KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

A Genetic Algorithm For Option Pricing

By

Andam Perpetual Saah (MSc. Mathematical Sciences)

A THESIS SUBMITTED TO THE DEPARTMENT OF MATHEMATICS, KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY IN PARTIAL FUFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF PHILOSOPHY (APPLIED MATHEMATICS)

April 25, 2014

Declaration

I hereby declare that this submission is my own work towards the award of the M. Phil degree and that, to the best of my knowledge, it contains no material previously published by another person nor material which had been accepted for the award of any other degree of the university, except where due acknowledgement had been made in the text.

Andam Perpetual Saah	....	....
Student	Signature	Date

Certified by:

Dr. J. Ackora-Prah	....	....
Supervisor	Signature	Date

Certified by:

Prof. S. k. Amponsah	....	....
Head of Department	Signature	Date

Dedication

I dedicate this work to my lovely families and friends and all those who contributed towards my work.

Abstract

The search for a better option pricing model continues to find the one that outperforms the existing ones in the financial market. We present a Genetic Algorithm to price a fixed term American put option when the underlying asset price is Geometric Brownian Motion. The Genetic Algorithm has a better approximation of the relationship between the option price and its contract terms. Our method produces a perfect and a minimum option price that outperforms other models like the Black-Scholes under the same conditions. Our method requires minimum assumptions and can easily adapt to changes and uncertainties in the financial environments.

Acknowledgements

I give thanks to the Almighty God for His guidance and protection. Moreover, I want to express my gratitude to Dr. J. Ackora-Prah, for his great efforts of supervising and leading me to accomplish this project. Furthermore, I want to thank Professor Francis .K. Allotey and Professor S.K Amponsah for their kind hearted words of encouragement and support and AIMS-Ghana for this wonderful opportunity. To all my lovely families and friends especially Gyamera Asante, they were a great source of support throughout this phase. To every person who gave me something to light my pathway, I say thanks to you for contributing to make my thesis a success.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background of Study

Aldous Huxley said "Facts do not cease to exist because they are ignored". When you are ignorant of type of investment you do, it puts you into a weak position. Options gives a world of opportunities to investors. It provides the power to the investor to adjust the position of the situation in the market. It can be conserved or speculated depending on the investor, who can choose to protect the position from falling from the movement of a particular stock in the market. It can also be hedged to minimize risk of the movement of assets in the market.

In 332BC, a man known as Thales bought the right to buy olive oil prior to a harvest. Then came the tulip of mania which turned up in 1636, which were bought in order to speculate the prices of tulip. In the 17th century, they formed a market in London to trade on options and this was the first instance of trading. Russel Sage in 1872 came by with over the counter trading for both call and put options. It was unstandardised and this took place in United States. In 1973, the Chicago Board Options Trade (CBOT) marked the first landmark in the development of this market which brought forth the Chicago Board Options Exchange (CBOE). They regularly published new options quotes which were ready for the sellers or buyers in which it created the first market maker system. Then the Chicago Board Options Exchange (CBOE) and Options Clearing Corporation (OCC) made new exchange. They traded options which was standardized in terms and conditions. In that same year, Fischer Black and Myron Scholes published 'The option pricing model' Black and Scholes (1973). In 1975 the Chicago

Board of Trade (CBOT) created the first interest rate future contract. The 1980s marked the beginning of swaps and other over-the-counter derivatives. In order to eliminate the discrepancies and to strengthen the options made by traders in the financial markets, it was necessary to standardize the exercise prices, expiry dates and option prices.

Options are used by many major financial institutions because they gain a lot when they invest in it. Investing in options makes one takes a vital step in positioning oneself in the option market so that the person does not loose totally in the option market. They are cost efficient and flexible to practice. Also, it is a zero-sum game where both parties benefits.

1.2 Statement of the Problem

Fixed term American put option as compared to American call, European put and European call option is not easy to price. Numerous studies have done on pricing a European call, European put and American call options. For American put option the most priced ones are the perpetual American put option. The price of a European call option is the same as the price of an American call option because they are all exercised at the maturity time but the price of an American put option differs from the price of a European put option. This makes it interesting to focus on the American put option. Observations has been made that pricing an American put option in a continuous time modelling encounters a lot of difficulties. Stopping at any time to exercise an option can affect the investor, this is because the investor can loose massively or gain massively in the market.

Many researches have been done on pricing options which did not consider pricing a fixed term American Put options with Genetic Algorithm. We seek to examine and compare the famous Black-Scholes model and Genetic Algorithm in relation to the way it affects option pricing and to determine the best method that gives the perfect price of a fixed term American put option under the same conditions.

1.3 Objective of the study

This study seeks to examine the analysis of the famous Black-Scholes model and Genetic Algorithm in relation to the way it affects the option pricing. The main aim of this thesis is to determine the best method that gives the perfect price of a fixed term American put option under the same conditions.

1.4 Specific Objectives

The specific objectives of this study are as follows;

1. To calculate the option price
2. To calculate the optimal stopping time.
3. To compute the option price using Black-Scholes model and Genetic Algorithm.

1.5 Methodology

Geometric Brownian motion was used to simulate the underlying asset. Genetic Algorithm and Black-Scholes model were used to calculate the option price and the optimal stopping time of a fixed term American put option.

The programming language used was Python 2.7.3.

1.6 Justification of work

This study will help decision makers who invest in options to find the optimal time to stop and exercise the contract for them to gain and also reduce risk in the financial market. It will also make it attractive to new investors.
In addition, it will help the investors determine how much they are worth so that they can be bought and sold with confidence. To researchers, it will be helpful to research more on Genetic Algorithm applied to different types of options in the financial markets.

1.7 Organisation of the study

This chapter presents the introduction of the work. Chapter two discusses the literature review. Chapter three deals with the basic definitions, mathematical preliminaries and the methodology of the work. Chapter four demonstrates the pricing of the American option and the analysis of the work. Chapter five is the conclusion and the recommendation. This project ends with references and appendices which supports the study.

Chapter 2

Literature Review

2.1 Introduction

This chapter provides a brief survey on previous works in context of the study. The aim of this chapter is to review relevant literature about pricing options and various application of genetics in finance.

2.2 Review of Literature

Grace (2000) priced Black-Scholes option via Genetic Algorithm. She argued that Genetic Algorithm can be more effectively in finding the accurate values of the implied volatilities than the calculus-based methods. Moreover, she found out that the implied volatilities with the Genetic Algorithms produced option prices which were much closer to the theoretical values than the Newton-Raphson implied volatilities. She compared the performances of Newton-Raphson method and Genetic Algorithm for finding the implied volatilities for the Black-Scholes model. She used data from the Standard and Poor 100 Index option which she obtained from the resorted format of the Berkeley option tapes. Furthermore, she noticed that the Newton-Raphson method allowed 1000 iterations at maximum while the Genetic Algorithm was set at 99. Also, she presented that the Newton-Raphson method failed to converge for 8 out of 500 observations, even with those eight occurrences, she noticed that four produced negative option values and the other four produced zero implied volatilities. She showed that for all the 500 observations the Genetic Algorithm produced implied volatilities. Moreover, she discussed that the average difference between the actual call and the Black-Scholes call using genetic algorithm for the implied volatilities was 0.000020325 whiles the

Newton-Raphson method produced the average difference of 0.0087805. She presented that the genetic algorithm exactly matched the actual values while the Newton-Raphson method did not perform well.

Chen et al. (2010) focused on the trading behaviour of the option market and the importance of the use of dynamic character for pricing option. Furthermore, they proposed a hybrid model by combining Genetic Algorithm and the Black Scholes model to produce a Genetic-Black Scholes model in order to estimate the price of the option more accurately. They compared the Black Scholes model, the Gram-Charlier garch model and the Genetic-Black Scholes pricing model. They realized that the Gram-Charlier garch model output gave the worst output during the simulation. They then noticed that Genetic-Black Scholes model verified that the dynamic option pricing model was applicable. The simulation result indicated that Genetic-Black Scholes model estimated better and more exact accurate than Black Scholes model even the Gram-Charlier garch model. They concluded that the Genetic-Black Scholes pricing model was exactly practical.

Shu-Cheng and Lee (1997) demonstrated how Genetic Algorithm was helpful in dealing with option pricing, the case of European style. They tested the performance of Genetic Algorithm in determining the prices of European call options whose exact solution was known from the Black Scholes option pricing theory. They use GENESIS 5.0 software which was developed by John Grefenstette for the study of Genetic Algorithm for function optimisation. They noticed that the boundary conditions using the Genetic Algorithm was arbitrarily imposed and it only satisfied the case when the stock price was greater than the exercise price. Moreover, when the stock price was less than the exercise price at the date of expiry, some of the call prices appeared to be negative. The solutions that they found using the basic Genetic Algorithm were compared to the exact solution and their results showed that Genetic Algorithm was a powerful tool for option pricing.

Sidarto (2006) used a simple Genetic Algorithm to find an approximate solution of the non-linear equation. His results showed eight different strike prices, its corresponding call option prices and the implied volatility computed for each of the eight different strike prices using Genetic Algorithm. He also computed the Black Scholes value for the 'market' data for a European call option. He plotted a graph of implied volatility against the strike price and they found out that the curve was convex in shape, rather than straight horizontal line as suggested by the Black Scholes formula. He noticed that if the Black Scholes formula was valid, then the volatility would be the same for each strike price. He concluded that the method was easy to implement without using much mathematical requirements.

Rimcharoen et al. (2007) focused on the compact Genetic Algorithm which employed the probability vector as a model that scales well with the problem size. Furthermore, they found the optimal stopping time of trap problems and they proposed an optimal stopping criterion as a decision which provided a stopping boundary, where the termination was optimal on one side and continuation is on the other which helped save the computational effort by stopping early. Furthermore, they showed the evolutionary process reached a higher solution quality when the reset method was incorporated. For the case of early stopping problems, they employed the stopping region as a policy when the algorithm stops running which suggested that it was unlikely to achieve a good solution with the fitness value that falls within this stopping region. They improve the quality of the solution by reversing the probability vector when the fitness fell into the stopping region. Furthermore, they noticed that when each dimension was reversed from the probability vector by 1 minus probability in that dimension, the algorithm explored the solution in the counterpart and provided an opportunity to search more candidate solutions when the model seemed to go bad. They noticed that the method that they proposed can also be applied to analyse other problems.

