Options Pricing Using Combinatoric Methods
18.204 Postnikov — Final Paper

Annika Kim

May 7, 2018

Contents

Abstract

There are many complex options that have come about in the past couple of decades. The question is how we can price these options quickly and accurately Black and Scholes answered this question for vanilla options, the simplest kind of option that gives the owner the right to buy or sell a stock at the strike price, with a closed-form solution. However, there is no closed-form solution for other kinds of options, especially those with unusual features. These other kinds of options must be priced using numerical methods such as the lattice method. Combinatoric methods such as the Reflection Principle and Principle of Inclusion and Exclusion can be used instead of the lattice method to price options. Compared to the lattice method, which takes at least quadratic time to price these options, we show in this paper than combinatorial methods can price these options in linear time.

1 Introduction

Combinatorial algorithms can dramatically help improve the performance in pricing derivatives such as options. A derivative is a financial contract that derives its values from the performance of an underlying. They include financial items such as stocks, indices, interest rates, commodities, foreign exchange rates. Another popular type of derivative is called an option, which are a type of derivative that gives the buyer the right, but not the obligation, to buy or sell the option at the strike price. Various types of options exist such as:

- Vanilla options: standard options that give the holder the option to buy or sell the underlying asset at a set strike price in a given timeframe.
- Power options: exotic options that give the holder a payoff determined by the difference between the n-th power of the underlying asset and a fixed strike price.
- Single-Barrier options: similar to vanilla options but the price must hit a certain level (the barrier) for the option to become activated.
- Double Knock-in options: Similar to single barrier options but the price must hit at least one of two possible levels for the option to become activated.
- Lookback options: give the holder the option to buy (or sell) the underlying asset at the most beneficial price reached over the lifetime of the option.

In this paper, we will use combinatorial pricing algorithms to price barrier and double knock-in options in linear-time. These algorithms compare favorably against pricing algorithms with the lattice method.

2 The Lattice Model

2.1 Overview

A lattice divides a certain time interval into n discrete time steps and simulates the stock price discretely at each step. At each time step, the stock can either go up by u or down by d. The parameters u and d are fixed factors. It is is popular in the industry, but takes at least quadratic time. Below is an example of a 3-step Cox-Ross-Rubinstein (CRR) binomial lattice. S_0 indicates the initial stock price, u indicates upwards with probability p, and d indicates downwards with probability $1 - p$.

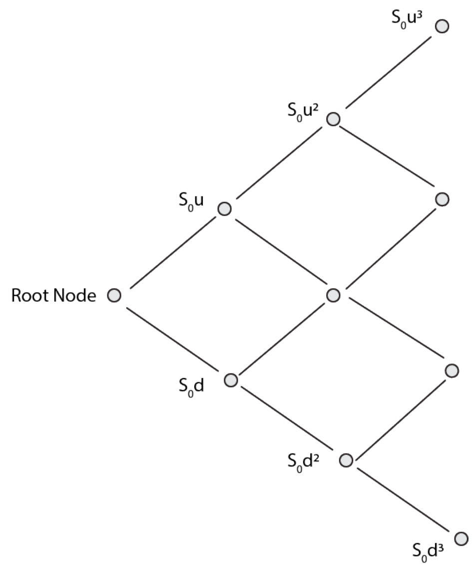


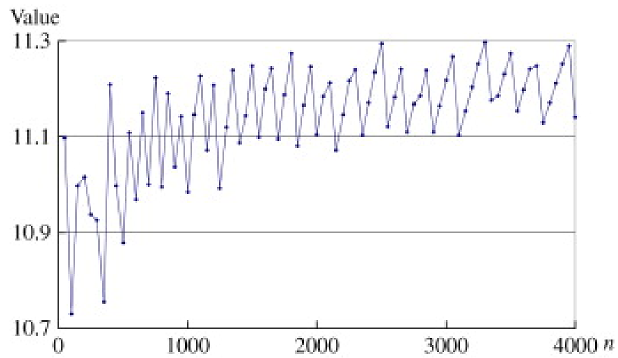
Figure 1: Example of a 3 time-step CRR lattice.

