the piecewise linear decision rule pricing model. We recommend Georghiou, Wiesemann and Kuhn [23] to the reader interested in this topic.

$$\Delta = \phi(d_1), \quad d_1 = \frac{log(S/K) + \left(r + \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{T-t}} \ , \tag{8.5}$$

where ϕ is the standard normal distribution. An explanation of each variable is provided in Section 2.2.2, where we review the Black-Scholes model, since continuous delta-hedging is a key concept to derive the Black-Scholes formula. For more details about delta-hedging, we recommend, for example, Higham [25] and Hull [26].

In the case of European call options, from the experiments we encountered, the portfolio suggested by the robust pricing paradigm behaved similar to static hedging strategy especially when the option considered was deep in the money or deep out of the money. In several cases, the model decided on the portfolio's components here and now and often made only negligible changes to the portfolio in subsequent periods. This can be explained by the following argument. Consider the final wealth of the portfolio which is given by

$$W_T = \left(n_0^S + \sum_{t=1}^{T-1} u_t\right)\tilde{S}_T + \left(n_0^B - \sum_{t=1}^{T-1}\frac{u_t\tilde{S}_t}{B_t}\right)B_T. \tag{8.6}$$

Rearranging terms in the equation, we have

$$W_T = n_0^B B_T + \left(\sum_{t=1}^{T-1}\frac{-u_t B_T}{B_t}\tilde{S}_t\right) + \left(n_0^S + \sum_{t=1}^{T-1} u_t\right)\tilde{S}_T. \tag{8.7}$$

A set of all possible values of the final portfolio wealth for a fixed path of asset price $\varsigma = \{\tilde{S}_t\}_{t=1}^T$ is thus given by

$$\varpi_\varsigma = \left\{ W \in \mathbb{R} \;\middle|\; W = n_0^B B_T + \left(\sum_{t=1}^{T-1}\frac{-u_t B_T}{B_t}\tilde{S}_t\right) + \left(n_0^S + \sum_{t=1}^{T-1} u_t\right)\tilde{S}_T, \right.$$
$$\left. n_0^S, n_0^B, u_1, u_2, \ldots, u_{T-1} \in \mathbb{R} \right\}. \tag{8.8}$$

Observe that the term W is a summation of $T+1$ terms, and there are $T+1$ degrees of freedom: n_0^S, n_0^B, and $\{u_t\}_{t=1}^{T-1}$. It is therefore possible to conclude that ϖ_ς is equivalent to

$$\varpi'_\varsigma = \left\{ W \in \mathbb{R} \;\middle|\; W = \pi_0 + \sum_{t=1}^{T}\pi_t\tilde{S}_t,\ \pi_0, \pi_1, \ldots, \pi_T \in \mathbb{R} \right\}. \tag{8.9}$$

Hence, an alternate way of interpreting the robust pricing model is to determine a set of optimal decision variables $\{\pi_t^*\}_{t=0}^T$ such that $W^* = \pi_0^* + \sum_{t=1}^{T}\pi_t^*\tilde{S}_t$ matches most closely to the option payoff in the worst-case sense.

Indeed, for a European call option, a best matching replicating portfolio in the worst-case sense is likely to have $\pi_t^* \approx 0$, $\forall t, 1 \le t \le T-1$, because its payoff, which is given by $(S_T - K)^+$, does not depend on $\{\tilde{S}_t\}_{t=1}^{T-1}$. Having non-zero π_t^*, $1 \le t \le T-1$, might just add an extra uncertain element contributing to the greater difference between the option payoff and the final portfolio wealth in some scenarios. In the case that portfolio adjustments are noticeable, they still usually occur in early stages, far from option expiry date, which probably results from a degree of uncertainty in the asset's price being enlarged as time progresses. The argument also seems to be supported by the Asian call option pricing model, where the output portfolio is evidently dynamically rebalanced, because payoffs of Asian call options depend on the whole price path $\{\tilde{S}_t\}_{t=1}^T$.

