

PRICING LADDER OPTIONS WITH COMBINATORICS

B. Q. LI¹, H. J. ZHAO² and J. LEI¹

¹Department of Risk Management and Insurance
Nankai University
Tianjin 300071
P. R. China
e-mail: libq@nankai.edu.cn;
mathecholo29@hotmail.com

²Center for Combinatorics
Nankai University
Tianjin 300071
P. R. China
e-mail: 1120080017@mail.nankai.edu.cn

Abstract

Exotic options are popular financial derivatives that play essential roles in financial markets. How to price them efficiently and accurately is very important both in theory and practice. The lattice model is usually used to price them. The prices computed by the lattice converge to the theoretical value under the continuous-time model. But the lattice model may produce quite slow convergence; and when it comes to such options as barrier options, the lattice often produces wild oscillation and huge amounts of computational time are required to achieve acceptable accuracy. This paper introduces combinatorial techniques to help improve the performance in pricing a special barrier option, the ladder option. Through a computer experiment, it is proved that our

2000 Mathematics Subject Classification: 06xx.

Keywords and phrases: ladder option, lattice, reflection principle.

Received October 25, 2008

algorithm based on combinatorics compares favorably against popular lattice methods, which take at least quadratic time.

1. Introduction

Options are financial instruments that convey the right, but not the obligation, to engage in a future transaction on some underlying security, or in a futures contract. With the rapid growth and the deregulation of financial markets, many complex options have been structured to meet specific financial goals. Although financial innovations make the market more efficient, they also give rise to new problems: How to price these options efficiently and accurately.

In 1973, Black and Scholes settled the pricing problem in a satisfying way first. They derived formulas for the vanilla option, which gives its owner the right to buy or sell stock for the exercise price and does not have other unusual features. Although an option must have a unique theoretical value, calculating that value, especially of the exotic option, may be intractable. Most options cannot be evaluated analytically and must be priced by numerical methods. Finding efficient and accurate numerical pricing methods is thus important in both theory and practice.

The lattice method is a popular numerical method for pricing options. A lattice divides a certain time interval into n discrete time-steps and simulates the stock price discretely at each time-step. The most famous lattice model is a binary tree model proposed by Cox, Ross and Rubinstein, CRR model for short. To calculate the option prices, the naive lattice algorithm calculates the option price for each node of the lattice, working backward in time. The time complexity of such an algorithm is $O(n^2)$ since there are $(n+1)(n+2)/2$ nodes on a lattice. However, the prices of the CRR method may converge slowly or even oscillate wildly. And CRR model easily produces successive deviation when it is used to price barrier options [2]. The oscillation phenomenon for pricing options by the lattice model has been studied by many people. Boyle and Lau suggest picking proper n 's to reduce the oscillation for single-barrier options [2]. Alternatively, Ritchken provides a novel trinomial lattice

model for pricing both single-and double-barrier options [12]. But his approach is costly when the barrier is very close to the initial stock price. That is also called the "barrier-too-close" problem. Although the above-mentioned approaches can partially alleviate the price oscillation problem, they are not efficient as they all run in $O(n^2)$. Dai and Lyuu use linear-time pricing algorithms to price single and double-barrier options. These algorithms accelerate convergence and reduce oscillation. Moreover, the time complexity is only $O(n)$ [5, 6, 10].

This paper develops linear-time combinatorial algorithms for pricing ladder options. A ladder option is an option that locks in gains once predetermined price levels are reached by the underlying security. The pricing of the ladder option is similar to the pricing of the look back option, while the analytical solution is more complex. And it is quite difficult to price the ladder option by means of the standard CRR model [13]. Monte Carlo simulation is also suggested to price the ladder option, but it produces very wild oscillation. In order to achieve proper accuracy, a rather large n is required. Besides, the time complexity of pricing a ladder option with m rungs comes to $O(mn^2)$, which is a huge calculating. This paper develops a linear-time combinatorial algorithm for pricing ladder options. The payoff of a ladder option depends on numbers of paths getting round of "rungs". Thus a way is needed to efficiently count the number of those special paths. The result of the combinatorial algorithm reflects that the algorithm can produce quick convergence, and it runs in $O(mn)$, Which is proved a efficient algorithm.