From the root node with stock price S_0, the lattice shows how the stock price moves after each time-step by calculating the option price at each node of the lattice and working backwards. Since there are $\frac{(n+1)(n+2)}{2}$ nodes of the lattice, the time complexity would be $O(\frac{n^2+3n+2}{2})$ which simplifies to $O(n^2)$. However, more time is required for complex options that have more than one possible state per node. The pricing results generated by the lattice model converge to the theoretical option price as the number of time steps n approach ∞.

2.2 Limitations of the Lattice Model

Although these prices converge to the theoretical option value in continuous time, they tend to converge slowly. But worse, for some options such as barrier options, the prices can even oscillate wildly; this means that large amounts of computational time are required to achieve acceptable accuracy.

For example, the oscillation phenomena of the lattice method can be seen even when $n = 4000$ for barrier options. Figure 2 below illustrates this phenomena.



(a) Source: Dai, et al.

Figure 2: Oscillation phenomena from lattice method

It is clear that there is slow convergence with the lattice model and a more efficient pricing algorithm would be useful. In the next section we will show how to price barrier options and knock-in options faster.

3 Barrier Options

To expand on definitions stated in the *Introduction* section of the paper, a barrier option is an option whose payoff depends on whether or not the stock's price path ever hits certain price levels called barriers. A single-barrier option contains only one barrier, whereas a double-barrier option contains two barriers.

Since there is no simple, closed-form solution to price these options, currently, industry standards are to use a formula that expresses the theoretical prices of these options as an infinite series of cumulative normal distributions. This implies that there must be truncation of the series at some point, but that it may also lead to large pricing errors.

3.1 Definition

Assume that there is a barrier H and an initial stock price S_0.

$$\begin{aligned} H &> S_0 \\ S_{sup} &= sup_{o \leq t \leq T} S(t) \end{aligned} \tag{1}$$

Then, the payoff of a single-barrier option at maturity T and strike price X is as follows.

$$payoff = \begin{cases} max(\theta S(T) - \theta, 0), if S_{sup} \geq H \\ 0, otherwise \end{cases} \tag{2}$$

where $\theta = 1$ for call options and $\theta = -1$ for put options.

3.2 The Reflection Principle

We will use the reflection principle to count the number of paths that hit a specific price level before reaching a certain node at maturity. The reflection principle state that the set of all paths from $(1, 1)$ to $(a+b, a-b)$ that touh the x-axis has a one-to-one correspondence with the set of all paths from $(1, -1)$ to $(a + b, a - b)$.

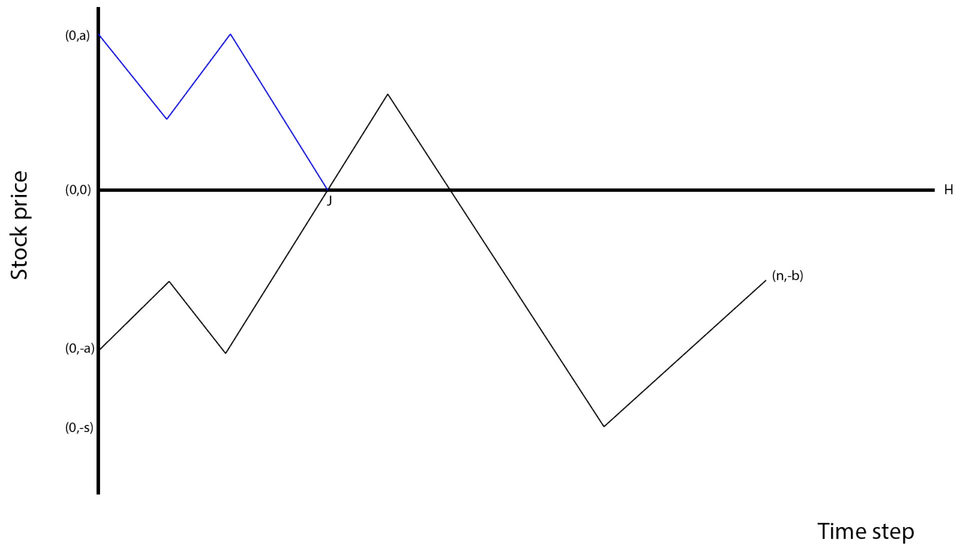


Figure 3: Reflection principle illustration.