Furthermore, the equivalence between ϖ_ς and ϖ'_ς also explains why the European call option robust pricing model and the Asian call option robust pricing model perform well and produce the results similar to the Black-Scholes formula, especially when the option is currently deep in the money or deep out of the money. In the case that the option considered is deep out of the money, it may be deemed worthless with certainty, and the output portfolio of the robust pricing model is thus likely given by $\pi_0^* = \pi_1^* = \ldots = \pi_T^* = 0$. For a European call option that is deep in the money, the European call option robust pricing model has a tendency to output a portfolio given by $\pi_0^* = -K$, $\pi_1^* = \pi_2^* = \ldots = \pi_{T-1}^* = 0$, and $\pi_T^* = 1$. Similarly, for an Asian call option which is deep in the money, the Asian call option robust pricing model tends to output a portfolio given by $\pi_0^* = -K$ and $\pi_1^* = \pi_2^* = \ldots = \pi_T^* = 1/T$.

Unfortunately, a similar argument cannot be applied to a lookback call option whose payoff is determined by the optimal asset price and an American put option where its holder has a flexible exercising policy. The complexity of their payoffs makes the replicating portfolio more subtle. It is seen from the experiment results that, using the robust pricing framework, a perfect hedge cannot be obtained when an option of these two types is deep in the money. It can be further observed that the robust pricing model tends to underprice an American put option as compared with the binomial options pricing model. This is expected since the optimal exercising strategy is not considered in the robust pricing model as it solely tries to minimise arbitrage error in the worst-case scenario. Believing that the price of the underlying asset will rise (positive growth rate), Figure 8.8 presents a worth discussing result. When the strike price K is far below the current asset price S_0, i.e., the American put option considered is deep out of the money, setting risk-aversion parameters $(\Gamma, \{\Gamma_t\}_{t=1}^T)$ to one and setting them to two agree to price the option at around zero, i.e., the option is considered worthless. When around-the-money options are considered, the larger the risk-aversion parameters are, the higher the output price becomes. This phenomenon is due to the increased probability of the option eventually becoming profitable. However, if the option currently is (slightly) in the money, setting risk-aversion parameters to high values seems to decrease the output price, as the bigger the uncertainty set is, the more probable the option will become less profitable.

Result from the experiment on the super- and the sub-replication robust pricing models is in line with our intuition about the market. For a particular derivative, namely option contract, its price from the seller's (option writer's) perspective should be higher than that from the buyer's perspective. As confirmed by the example of the butterfly spread option, the robust pricing model can also be used to price a variety of options. In fact, the idea of minimising worst-case arbitrage error ϵ is simply denoted by a constraint

$$|P(\tilde{S}, K) - W_T| \leq \epsilon. \tag{8.10}$$

If this constraint can be equivalently replaced by a finite set of (uncertain) linear constraints, then the robust pricing model can be readily constructed for the pricing purpose. For European options and Asian options, the sizes of the linear programs corresponding to the robust pricing problem grow linearly with the number of time periods. For lookback options and American options, this number grows quadratically with the number of time periods. Such polynomial growth rates encourage the problem modeller to imitate continuous-time model via discretisation.

Although basket call option and European call option have a very similar structure. From our experiment, it seems like the robust pricing model had difficulty pricing an option with multiple underlying assets. When $\Gamma = \Gamma_t = 2$, in the case of European call option the highest value of ϵ^* from Table 8.4 is 8.550 while in the case of basket call option the highest ϵ^* from Table 8.11 is 17.992. We claim that this undesirable figure is a result of a well-known shortcoming of the robust optimisation, that is, the extreme conservatism. However, this problem is lessened after the introduction of a factor model which we use for covariance matrix estimation (see Table 8.15 and Table 8.16).

Let Γ^* be the value of the risk-aversion parameters that makes the model's output price, when $\Gamma = \Gamma_1 = \Gamma_2 = \ldots = \Gamma_T = \Gamma^*$, equal to the corresponding market price. From the comparison with market prices, it can be observed that Γ^* should be around 2 and 2.5. When the option considered has no intrinsic value and is deep out of the money, it may be valued by the proposed robust pricing model with such level of risk-aversion at zero; however, the market price is not exactly zero because it is still possible for the underlying asset's price to make a major move resulting in the option becoming lucrative. This situation may not be accounted by the uncertainty set associated with the pricing model. Γ^* for deep-out-of-the-money options should thus be unusually high. The same goes for deep-in-the-money options, which might be because of their being appealing to a lot of investors. Indeed, options which are deep in the money can help investors enhance their portfolios. Option trading strategies can be reviewed from many informative sources, for example, Lowell [33].