Lazo et al. (2003) proposed a model which was based on Genetic Algorithms and Monte Carlo simulation to find an optimal decision rule for oil field development alternatives which was under market uncertainty. They presented that this method would help decision-making with regard to the development of a field immediately or waiting until market conditions are more favourable. They used three mutually exclusive alternatives to form the optimal decision rule which described three exercise regions along time up to the expiration of the concession of the field. They also used the Monte Carlo simulation, which was employed within the Genetic Algorithm for the purpose of simulating the possible paths of oil prices up to the expiration date and they assumed that oil prices follow a Geometric Brownian Motion. The result they obtained showed that Net Present Value decreases as the number of iterations in the Monte Carlo Simulation increases. Also they noticed that when the model was used with the Genetic Algorithm in the analysis of development alternatives, it proved to be more flexible. They found out that, it was possible to introduce a greater number of investment alternatives to change the stochastic process or to introduce other uncertainties with minor modifications. Also, the Genetic Algorithm made it possible for them to obtain optimal or suboptimal decision rules and to avoid the need to solve partial differential equations.

Jackson (2000) applied Genetic Algorithm to the problem of asset allocation by using traditional mean variance approach and also used direct utility maximization method for a step utility function. He compared the performance of Genetic Algorithm with an alternative method of optimization which was Newton's method. He noticed that Genetic Algorithm can be used for asset allocation problem by specifying an objective function for the fund, also by specifying the constraints on the asset weight and specifying the penalty function.

Wu (1997) in the paper " Using Genetic Algorithm to determine near-optimal pricing, investment and operation Strategies in the electric power industry" found out that Genetic Algorithm provided feasible solutions and close solutions to the optimal when the optimal solution is known. He focused primarily on exploring Genetic Algorithm as a method for solving the first-best welfare maximization problem for a given network structure in which Genetic Algorithm was used to find the optimal network structures. He proposed that Genetic Algorithm approach can only be tested by pushing the limits of currently available non-linear optimization software. He noticed that, the performance was so far better. The result he obtained showed that, using Genetic Algorithm for a realistic systems has been without assurance of feasibility or benchmark test against optimal solutions.

Heigl (2008) found a closed-form solution for the price of European call options in which the underlying securities follow a Generalized Autoregressive Conditional Heteroskedastic process. He simulated a data over a wide range to cover a lot of existing options in one single equation. He used Genetic Programming to generate the pricing function from the data in which the resulting equation was found via a heuristic algorithm. He noticed that, the resulting equation could be used to calculate the price of an option in the given range with minimal errors. He presented that the equation was well behaved and could be used in standard spread sheet programs which offered a wider range of utilization or a higher accuracy respectively than other existing approaches. He noticed that, the situation that all European call options with an expiration time of up to 90 trading days are covered. Also, he stated that one can find a lot of empirical evidence that the underlying following a Generalized Autoregressive Conditional Heteroskedastic process was realistic.

Smith and Hussain (2012) proposed a real coded Genetic Algorithm particle filter for the dual estimation of stochastic volatility and parameters of a Heston type stochastic volatility model. They compared the performance of their hybrid particle filter with a parameter learning particle filter. They noticed that, their algorithm performed well for both the volatility and parameter estimation. They presented the recombination operator for being responsible for the improved performance of the particle filter.

Kaboudan (2000) proposed a profitable trading strategy based on predictions of stock-prices. He used a metric that quantified the probability that a specific time series was Genetic Programming predictable was used to show that stock prices are predictable. He noticed that Genetic Programming evolved regression models that produce reasonable one-day-ahead forecasts only. He presented that, the limited ability was led to the development of a single day-trading strategy in which trading decisions are based on Genetic Programming forecasts of daily highest and lowest stock prices. He found that the single day-trading strategy which was executed for fifty consecutive trading days of six stocks yielded relatively high returns on investment. His results clearly suggested that Genetic Programming predictability of price levels and not returns in which he explained the difference as perhaps the natural ability to identify variables that explain price level variations over time. He presented the evidence that Genetic Programming may forecast out-of-sample price levels better.

Chidambaran et al. (1998) proposed a methodology of Genetic Programming to approximate the relationship between the option price, its contract terms and the properties of the underlying stock price. They used Monte Carlo simulations to show that the Genetic Programming model approximates the true solution better than the Black-Scholes model when stock prices follow a jump-diffusion process. They showed that Genetic Programming model outperformed various other models in many different settings. They presented genetic programs as flexible, self-learning and self-improving and an ideal tool for practitioners. They suggested that the Genetic Programming approach works well in practice. They showed that Genetic Programming formulas beat the Black-Scholes model in 10 out of 10 cases in a simulation study where the underlying stock prices were generated using a jump diffusion process.

Yin et al. (2007) illustrated the application of an adaptive form of Genetic Programming, where the probability of crossover and mutation was adapted dynamically during the Genetic Programming run, to the important real-world problem of options pricing. They carried out their tests using market option price data and the results they obtained illustrated that the new method yields better results than the ones obtained from Genetic Programming with fixed crossover and mutation rates. They presented that the developed method had a potential for implementation across a range of dynamic problem environments.

Sunzu (2007) noticed that an American put option can be treated as optimal stopping problem in which he used boundary conditions to find the solutions to the optimal stopping problem. He noticed that pricing a perpetual American put option is always an optimal stopping problem regardless of whether the stock pays dividends or not. Furthermore, he presented that it is not in the case of American call option which can be considered an optimal stopping problem only when the stock pays dividends. He found that optimal stopping problems do not only solve derivative securities like options problems but it is powerful tool which solves many mathematical problems. He presented that American put option can be of finite or infinite horizon.

Mcwilliams (2005) priced American options using Monte Carlo simulation. He designed an algorithm to price an American put option using the Black-Scholes model, the binomial tree model and Monte Carlo simulation. He found the estimated option price and the average optimal stopping time for an American option. He presented that for the Black-Scholes model, in theoretical aspect, pricing American options using optimal stopping times can be done while in the practical aspect it becomes challenging. Also, he found that for binomial tree method of Cox, Ross and Rubinstein, it only considers a discrete set of stock price values at each time period in American options and so it can be generalised to a continuous interval of possible stock prices by increasing the number of periods. Lastly, he presented Monte Carlo simulation as a significant tool in pricing an American option.

Haugh and Leonid (2004) priced American option and portfolio optimization problems when the underlying state space is high-dimensional. They used programming and dual-based methods for constructing and evaluating good approximate solutions to the problems. General algorithm was written by them for constructing upper and lower bound on the true price of the option using any approximation to the option price. They found out that the bounds were tight so that if the initial approximation was close to the true price of the options then the bounds were also to be close to it. Simulation of the suboptimal exercise strategy was used by the approximate option price. They computed the upper bound using Monte Carlo simulation. Their results presented their algorithm proven to be accurate by using a set of sample problems where they priced the call option on the maximum and the geometric mean of a collection of stocks.

Shreve (2004) priced a perpetual put American option when an underlying asset price is a Geometric Brownian Motion. He presented the theory of Laplace transform for first passage time of drifted Brownian motion to price the perpetual American put option. He also used the same theory to find a discounting factor. Furthermore, he found a price which is lower than the striking price at which the holder of the put would exercise at that price to get a maximum value.

Bjerksund and Stensland (1991) found a simple and intuitive approximation of American call and put value. The approximations that they used generalized the model they had by dividing time to maturity into two periods with each of them being a flat early exercise boundary. Moreover, they approximated the American option which was accurate and computer efficient value by imposing a feasible but non-optimal exercise strategy. They also assumed early boundary exercise and a lower bound to the true option value was obtained. Also they used some numerical investigations which indicated that the method was accurate and extremely computer efficient approximation to the American option values.

Baxter and Rennie (1993) presented hedging and pricing by arbitrage in the discrete time setting of binary trees. They focused on extending the same idea under continuous time setting. They used Brownian motion which was brought out as well as Itô calculus which was needed to manipulate it and came into completion in the derivation of the Black-Scholes formula. Also, they built foundations on probability to find the strongest possible links between claims and their random underlying stocks. More-so, they explored the limits of arbitrage and put together a mathematical framework that was strong enough to be a realistic model of the real financial market in which they structured to support construction techniques. They presented the actual financial instruments such as dividend paying equities, currencies and coupon paying bonds and adapts the Black-Scholes to each in turn.

Doubt was taken away about the prevailing belief that American-style options which cannot be valued efficiently in a simulation model by Tilley (1993). He removed the idea that has been considered as a major hindrance to the use of simulation models for valuing financial instruments. He came out with a general algorithm for estimating the value of American options on an underlying instrument for which the arbitrage-free probability distribution of paths through time can be simulated. Furthermore, he used an example to test the general algorithm for which the exact option premium can be determined.

Chapter 3

Methodology

3.1 Introduction

Investors and traders must be aware of the level of risk in the market whether the prices are rising or falling in order to control the flow of cash to prevent them from loosing much money.
In this view that this chapter demonstrates how to calculate the underlying asset price given by the Geometric Brownian Motion model. It also presents the methodology, the basic concepts, some definitions and mathematical preliminaries.

3.2 Basic Definition and Mathematical Preliminaries

3.2.1 Derivative Security

A derivative security is a financial instrument whose value is completely dependent on the price of the underlying asset. Examples are options, swaps, forward contract and future contract (Hull, 1997).

3.2.2 Futures contract

Future contract is a standardized contract between two parties to trade a specified asset of standardized quantity and quality for a strike price with delivery and payment occurring at a specified future date which is known as the delivery date (Hull, 1997).

3.2.3 Forward contract

Forward contract is a non-standardized contract between two parties to trade an asset at a specified future time at a strike price. The price agreed upon is known as the delivery price which is equal to the forward price the contract is entered into (Hull, 1997).

3.2.4 Swap

Swap is a derivative in which two counter-parties exchange cash flows of one party's financial instrument for those of the other party's financial instrument (Hull, 1997).

3.2.5 Options

Options are contract that gives the holder the right but not under obligation to buy or sell an underlying asset at a prescribed price known as the strike price before or at the expiry date or the maturity date (Black and Scholes, 1973).

If the seller of the option exercises the option then the buyer must fulfil the obligation to the transaction that has taken place.
The buyer pays a premium (which is a small fee of redeeming a proof of purchase of an underlying asset) to the seller.
There are two types of options; the call option and the put option.

Call Option

The call option is the right but not under obligation to buy an underlying asset at a prescribed price known as the strike price before or at the expiry date.

Put Option

The put option is the right but not under obligation to sell an underlying asset at a prescribed price known as the strike price before or at the expiry date.

3.2.6 Option Pricing Model

An option pricing model is any model that seeks to determine the proper valuation of an option and it attempts to set accurate prices for options using available information (Black and Scholes, 1973).

