Combinatorial Methods for Double-Barrier Option Pricing

Kun-Yuan Chao
Department of Computer Science and Information Engineering
National Taiwan University

Contents

List of Figures

Abstract

The space of financial innovation continues unabated. In particular, numerous variations on standard option payoffs have been proposed and traded in the over-the-counter market. Barrier options represent one of the more popular forms of those so-called *exotic* options. A double barrier option is an option which is knocked in or out the first time that the underlying asset touches one of two barriers prior to expiration.

The main purpose of this thesis is to use the combinatorial method to value double barrier options by the use of reflection principle. With this method, the convergence speed in the double-barrier case is faster than the single-barrier case. We implement algorithms in the C language and compile them by the Sun CC compiler running under Solaris 5.6. The data are generated on the **UltraSparc II** workstation with 300 MHz CPU and 256 MB DRAM. The resulting algorithm is highly efficient for pricing this type of barrier option, which is at least an order of magnitude faster than the binomial tree method with backward induction. The binomial tree method with backward induction takes about 10 times than the combinatorial method with identical parameters. The combinatorial method is clearly superior.

Chapter 1

Introduction

1.1 Puts and Calls

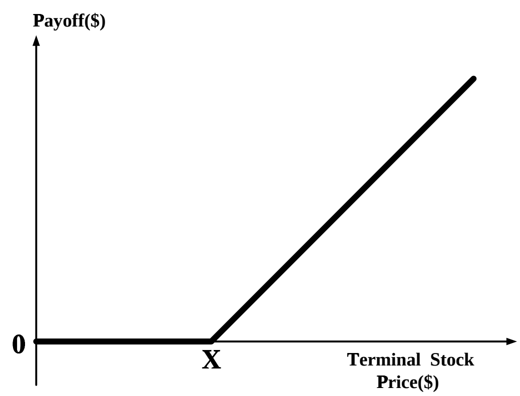

Figure 1.1: Call Option.

Put and call are two basic types of financial options. They were first traded on an organized exchange in 1973. A call option gives the holder the right to buy the underlying asset by a certain date called the expiration date for a certain price. A put option gives the holder the right to sell the underlying asset by a certain date for a certain price. This certain price is called the strike price. Options can be classified as American or European. All the options discussed in the thesis are European, which can be exercised only on the exercise date but not before, namely not early exercise.

Let X be the strike price and S_T the final price of the underlying asset. The payoff from holding a European call option is

$$\max(S_T - X, 0).$$

This is because, if $S_T > X$, the holder will exercise and receive $S_T - X$ in effect, while if $S_T \leq X$, the holder will not exercise and the call is worthless (Figure 1.1).

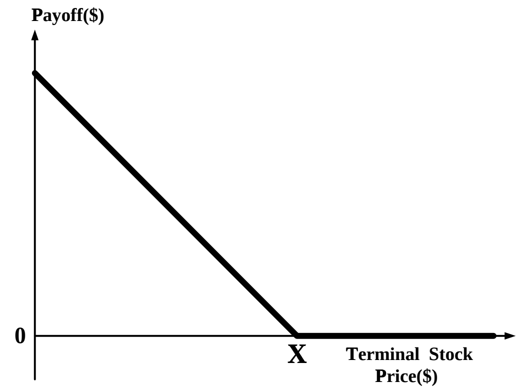

Figure 1.2: PUT OPTION.

Similarly, the European put option has a terminal value of

$$\max(X - S_T, 0).$$

This is because, if $X > S_T$, the holder will exercise the right and receive $X - S_T$ in effect, while if $X \leq S_T$, the holder will not exercise the right and the put option is worthless (Figure 1.2).

1.2 Barrier Options

Barrier option is a kind of exotic path-dependent option. It differs from a standard option in that it may expire before expiration or come into existence before expiration, depending on the specification, when certain barriers are hit. In this thesis, barrier price is constant, it never move during the option's life.

There are two types of barrier options: knock-out and knock-in options. A knock-out option is similar to a standard option except that, when the underlying asset's price reaches a certain barrier H, the option ceases to exist. In the case of a call knock-out, the barrier is generally below the strike price ($H < X$). This option is sometimes referred to as a down-and-out option. In the case of a put knock-out with $H > X$, the option is sometimes referred to as an up-and-out option. The knock-in option is similar. With reference to Figure 1.3, a down-and-in option is a call that comes into existence only when the barrier H ($H < X$) is reached, and an up-and-in option is a put that comes into existence only when the barrier H ($H > X$) is reached.

Generally speaking, the price of a barrier option is cheaper than a standard option. This is because additional conditions are imposed on the barrier option (i.e., stock price must either touch or not touch the barrier during the option's life for it to be eligible for exercise at expiration). An important issue is the frequency with which the asset price S is observed for the purpose of testing whether the barrier has been reached. Often the terms of a contract state that S is observed once a day.

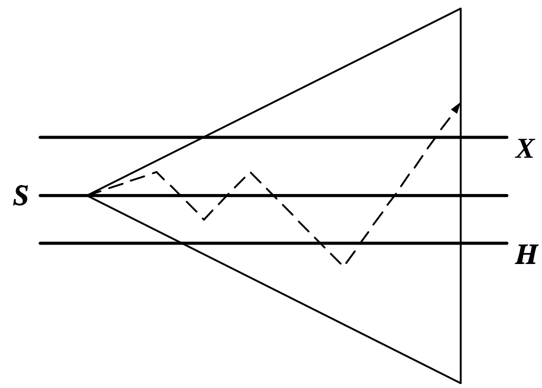

Figure 1.3: Down-and-In Call and Binomial Tree.