Additionally, from the comparison between the European call option robust pricing model and the market prices of the RIMM American call options, Γ^* of the European call option robust pricing model should approximately be around 2.3. Besides, from the comparison with the market prices of the RIMM American put options, Γ^* of the American put option robust pricing model is expected to be close to 2.5. That Γ^* of the American put option robust pricing model is higher than that of the European call option robust pricing model is anticipated since the American put option robust pricing model appears to be prone to underpricing. Determining the appropriate values of risk-aversion parameters is not a completely trivial task. Consequently, we recommend the reader to carry out an empirical experiment to determine Γ^* for a particular option before using our proposed robust pricing models in order to more reasonably adjust the risk-aversion parameters.

8.5 Conclusions

In this chapter, we evaluate the proposed robust pricing model for several types of option: European call option, Asian call option, fixed strike lookback call option, and American put option. Also, we show that the robust pricing model can be used to price other options, such as butterfly spread option. The output price of the European call option robust pricing model and that of the Asian call option robust pricing model are fairly close to the corresponding Black-Scholes prices, which confirms the reliability of the proposed robust pricing model. Moreover, by adjusting the model's risk-aversion parameters appropriately, the output price of the robust pricing model coincides with the observed market price. Such values of the risk-aversion parameters should be used as a bottom line for the future employment of the robust pricing model. For options with multiple underliers, we suggest using the factor model to remedy the conservatism issue of the robust optimisation. This method produces a very nice result as confirmed by our experiment. Last but not least, we show that there is not a unique way to formulate a pricing model as an application of the robust optimisation. Specifically in this thesis, we propose three ways to formulate the pricing problem: the minimum-arbitrage robust pricing model, the super-replication robust pricing model, and the sub-replication robust pricing model. All of them are evaluated; tabular and graphical evaluation results are reported in this chapter.

Chapter 9

Conclusions

We started this work by reviewing theories and applications of both deterministic optimisation and optimisation under uncertainty. Many classes of deterministic convex optimisation problems can be efficiently solved by, for example, interior point method; however, they are not practical for a number of real world applications as the deterministic optimisation is not specially designed for tackling uncertain factors that can arise in the optimisation problems. There are two main competing approaches, namely the stochastic programming approach and the robust optimisation approach, to tackle the optimisation problems which are subject to uncertainty. Each of them has its own advantages and disadvantages. In the context of option pricing, Chen [15] proposed a robust optimisation model that can be used to price several options, for example, European call options and Asian call options. The concept behind his model is that he translated the arbitrage-free assumption to an arbitrage opportunity minimisation problem. One nice feature of Chen's model is that it requires fewer assumptions about the market as compared with other famous approaches, for example, the Black-Scholes model. The model also inherits the tractability of the robust linear optimisation. Chen showed that typically the size of the deterministic linear program for option pricing grows polynomially with the number of time periods, which is considerably slower than the binomial options pricing model, which suffers from the exponential growth rate.

In the option pricing problem, the uncertain parameters are the future prices of the underlying asset(s). In the model that Chen proposed, the uncertainty set describing these uncertain parameters is designed as a polyhedron. This choice of uncertainty set enabled him to derive a deterministic equivalent of a particular robust pricing problem using duality in linear programming. We identified that there are unnecessary steps in the derivation of Chen's model. The correction we made leads to a new formulation of the robust pricing problem, which welcomes further analysis and improvement.

Motivated by the success of linear decision rules and piecewise linear decision rules illustrated in Georghiou, Wiesemann and Kuhn [23] and Georghiou, Wiesemann and Kuhn [24], we employ these two types of decision rules in the proposed robust pricing model. Although to the best of our knowledge, to date, there is no equivalent deterministic formulation of the robust pricing model whose decision variables are modelled by the linear decision rules or the piecewise linear decision rules, we successfully derive the approximate deterministic version, which is a semidefinite program, of the robust counterpart. We also establish theories reassuring that, after employing the linear decision rules or the piecewise linear decision rules, the arising robust pricing model performs at least as good as the original robust pricing model with no embedded decision rules, i.e., the one with every decision variable modelled by a constant decision rule, in terms of the optimality of the generated solution. The experiment however detected no noticeable improvement, which made us claim that the robust pricing paradigm suggests a portfolio rebalancing strategy differently from the classic delta-hedging model.