The option pricing theory will need to take into account the past information, current prices and more likely future performance about the asset for the call and put option in order for the seller or the buyer to arrive at a fair valuation. It also need to record the length of time the option lasts in order for to know whether there is a lost or gain.

3.2.7 Option Styles

In this Project, only two of the option styles; American option and European option are focused on irrespective of the other option styles.

3.2.8 American Option

An American option is a financial instrument that gives its holder the right to trade an underlying asset at any time before or at the maturity time ($t \leq T$) at prescribed price K without being obliged to do so.

American option is consist of two types: the American call option and the American put option.

American Call Option

An American call option is an option that provides the holder the right to buy an underlying asset at any time before or at maturity ($t \leq T$) at a strike price K without being under obligation to do so.

Suppose S_t is the underlying asset price and K is the strike price or the prescribed price then, we expect the payoff of the holder of an American call option to be

$$P_t = \begin{cases} S_t - K, \quad S_t > K, t \leq T & \text{if exercised} \\ \\ 0 & \text{if not exercised.} \end{cases}$$

American Put Option

An American put option is an option that provides the holder the right to sell an underlying asset at any time before or at maturity ($t \leq T$) as soon as the underlying asset falls "far enough" below the strike price K without being under obligation to do so.

Suppose S_t is the underlying asset price and K is the strike price or the prescribed price then, we expect the payoff of the holder of an American put option to be

$$P_t = \begin{cases} K - S_t, \quad S_t < K, t \leq T & \text{if exercised} \\ \\ 0 & \text{if not exercised.} \end{cases}$$

3.2.9 European Option.

A European option is a financial instrument that gives its holder the right to trade an underlying asset at maturity time at a prescribed price K without being obliged to do so.

There are two types of European option: the European call option and the European put option.

European Call Option

A European call option is an option that provides the holder the right to buy an underlying asset at maturity time at a strike price K without being under obligation to do so.

Suppose S_T is the underlying asset price at maturity and K is the strike price or the prescribed price then, we expect the payoff of the holder of a European call option to be

$$P_T = \begin{cases} S_T - K, \quad S_T > K, & \text{if exercised} \\ \\ 0 & \text{if not exercised.} \end{cases}$$

European Put Option

A European put option is an option that provides the holder the right to sell an underlying asset at maturity time at a strike price K without being under obligation to do so.

Suppose S_T is the underlying asset price and K is the strike price or the prescribed price then, we expect the payoff of the holder of a European call option to be

$$P_T = \begin{cases} K - S_T, \quad S_T < K, & \text{if exercised} \\ \\ 0 & \text{if not exercised.} \end{cases}$$

This is when the holder of this European put option expects the price of the underlying asset to fall at the expiry date.

The total value of an option consists of intrinsic value (how far in-the-money an option is) and time value (the difference between the price paid and the intrinsic value). The time value approaches zero as the date of expiration draws near the time value which is also called option price.

3.3 Mathematical Preliminaries

A stochastic process $\{S_t\}_{0 \leq t \leq T}$ is a collection of random variables on $(\Omega, \mathcal{F})$. For each fixed $w \in \Omega$, $S_t(w)$ is the sample path of S_t associated with w (Mcwilliams, 2005).

Let $(\Omega, \mathcal{F}, Q)$ be a probability space, then a standard one dimensional Brownian motion is a continuous stochastic process $\{W_t\}_{0 \leq t \leq T}$ such that,

- $W_0 = 0$.
- For $0 \leq s < t \leq T$, $W_t - W_s$ is independent of $\mathcal{F}_s$ and is normally distributed with mean 0 and variance $t - s$.
- For $0 = t_0 < t_1 < \cdots < t_d = T$, the increments $W_{t_j} - W_{t_{j-1}}$ are independent and their distribution depends only on the difference $t_j - t_{j-1}$. $\{W_t\}$ is said to independent and stationary increments (Mcwilliams, 2005).

Let $\Omega = \{\omega_1, \omega_2, \cdots\}$ be a sample space. Then a σ-algebra, $\mathcal{F}$, is defined as the set of all observable events for a single trail satisfying the properties;

- $\emptyset, \Omega \in \mathcal{F}$.
- If $A \in \mathcal{F}$, then $\Omega \setminus A \in \mathcal{F}$.
- If $A_1, A_2, \cdots \in \mathcal{F}$ then $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$

$\mathbb{P}$ is the probability measure on $\mathcal{F}$ where, $\mathbb{P}(A) \in [0,1], \forall A \in \mathcal{F}$ and $\mathbb{P}(\Omega) = 1$. The probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is equipped with a filtration that is a collection of σ-algebras $\{\mathcal{F}_t\}_{0 \le t \le T}$ where $\mathcal{F}_0 \subseteq \mathcal{F}_s \subset \mathcal{F}_t \subseteq \mathcal{F}_T$, for $0 \le s < t \le T$ (Mcwilliams, 2005).

Consider the probability space (Ω, A, Q), where Q is equivalent to $\mathbb{P}$, a filtration $\mathcal{F}_t$ for $t \ge 0$ is an increasing family of a σ-algebra included in A.

The σ-algebra $\mathcal{F}_t$ represents the information available at time t. A process S_t is adapted to $\mathcal{F}_t$, for $t \ge 0$, if for any t, S_t is $\mathcal{F}_t$ measurable which means by that time t an investor has all the information to calculate the value of the underlying asset price S_t (Lamberton and Lapeyre, 1996).

Lets assume that the current asset price S_t is $\mathcal{F}_t$ measurable that is within the period of $[0, t]$. S_t is adapted to $\{\mathcal{F}_t\}_{0 \le t \le T}$ since the previous and the present values are known not the future ones (Mcwilliams, 2005).

- S_t is said to be a martingale if for every $s < t, \quad E[S_s \mid \mathcal{F}_t] = S_s$.
- S_t is said to be a supermartingale if for every $s < t, \quad E[S_s \mid \mathcal{F}_t] \le S_s$.
- S_t is said to be a submartingale if for every $s < t, \quad E[S_s \mid \mathcal{F}_t] \ge S_s$.

An Itô integral is defined as,

$$\int_0^T S_t dW_t = \sum_{i=0}^{n-1} S_i (W_{t_{i+1}} - W_{t_i}),$$

where W_t is a standard Brownian motion adapted to the filtration $\mathcal{F}_t$.

3.3.1 Definition of Standard Brownian Motion Model

We consider a standard Brownian motion as a random process $W = \{W_t : t \in [0, \infty)\}$ with a space $\mathbb{R}$ that satisfies the following properties:

- $W_0 = 0$
- W has stationary increments, $\forall s, t \in [0, \infty)$ with $s < t$, the distribution depends only on the difference $W_t - W_s$ that is (W_{t-s})
- W has independent increments, $t_1, t_2, \cdots t_d \in [0, \infty)$ with $t_1 < t_2 < \cdots t_d$, then $W_{t_1}, W_{t_2} - W_{t_1}, \cdots, W_{t_d} - W_{t_{d-1}}$ are independent.
- W_t is normally distributed with mean 0 and variance t, $\forall\, t \in [0, \infty)$
- W_t is continuous on $[0, \infty)$.

Let $\tilde{B} : [0, T] \times \Omega \to \mathbb{R}$ be a Wiener process defined up to $T > 0$ and let $S : [0, T] \times \Omega \to \mathbb{R}$ be a stochastic process that is adapted to the natural filtration $\mathcal{F}$ of a Wiener process (Oksendal, 2003). Then

$$E\left[\left(\int_0^T S_t d\tilde{B}_t\right)^2\right] = E\left[\int_0^T S_t^2 dt\right].$$

A stopping time τ is a random variables taking values in $[0, \infty]$ and satisfying $\{\tau \leq t\} \in \mathcal{F}(t), \quad \forall t \geq 0$ (Shreve, 2004).

3.3.2 Itô Lemma

Let S_t be a stochastic process and let $f(a,t)$ be a measurable function with a continuous partial derivatives up to the second order term then,

$$df(t,S_t) = \frac{\partial f}{\partial t}(t,S_t)dt + \frac{\partial f}{\partial a}(t,S_t)dS_t + \frac{1}{2}\frac{\partial^2 f}{\partial a^2}(t,S_t)(dS_t)^2.$$

3.4 The Model of the Underlying Asset Price

Models are important for investors and those who trade in the financial world in order to determine the accurate prediction of the future behaviour of prices of an asset and fluctuations of their prices in the market so that when they intend to buy or sell they do that at the right time.

In this project the model used is Geometric Brownian Motion. Most economists prefer Geometric Brownian Motion as a model for market prices because it is everywhere positive. It always has a probability of 1.
Suppose that the price of the underlying asset S_t is given by a Geometric Brownian Motion Model,

$$\frac{dS_t}{S_t} = \rho dt + \sigma dW_t. \tag{3.1}$$

From the model we will determine the underlying asset price using the Itô formula. The following assumption taken is; $W(t)$ is a one-dimensional standard Brownian motion with respect to the risk neutral measure Q and $dW_t \sim N[0, dt]$. The volatility rate, σ is constant, ρ is the constant force of interest. We note that S_t is the underlying asset price at time t.

Let $W_t, \quad 0 \leq t \leq T$ be a standard one dimensional Brownian motion on a probability space $(\Omega, \mathcal{F}, Q)$ and $\{\mathcal{F}(t), 0 \leq t \leq T\}$ be a filtration for this Brownian motion where T is a fixed final time.

Then from (3.1) we can solve it as,

$$\frac{dS_t}{S_t} = \rho\, dt + \sigma\, dW_t\,, \qquad \text{assuming } W_0 = 0.$$

$$dS_t = S_t\,[\rho\, dt + \sigma dW_t.]$$

Solving for S_t we apply Itô formula to $d\ln S_t$,

$$\begin{aligned}
d\ln S_t &= \frac{1}{S_t} dS_t - \frac{1}{2S_t^2} dS_t^2, \\
d\ln S_t &= \frac{1}{S_t} S_t[\rho\, dt + \sigma dW_t] - \frac{1}{2S_t^2} S_t^2[\sigma^2 dW_t^2], \\
d\ln S_t &= \rho\, dt + \sigma dW_t - \frac{1}{2}\sigma^2 dt.
\end{aligned}$$

Then it follows that, integrating and applying fundamental theorem of calculus to it we get,

$$\ln S_t - \ln S_0 = \left(\rho - \frac{\sigma^2}{2}\right) t + \sigma W_t.$$

$$S_t = S_0 \exp\left[\left(\rho - \frac{\sigma^2}{2}\right) t + \sigma W_t.\right] \tag{3.2}$$

Simulating the underlying asset price using this equation (3.2), we will use some numerical values to simulate the underlying asset price (S_t) over the period of $[0, T]$. These are the following values that will be used; initial underlying asset price $S_0 = 100$, the volatility rate $\sigma = 0.35$,the interest rate $\rho = 0.1$ and the maturity time from now $T = 4$ years.