In this thesis, we propose a combinatorial method to price barrier options is due to Lyuu [9]. We then compare its performance against that of the standard backward implementation. Numerical experiments show that Lyuu's algorithm is substantially better.

1.3 Double-Barrier Options

Some barrier options contain two barriers H and L with $L < H$ (See Figure 1.4). Depending on how the barriers affect the existence of the options, various barrier options can be defined. For instance, the option may come into existence only if both barriers are hit.

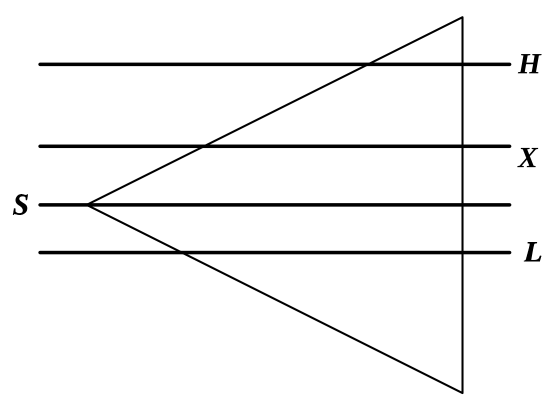

Figure 1.4: Double-Barrier Option and Binomial Tree.

We will restrict attention here to the case that the option comes into existence only if the underlying asset touches either barrier, and call it double barrier call if the option is a call. Other variations include options that come into existence only if both barriers are hit and options that knock-out when one or both barriers are hit. These variations can all be solved by the same method. We shall demonstrate in this thesis that the reflection principle coupled with the inclusion-exclusion principle can be applied to derive combinatorial formulae for double barrier options. These formulae load to efficient algorithms.

1.4 Structure of This Thesis

There are some chapters in this thesis. We introduce the basic concepts and explain the ideas in details in this Chapter. In order to understand the pricing methodology, we also add background information about finance and mathematics and computer science in Chapter 2. The kernel of the thesis, the combinatorial method and the binomial tree method with backward induction used in pricing double barrier options, appears in Chapter 3. We then have computational results about the combinatorial method and compare the accuracy and efficiency of these two methods in Chapter 4. Finally, we give conclusions and future work in Chapter 5.

Chapter 2

Some Backgrounds

2.1 The Black-Scholes Option Pricing Model

The year 1973 was a milestone in finance. In the year, the Chicago Board Options Exchange was founded and became the first organized facility for options trading. Also, two professors at the Massachusetts Institute of Technology, Fischer Black and Myron Scholes, published an article in the *Journal of Political Economy* that contained for option pricing formulae. The celebrated Black-Scholes option pricing model was one of the most significant developments in the pricing of financial instruments. The mathematics of this formula's derivation is quite complex, so we shall omit it here. See [5] for more detailed information. We review the model's assumptions below.

1. The stock price follows the log-normal distribution. This means that the logarithm of the stock price follows the normal distribution. The log-normal distribution is a convenient and realistic characterization of stock price because it reflects stockholders' limited liability.
2. There are no taxes or transaction costs.
3. There are no dividends during the life of the option.
4. There are no risk-less arbitrage opportunities.
5. The risk-free rate of interest, r, is constant.
6. The options are European.

Black and Scholes derived the following formula for the call option:

$$C = SN(d_1) - Xe^{-r_c\tau}N(d_2),$$

and the following formula for the put option:

$$P = Xe^{-r_c\tau}N(-d_2) - SN(-d_1)$$

where

$$d_1 = \frac{\ln(S/X) + (r_c + \sigma^2/2)\tau}{\sigma\sqrt{\tau}}$$

$$d_2 = d_1 - \sigma\sqrt{\tau}$$

$N(x)$ = cumulative normal probability
σ^2 = annualized variance of the continuously compounded return on the stock
r_c = continuously compounded risk-free rate
S = current stock price
X = strike price
τ = time to expiration of an option

2.2 The Merton Barrier Option Pricing Model

The value of a European down-and-in call is

$$Se^{-q\tau}(H/S)^{2\lambda}N(x) - Xe^{-r_c\tau}(H/S)^{2\lambda-2}N(x - \sigma\sqrt{\tau})$$

where

$$x = \frac{\ln[H^2/(SX)] + (r_c - q + \sigma^2/2)\tau}{\sigma\sqrt{\tau}}$$

$$\lambda = \frac{r_c - q + \sigma^2/2}{\sigma^2}$$

where $S \geq H$; q is the stock's dividend yield; and τ is the time to maturity.

Equation above assumes that the underlying asset price follows geometric Brownian motion and is due to Merton [10]. A European down-and-out call can be priced via the in-out parity relation. The value of a European up-and-in put is

$$Xe^{-r_c\tau}(H/S)^{2\lambda-2}N(-x + \sigma\sqrt{\tau}) - Se^{-q\tau}(H/S)^{2\lambda}N(-x)$$

Although closed-form solutions exist, the study of numerical methods based on binomial models is still useful for the new insights it brings. It also has applications to exotic options where the terminal payoff function is non-standard, and closed-form solutions are hard to come by.