In the case that the option considered is tied to more than one underlier, we encountered a situation where the robust pricing model was overly conservative. This is expected since in reality it is unlikely for the price of each individual asset to move freely with no regards to the other assets. However, it is also not trivial to impose restrictions on the movements of assets' prices as a result of a sample covariance matrix becoming not invertible when the number of underliers is large and the size of available historical data is small. We then propose a way to lessen such excessive conservatism by using another covariance matrix estimator, namely the factor model. The experiment result is indeed satisfactory. We also elaborate on the development of robust pricing model from option writer's perspective and option holder's perspective as it is usually the case that an option is valued differently by the seller and the buyer.

9.1 Advantages of the Robust Pricing Model

We believe that the main advantages of the robust pricing model are its flexibility and its tractability. Tractability of the robust pricing model is determined by solvability of deterministic linear programs, deterministic second-order cone programs, and deterministic semidefinite programs. The robust pricing model therefore offers a high degree of scalability. Consequently, it can very well be used to price an option where the price of its underlying security constantly fluctuates. It is observed that when the price of the underlying security obeys the geometric Brownian motion, the output price of the robust pricing model is fairly close to the Black-Scholes formula. This result confirms the reliability of the robust pricing model.

Besides, the robust pricing model, from our point of view, is very flexible. Problem modeller can amend the robust pricing model to simulate other market conditions, for example, transaction costs and short-selling prohibition. In order to confirm the model's flexibility, below we show how one can model the transaction costs and include them in the proposed pricing model.

9.1.1 Transaction Costs

The proposed robust pricing model can be generalised to take into account transaction costs as a result of portfolio adjustments. Consider the original self-financing constraint in the robust pricing model (see the model (5.5))

$$u_t\tilde{S}_t + v_tB_t = 0, \tag{9.1}$$

which can be modified as

$$\begin{gathered} \left(u_t^+ - u_t^-\right)\tilde{S}_t + v_tB_t = 0 \\ u_t^+, u_t^- \geq 0 \end{gathered} \tag{9.2}$$

so as to differentiate buying activities (u_t^+) from selling activities (u_t^-). In the latter form, linear transaction costs can be included in the robust pricing model by replacing the original self-financing constraint with

$$\begin{gathered} \left(u_t^+ - u_t^-\right)\tilde{S}_t + v_tB_t + c^{S+}u_t^+\tilde{S}_t + c^{S-}u_t^-\tilde{S}_t = 0 \\ u_t^+, u_t^- \geq 0, \end{gathered} \tag{9.3}$$

where c^{S+} and c^{S-} are the transaction cost rates associated with buying and selling the underlying asset respectively.

We repeat the experiment on the Asian call options in Chapter 8 with the inclusion of the transaction costs. In Figure 9.1, we present the result of this experiment.