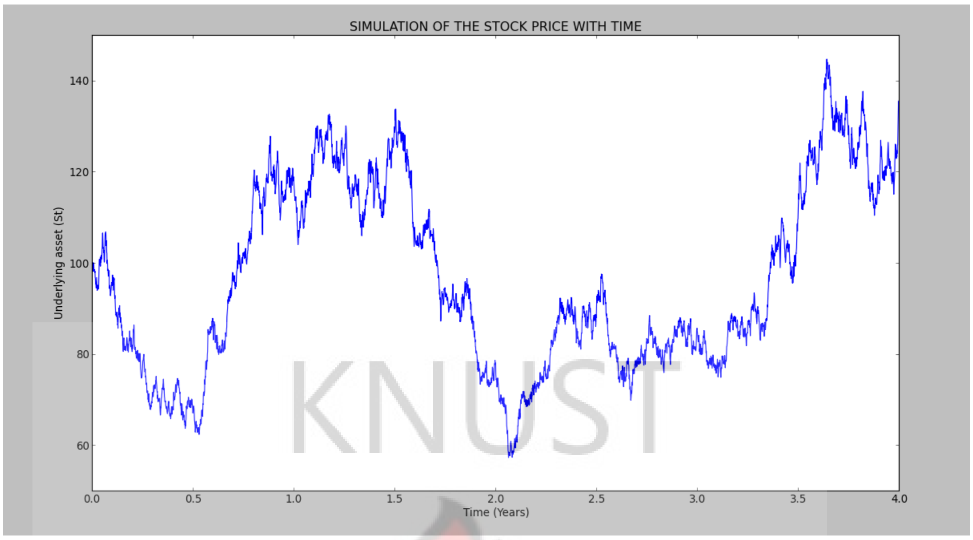

Figure 3.1: A Simulated Underlying Asset Price that Follows a Geometric Brownian Motion Model.

The graph above shows a simulated underlying asset price over the period of $[0, T]$. It can be observed from the graph that the underlying asset price is positive everywhere and also it changes along with time. A put option can be exercised at $t = 3.6$years. Examples of some assets whose values follows the Geometric Brownian Motion are; gold, diamond, tomatoes etc.

3.4.1 Finding the Expectation of the Underlying asset

From (3.2) we have,

$$S_t = S_0 \exp\left[\left(\rho - \frac{\sigma^2}{2}\right) t + \sigma W_t.\right]$$

$$E[S_t] = E\left[S_0 \exp\left[\left(\rho - \frac{\sigma^2}{2}\right) t + \sigma W_t\right]\right]$$

$$E[S_t] = S_0 \exp\left[\left(\rho - \frac{\sigma^2}{2}\right) t\right] E[e^{\sigma W_t}.]$$

But we know that $W_t = W_t - W_0$ which follows a normal distribution $N(0,t)$, then it follows that,

$$E[e^{\sigma W_t}] = \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{\infty} e^{\sigma x} e^{-\frac{x^2}{2t}} dx.$$

But we have,

$$-\frac{(x-\sigma t)^2}{2t} = -\frac{1}{2t}(x^2 - 2x\sigma t + \sigma^2 t^2) = -\frac{x^2}{2t} + \sigma x - \frac{\sigma^2 t}{2},$$

then it follows that we have,

$$e^{\sigma x} e^{-\frac{x^2}{2t}} = e^{-\frac{(x-\sigma t)^2}{2t}} e^{\frac{(\sigma^2 t)}{2}},$$

which gives,

$$E[e^{\sigma W_t}] = \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{\infty} e^{-\frac{(x-\sigma t)^2}{2t}} e^{\frac{\sigma^2 t}{2}} dx,$$
$$E[e^{\sigma W_t}] = e^{\frac{\sigma^2 t}{2}} \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{\infty} e^{-\frac{(x-\sigma t)^2}{2t}} dx,$$

but

$$\frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{\infty} e^{-\frac{(x-\sigma t)^2}{2t}} dx = 1$$

then,

$$E[e^{\sigma W_t}] = e^{\frac{\sigma^2 t}{2}}.$$

Then it follows that,

$$E[S_t] = S_0 e^{(\rho - \frac{\sigma^2}{2})t} e^{\frac{\sigma^2 t}{2}},$$

$$E[S_t] = S_0 e^{\rho t}.$$

Therefore the expectation of the underlying asset is

$$E[S_t] = S_0 e^{\rho t}. \tag{3.3}$$

3.4.2 Finding the variance of the Underlying asset

From (3.3) we have,

$$(E[S_t])^2 = S_0^2 e^{2\rho t},$$

$$E[S_t^2] = E\left(\left(S_0 \exp\left[\left(\rho - \frac{\sigma^2}{2}\right)t + \sigma W_t\right]\right)^2,\right)$$
$$E[S_t^2] = \left(S_0 \exp\left(\rho - \frac{\sigma^2}{2}\right)t\right)^2 E[e^{2\sigma W_t}].$$

$$E[e^{2\sigma W_t}] = \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{\infty} e^{2\sigma x} e^{-\frac{x^2}{2t}} dx.$$

But we have,

$$-\frac{(x - 2\sigma t)^2}{2t} = -\frac{1}{2t}(x^2 - 4x\sigma t + 4\sigma^2 t^2) = -\frac{x^2}{2t} + 2\sigma x - 2\sigma^2 t,$$

then it follows that we have,

$$e^{2\sigma x} e^{-\frac{x^2}{2t}} = e^{-\frac{(x-2\sigma t)^2}{2t}} e^{2\sigma^2 t},$$

which gives,

$$E[e^{2\sigma W_t}] = \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{\infty} e^{-\frac{(x-2\sigma t)^2}{2t}} e^{2\sigma^2 t} dx,$$
$$E[e^{2\sigma W_t}] = e^{2\sigma^2 t} \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{\infty} e^{-\frac{(x-2\sigma t)^2}{2t}} dx,$$

but

$$\frac{1}{\sqrt{2\pi t}}\int_{-\infty}^{\infty} e^{-\frac{(x-2\sigma t)^2}{2t}}dx = 1.$$

then,

$$E[e^{2\sigma W_t}] = e^{2\sigma^2 t}.$$

Then it follows that,

$$E[S_t^2] = \left(S_0 e^{(\rho-\frac{\sigma^2}{2})t}\right)^2 e^{2\sigma^2 t},$$
$$E[S_t^2] = S_0^2 e^{2\rho t} e^{-\sigma^2 t} e^{2\sigma^2 t},$$
$$E[S_t^2] = S_0^2 e^{(2\rho+\sigma^2)t}.$$

Then it follows that,

$$Var(S_t) = E[S_t^2] - (E[S_t])^2,$$
$$Var(S_t) = S_0^2 e^{(2\rho+\sigma^2)t} - S_0^2 e^{2\rho t},$$
$$Var(S_t) = S_0^2 e^{2\rho t}\left(e^{\sigma^2 t} - 1\right).$$

Hence the variance of the underlying asset is

$$Var(S_t) = S_0^2 e^{2\rho t}\left(e^{\sigma^2 t} - 1\right).$$

3.5 Overview of Genetic Algorithms

3.5.1 Introduction to Genetic Algorithm

Since creation, human beings still learn to understand the world and the role every item in it plays in our everyday life. This has made it to the extent that some scientist can even predict some natural phenomena which will happen in the future. We have even come to understand some fundamental limits to our abilities to predict these natural phenomena. Sometimes we learn even hard way to control some natural phenomena to the extent that some still can't be controlled. All these are the blessings we get from our creator. This is to show us that the world is in order not chaos.

Electronic computers have been the most revolutionary development in the history of science and technology. This has increased our ability to predict and control nature which came in the form of computer programming.

Genetic Algorithm are the numerical optimizations algorithm which are inspired by both natural and artificial genetics (Holland, 1975). Using population of solutions to solve practical optimization problems was brought up several times in the 1950s and 1960s. Nils Aall Barricelli in 1954 used computer simulations of evolution in artificial life research which his publish was not widely noticed. During 1957, Alex Fraser, who was an Australian quantitative genetics, published some papers on simulation of artificial selection of organisms with multiple loci to control a measurable trait. His simulations included all essential elements of modern Genetic Algorithms. John Henry Holland in 1960s invented the Genetic Algorithms. This brought up many insight in using Genetic Algorithms to solve practical problems. He published a book known as "Adaptation in natural and artificial systems" in 1975. He introduced a framework for predicting the quality of the next generation called the Holland's Schema Theorem (It states that low-order schemata with above average fitness increases exponentially in successive generations (Holland, 1975)).

Research in Genetic Algorithms remained theoretical until the first conference on Genetic Algorithm was held in Pittsburgh Pennsylvania in the mid 1980s. It then grew popular until late 80s where the General Electric sold the mainframe-based toolkit which was designed for industrial processes which was also the first Genetic Algorithms product. Then came the Evolver (Evolver is a software package that allows users to solve a wide variety of optimization problems using Genetic Algorithms) in 1989 by Axcells Incorporated. It was the first commercial Genetic Algorithms product for desktop computers and remained commercial until 1995 when the Evolver was translated to its many languages.

Several years have past away and there has still been a widespread of interaction among researchers researching on evolutionary programming and other evolutionary approaches. Recently researchers uses Holland's Genetic Algorithms to research in image processing, facial recognition, laser technology, medicine, spacecraft trajectories, analysis of time series, robotics, jobshop scheduling, stock prediction and also solve some problems in other fields of study.

3.5.2 Some Basic terms in Genetic Algorithms

Search Space: It is a space of all feasible solutions. Genetic Algorithms is about exploiting random search which are used to solve optimization problems. Each point in the search space represents one feasible solution and each feasible solution is marked by its fitness for the problem at hand. Each individual is coded into binary alphabets which are represented as $\{0, 1\}$ and they behave like chromosomes (Holland, 1975) .

Chromosomes: They are solutions composed of several genes which can be also called variables.
A fitness score is assigned to each solution which represents the abilities of an individual to compete with each other (Holland, 1975).

Locus: This is when each gene has its own position in a solution (Holland, 1975).