2.3 The Wiener Process

Models of stock price are usually in terms of the Wiener process. The Wiener process is a particular type of Markov stochastic process. It has been used before in physics to describe the motion of a particle that is subject to a large number of small molecular shocks and is sometimes referred to as normalized Brownian motion.

A process, z, which follows the Wiener process can be understood by considering the changes in its value in small intervals of time. Let $\triangle t$ be the length of a small interval of time and $\triangle z$ be the change in z during $\triangle t$. To follow the Wiener process, there are two basic properties $\triangle z$ must satisfy

Property 1.

If z is a variable which follows the Wiener process, then $\triangle z$, the change of variable z during a small interval of time $\triangle t$, satisfies

$$\triangle z = \epsilon\sqrt{\triangle t}$$

where ϵ is a random drawing from the standardized normal distribution.

Property 2.

The value of $\triangle z$ for any two different short intervals of time $\triangle t$ are independent. By property 1, $\triangle z$ itself has a normal distribution, while Property 2 implies that z follows a Markov process. The stock price under the Black-Scholes model then follows

$$dS = S\mu dt + S\sigma dz.$$

This process is called the geometric Brownian motion [8]. It was used to structure the binomial model in next section.

2.4 The Binomial Model

First, we review the discrete-time approximation to the geometric Brownian motion, $dS = S\mu dt + S\sigma dz$. For brevity, we use S in place of $S(0)$, the current stock price.

Consider the stock price $\triangle t$ time from now (time zero) $S(\triangle t)$. Under the geometric binomial random walk model, in a period of $\triangle t$, the stock price either increases to Su with probability p or decreases to Sd with probability $1-p$ (See Figure 2.1).

First, we impose that

$$E[S(\triangle t)] = Se^{\mu\triangle t} \text{ and } Var[S(\triangle t)] = S^2(e^{\triangle t\sigma^2}-1)e^{2\triangle t\mu} \rightarrow S^2\sigma^2 \triangle t.$$

The expected stock price at time $\triangle t$ is given by

$$pSu + (1-p)Sd.$$

Hence, our first requirement is the equality of expected price,

$$pSu + (1-p)Sd = Se^{\mu\triangle t}.$$

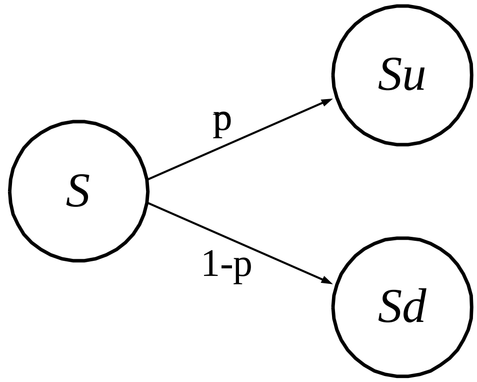

Figure 2.1: BINOMIAL MODEL. After $\triangle t$, the stock price S either moves to Su with probability p or Sd with probability $1-p$.

The variance of the binomial stock price at $\triangle t$ is given by

$$p(Su)^2 + (1-p)(Sd)^2 - (Se^{\mu \triangle t})^2.$$

So our second requirement is the equality of variance,

$$p(Su)^2 + (1-p)(Sd)^2 - (Se^{\mu \triangle t})^2 = S^2 \sigma^2 \triangle t.$$

or

$$pu^2 + (1-p)d^2 - e^{2\mu \triangle t} = \sigma^2 \triangle t.$$

Imposing $ud = 1$, we can get the solution

$$u = e^{\sigma \sqrt{\triangle t}}, d = e^{-\sigma \sqrt{\triangle t}} \text{ and } p = \frac{e^{\mu \triangle t} - d}{u - d}.$$

In a risk-neutral economy, $\mu = r$ and

$$p \to (1/2) + (1/2) \frac{r - \sigma^2/2}{\sigma} \sqrt{\triangle t}.$$

If we partition the time to expiration T into n periods, then $\triangle t = \frac{T}{n}$. See [8] for more detailed analysis.

Chapter 3

Combinatorial Methods

3.1 The Binomial Tree Method with Backward Induction

The binomial method is widely used in pricing options. A standard algorithm for the binomial model is the binomial tree method with backward induction. This method values the option by running from the terminal nodes backward in time through the tree. The value of double barrier options can be computed by this method as follows. We will use the double barrier call as an example.

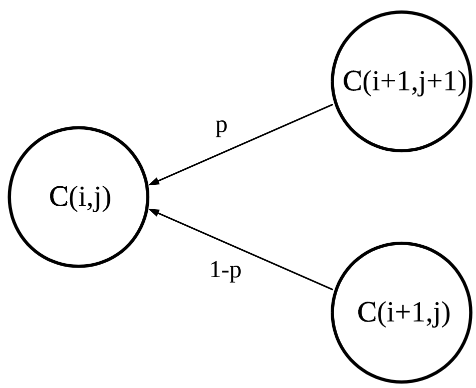

Figure 3.1: Backward Induction for the Binomial Model. The first parameter inside the parentheses denotes time, while the second denotes state.

Let $C(i,j)$ denote the call value at time i and state j. Given $C(i+1,j+1)$ and $C(i+1,j)$, backward induction for the binomial model (see Figure 3.1) says $C(i,j)$ is

$$e^{-r\triangle t}(pC(i+1,j+1)+(1-p)C(i+1,j))$$

if barriers are not involved. If out state j corresponds to a stock price of H or L, then $C(i,j)$ should be zero instead. This incidentally shows the value of the double barrier call at (i, j) is equal to that of the standard call minus that of the double barrier knock-out call via the in-out parity. Backward induction will eventually reach the initial node at $(0, 0)$, and our double barrier call is priced.