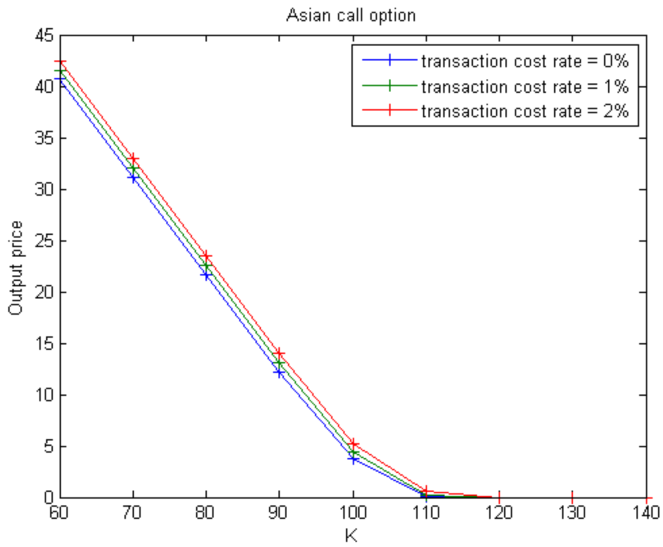

Figure 9.1: Option pricing in the presence of transaction costs

9.1.2 Pricing Other Exotic Options

The robust pricing model can also be used to price more exotic types of option, which again confirms the flexibility of the robust pricing model. Two case studies are presented in this part. In the first case study, we modify the European call option robust pricing model to price binary options. We assume that a binary option of our interest pays a fixed amount of £1 if the price of the underlying asset ends up above the strike price. The result of this study is presented in Figure 9.2.

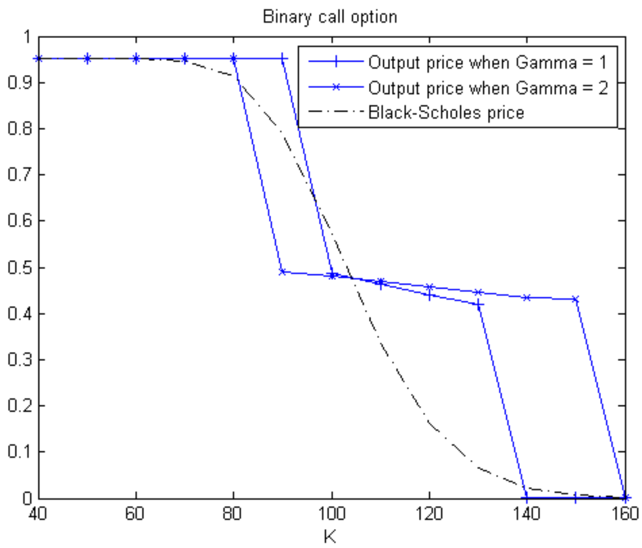

Figure 9.2: Valuation of binary options

In the second case study, we develop a robust pricing model for a certain type of barrier option, namely down and out call option. A down and out call option is a regular call option which automatically becomes worthless if the asset price reaches a barrier level at any time during the option's lifetime. This type of contract is interesting because its payoff is similar to that of the corresponding vanilla option and it is cheaper. The comparison between the Black-Scholes model and the robust option pricing model for down and out call options is illustrated in Figure 9.3.

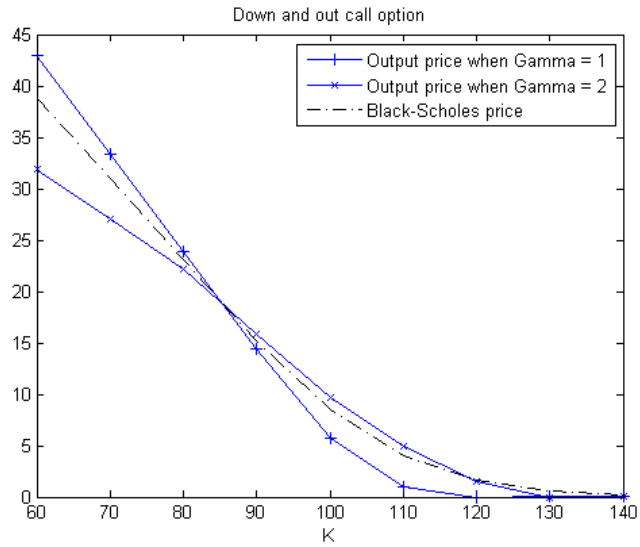

Figure 9.3: Valuation of down and out call options

9.2 Future Research

We note in this section potential extensions that could be made to the robust pricing model.

- **Probabilistic interpretation:** One disadvantage of the robust pricing model that we foresee is that it is not easy to reasonably assign values to the risk-aversion parameters. Indeed, we know that the larger such values are, the more risk-averse the investor is. However, the meaning of the magnitudes of these parameters is still ambiguous. Lately, there has been research focusing on finding probabilistic interpretation of the uncertainty set of a specific size and shape specified in the robust optimisation problem (see, for example, El Ghaoui, Oks and Oustry [21], Zymler, Kuhn and Rustem [51] and Zymler, Kuhn and Rustem [52]). It would be desirable if this idea can also be applied to the robust pricing model.