Genome: It is a complete set of genetic material that is all chromosomes (Holland, 1975).

Fitness: This measures the success of the organism in its life, that is the survival of the organism (Holland, 1975).
Gene: This is a molecular unit of heredity of a living organism (Holland, 1975).

Fitness function: This is a particular type of objective function that is used to summarise how close a given design solution is to achieving the set aims (Holland, 1975).

Genetic operators: They are used in Genetic Algorithms to generate diversity and also to combine existing solutions into others (Holland, 1975).

Schema: It is a template in computer science used in the field of Genetic Algorithms that identifies a subset of strings with similarities at certain string positions (Holland, 1975).

3.6 Genetic Operators

3.6.1 Selection

Selection is the process that takes place in a genetic algorithm in which individual genomes are chosen from a population which are bred later. The breeding can be in the form of mutation or crossover.

3.6.2 Methods of selection

There are several methods of selecting of chromosomes for mutation or crossover. All the methods have a unique way of selecting chromosomes. The following are some of the selection methods that are used for mutation or crossover.

Roulette wheel selection

In roulette wheel selection, individuals that are potentially useful for crossover are selected. Here, the fitness level associated to a probability of selection with each chromosome which assigns fitness to possible solutions within the population. Let's say we have f_i to be the fitness of individual (i) in the population (N) then the probability of individuals being selected will be $p_i = \dfrac{f_i}{\Sigma_{j=1}^{N} f_j}$.

This is done by a proportion of the wheel being assigned to each of the possible selections based on their fitness value and then a random selection is made. This is like playing game on the roulette wheel in which each candidate is drawn independently. Then solutions that appears to have a higher fitness will be less likely to be eliminated, even though there is still a chance that they may be eliminated. In this process, there is a chance some weaker solutions may survive the selection process.

Tournament selection

This process is like running several "tournaments" among which a few individuals are chosen at random from the population. Here, the winner which is the one with the best fitness of each tournament is selected for the next process which is crossover. When the tournament size is very large then the individuals that are weak have a smaller chance to be selected.
We can select best individual in any tournament or select individuals at random. The individual that wins the tournament can be removed from the population that the selection is made from, otherwise individuals can be selected more than once for the next generation.

Truncation selection

In this process, the individual solutions are ordered by their fitness which is in accordance to some proportion p of which individuals that is the fittest are selected and reproduced by $\frac{1}{p}$ times.

Rank selection

Here, each individual in the population is assigned a numerical rank based on fitness and then it is selected based on the ranking.

Steady-state selection

In this process, the offspring of the individuals selected from each generation go back into the pre-existing gene pool which they were selected from and this replaces some of the less fit members of the previous generation. During this process some of the individuals are retained between the generations.

Local selection

Here, the individual fitnesses are compared to a fixed threshold, rather than comparing it to each other, to decide who gets to reproduce. Every individual stays inside an environment which is constrained called the local neighbourhood in which the individuals interact only with individuals inside this region.

Boltzmann selection

In this process, a continuously varying "temperature" controls the rate of selection according to a preset schedule. Suppose there is a high temperature start then the selection pressure will be low which means that every individual has some probability of reproducing while if there is a low temperature then the selection pressure increases gradually.

3.6.3 Crossover

It is a genetic operator that is used to vary the programming of chromosomes from one generation to the other. In crossover more than one parent solutions are used to produce a child solution from them.

3.6.4 Crossover techniques

These are techniques used to perform crossover to produce offsprings. The following are some crossover techniques used to produce offsprings:

One-point crossover

It is a single crossover point in which both parents' organism strings is selected. Here, the selected string are swapped between the two parent organisms to produce the offsprings.

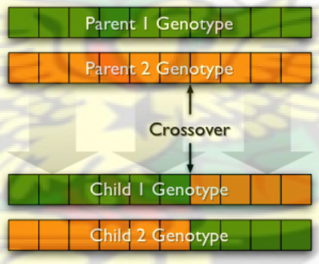

Figure 3.2: A Single point Crossover

Two-point crossover

In this process, two points are selected on the parent organism strings and then everything between the two points is swapped between the parent organisms to produce the offspring.

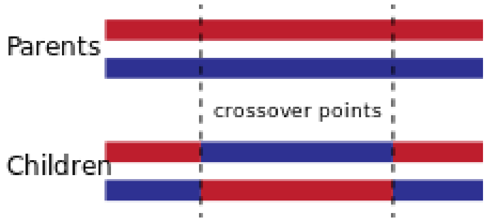

Figure 3.3: Two point crossover

Cut and splice

In this process, there is a change in length of the children strings because each parent string has a separate choice of crossover point to produce the offsprings.

Uniform Crossover and Half Uniform Crossover

The Uniform Crossover uses a fixed mixing ratio between two parents which enables the parent chromosomes to contribute the gene level rather than the segment level. During this process the individual bits in the string are compared between two parents. Lets say we have a mixing ratio to be 0.5, then the offspring has approximately half of the genes from first parent and the other half from second parent. The bits are swapped with a fixed probability.

For the half uniform crossover, exactly half of the non-matching bits are swapped that is the number of bits that are different is calculated and then it is divided by two. Then the resulting number is how many of the bits that do not match between the two parents will be swapped.

Three parent crossover

Here, three parents are chosen randomly and the child is obtained from the three parents. Each bit of first parent is checked with bit of second parent whether they are the same or not. If they are the same then the bit is taken for the offspring otherwise the bit from the third parent is taken for the offspring.

Crossover for Ordered Chromosomes

For the ordered chromosomes a direct swap may not be possible because it depends on how the chromosomes represents the solution. There are so many way that ordered chromosomes can be achieved.

It can be Partially matched crossover, where two crossover points are selected at random and then partially matched crossover is implemented by position wise exchanges which gives a matching selection.

It can also be a Cycle crossover, lets say we have any gene i in the first parent, then the i^{th} gene in the second parent becomes replaced by the one in the first parent. It is then repeated for the displaced gene until the gene which is equal to the first inserted gene becomes replaced.

Order crossover operator can also be done in the sense when a portion of one parent is mapped to a portion of the other parent. From the replaced portion on, the rest is filled up by the remaining genes, where already present genes are omitted and the order is preserved.

Order-based crossover operator can also take place when the individuals are selected from one of the parent and then they are copied to the offsprings. Then the missing individuals are taking from the other parents in order.

Position-based crossover operator could be done when random set of positions that are selected in the parent to produce the offsprings.

It can also be Voting recombination crossover operator where one child is produced from some parent based on a certain threshold value.

Alternating-position crossover operator can also be done when an offspring is created by selecting alternative the next individual of the first parent and the next individual of the second parent ignoring the individuals already present in the offsprings.

Sequential constructive crossover operator is also possible when better edges from the two parents are selected and used it to form the offsprings.

Arithmetic Crossover

It is a crossover operator that combines two parent chromosome vectors in a linear form to produce two new offspring. This equation is used to obtain the arithmetic crossover,

$$\text{Offspring A} = x * \text{Parent A} + (1 - x) * \text{Parent B}.$$

$$\text{Offspring B} = (1 - x) * \text{Parent A} + x * \text{Parent B}.$$

Heuristic Crossover

It is a crossover operator that uses the fitness values of the two parent chromosomes to determine the direction of the search. This equation is used to obtain the heuristic crossover, let α be a random number between 0 and 1 then we have,

$$\text{Offspring A} = \text{ BestParent} + \alpha * (\text{BestParent - WorstParent}),$$

$$\text{Offspring B} = \text{BestParent}.$$

3.6.5 Mutation

It is a genetic operator used to maintain genetic diversity from one generation of population of chromosomes to the other generation. It alters one or more genes in a chromosome from its initial state. The solution can entirely change from the previous one during mutation. The mutation probability should be set low so as to prevent basic random search. Mutation should be allowed in the algorithm in order to avoid local minima by preventing the chromosomes from becoming similar.

Mutations consist of so many types which are;

i. Bit String mutation: This changes several bits in the bit string at random and it depends on the probability that is chosen.

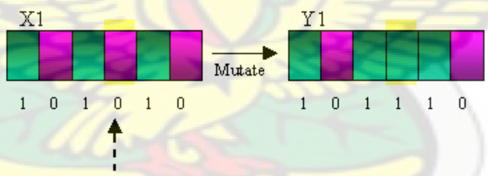

Figure 3.4: A Bit String mutation

ii. Flip Bit mutation: Here, the mutation operator takes the chosen genome and inverts the bit.

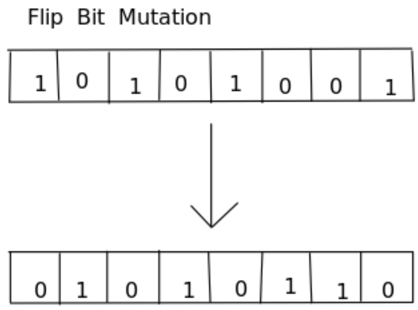

Figure 3.5: A Flip Bit mutation

iii. Boundary mutation: In this type of mutation, the mutation operator replaces the genome with either lower bound or upper bound.

iv. Non-Uniform mutation: Here, the probability that the amount of mutation will go to 0 with the next generation is increased. This keeps the population from remaining stagnant from the early stages of the evolutionary process.

v. Uniform mutation: This replaces the chosen gene with a uniform random value which is selected between what the user chooses for the upper and lower bound for that gene.

vi. Gaussian mutation: In this process, a unit of Gaussian distributed random value is added to the chosen gene. It falls outside of what the user chooses for the lower or upper bounds for that gene, the new gene value is clipped.

3.6.6 Elitism

Elitism can be defined as the most fit individuals that are guaranteed a place in the next generation without undergoing mutation.

In this process, a very successful or slight variant of the general process of constructing a new population is to allow some of the better organisms from the current generation to carry over to the next which is not altered. The more the search space is exploit, the faster the progress of the algorithm but the greater the possibility of the algorithm failing to finally locate the true global optimum or even converge. The search speed can be greatly improved by not losing the elite member that is between the generations. In Elite process, it is required that not only is the elite member selected but also a copy of it does not become disrupted by crossover or mutation. Elitism can take the value of 0 that is if elitism is not applied or 1 that is if elitism is applied.

3.6.7 Holland's schema theorem

Holland's schema theorem states that short, low-order schemata with above-average fitness increase exponentially in successive generations (Holland, 1975).

Let's say we have binary strings of length 6. Then the schema $1*10*1$ describes the set of all strings of length 6 with $1's$ at positions 1, 3 and 6 and a 0 at position 4. This symbol $*$ represents a wildcard symbol (a wildcard is a character that may be substituted for any of a defined subset of all possible characters) meaning positions 2 and 5 can have a value of either 1 or 0.