3.2 The Reflection Principle

A tool we should know to understand the combinatorial method is the reflection principle [9].

Imagine a particle starts at position $(0, -a)$, on the integral lattice and wishes to reach $(n, -b)$. Without loss of generality, assume $a, b \geq 0$. The particle is constrained to move to $(i+1, j+1)$ or $(i+1, j-1)$ from (i, j), the very way the price under the binomial model is supposed to evolve:

$(i, j) \rightarrow (i+1, j+1)$ associated with the up move, $S \rightarrow Su$
$(i, j) \rightarrow (i+1, j-1)$ associated with the down move, $S \rightarrow Sd$

How many such paths can the particle take that touch or cross the x-axis? This question can be rephrased as a variant of the ballot problem. Given that a candidate starts with a fewer votes than the opponent (which is not uncommon in many parts of the world) and ends up with b fewer votes, how many ways can the votes be counted in which the winner's vote count equals the opponent's at least once?

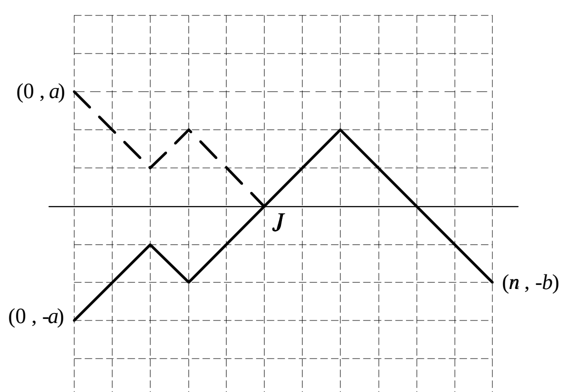

Figure 3.2: THE REFLECTION PRINCIPLE FOR BINOMIAL RANDOM WALKS.

Consider any legitimate path from $(0, -a)$ to $(n, -b)$ that either touches or crosses the x-axis. Let J denote the first position it happens. By reflecting the portion of the path from $(0, -a)$ to J, a path from $(0, a)$ to $(n, -b)$ is thus constructed. Note that this path crosses the x-axis at J. See Figure 3.2 for illustration. A moment's reflection leads to the conclusion that the number of paths from $(0, -a)$ to $(n, -b)$ that touch the x-axis is exactly the number of paths from $(0, a)$ to $(n, -b)$. This is the celebrated reflection principle of André (1840-1917) published in 1887 (Lint and Wilson[7]) [9].

Since any such path consisting of n moves must have $b + a$ more down moves ("-1"s) than up moves ("+1"s), the desired number equals the number of ways to permute $(n - a - b)/2$ "+1"s and $(n + a + b)/2$ "-1"s, which is equal to

$$\binom{n}{\frac{n+a+b}{2}} \text{ for even, non-negative } n + a + b$$

and zero otherwise. The negative $n + a + b$ case can be disregarded under the convention,

$$\binom{n}{k} = 0 \text{ for } k < 0 \text{ or } k > n$$

3.3 The Idea of Combinatorial Methods

An alternative pricing method uses combinatorics. Counting the number of valid paths that lead to a particular terminal price is the idea behind the highly efficient combinatorial method to price European double barrier options. Its performance and accuracy will be documented in Chapter 4. We derive below combinatorial formulae for options that come into existence only if either barrier is hit.

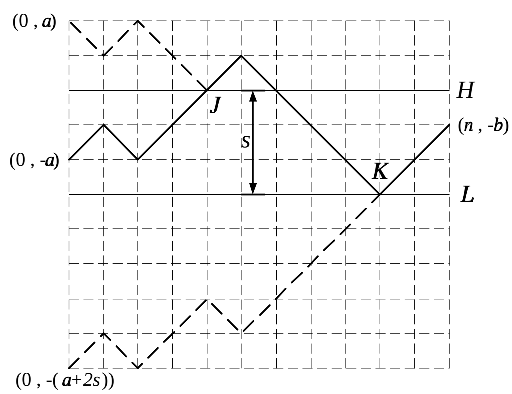

Figure 3.3: Repeated Applications of the Reflection Principle. The random walk from $(0, -a)$ to $(n, -b)$ must hit either barrier, and there must exists an L-hit preceded by H-hit. In counting the number of such walks, reflect the path first at J and then at K.

Like the reflection principle described earlier, consider a particle starts at position $(0, -a)$ on the integral lattice and is destined for $(n, -b)$ which lies between the two barriers. Without loss of generality, assume $a, b \geq 0$. As before, the particle is constrained to move to $(i + 1, j + 1)$ or $(i + 1, j - 1)$ from (i, j). We claim that the number of paths in which a hit of the x-axis (i.e., the H-line) appears before a hit of the L-line $x = -s$ is

$$\binom{n}{\frac{n+a-b+2s}{2}} \text{ for even, non-negative } n+a-b \tag{3.1}$$

In the above, we assume $s > b$ and $s > a$ to make both barriers effective.