- **Choices of objective function:** We describe in this work that there are several ways to define the objective function in the robust pricing model. For example, the super-replication and the sub-replication robust pricing models employ different forms of the objective function, and both of them are also different from the original objective function, which is an arbitrage error ϵ. There are several ways to define the measure of closeness between the final wealth of the portfolio and the option payoff. Different measures lead to different robust counterparts. Under a given setting, some of them may be more appropriate than others.

- **Choices of uncertainty set:** Chen used a polyhedral uncertainty set in his robust pricing model, and we also use the same type of uncertainty set in the proposed robust pricing model. That Chen decided to use a polyhedron was probably because he wanted to use the duality in linear programming to derive the linear deterministic equivalent of the robust pricing problem. We however use several other techniques, for instance, the approximate S-lemma and the duality in second-order cone programming, to determine the deterministic version of the robust counterpart. Consequently, it is possible to define the uncertainty set differently.

- **Reduction of arbitrage:** After having introduced a couple of the decision rules to the robust pricing model, the minimum arbitrage error ϵ^* still remains roughly the same. It would of course be preferable to reduce the arbitrage error as much as possible. One possibility that might help is to introduce new forms of decision rules. If this could be done, the robust pricing model would be considered less conservative and more reliable.

Bibliography

[1] Bampou, D. and Kuhn, D. (2011): Scenario-free stochastic programming with polynomial decision rules. *In Proceedings of IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Florida.

[2] Bandi, C. and Bertsimas, D. (2012): Tractable stochastic analysis in high dimensions via robust optimization. *Mathematical Programming*, 134, 23 – 70.

[3] Ben-Tal, A., El Ghaoui, L. and Nemirovski, A. (2009): *Robust Optimization.* Princeton University Press, New Jersey.

[4] Ben-Tal, A., Goryashko, A., Guslitzer, E. and Nemirovski, A. (2004): Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99, 351 – 376.

[5] Ben-Tal, A., Margalit, T. and Nemirovski, A. (2000): Robust modeling of multi-stage portfolio problems. *In Frenk, H., Roos, K., Terlaky, T. and Zhang, S., Eds. High-Performance Optimization*, Kluwer Academic Publishers, Dordrecht, 303 – 328.

[6] Ben-Tal, A. and Nemirovski, A. (1999): Robust solutions of uncertain linear programs. *Operations Research Letters*, 25, 1 – 13.

[7] Bertsimas, D., Kogan, L. and Lo, A. (2001): Hedging derivative securities and incomplete markets: an $\epsilon-$arbitrage approach. *Operations Research*, 49, 372 – 397.

[8] Bertsimas, D. and Pachamanova, D. (2008): Robust multiperiod portfolio management in the presence of transaction costs. *Computers and Operations Research*, 35, 3 – 17.

[9] Bertsimas, D., Pachamanova, D. and Sim, M. (2004): Robust linear optimization under general norms. *Operations Research Letters*, 32, 510 – 516.

[10] Bertsimas, D. and Sim, M. (2004): The price of robustness. *Operations Research*, 52, 35 – 53.

[11] Birge, J. and Louveaux, F. (1997): *Introduction to Stochastic Programming.* Springer, New York.

[12] Boyd, S. and Vandenberghe, L. (2004): *Convex Optimization.* Cambridge University Press, Cambridge.

[13] Boyle, P. (1977): Options: a Monte Carlo approach. *Journal of Financial Economics*, 4, 323 – 338.

[14] Bradley, S., Hax, A. and Magnanti, T. (1977): *Applied Mathematical Programming.* Addison-Wesley, Massachusetts.

[15] Chen, S. (2008): Robust option pricing: an ϵ-arbitrage approach. Master's thesis, Massachusetts Institute of Technology.

[16] Chen, X., Sim, M., Sun, P. and Zhang, J. (2008): A linear decision-based approximation approach to stochastic programming. *Operations Research*, 56, 344 – 357.

[17] Dantzig, G., Orden, A. and Wolfe, P. (1955): The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pacific Journal of Mathematics*, 5, 183 – 195.

[18] Dantzig, G. and Thapa, M. (1997): *Linear Programming 1: Introduction*. Springer, New York.