The order of a schema $O(S)$ of $1*10*1$ is 4 which is defined as the number of fixed positions in the template and also the defining length which is represented by $\delta(S)$ of $1*10*1$ is 5 which is the distance between the first and last specific positions.

The fitness of a schema can be defined as the average fitness of all strings that matches the schema. The fitness of a string is defined as a measure of the value of the encoded problem solution, as computed by a problem-specific evaluation function. The Schema theorem can be expressed as:

$$m(S, t+1) \geq \frac{m(S,t)f(S)}{a_t}\left[1-p\right],$$

where $m(S,t)$ represents the number of strings belonging to schema S at generation t, $f(S)$ is the observed fitness of schema S and a_t is the observed average fitness at generation t. Let p be the probability of disruption which is the probability that crossover or mutation will destroy the schema S can be written as:

$$p = \frac{\delta(S)}{l-1}p_c + O(S)p_m$$

where $O(S)$ is the order of the schema, l is the length of the code, p_m is the probability of mutation and p_c is the probability of crossover. So a schema with a shorter defining length $\delta(S)$ is less likely to be disrupted.
The Schema Theorem is an inequality because the theorem neglects the small, yet non-zero probability that a string belonging to the schema S will be created "from beginning" by mutation of a single string or recombination of two strings that did not belong to the schema S in the previous generation (Holland, 1975).

3.6.8 The Building Block Hypothesis

Heuristic which is described by Goldberg is a process where a short, low order, and highly fit schemata are sampled, crossed over and re-sampled to form strings of potentially higher fitness. When using the building blocks, it reduces the complexity of the problem. Also a better strings from the best partial solutions of past samplings is made. Also genetic algorithm seek near optimal performance through the adjoining of short, low-order, high-performance schemata or building blocks.

The building block hypothesis is a description of a heuristic that performs adaptation by identifying and recombining "building blocks", which means low order, low defining-length schemata with above average fitness.

3.7 The Black Scholes Model

In this project the Black Scholes model will be the other alternative method that will be used to price options. Black Scholes model can be deduced from the mathematical module, that gives a theoretical estimates of the price of a European option. It can also be used to price American options too. This formula has made a great increase in option trading which is widely used around the world by practitioners of option trading. This model has been well accepted and also earned a position among all the financial models.

This model has gone through researches and empirical test and it has been noticed that the price of this model is 'fairly close' to the prices being that have been observed in the financial markets. Black Scholes formula was published by Fischer Black and Myron Scholes in 1973 in the paper known as "The Pricing of Options and Corporate Liabilities" and derived the Black Scholes equations (Black and Scholes, 1973). The Black Scholes model estimates the price of the option over time and this was done to prevent loss in the market. Prevention of loss can be done by means of hedging the option through buying and selling of the underlying asset at the right way and the right time.
Robert C. Merton was the first to publish a paper which expanded the mathematical understanding of the option pricing and created a new expression from the Black Scholes Option pricing model.

3.7.1 Assumption of Black Scholes Model

The Black Scholes formula has some assumptions by Black and Scholes (1973) which are as follows;

- The stock does not pay a dividend or distribution.
- The log returns of the stock price is infinitesimal random walk with drift. It is a Geometric Brownian motion and we assume its drift and volatility is constant
- The short term interest rate is known and constant through time.
- There are no transactions cost in buying or selling the stock or the option.
- It is possible to to borrow any fraction of the price of a security to buy it or to hold it at the short term interest rate.
- There are no penalties in short selling that is it is possible to buy and sell at any amount of the stock.

3.7.2 The Black Scholes Equation

The Black Scholes equation for a European call or put on an underlying stock that does not pay any dividends, is given as;

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0,$$

where V is the price of derivative, r is the risk-free interest rate, σ is the volatility of the stock return.

This means practising how to eliminate risk of gaining or loosing can be done because every loss or gain of the value of the option can be compensated by the opposite gain or loss due to a proportionate change of the value of the underlying asset (Black and Scholes, 1973).

Let C(S,t) be the price of the European call option and let $N(x)$ be the standard normal cumulative distribution function, then it follows that we have,

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}z^2} dz$$

and

$$N'(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{1}{2}x^2}$$

is referred to as the probability density function. Let K be the strike price and t be the time and T be the maturity time or time of expiry (Black and Scholes, 1973). Then the value of a call option that pays no dividend of an underlying stock is;

$$C(S,t) = N(d_1)S - N(d_2)Ke^{-r(T-t)}$$

where,

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right]$$

and

$$d_2 = \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)(T-t)\right]$$

or

$$d_2 = d_1 - \sigma\sqrt{T-t}.$$

Let $P(S,t)$ be the price of a European put option then the value of a European put option is given as;

$$P(S,t) = Ke^{-r(T-t)} - S + C(S,t)$$

or

$$P(S,t) = N(-d_2)Ke^{-r(T-t)} - N(-d_1)S.$$

For the American options, since it can be exercised at any time before or at the expiry date, the Black-Scholes formula becomes,

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV \leq 0.$$

The American call option that pays no dividend is equal to the European call option that pays no dividend but in the case of American put option it is not true (Black and Scholes, 1973).

3.7.3 Derivation of the Black- Scholes Formula

Suppose that the price of the underlying asset follows a geometric Brownian motion, then it follows that we have;

$$\frac{dS}{S} = \rho dt + \sigma dW_t, \tag{3.4}$$

where W_t is the Brownian motion which goes up and down in a random way and it has an expected value of ρdt and a variance of $\sigma^2 dt$. Then we have the payoff of an option depending on the price of the stock and the maturity time that is $V(S,T)$.
Finding its value, we use It$\hat{o}$ lemma then we get;

$$dV = \left(\rho S\frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right) dt + \sigma S \frac{\partial V}{\partial S} dW. \tag{3.5}$$

Suppose we have a portfolio of a short one option and long $\frac{\partial V}{\partial S}$ shares at time t (Black and Scholes, 1973). Then the value of these holding is given as,

$$\alpha = -V + \frac{\partial V}{\partial S} S$$

over $[t, t+\Delta t]$, then from that we get the total profit or loss as,

$$\Delta\alpha = -\Delta V + \frac{\partial V}{\partial S}\Delta S.$$

From equation (3.4) and (3.5) when it is discretized we have,

$$\Delta S = \rho S \Delta t + \sigma S \Delta W$$

and

$$\Delta V = \left(\rho S\frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right)\Delta t + \sigma S \frac{\partial V}{\partial S}\Delta W$$

respectively. Substituting them into $\Delta\alpha$ it follows that we have,

$$\Delta\alpha = \left(-\frac{\partial V}{\partial t} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right)\Delta t.$$

Now let us assume that we have a risk-free rate of return r over $[t, t+\Delta t]$ then it follows that we get;

$$r\alpha\Delta t = \Delta\alpha.$$

Simplifying and solving it further we get,

$$\left(-\frac{\partial V}{\partial t} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right)\Delta t = r\left(-V + S\frac{\partial V}{\partial S}\right)\Delta t$$

then we get,

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0.$$

A random population of stock prices will be generated using the Geometric Brownian Motion model. Then the genetic operators will be applied to it to find the optimal stopping time and the option price using our Genetic Algorithm to price the option when the underlying asset is Geometric Brownian Motion. Also the Black Scholes method will be used to find the option price when the underlying asset is Geometric Brownian Motion. These two methods will be under the same conditions.

Chapter 4

Analysis and Results

4.1 Introduction

Black Scholes model have been seen to be the most popular tool for option pricing in the financial market and the world as a whole. Genetic Algorithm on the other hand have been applied to a different range of problems and that financial practitioners wants to widen the application to solve financial problems.

We will test the performance of Black Scholes model and the performance of Genetic Algorithm to determine the one that gives the minimum price of an American put option under the same conditions. Here, Geometric Brownian Motion model is used to simulate the underlying asset price. For an American option, using Genetic Algorithm, values are assigned to it, giving it a starting time and a stopping time. If it is possible to exercise early then we assign a stopping time in which the option could be exercised.

Suppose, K is the strike price and S_t is the underlying asset price at time t. Then if the strike price exceeds the underlying asset price, that is $K > S_t$, then the investor would exercise a put option. Then we will have the payoff of the exercise put option as,

$$P_t = (K - S_t)^+.$$

If the underlying asset price exceeds the strike price, that is $K < S_t$, then the investor would exercise a call option. Then we will have the payoff of the exercise call option as,

$$P_t = (S_t - K)^+.$$

If we have ρ as the risk-free interest rate, then the present value of the payoff is will be,

$$e^{-\rho t} P_t = e^{-\rho t}(S_t - K)^+.$$

4.2 How to Calculate for the option price

Option investment can turn into massive gains for the investor because it allows the investor to control the profit potential of an investment many times the size of the actual amount the investor has at risk by taking positions on the option market to minimise the risk.

Suppose we assume we have an equally spaced dates which is given as $t_0, t_1, \cdots, t_d$ where $t_0 = 0$, $t_d = T$ in which it is possible to exercise an option. We will also assume that the underlying asset price which is S_t is $\mathcal{F}_t$ measurable and S_t is said to be adapted to $\{\mathcal{F}_t\}_{0 \leq t \leq T}$.