We prove (3.1) with reference to Figure 3.3. Consider any legitimate path from $(0, -a)$ to $(n, -b)$ that hits H. Let J denote the first position this happens. (The path may have hit the L-line earlier.) By reflecting the portion of the path from $(0, -a)$ to J, a path from $(0, a)$ to $(n, -b)$ is thus constructed. Note that this path hits H at J. The number of paths from $(0, -a)$ to $(n, -b)$ in which an L-hit is preceded by an H-hit is exactly the number of paths from $(0, a)$ to $(n, -b)$ that hits L. The desired number is thus as claimed by applying the reflection principle.

Equation (3.1) can be generalized. Let A_i denote the set of paths that hit the barriers with a sequence that contains $\overbrace{H^+L^+H^+\cdots}^{i}$ with $i \geq 1$. Here, L^+ denotes a sequence of Ls, and H^+ denotes a sequence of Hs. For instance, a path with the hit pattern $LLHLLHH$ belongs to A_3. Similarly, let B_i denote the set of paths that hit the barriers with a sequence that contains $\overbrace{L^+H^+L^+\cdots}^{i}$ with $i \geq 1$. Note that $A_i \bigcap B_i$ may not be empty. The number of paths that hit either barrier now equals

$$\mathbf{N}(a, b, s) = \sum_{i=1}^{n} (-1)^{i+1} (|A_i| + |B_i|). \tag{3.2}$$

The value of the double barrier call is now within reach. Let us first take care of degenerate cases. If $S \leq L$, then the double barrier call is reduced to a standard call. Similarly, if $S \geq H$, then the double barrier call is reduced to a knock-in call with a single barrier H. So we will assume $L < S < H$ from now on. Under this assumption, it is again easy to check that the double barrier option is reduced to simpler options unless $L < X < H$. So we further assume $L < X < H$. Let

$$a = \left\lceil \frac{\ln(X/(Sd^n))}{\ln(u/d)} \right\rceil = \left\lceil \frac{\ln(H/S)}{2\sigma\sqrt{\Delta t}} + \frac{n}{2} \right\rceil$$

$$h = \left\lceil \frac{\ln(H/(Sd^n))}{\ln(u/d)} \right\rceil = \left\lceil \frac{\ln(H/S)}{2\sigma\sqrt{\Delta t}} + \frac{n}{2} \right\rceil$$

$$l = \left\lfloor \frac{\ln(L/(Sd^n))}{\ln(u/d)} \right\rfloor = \left\lfloor \frac{\ln(L/S)}{2\sigma\sqrt{\Delta t}} + \frac{n}{2} \right\rfloor$$

The barriers will be replaced by the effective barriers $\tilde{H} \equiv Su^h d^{n-h}$ and $\tilde{L} \equiv Su^l d^{n-l}$.

Note that by using (3.2), these terminal nodes were supposed to lie between $\tilde{L}$ and $\tilde{H}$ (inclusive). These terminal nodes together contribute

$$A \equiv \hat{R}^{-n} \sum_{j=a}^{n} \mathbf{N}(2h-n, 2h-2j, 2(h-l)) p^j (1-p)^{n-j} (Su^j d^{n-j} - X) \tag{3.3}$$

to the option value by the risk-neutral methodology, where $\hat{R} = e^{\hat{r}} = e^{rT/n}$ is the riskless return per period. Here, apply the reflection principle repetitively to calculate A_i and B_i as

$$A_i = \begin{cases} \binom{n}{\frac{n+a+b+(i-1)s}{2}} & \text{for odd i} \\ \binom{n}{\frac{n+a-b+is}{2}} & \text{for even i} \end{cases}$$

$$B_i = \begin{cases} \binom{n}{\frac{n-a-b+(i+1)s}{2}} & \text{for odd i} \\ \binom{n}{\frac{n-a+b+is}{2}} & \text{for even i} \end{cases}$$

for even $n - a + b$ [9].

Equation (3.3) is the basis for our combinatorial method. We comment that, in general, the combinatorial method implies an algorithm that uses $O(n^2)$ arithmetic operations. This is in contrast to the single-barrier case, which allows an algorithm that uses only $O(n)$ operations [9].

See Figure 3.4 for the choice of arguments in $\mathbf{N}(\cdot)$ above. As for these terminal nodes outside the range, they constitute a standard call with a strike price of $\tilde{X}$. Let its value be D. The double barrier call thus has value $A + D$.

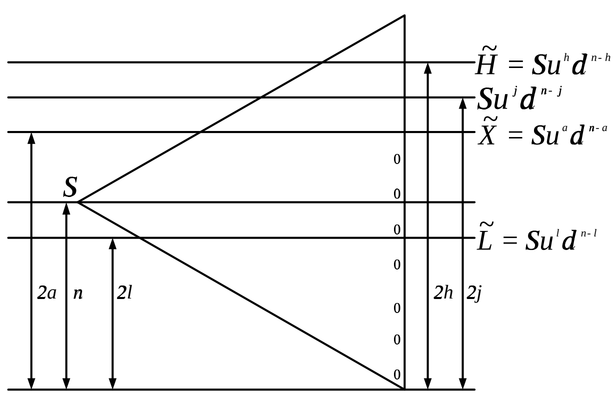

Figure 3.4: Barrier Call Option with Two Barriers under Binomial Model.

Chapter 4

Computational Results

4.1 Convergence Comparison

This Chapter compares the combinatorial method and the binomial tree method with backward induction in terms of their actual performance when realized as programs. Besides, there is a result worth to note. Let's first see the comparison between single-barrier and double-barrier cases.