[19] DeMiguel, V., Martín-Utrera, A. and Nogales, F. (2011): Size matters: optimal calibration of shrinkage estimators for portfolio selection. *E-print available at: http://ssrn.com/*.

[20] Edirisinghe, C., Naik, V. and Uppal, R. (1993): Optimal replication of options with transaction costs and trading restrictions. *Journal of Financial and Quantitative Analysis*, 28, 117 – 138.

[21] El Ghaoui, L., Oks, M. and Oustry, F. (2003): Worst-case value-at-risk and robust portfolio optimization: a conic programming approach. *Operations Research*, 51, 543 – 556.

[22] Fan, J., Fan, Y. and Lv, J. (2008): High dimensional covariance matrix estimation using a factor model. *Journal of Econometrics*, 147, 186 – 197.

[23] Georghiou, A., Wiesemann, W. and Kuhn, D. (2010): Generalized decision rule approximations for stochastic programming via liftings. *E-print available at: http://www.optimization-online.org*.

[24] Georghiou, A., Wiesemann, W. and Kuhn, D. (2011): The decision rule approach to optimisation under uncertainty: methodology and applications in operations management. *E-print available at: http://www.optimization-online.org*.

[25] Higham, D. (2004): *An Introduction to Financial Option Valuation: Mathematics, Stochastics and Computation*. Cambridge University Press, Cambridge.

[26] Hull, J. (2009): *Options, Futures and Other Derivatives*, 7th ed. Pearson Prentice Hall, New Jersey.

[27] Kuhn, D., Wiesemann, W. and Georghiou, A. (2011): Primal and dual linear decision rules in stochastic and robust optimization. *Mathematical Programming*, 130, 177 – 209.

[28] Ledoit, O. and Wolf, M. (2003): Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, 10, 603 – 621.

[29] Li, z. and Floudas, C. (2012): Robust counterpart optimization: uncertainty sets, formulations and probabilistic guarantees. *In Proceedings of the Foundations of Computer-Aided Process Operations (FOCAPO) Conference*, Georgia.

[30] Lobo, M., Vandenberghe, L., Boyd, S. and Lebret, H. (1998): Applications of second-order cone programming. *Linear Algebra and its Applications*, 284, 193 – 228.

[31] Löfberg, J. (2004): YALMIP: a toolbox for modeling and optimization in MATLAB. *In Proceedings of IEEE International Symposium on Computer-Aided Control Systems Design (CACSD)*, Taipei.

[32] Löfberg, J. (2012): Automatic robust convex programming. *Optimization Methods and Software*, 27, 115 – 129.

[33] Lowell, L. (2009): *Get Rich with Options: Four Winning Strategies Straight from the Exchange Floor*, 2nd ed. John Wiley, New Jersey.

[34] Luenberger, D. (1998): *Investment Science*. Oxford University Press, Oxford.

[35] Mehrotra, S. (1992): On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2, 575 – 601.

[36] Musiela, M. and Rutkowski, M. (2005): *Martingale Methods in Financial Modelling*, 2nd ed. Springer, Berlin.

[37] Pachamanova, D. (2002): A robust optimization approach to finance. Ph.D. thesis, Massachusetts Institute of Technology.

[38] Pólik, I. and Terlaky, T. (2007): A survey of the S-lemma. *SIAM Review*, 49, 371 – 418.

[39] Ruszczyński, A. and Shapiro, A., Eds. (2003): *Stochastic Programming.* Elsevier, Amsterdam.

[40] Shah, A. (1997): Black, Merton and Scholes: their work and its consequences. *Economic and Political Weekly*, 32, 3337 – 3342.

[41] Shapiro, A., Dentcheva, D. and Ruszczyński, A. (2009): *Lectures on Stochastic Programming: Modeling and Theory.* SIAM, Philadelphia.

[42] Shapiro, A. and Nemirovski, A. (2005): On complexity of stochastic programming problems. *Applied Optimization*, 99, 111 – 146.

[43] Shapiro, A. and Philpott, A. (2007): A tutorial on stochastic programming. *E-print available at: http://stoprog.org.*

[44] Toh, K., Todd, M. and Tütüncü, R. (1999): SDPT3 — a Matlab software package for semidefinite programming, version 2.1. *Optimization Methods and Software*, 11, 545 – 581.