Figure 3 shows how we can use the Reflection Principle to count the number of paths that hit a specific price level H before reaching a certain node $B(n, -b)$ at maturity.

Each step on Figure 3 from vertex (i, j) will either be an up-move or a down-move. An up-move would bring (i, j) to $(i + 1, j + 1)$, whereas a down-move would bring (i, j) to $(i + 1, j - 1)$. We want to find how many paths that go from node $A(0, -a)$ to $B(n, -b)$ will hit barrier H at some point.

$\widehat{AJB}$ is one path that hits H for the first time at J. If we reflect $\widehat{AJ}$, we get $\widehat{A_1J}$ and since each path from node A to node J maps to a unique path from node A_1 to node J and vice versa, the number of paths from $A$to J is equal to the number of paths from A_1 to J. Therefore, we use the reflection principle to state that the number of paths from A to B that hits H is simply equal to the number of paths from A_1 to B.

Assume that x up-moves and y down-moves are needed to go from A_1 to B. Then,

$$
\begin{aligned}
x + y &= n \\
x - y &= a - b
\end{aligned} \tag{3}
$$

and we get that

$$
\begin{aligned}
x &= \frac{n - a - b}{2} \\
y &= \frac{n + a + b}{2}.
\end{aligned} \tag{4}
$$

So, the number of paths that hit H before reaching the terminal node B is:

$$
\binom{n}{\frac{n-a-b}{2}}. \tag{5}
$$

3.3 Mapping the Stock Price onto a Grid

Next, we will map the stock price using a grid as shown on Figure 4. The option value will be obtained by taking the discounted value of the expected payoff of the option at maturity. We will do this by adding up the values contributed by each of the terminal nodes (seven in the case of the grid on Figure 4 below).

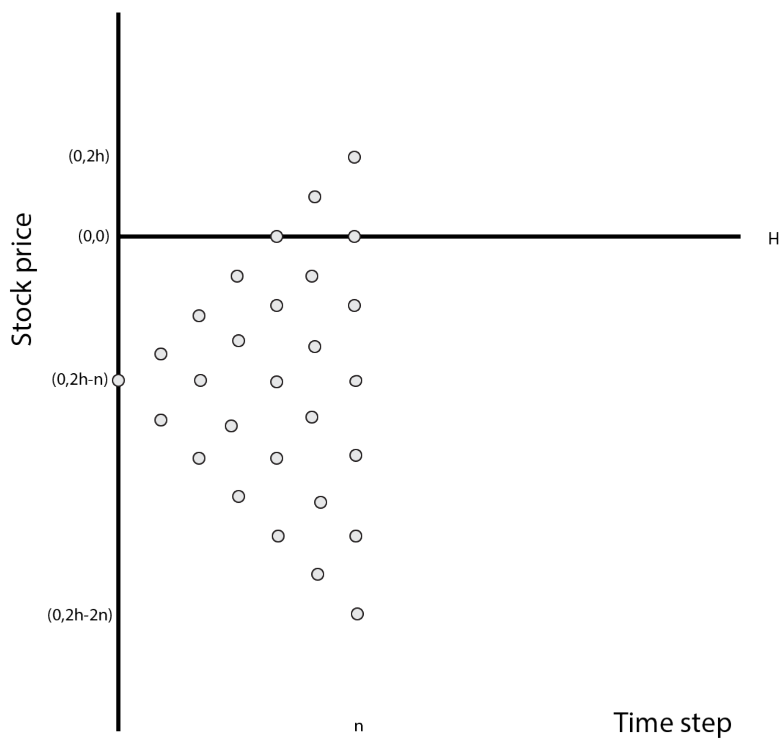


Figure 4: Stock price grid.