Since $P_t = (K - S_t)^+$ is the payoff from exercising the American put option, then we let $M_t = M(S_t)$ be the option price at time t, then when $t_d = T$, the payoff is $M_T = P_T$. At t_{d-1} the option holder can exercise immediately and get intrinsic value of $P_{t_{d-1}}$ or wait till the maturity time t_d and exercise it. Then the value of the American put option at that time t_{d-1} is the maximum between intrinsic value and the discounted expected future value of the option. This will help in calculating the option price. The option price the holder gets at the time before maturity t_{d-1} is,

$$M_{t_{d-1}} = \max(P_{t_{d-1}}, e^{-\rho \delta t} E[M_{t_d} | \mathcal{F}_{t_{d-1}}]), \tag{4.1}$$

where $\delta t = \frac{T}{d} = t_j - t_{j-1}$ is the length of time from one step to the other. Using backward induction it can be solved as,

$$e^{-\rho t} P_T = e^{-\rho t}(K - S_T)^+,$$

but we have the payoff of the put option at maturity to be $P_T = (K - S_T)^+$, then it follows that we have,

$$\begin{aligned}
\tilde{P}_T &= e^{-\rho t} P_T, \qquad \tilde{M}_{t_d} = e^{-\rho t_d} M_{t_d} \\
\tilde{M}_{t_{j-1}} &= M_{t_{j-1}} e^{-\rho t_{j-1}} \\
\tilde{M}_{t_{j-1}} &= \max(P_{t_{j-1}}, e^{-\rho \delta t} E[M_{t_j} | \mathcal{F}_{t_{j-1}}]) e^{-\rho t_{j-1}} \\
\tilde{M}_{t_{j-1}} &= \max(P_{t_{j-1}} e^{-\rho t_{j-1}}, e^{-\rho(\delta t + t_{j-1})} E[M_{t_j} | \mathcal{F}_{t_{j-1}}]),
\end{aligned}$$

but we have $\delta t = t_j - t_{j-1} \Rightarrow t_j = \delta t + t_{j-1}$ then it follows that,

$$\begin{aligned}
\tilde{M}_{t_{j-1}} &= \max(\tilde{P}_{t_{j-1}}, e^{-\rho t_j} E[M_{t_j} | \mathcal{F}_{t_{j-1}}]) \\
\tilde{M}_{t_{j-1}} &= \max(\tilde{P}_{t_{j-1}}, E[e^{-\rho t_j} M_{t_j} | \mathcal{F}_{t_{j-1}}]) \\
\tilde{M}_{t_{j-1}} &= \max(\tilde{P}_{t_{j-1}}, E[\tilde{M}_{t_j} | \mathcal{F}_{t_{j-1}}]),
\end{aligned}$$

then we get,

$$\tilde{M}_{t_{j-1}} = \max(\tilde{P}_{t_{j-1}}, E[\tilde{M}_{t_j} | \mathcal{F}_{t_{j-1}}]), \qquad j = 1, \cdots, d.$$

$$\Longrightarrow \tilde{M}_T = M_T e^{-\rho T}, \qquad \tilde{P}_T = P_T e^{-\rho T} \Rightarrow \tilde{M}_T = \tilde{P}_T.$$

$$M_0 = \max(P_0, e^{-\rho \delta t} E[M_1 | \mathcal{F}_0]), \qquad \tilde{M}_0 = \max(\tilde{P}_0, E[\tilde{M}_1 | \mathcal{F}_0]) \Rightarrow M_0 = \tilde{M}_0.$$

Then we get $\{M_{t_j}\}_{j=0}^{d}$ to be a supermartingale which implies that

$$E[\tilde{M}_t] = E[\tilde{M}_t|\mathcal{F}_0] \leq M_0, \quad t \in \{t_0, \cdots, t_d\}.$$

4.3 How to Calculate stopping times from a martingale

Suppose τ is the stopping time for an option price then we have $\{\tau \leq t_j\}$: $\{w|\tau(w) \leq t_j\} \in \mathcal{F}_{t_j}, \quad j = 0, 1, \cdots, d$ and also we have $h_{0,T}$ to be the set of all stopping times τ of an option price over the interval $[0, T]$. If $\tau = t_k$ then we have,

$$\tilde{M}_{t_j}^{\tau} := \begin{cases} \tilde{M}_{t_j}, & 0 \leq j < k \\ \tilde{M}_{t_k}, & k \leq j \leq T \end{cases},$$

where $\tilde{M}_{t_j}^{\tau}$ denotes the present value of the option price at time t_j and τ is the time where it is possible to exercise the option. Also, we let $c = \inf\{\tau \in \mathcal{T}_{0,T}|\tilde{M}_\tau = \tilde{P}_\tau\}$ be the optimal stopping time which is the smallest possible time for the exercising the option.

Then it follows that, $\{\tilde{M}_{t_j}^{c}\}_{j=0}^{d}$ is a martingale. Then we have,

$$M_0 = E[\tilde{M}_T^c] = E[\tilde{M}_c] = E[\tilde{P}_c] = \sup_{\tau \in \mathcal{T}_{0,T}} E[\tilde{P}_\tau].$$

The optimal stopping is the smallest time for a path of the simulated underlying asset price $S_t(w_i)$ at the point where the option holder should exercise the option. When the option value equals the intrinsic value then we get the optimal stopping time.

4.4 Using A Genetic Algorithm for Option Pricing

Numerical values were assigned to the parameters to find the value of the option price and then the optimal stopping time of a fixed term American put option when the underlying asset price follows a Geometric Brownian Motion. We used the initial underlying asset price as, $S_0 = 100$, the strike price, $K = 120$, the volatility rate $\sigma = 0.35$, the interest rate, $\rho = 10\%$ and the maturity time $T = 4$ years from now.

A random population of stock prices were generated. A fitness function of $(\max\{K - S_t, 0\})$ was used to select the fittest organisms that will survive the process. Roulette wheel was used to select individuals at random for crossover and mutation. One-point crossover was applied to the individuals selected. Then, flip-bit mutation was applied. The stopping time and the option price were found and calculated.

We note a summary of the Genetic Algorithm to price a fixed term American put option when the underlying asset price is Geometric Brownian motion as follows:

Finding the option price at maturity

1. Generate a random population of Underlying asset prices at maturity time using (3.2)
2. Test each individual for fitness using $(\max\{K - S_t, 0\})$.
3. Using Roulette wheel selection, select individuals from the population generated.
4. Decode each individual into binary form.

5. Select pairs of individuals and crossover (One-point crossover).
6. Apply the mutation operator to every individual (flip-bit mutation).
7. Encode each new generations into real numbers.
8. Set the option value at maturity time to be equal to the intrinsic value at maturity time.

Finding the option price and the optimal stopping time before maturity

9. Calculate for $E = [M_{t_{j+1}}|\mathcal{F}_{t_j}] = \frac{1}{n1}\sum M_{t_{j+1}}$.
10. Calculate for $t_j = j \times \delta t$.
11. Compute the underlying asset price at time t_j, which is before maturity.
12. Test each individual for fitness using $(\max\{K - S_t, 0\})$.
13. Using Roulette wheel selection, select individuals from the population generated.
14. Decode each individual into binary form.
15. Select pairs of individuals and crossover (One-point crossover).
16. Apply the mutation operator to every individual (flip-bit mutation).
17. Encode each new generations into real numbers.
18. Set the option value before maturity time to be equal to the intrinsic value before maturity time.
19. Compute $M_{t_{d-1}} = \max(P_{t_{d-1}}, e^{-\rho\delta t}E[M_{t_d}|\mathcal{F}_{t_{d-1}}]$.
20. If $M_{t_i} = P_{t_j}$, then that is the stopping time.
21. Set $\tau_i = t_j$ to be the stopping time.
22. Calculate for the Option Prices and find the average to obtain the perfect price of the option.

23. Calculate for the stopping times and find the average to obtain the optimal stopping time.

Coding the algorithm above in python 2.7.3 software, the output gives the option price as GH ¢4.12 and the optimal stopping time as 3.6 years. When the equation of the underlying asset (3.2) is used to calculate the underlying asset price at time $\tau = 3.6$,we obtain the the price of the underlying asset as GH¢116.23. This means that at $\tau = 3.6$years, the option seller will make a profit GH¢3.77 since the price at that time at the market will rise to GH¢116.23.

4.4.1 The Zero sum Game

If the holder of the option pays an option price of GH ¢4.12 at $t = 0$ and exercises at time $\tau = 3.6$, then the option holder gets intrinsic value of GH ¢3.77. The seller of the option buys α assets and β bonds at time $t = 0$ with the GH ¢4.12 received.

Using the fixed values which are, $S_0 = 100$, $K = 120$, $\sigma = 0.35$, $\rho = 10\%$,$T = 4$ the option buyer of an American put pays an option price of GH ¢4.12 at $\tau = 3.6$years. At the optimal stopping time $\tau = 3.6$, the underlying asset S_τ = GH ¢116.23 and the intrinsic value $P_\tau = (K - S_\tau)$ = GH ¢3.77.

We have the equation of the bond to be,

$$dX_t = \rho X_t dt,$$

where ρ is the interest rate and X_t is the bond price at time t. When we solve the equation we get,

$$X_t = X_0 e^{\rho t}.$$

Let X_0 be the initial bond at time $t = 0$. At time $t = 0$, the seller of the American put option receives the GH ¢4.12 and invests in α assets and β bonds. Suppose

$X_0 =$ GH ¢1 then we get,

$$\alpha S_0 + \beta X_0 = 4.12.$$

$$\alpha S_{3.6} + \beta e^{3.6\rho} = 3.77.$$

When these two equation are solved we get $\alpha = 0.08$ and $\beta = -3.76$ meaning that the seller of the put option should have bought 0.07 assets and sold 3.76 bonds to get GH ¢3.77 (which is the same as the payoff as the option buyer) at $\tau = 3.6$. Then it becomes a zero sum game that is in this case both benefits.

4.5 Using Black Scholes Model for Option Pricing

We use the same numerical values to find the option price using the Black Scholes model. We note $S_0 = 100$, $K = 120$, $\sigma = 0.35$, $\rho = 10\%$ and using $\tau = 3.6$years which was the optimal stopping time obtained in our Genetic Algorithm, we now use the equation,

$$\text{Put} = N(-d_2)Ke^{-\rho(\tau)} - N(-d_1)S$$

to price the American put option where $N(\cdot)$ is the standard normal cumulative distribution function,

$$d_1 = \frac{1}{\sigma\sqrt{\tau}}\left[\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(\tau)\right]$$

and

$$d_2 = \frac{1}{\sigma\sqrt{\tau}}\left[\ln\left(\frac{S}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)(\tau)\right]$$

We programmed this formula in python 2.7.3 software using equation (3.2) and the formula of the put option we obtained GH ¢11.91 at $\tau = 3.6$ years.

4.6 Comparing Genetic Algorithm and Black Scholes Model

In the Black Scholes Model, the option price was GH ¢11.91 in the 3.6th year whiles the Genetic Algorithm gives an option price to be GH ¢4.12 in the 3.6th year without change in conditions.

This means that it is better to price an American put option using Genetic Algorithm than using Black Scholes model. From the results obtained, early exercise is the best for an American put option when the underlying asset price is Geometric Brownian Motion.

Also any investor will prefer a minimum option price at the under the same conditions than a higher price in option. Option investors should practise the use of Genetic Algorithm to obtain a perfect minimum price in option which will give them the opportunity to gain more and make option pricing more interesting.