The results in Figure 4.1 and Figure 4.2 work with identical parameters except that the double-barrier case has an additional barrier ($H = 120$). The convergence speed in the double-barrier case seems faster than the single-barrier case, as can be easily confirmed by looking at the figures.

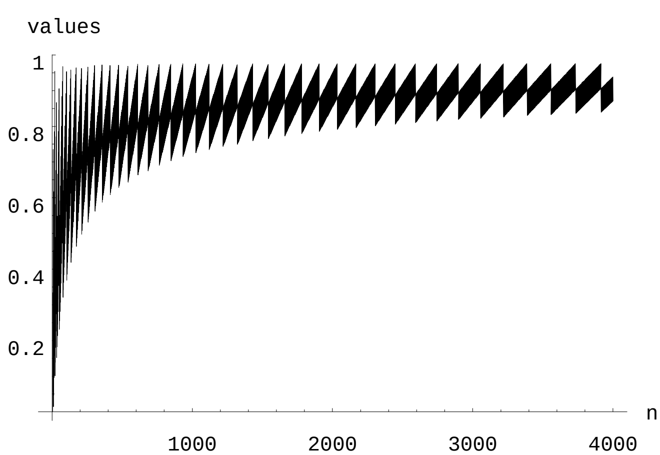

Figure 4.1: SINGLE-BARRIER CALL WITH COMBINATORIAL METHODS. The option uses $S = 95$, $Barrier = 80$, $X = 97$, $\sigma = 0.25$, $T = 1$ (year), $r = 15\%$, $q = 5\%$ (continuously compounded). The analytical value is about 0.9760.

The option values of the combinatorial method oscillate as we increase number of time periods n, because the method is based on the binomial model [3]. It leads to a sawtooth-like convergence. As with the binomial option pricing method, the main reason of the swings is that the barrier lays between the binomial lattices. Our choices of a, h and l introduce specification errors in that effective exercise price, $Su^a d^{n-a}$, and effective barriers, $Su^h d^{n-h}$ and $Su^l d^{n-l}$, are not exactly the original exercise price, X, and barriers, H and L.

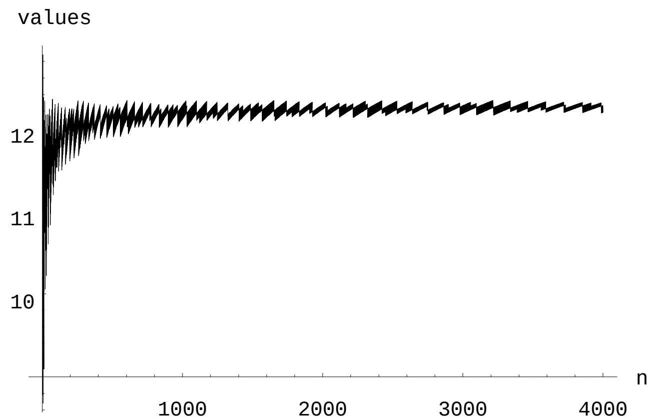

Figure 4.2: Double-Barrier Call with Combinatorial Methods. The option uses $S = 95$, $H = 120$, $L = 80$, $X = 97$, $\sigma = 0.25$, $T = 1$ (year), $r = 15\%$, $q = 5\%$ (continuously compounded). The analytical value is about 12.30.

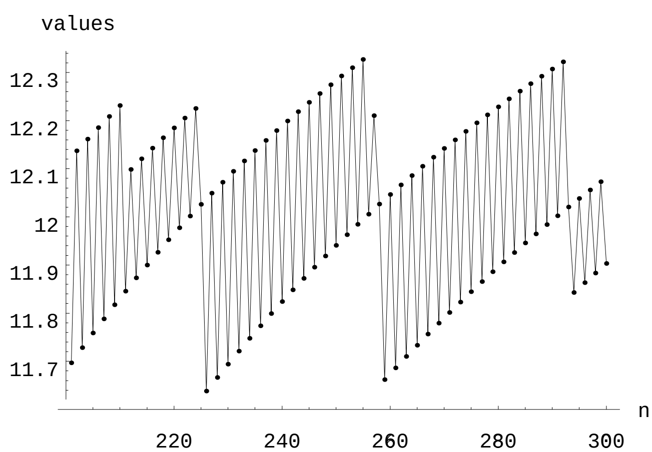

Figure 4.3: Double-Barrier Call with Combinatorial Methods (Magnified). The number of periods runs from 201 to 300 in Figure 4.2.

Let's next see the accuracy of the combinatorial method and the backward induction method. The backward induction method has been introduced in section 3.1. See Figure 4.4 for computational results of the binomial tree method with backward induction.