Our paper is organized as follows. Background knowledge, like the assumption of the stock price process, the definitions of the options mentioned in this paper, the method to price an option under the risk-neutral probability, and the required combinatorial techniques, are introduced in Section 2. A linear-time algorithm for pricing ladder options is derived in Section 3. In Section 4, numerical results of computer simulation and the time complexity of the algorithm are given. Section 5 concludes the paper. Underlying assets concerned in our paper is stocks without dividends.

2. Financial Background

2.1. Stock price process

Let S_T denote the stock price at time T . S_T follows the log-normal diffusion process:

$$\ln S_T \sim \Phi[\ln S + (\mu - \sigma^2 / 2)T, \sigma\sqrt{T}],$$

where μ denotes the expected return rate of the stock, and σ denotes the volatility of the stock price. Assume it is risk neutral, thus the expected return rate equals the risk-free interest rate, $\mu = r$. Then we can conclude $E(S_T) = Se^{rT}$ and $\text{var}(S_T) = S^2 e^{2rT} (e^{\sigma^2 T} - 1)$.

2.2. CRR lattice

A CRR lattice model divides a certain time interval from 0 to T into n equal time- steps, Δt . In the period of Δt , the stock price either goes up to Su with the probability of p or comes down to Sd with the probability of $1 - p$ (Figure 1), and $ud = 1$. Then, the expected stock price is $pSu + (1 - p)Sd$ at Δt . Combining it with the above-mentioned expected value and variance of the stock price, we can have $u = e^{\sigma\sqrt{\Delta t}}$, $d = e^{-\sigma\sqrt{\Delta t}}$, $p = \frac{e^{r\Delta t} - d}{u - d}$.

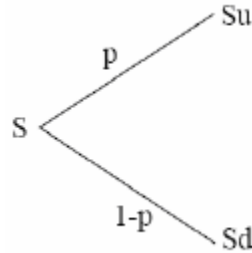


Figure 1. Binomial model. After Δt , the stock price S either moves to Su with probability p or Sd with probability $1 - p$.

2.3. Recurrence relations

The theoretical value of an option equals the discounted expected payoff of the option, $e^{-rT}E$ (payoff). Take a euro call option as an example. Assume that $N(n, j)$ denotes the final node reached after $n - j$ up-moves and j down-moves from the initial stock price. Since the probability of reaching node $N(n, j)$ is $\binom{n}{j}p^{n-j}(1-p)^j$, the price of the n -time-step CRR lattice can be derived directly as

$$e^{-rT} \sum_{j=0}^n \binom{n}{j} p^{n-j} (1-p)^j \max(Su^{n-j}d^j - X, 0).$$

The time complexity of it is $O(n)$ [5].

2.4. Ladder option

A ladder option is an option that locks in gains once predetermined price levels are reached by the underlying security. The set of predetermined price levels is called “ladder” and the price level in the set are called “rungs”. Typically, the rungs are equally spaced above the option strike with the distance equals to some percentage (e.g. 10%) of the strike. At maturity, the option pays the difference between the highest rung reached during the option life and strike or expires worthless if no rung has been reached. Take a ladder call for example. The rungs of the ladder call are represented by K_1, \dots, K_n . X denotes the strike. S_{max} is the highest price reached by the underlying stock during the option life. At maturity, the payoff does not depend on the final price of the underlying stock but the highest price during the option life S_{max} . That is: the payoff is $K_i - X$ if $K_i \leq S_{max} < K_{i+1}$, $1 \leq i < n$; it is $K_n - X$ if $S_{max} \geq K_n$; it is 0 if $S_{max} < K_1$.

2.5. The reflection principle

The reflection principle can help us to efficiently count the number of paths that hit a specific price level before reaching a certain node at maturity in a CRR lattice. We now derive a useful combinatorial formula with the help of the grid in Figure 2. This grid reflects the structure of a CRR lattice: The x -coordinate denotes the time-step of the CRR lattice,

and the y -coordinate denotes the stock price level. Each step on the grid from vertex (i, j) can either go to vertex $(i + 1, j + 1)$ (the up-move) or vertex $(i + 1, j - 1)$ (the down-move). The question is, how many paths from node $A(0, a)$ that end at node $B(n, b)$ will hit barrier $y = h$?