First, consider the terminal nodes above the barrier H. Starting at the initial stock price $S_0(0, 2h-n)$, we see that all paths that reach noes above H must hit H. The value contributed by these nodes are as follows.

$$\sum_{i=0}^{h} \binom{n}{i} p^{n-i}(1-p)^i e^{-rT} max(S_0 u^{n-i} d^i - X, 0). \quad (6)$$

Next, consider the terminal nodes below the barrier H. The number of paths that hit H before reaching the terminal node can be computed as follows.

$$\sum_{i=h+1}^{n} \binom{n}{2h-i} p^{n-i}(1-p)^i e^{-rT} max(S_0 u^{n-i} d^i - X, 0). \quad (7)$$

Finally, sum equation (6) and (7) to get the price of a single barrier option. Next, we will discuss double knock-in options.

4 Double Knock-In Options

Double knock-in options are a type of option whose payoff depends on whether or not the stock's price path ever hits certain price barriers. They are activated when the stock hits either of the two barriers. To price these types of options, we will utilize both the Reflection Principle and Principle of Inclusion-Exclusion.

4.1 Applying the Reflection Principle

As we did when pricing single-barrier options, we will count the number of paths that hit barriers L or H, denoting the lower and higher barriers, respectively, before reaching a certain node at the nth time-step. Figure 5 on the following page shows a sample stock price path.

Referring to Figure 5,to determine how many price paths from node A to node B hit barrier H before hitting barrier L, we will first reflect $\widehat{AJ}$ to obtain $\widehat{A_1J}$. Utilizing the reflection principle, we can state that the number of paths from A to B that hit H is equal to the number of paths from A_1 to B. Applying the reflection principle again, we can state that the number of paths from A_1 to B that hit L is equal to the number of paths from A_2 to B. Therefore, the number of paths from A to B that hit H before L is equal to the number of paths from A_2 to B.

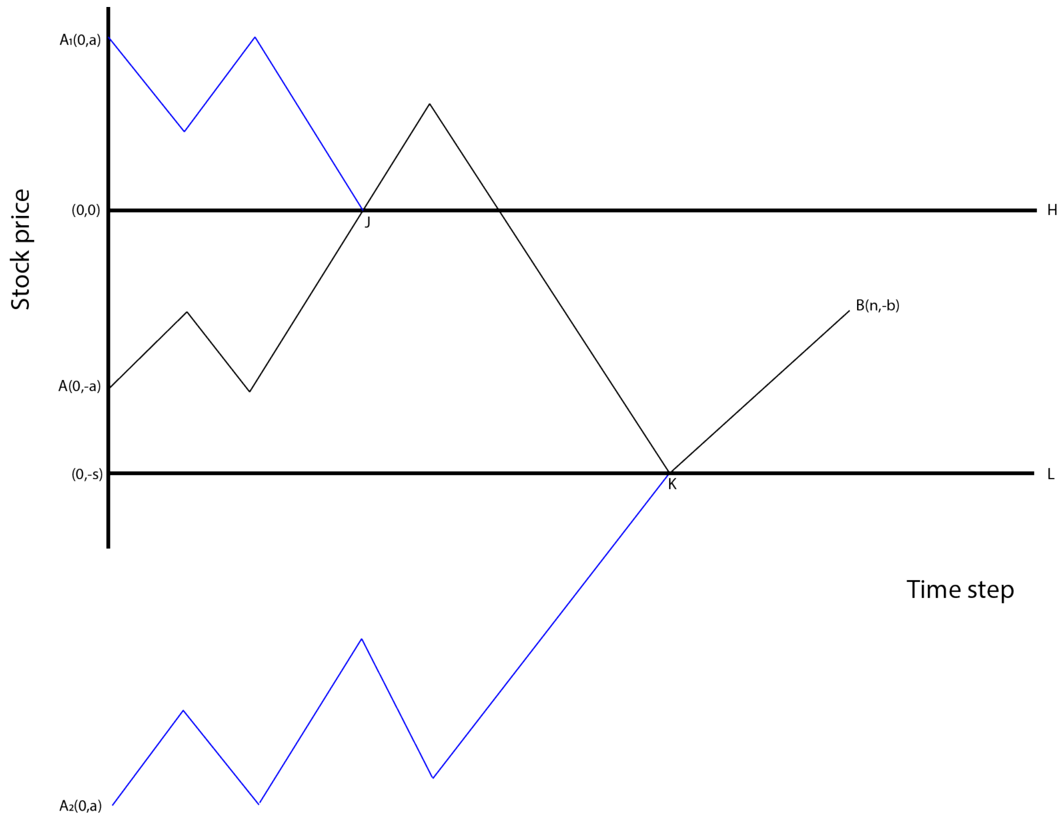


Figure 5: Reflection principle illustration for double barriers.