The result in Figure 4.4 works with identical parameters in Figure 4.2. As expected, the convergence rate of the combinatorial method is slightly slower than backward induction at the beginning. The reason is that we used the original L and

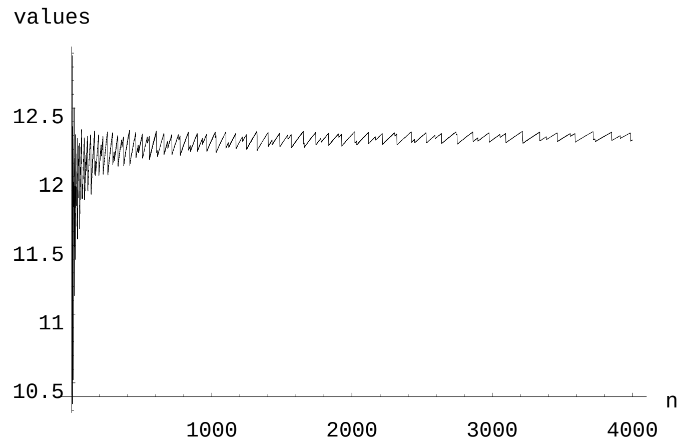

Figure 4.4: DOUBLE-BARRIER CALL WITH BACKWARD INDUCTION. The option uses $S = 95$, $H = 120$, $L = 80$, $X = 97$, $\sigma = 0.25$, $T = 1$ (year), $r = 15\%$, $q = 5\%$ (continuously compounded). The analytical value is about 12.30.

H for the backward induction scheme. The combinatorial method and the binomial tree backward induction algorithm do converge as n increases.

Basically, the principle of the combinatorial method is identical with the backward induction method, they are structured with binomial mode. The computations of these two methods are dependent on the value of worthy paths on binomial tree. Understandably, the computational result of the backward induction method will closely approximate to that of the combinatorial method while the backward induction method also uses effective barriers, $Su^h d^{n-h}$ and $Su^l d^{n-l}$, are not exactly the original barriers, H and L. Let's see some computational results of these two methods in Table 4.1.

NUMBER OF PARTITIONS	COMBINATORIAL	BACKWARD INDUCTION
1	14.602622	14.602622
2	12.480741	12.480741
3	8.780143	8.780143
4	12.882859	12.882859
5	10.896411	10.611719
6	8.680337	8.680337
7	11.851448	11.851448
8	10.362529	10.362529
.	.	.
.	.	.
.	.	.
4000	12.268334	12.268334

Table 4.1 SOME COMPUTATIONAL RESULTS OF THE TWO METHODS.

These two methods use effective barriers, $Su^h d^{n-h}$ and $Su^l d^{n-l}$.

Visibly, most computational results of these two methods are identical while both use effective barriers, $Su^h d^{n-h}$ and $Su^l d^{n-l}$. See Figure 4.5 for more detailed computational results of backward induction method with effective barriers, $Su^h d^{n-h}$ and $Su^l d^{n-l}$.

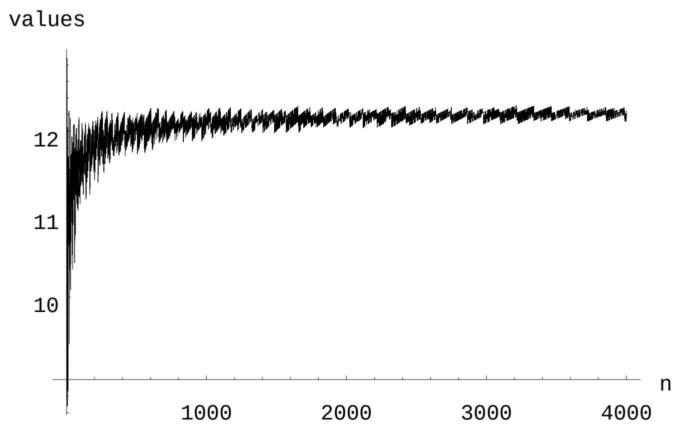

Figure 4.5: Double-Barrier Call with Backward Induction (Modified). The original barriers, H and L, are adjusted to effective barriers, $Su^h d^{n-h}$ and $Su^l d^{n-l}$. The option uses $S = 95$, $H = 120$, $L = 80$, $X = 97$, $\sigma = 0.25$, $T = 1$ (year), $r = 15\%$, $q = 5\%$ (continuously compounded). The analytical value is about 12.30.

4.2 Quadratic Running Times

We implement algorithms in the C language and compile them by the Sun CC compiler running under Solaris 5.6. The data are generated on the **UltraSparc II** workstation with 300 MHz CPU and 256 MB DRAM. See Figure 4.6 for quadratic running times of the combinatorial method.

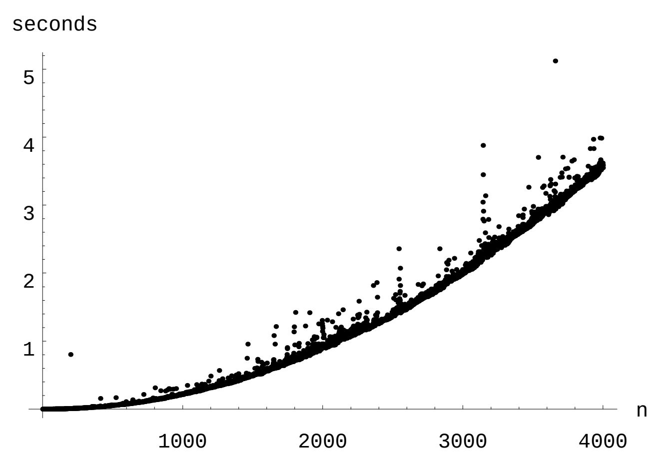

Figure 4.6: Quadratic Running Times. The option uses $S = 95$, $H = 120$, $L = 80$, $X = 97$, $\sigma = 0.25$, $T = 1$ (year), $r = 15\%$, $q = 5\%$ (continuously compounded).

Different machines may lead to different running times. The calculating time of the number of paths that hits either barrier for any terminal node is proportional to n, while the number of terminal nodes is proportional to n. Consistent with our theoretical analysis, the running time is quadratic in n.