Chapter 5

Conclusion and Recommendation

5.1 Conclusion

We used Geometric Brownian motion to simulate the underlying asset. Genetic Algorithm and Black Scholes model were used to calculate the option price and the optimal stopping time of a fixed term American put option with the help of Python 2.7.3 software for the programming. We analysed the performance of Black Scholes model and the performance of Genetic Algorithm to determine the one that gives the minimum price of a fixed term American put option under the same conditions. For an American put option, using Genetic Algorithm, we assigned values giving it a starting time and a stopping time. If it is possible to exercise early then we assign a stopping time in which the option could be exercised.

A Genetic Algorithm has been proposed to price options and find the optimal stopping time. A perfect price of an American put option was obtained using Genetic Algorithm which was lower than the American put option using the Black Scholes model. For American put option exercising early is the best way. The results were strongly encouraging and suggested that the Genetic Algorithm approach performed better than the Black Scholes model.

5.2 Recommendation

After careful analyses of the study, we recommend that further research should be conducted on some other mutations and crossovers applied to option pricing to see the effect. Also Genetic algorithm should be applied to other models (Mean Reversion Model) to obtain the price of other options.

REFERENCES

Baxter, M. and Rennie, A. (1993). *Financial Calculus: An Introduction to derivative pricing*. Cambridge University Press, New York.

Bjerksund, P. and Stensland, G. (1991). *Closed form Valuation of American Options*. Institute of Finance and Management Science.

Black, F. and Scholes, M. (1973). Pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–654.

Chen, Y.-C., Chang, S.-L., and Wu, C.-C. (2010). A dynamic hybrid option pricing model by genetic algorithm and black- scholes model. *World Academy of Science, Engineering and Technology*, 70:715–718.

Chidambaran, N. K., Lee, C.-W. J., and Trigueros, J. R. (1998). An adaptive evolutionary approach to option pricing via genetic programming. *Conference on Computational Intelligence for Financial Engineering.*

Grace, B. K. (2000). Black-scholes option pricing via genetic algorithms. *Applied Economic Letters*, 7:129–132.

Haugh, M. B. and Leonid, K. (2004). Duality theory and approximate dynamic programming for pricing american options and portfolio optimization. *Operations Research*, 52:258–270.

Heigl, A. (2008). Option pricing by means of genetic programming. Master's thesis, Vienna University of Technology.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. The University of Michigan Press.

Hull, J. C. (1997). *Options, Futures, and other Derivatives (4th Edition)*. Printice-Hall international limited,.

Jackson, A. (2000). Genetic algorithms for use in financial problems. *Lend Lease Investment Management*, pages 481–503.

Kaboudan, M. A. (2000). Genetic programming prediction of stock prices. *Computational Economics*, 16:207–236.

Lamberton, D. and Lapeyre, B. (1996). *Introduction to stochastic calculus applied to finance*. Chapman and Hall.

Lazo, J. G., Pacheco, M. A. C., and Vellasco, M. M. B. R. (2003). Real option decision rules for oil field development under market uncertainty using genetic algorithms and monte carlo simulation. *The Seventh Annual Real Options Conference*.

Mcwilliams, N. (2005). Pricing american options using monte carlo simulation. Master's thesis, Summer project.

Oksendal, B. (2003). *Stochastic Differential Equations: An introduction with applications (6th Edition)*. Springer-Verlag, New York,.

Rimcharoen, S., Sutivong, D., and Chongstitvatana, P. (2007). A synthesis of optimal stopping time in compact genetic algorithm based on real options approach. *GECCO*.

Shreve, S. E. (2004). *Stochastic Calculus for finance I and II*. Springer Science and Business Media New York,.

Shu-Cheng and Lee, W.-C. (1997). Option pricing with genetic algorithm: The case of european - style options. *Proceedings of the seventh International Conference on Genetic Algorithms*, pages 704–710.

Sidarto, K. A. (2006). On the calculation of implied volatility using a genetic algorithm. *Seminar Nasional Aplikasi Teknologi Informasi*.

Smith, R. and Hussain, M. S. (2012). Genetic algorithm sequential monte carlo methods for stochastic volatility and parameter estimation. *Proceedings of the World Congress on Engineering*, 1.

Sunzu, J. (2007). American options and optimal stopping problems. Master's thesis, African Institute for Mathematical Sciences.

Tilley, J. A. (1993). Valuing american options in a path simulation model. *TSA*, pages 499–549.

Wu, D. (1997). *Using Genetic Algorithms to Determine Near-Optimal Pricing, Investment and Operating Strategies in the Electric Power Industry.* PhD thesis, University of Pennsylvania.

Yin, Z., Brabazon, A., and O'Sullivan, C. (2007). Adaptive genetic programming for option pricing. *GECCO*, pages 2588–2594.

Appendix A

This Python code is used to simulate the underlying asset price over the period of [0, T] to obtain the Graph.

```
import matplotlib.pyplot as plt
import numpy as np

S0 = 100
mu = 0.1
sigma = 0.35
T=4
dt = 0.001
t = np.linspace(0, T, N)
W = np.random.standard_normal(size = N)
W = np.cumsum(W)*np.sqrt(dt) # standard brownian motion ###
X = (mu-0.5*sigma**2)*t + sigma*W
S = S0*np.exp(X) # geometric brownian motion #
plt.title('SIMULATION OF THE STOCK PRICE WITH TIME')
plt.xlabel('Time (Years)')
plt.ylabel('Underlying asset (St)')
plt.plot(t, S)
plt.show()
```

Appendix B

The code below using python 2.7.3 software price a fixed term American put option using our Genetic Algorithm when the underlying asset price is Geometric Brownian Motion.

```
from __future__ import division
import math
import random
import pylab
import numpy as np
S0=100
K=120
p=0.1
sigma=0.35
T=4
m=4
dt = 0.001
n1=10
ut=[]
at=[]
bt=[]
lt=[]
l=[]
qt=[]
v=[S0]
al=[]
```

```
for i in range (n1):
W = np.random.standard_normal(size = n1)
W = np.cumsum(W)*np.sqrt(dt)
X = (p-0.5*sigma**2)*T + sigma*W
Stj=S0*np.exp(X)
l.append(Stj)
Pt= np.max (K-Stj, 0)
ut.append(Pt)
ints = [int(float(num)) for num in ut]
al.append(ints)
for a in al:
for b in al:
a!=b
for h in a:
x1 = bin(h)
for g in b:
y1 = bin(g)
d1 = list(x1)
d2 = list(y1)


del d1[0:2]
del d2[0:2]
if len(d1)%2 != 0:
d1.insert(0,'0')
if len(d2)%2 != 0:
d2.insert(0,'0')
```

```
offspring1 = []
offspring2 = []
m1 = len(d1)
m11 = m1/2
m111 = int(m11)
m2 = len(d1)
n11 = len(d2)/2
n111 = int(n11)


for i in range(n111):
offspring1.append(d2[i])

for i in range(m111,m2):
offspring1.append(d1[i])



for i in range(m111):
offspring2.append(d1[i])

for i in range(n111,len(d2)):
offspring2.append(d2[i])

stroffspring1 = ""
stroffspring2 = ""
strflip1 = ""
strflip2 = ""
flipoffspring1 =[]
flipoffspring2 = []
```

```
for x in offspring1:
stroffspring1 = stroffspring1 + x

for x in offspring2:
stroffspring2 = stroffspring2 + x


for i in range(len(offspring1)):
if offspring1[i] =="0":
flipoffspring1.append("1")
elif offspring1[i] =="1":
flipoffspring1.append("0")

for i in range(len(offspring2)):
if offspring2[i] =="0":
flipoffspring2.append("1")
elif offspring2[i] =="1":
flipoffspring2.append("0")

for x in flipoffspring1:
strflip1 = strflip1 + x


for x in flipoffspring2:
strflip2 = strflip2 + x


intoffspring1 = int(stroffspring1, base =2)
```

```
intoffspring2 = int(stroffspring2, base =2)


intflip1 = int(strflip1, base =2)

intflip2 = int(strflip2, base =2)


Pts= max (intflip1,intflip2)

lt.append (Pts)

li=[]

uti=[]

ali=[]

#stopping time

tau=[T for i in range(n1+1)]

for j in reversed(range(m)):

E=sum(lt)/(len(lt))

for i in range (n1):

tj=j*dt

W = np.random.standard_normal(size = n1)

W = np.cumsum(W)*np.sqrt(dt)

X = (p-0.5*sigma**2)*tj + sigma*W

Stj1=S0*np.exp(X)

li.append(Stj1)

Pt= np.max (K-Stj1, 0)

uti.append(Pt)

ints1 = [int(float(num)) for num in uti]

ali.append(ints1)

for aa in ali:



for bb in al:
```

```
aa!=bb
for hh in aa:
x11 = bin(hh)
for gg in bb:
y11 = bin(gg)
d11 = list(x11)
d22 = list(y11)


del d11[0:2]
del d22[0:2]
if len(d11)%2 != 0:
d11.insert(0,'0')
if len(d22)%2 != 0:
d22.insert(0,'0')

offspring11 = []
offspring22 = []
m11 = len(d11)
m111 = m11/2
m1111 = int(m111)
m22 = len(d11)
n111 = len(d22)/2
n1111 = int(n111)


for i in range(n1111):
offspring11.append(d22[i])
```

```
for i in range(m1111,m22):
offspring11.append(d11[i])



for i in range(m1111):
offspring22.append(d11[i])

for i in range(n1111,len(d22)):
offspring22.append(d22[i])

stroffspring11 = ""
stroffspring22 = ""
strflip11 = ""
strflip22 = ""
flipoffspring11 =[]
flipoffspring22 = []

for xi in offspring11:
stroffspring11 = stroffspring11 + xi

for xi in offspring22:
stroffspring22 = stroffspring22 + xi



for i in range(len(offspring11)):
if offspring11[i] =="0":
flipoffspring11.append("1")
elif offspring11[i] =="1":
flipoffspring11.append("0")
```

```
for i in range(len(offspring22)):
if offspring22[i] =="0":
flipoffspring22.append("1")
elif offspring22[i] =="1":
flipoffspring22.append("0")

for xi in flipoffspring11:
strflip11 = strflip11 + xi


for xi in flipoffspring22:
strflip22 = strflip22 + xi


intoffspring11 = int(stroffspring11, base =2)
intoffspring22 = int(stroffspring22, base =2)

intflip11 = int(strflip11, base =2)
intflip22 = int(strflip22, base =2)

Pts1= max (intflip11,intflip22)

lt[i]=max(Pts1,math.exp (-p*dt)*E)
if lt[i]==Pts1:
tau[i]=tj

print  'the optimal stopping time is', sum(tau)/n1
print 'the option price is',sum(lt)/(len(lt))
```