Since we need x up-moves and y down-moves to get from $A_2(0, -(a+2s))$ to $B(n, -b)$, we see that

$$\begin{aligned} x + y &= n \\ x - y &= a - b + 2s \end{aligned} \tag{8}$$

and we can solve for x and y to obtain

$$\begin{aligned} x &= \frac{n + a - b + 2s}{2} \\ y &= \frac{n - a + b - 2s}{2}. \end{aligned} \tag{9}$$

So, the number of paths from A to B that hit H before L is:

$$\binom{n}{\frac{n+a-b+2s}{2}}. \tag{10}$$

Next, consider the function f that maps a path to a string. For example, $f(AB) = HHL$ would indicate that a path $\widehat{AB}$ hits H before hitting L.

Let α_i be the set of paths whose f contains $H^+L^+H^+..., i \geq 1$. Then, $\widehat{AB}$ belongs to both α_1 and α_2. H^+ indicates a sequence of Hs and L^+ indicates a sequence of Ls. Next, let β_i be the set of paths whose f contains $L^+H^+L^+..., i \geq 1$. Then, $\widehat{AB}$ belongs to β_1. Note that each path that hits a barrier may belong to more than one set. In addition, note that no path can hit H and L alternatively more than $\lceil \frac{n}{s} \rceil$ ties since each path can move only n steps and the distance from H to L is s. Next, we will define sets of paths that hit H or L and utilize the principle of inclusion-exclusion to count the paths.

4.2 Applying the Principle of Inclusion-Exclusion

Let γ be the set of paths from A to B that hit H or L. Then,

$$\gamma = \alpha_1 \cup \beta_1 \tag{11}$$

because all paths in γ must hit H or L. The ones that hit H belong to set α_1 and the ones that hit L belong to set β_1.

Using the principle of inclusion-exclusion, we can state the following.

$$|\gamma| = |\alpha_1| + |\beta_1| - |\alpha_1 \cap \beta_1| \tag{12}$$

The set $|\alpha_1 \cap \beta_1|$ contain that paths that hit both H and L. Since $\alpha_1 \cap \beta_1 = \alpha_2 \cup \beta_2$, we can state that

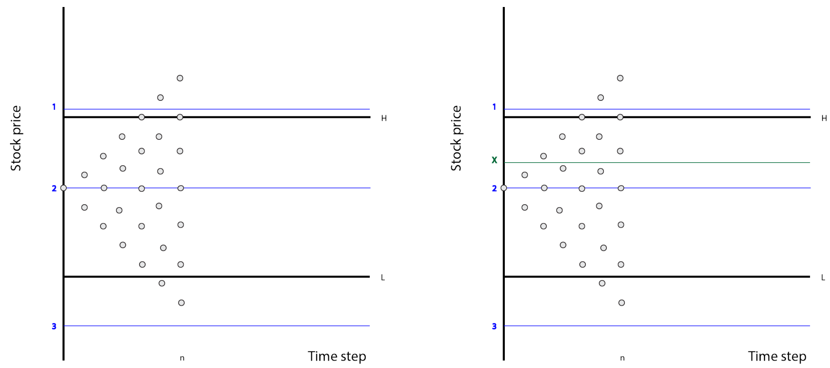
$$\begin{aligned} \alpha_1 \cap \beta_1 &= \alpha_2 \cup \beta_2 \\ &= |\alpha_2| + |\beta_2| - |\alpha_2 \cap \beta_2|. \end{aligned} \tag{13}$$

We apply equation 13 iteratively to get that

$$\begin{aligned} \gamma &= |\alpha_1| + |\beta_1| - |\alpha_2| - |\beta_2| + |\alpha_2 \cap \beta_2| \\ &= \sum_{i=1}^{\lceil \frac{n}{s} \rceil} (-1)^{i+1} (|\alpha_i| + |\beta_i|). \end{aligned} \tag{14}$$

4.3 Mapping the Stock Price onto a Grid

To price the option, will consider three cases: when the strike price X of the option exceeds H, when X is between L and H, and when X is below L.