[45] Toh, K., Tütüncü, R. and Todd, M. (2010): On the implementation and usage of SDPT3 — a Matlab software package for semidefinite-quadratic-linear programming, version 4.0. *E-print available at: http://www.optimization-online.org.*

[46] Tütüncü, R., Toh, K. and Todd, M. (2003): Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95, 189 – 217.

[47] Uğur, Ö. (2009): *An Introduction to Computational Finance.* Imperial College Press, London.

[48] Vayanos, P., Wiesemann, W. and Kuhn, D. (2011): Hedging electricity swing options in incomplete markets. *In Proceedings of the* 18^{th} *International Federation of Automatic Control (IFAC) World Congress*, Milan.

[49] Wiesemann, W., Kuhn, D. and Rustem, B. (2012): Robust Markov decision processes. To appear in *Mathematics of Operations Research.*

[50] Wong, H. and Kwok, Y. (2003): Sub-replication and replenishing premium: efficient pricing of multi-state lookbacks. *Review of Derivatives Research*, 6, 83 – 106.

[51] Zymler, S., Kuhn, D. and Rustem, B. (2011): Distributionally robust joint chance constraints with second-order moment information. To appear in *Mathematical Programming.*

[52] Zymler, S., Kuhn, D. and Rustem, B. (2012): Worst-case value-at-risk of non-linear portfolios. To appear in *Management Science.*

Appendix

A.1 Prerequisite Software

In order to use our proposed robust option pricing models, the following three components must already be installed on your machine.

- **Matlab**: Matlab is a programming environment specially developed for scientific purposes. It has a variety of built-in functions which can be used in a wide range of applications. The version of Matlab used in our experiment is R2011b.
- **SDPT3**: SDPT3 is a Matlab software package used for solving semidefinite programs. It is, in addition, capable of solving other classes of convex optimisation problems, for example, linear programs, quadratic constrained quadratic programs, and second-order cone programs. The version of SDPT3 used in our experiment is 4.0.
  SDPT3 can be downloaded from *http://www.math.nus.edu.sg/∼mattohkc/sdpt3.html*.
- **YALMIP**: YALMIP is a Matlab toolbox providing a simple way to model optimisation problems in Matlab. YALMIP itself is not a solver. It is often used with other solver packages, for example, SDPT3. The version of YALMIP used in our experiment is R20120420.
  YALMIP can be downloaded from *http://users.isy.liu.se/johanl/yalmip*.

A.2 Robust Pricing Software

We divide our implementation into three categories: single-underlier option pricing models, multiple-underlier option pricing models, and super- and sub-replication pricing models. Each function name is written as a combination of two parts: pricing model and option type. For example, lin_eco is an implementation of the linear decision rule robust pricing model for European call options.

Table A.1 contains a list of acronyms of the option types, and Table A.2 contains a list of acronyms of the pricing models.

Table A.1: Option acronyms

Acronym	Description
eco	European call option
aco	Asian call option
lco	Lookback call option
apo	American put option
bkt	Basket call option
but	Butterfly spread option

Table A.2: Pricing model acronyms

Acronym	Description
bls	Black-Scholes model
binomial	Binomial options pricing model
chen	Chen's ϵ-arbitrage robust pricing model
con	Constant decision rule robust pricing model
lin	Linear decision rule robust pricing model
pie	Piecewise linear decision rule robust pricing model
sup_con	Super-replication constant decision rule robust pricing model
sup_lin	Super-replication linear decision rule robust pricing model
sup_pie	Super-replication piecewise linear decision rule robust pricing model
sub_con	Sub-replication constant decision rule robust pricing model
sub_lin	Sub-replication linear decision rule robust pricing model
sub_pie	Sub-replication piecewise linear decision rule robust pricing model

Especially for the multiple-underlier option pricing models, *sample* refers to a model that takes as input a sample covariance matrix of assets' single-period returns. *Factor*, on the other hand, refers to a model that takes as input covariance matrices of τ-period cumulative returns ($1 \leq \tau \leq T$) estimated by a factor model.