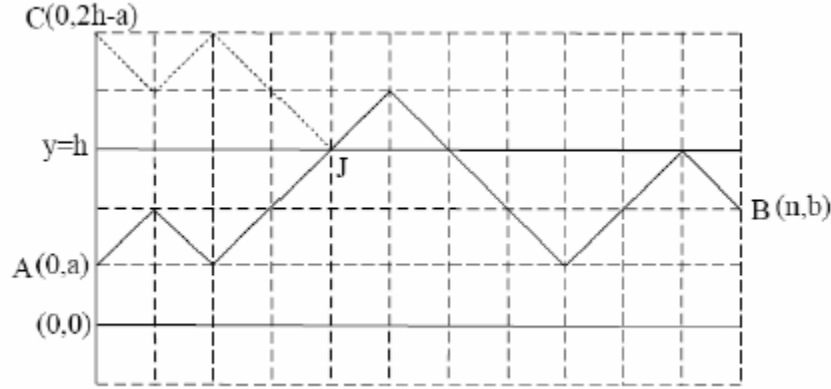


Figure 2. The reflection principle. Barrier (horizontal line $y = h$) is denoted by solid line.

Consider one such path, AJB , that hits barrier h at node J for the first time. We can reflect the initial path with respect to the h -axis to get CJ (the dashed curve). Each path from node A to node J maps to a unique path from node C to node J , and vice versa. Thus the number of paths from node A to node J equals the number of paths from node C to node J . As a result, the desired number of paths moving from node A to node B and hitting barrier h equals the number of paths from node C to node B . This is the celebrated reflection principle. Assume that x up-moves and y down-moves are required to go from node C to node B . Thus $x + y = n$ and $x - y = b - (2h - a)$. These two equations give $x = (n + a + b) / 2 - h$, $y = (n - a - b) / 2 + h$. Thus the number of paths that hit h before arriving at B is

$$\binom{n}{\frac{n+a+b}{2} - h}$$

for even, non-negative and zero otherwise.

3. Linear-Time Algorithms for Pricing Ladder Options

This section derives a useful combinatorial formula with the reflection principle to build up a pricing algorithm for ladder call options. Generally, the distance between rungs is equal, and we set it as l , $l > 0$. At first, we concentrate to develop a pricing algorithm for two-rung ladder call option. Then it is generalized to m -rung ladder call option.

3.1. 2-rung ladder call

At first, we have to make some explication about relative symbols in the paper.

$N(n, i)$ denotes the final node reached after $n - i$ up-moves and i down-moves from the initial price S_0 . And the final stock price is $S_0 u^{n-i} d^i$. Given $ud = 1$, the equation $S_0 u^{n-i} d^i = S_0 u^{n-2i}$ holds. K_1, K_2 denote the two-rungs, $K_1 < K_2$. X denotes the strike price. Generally, the distance between rungs is equal, and we set it as l , $l > 0$. Then $K_2 - K_1 = l$. Let

$$h_i = \left\lfloor \frac{\ln(S_0 u^n / K_i)}{\ln u} \right\rfloor, i = 1, 2$$

$$\widetilde{K}_i = S_0 u^{n-h_i}, i = 1, 2.$$

It is easy to see that \widetilde{K}_i is the price closest to, and not exceeded by K_i among all node prices. Then the role of the rung in determining the payoff of the option will be played by the effective line \widetilde{K}_i (Figure 3).