(a) Stock price path terminal values split into three cases.

(b) Construction of linear time pricing algorithm with three cases and strike price.

Figure 6: Mapping stock price path with three cases.

1) When $X \geq H$:

If the payoff of the option is greater than 0, then the barrier H must have been hit since the payoff of the option is $\max(S(T) - X, 0) > 0$. Then, the value of the double knock-in (call) option would be the following.

$$\begin{aligned} value &= e^{-rT} E(payoff) \\ &= e^{-rT} E(max(S(T) - X, 0) \end{aligned} \tag{15}$$

To value the option here, we are only considering the terminal nodes that are above the strike price since we want a non-negative payoff. This is illustrated in Figure 6. Note that in this case, the value of the double knock-in option is equivalent to the value of a vanilla (call) option.

Since the option value contributed by a price path depends on the terminal nodes, let $N(n, j)$ be the probability that it is the terminal node. This probability is equal to $p^{n-j}(1-p)^j$. The payoff at N(n,j) would be $max(S_0 u^{n-j} d^j, 0)$. If this price path hits L or H, its value would be $p(j) = e^{-rT} p^{n-j} (1-p)^j max(S_0 u^{n-j} d^j - X, 0)$. The values contributed by each terminal node would be $\binom{n}{j} p(j)$.

To price these options in linear time, we can let a be a unique integer such that $S_0 u^{n-a} d^a \leq X \leq S_0 u^{n-a+1} d^{a-1}$. Next, we can build a table that states the values of each $\binom{n}{k}$.

2) When $L < X < H$:

In this case, the value of the option would consist of the values of the terminal nodes between X and H and the values of the terminal nodes above H. Summing these together, we obtain that the value of the option is

$$value = \sum_{j=h+1}^{a-1} N(n-2h, 2j-2h, 2l-2h)p(j) + \sum_{i=0}^{h} \binom{n}{i} p(i), \qquad (16)$$

where $N(n-2h, 2j-2h, 2l-2h)$ denote the paths that hit L or H before reaching the terminal node.

3) When $X \leq L$:

In this last case, the value of the option would consist of the values of the terminal nodes between L and H, the values of the terminal nodes above H, and the values of the terminal nodes between L and X. Summing these together, we obtain that the value of the option the following.

$$value = \sum_{j=h+1}^{l-1} N(n-2h, 2j-2h, 2l-2h)p(j) + \sum_{i=0}^{h} \binom{n}{i} p(i) + \sum_{k=1}^{a} \binom{n}{k} p(k). \qquad (17)$$

The price of the double knock-in option is simply the sum of the three cases above.

5 Conclusion

In this paper, we priced single-barrier and double knock-in options in an approach that would require linear time, as opposed to the industry standard of quadratic time. In practice, fast options pricing algorithms are essential to traders who rely on speed and accuracy to execute successful trades. We used combinatorial methods, in particular the reflection principle and the principle of inclusion-exclusion, to value these options efficiently, thus demonstrating some of the power that combinatorics can bring to the field of financial algorithms. Furthermore, it is possible to use similar methods to price a variety of products and derivatives beyond those discussed in this paper. In light of this great applicability and the increasing importance of algorithms in many branches of finance, it is likely that such methods will remain of great interest in the near future.

6 Sources Consulted

1. Dai, Tian-Shyr, et al. Linear-Time Option Pricing Algorithms by Combinatorics. Computers Mathematics with Applications, vol. 55, no. 9, May 2008, pp. 21422157., www.sciencedirect.com/science/article/pii/S0898122107006931?via%3Dihub.

2. Options. Investopedia, IAC Publishing, www.investopedia.com/.

3. Rutkowski, Marek. The CRR Market Model. Financial Mathematics, University of Sydney, 2016, www.maths.usyd.edu.au/u/UG/SM/MATH3075/r/Slides_7_Binomial_Market_Model.pdf.

4. Reed, Bruce. Random Walks 1, Reflection and Reversal. Applied Stochastic Processes, McGil University, 2008, cgm.cs.mcgill.ca/ breed/MATH671/lecture2corrected.pdf.