The true comparison of algorithms should be based on the total running time. The combinatorial method has an edge here. Running times behind the data in Figure 4.4 are plotted in Figure 4.7.

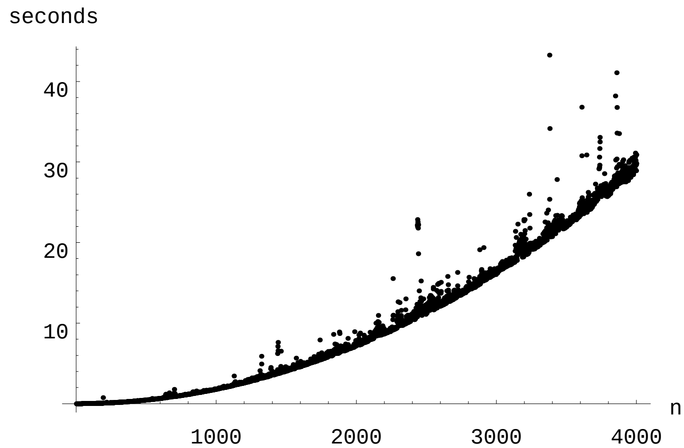

Figure 4.7: THE PERFORMANCE WITH BACKWARD INDUCTION. It takes much more time than the combinatorial method for the same number of periods.

The total time used by the combinatorial method is far less than that by the binomial tree method with backward induction algorithm. In the above example, the combinatorial method arrives at penny accuracy at n=3985 with a running time of 3517 ms, while the backward induction method takes about 10 times as much time. The combinatorial method is clearly superior.

4.3 More Discussions

1. In this thesis, we mainly consider double barrier knock-in option that comes into existence if *either barrier* is hit, and we use

$$N(a, b, s) = \sum_{i=1}^{n} (-1)^{i+1} (|A_i| + |B_i|)$$

to count the number of paths that lead to a particular terminal price between L and H and hit either barrier. Had the double barrier knock-in option been defined as an option that comes into existence only if *both barriers* are hit, then,

$$N(a, b, s) = \sum_{i=2}^{n} (-1)^{i} (|A_i| + |B_i|)$$

should have been used for similar purposes.

2. As the number of terms can be huge, the straight forward calculations of $N(a, b, s)$ and $p^j(1-p)^{n-j}$ will quickly generate computer overflow and underflow, respectively. Combining these two calculations with Napierian logarithm will avoid the problem. In the implementation of the combinatorial method, factorial computation and $p^j(1-p)^{n-j}$ use Napierian logarithm, and their computations are actually combined as $N(a, b, s)p^j(1-p)^{n-j}$.

Chapter 5

Conclusion and Future Work

Combinatorial methods have found wide applicability in many fields. This thesis extends their use in pricing European-style double barrier options. By the use of combinatorial methods, we succeed in developing fast algorithms to speed up the pricing of double barrier options. The combinatorial formula directly leads to highly efficient algorithms in terms of convergence and time. In particular, the total running time will be less if we use parallelizing techniques in our algorithm by dividing the work of (3.3) among processors.

We expect the combinatorial method to be similarly applicable to more sophisticated path-dependent derivatives. In addition, we also expect to improve the combinatorial method farther for that our algorithm's running process can been parallelized with a group of professional machines. The combinatorial method will be an applicative tool for various path-dependent derivatives.

Bibliography

[1] Bhagavatula, Ravi S., and Peter P. Carr. "Valuing Double Barrier Options with Fourier Series." Manuscript, September 12, 1997.

[2] Boyle, P. "A Lattice Framework for Option Pricing with Two State Variables." *Journal of Financial and Quantitative Analysis*, 35, No. 1 (1988), pp. 1–12.

[3] Boyle, Phelim, and Sok Hoon Lau. "Bumping Up against the Barrier with the Binomial Method." *Journal of Derivatives*, Summer 1994, pp. 6–14.

[4] Cheuk, Terry H.F., and Ton C.F. Vorst. "Complex Barrier Options." *The Journal of Derivatives*, Fall 1996, pp. 8–22.

[5] Hull, John C. *Options, Futures, and Other Derivatives* . 3rd ed. Englewood Cliffs, New Jersey: Prentice-Hall, 1997.

[6] Jason Z. Wei. "Valuation of Discrete Barrier Options by Interpolations." *The Journal of Derivatives*, Fall 1998, pp. 51–73.

[7] Lint, J.H van, and R.M. Wilson. *A Course in Combinatorics.* Cambridge: Cambridge University Press, 1994.

[8] Lyuu, Yuh-Dauh. *Financial Engineering and Computation: Principles, Mathematics, Algorithms.* Manuscripts. To be Published.

[9] Lyuu, Yuh-Dauh. "Very Fast Algorithms for Barrier Option Pricing and the Ballot Problem." *The Journal of Derivatives*, Spring 1998, pp. 68–79.

[10] Merton, Robert C. *Continuous-Time Finance.* Revised ed. Cambridge, Massachusetts: Blackwell, 1994.

[11] Ritchken, Peter. "On Pricing Barrier Options." *Journal of Derivatives*, Winter 1995, pp. 19–28.

[12] Rogers, L.C.G., and O. Zane. "Valuing Moving Barrier Options." *J. Computational Finance*, 1997, pp. 1–9.