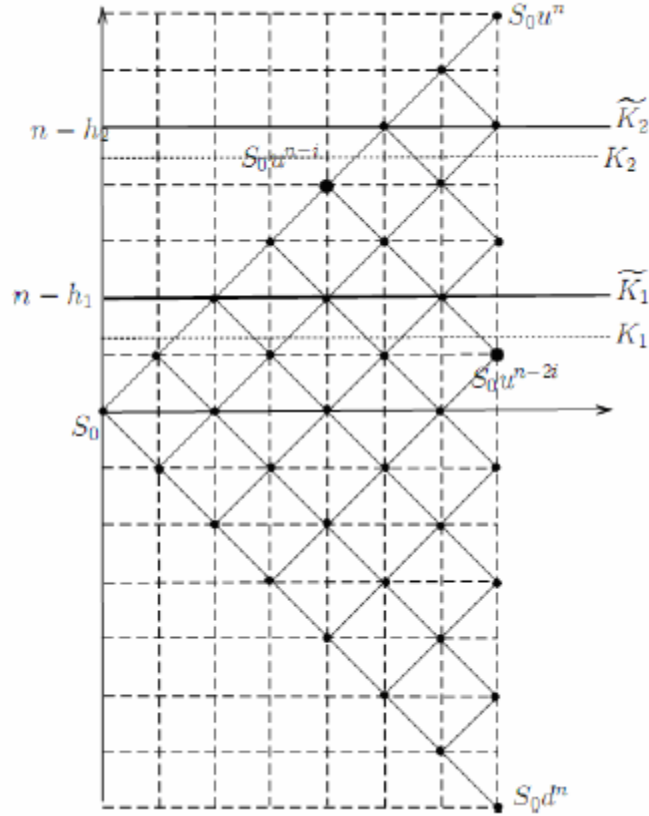


Figure 3. The initial stock price is S_0 . K_1, K_2 denote the two-rungs, and the effective barriers are the \tilde{K}_1, \tilde{K}_2 .

Define \wp as the set of paths beginning at S_0 and ending at $N(n, i)$. Obviously, $|\wp| = \binom{n}{i}$. Let $\Phi_i, i = 0, 1, 2$ denote the set of paths in which the highest rung touched is \tilde{K}_i ; and the paths in Φ_0 hit neither \tilde{K}_1 nor \tilde{K}_2 . Since paths in the same set result in the same payoff, the option value can be decomposed into three parts according to sets. And we just have to sum the three parts to get the option price. Next, we will calculate the three part of the option prices.

1. $2i \leq h_2$. Because all final nodes in the case are above \tilde{K}_2 (inclusive), each path ending at such nodes definitely hits \tilde{K}_2 . Then the

highest rung is \widetilde{K}_2 , and the payoff is $K_2 - X$. The number of all these paths is $\binom{n}{i}$. So the part of the option price is

$$F(i) = e^{-rT} p^{n-i} (1-p)^i \binom{n}{i} (K_2 - X).$$

2. $h_2 < 2i \leq h_1$. All final nodes are located between \widetilde{K}_1 (inclusive \widetilde{K}_1 ,) and \widetilde{K}_2 . Let

$$t = \min\left(\left\lfloor \frac{S_0 u^{n-i} - X}{l} \right\rfloor, 2\right).$$

If $t = 2$, the final node is high enough to allow the path hitting the highest rung. All paths in the case can be decomposed into two parts: (1) the highest rung hit is \widetilde{K}_2 , and (2) the highest rung hit is \widetilde{K}_1 . The number of paths in the first part is $\binom{n}{h_2-i}$, which is calculated according to the reflection principle. The option price in this part is

$$F(i, 2) = e^{-rT} p^{n-i} (1-p)^i \binom{n}{h_2-i} (K_2 - X).$$

Then, we have to calculate the number of paths which hit the highest rung \widetilde{K}_1 . Since any path that hits \widetilde{K}_2 definitely hits \widetilde{K}_1 , the number of paths in which the highest rung hit is \widetilde{K}_1 , equals $\binom{n}{i}$ minus the number of paths hitting \widetilde{K}_2 . That is $\binom{n}{i} - \binom{n}{h_2-i}$. Then the option price in this part is

$$F(i, 1) = e^{-rT} p^{n-i} (1-p)^i \left(\binom{n}{i} - \binom{n}{h_2-i} \right) (K_1 - X).$$

So when $t = 2$, the option price is $F(i) = F(i, 1) + F(i, 2)$.

If $t = 1$, all paths in which the highest rung hit is \widetilde{K}_1 total $\binom{n}{i}$. The option price is

$$F(i) = e^{-rT} p^{n-i} (1-p)^i \binom{n}{i} (K_1 - X).$$

3. $2i > h_1$. The definition of t still works here. If $t = 2$, following above routine, we have

$$F(i) = e^{-rT} p^{n-i} (1-p)^i \left(\binom{n}{h_2-i} (K_2 - X) + \left(\binom{n}{h_1-i} - \binom{n}{h_2-i} \right) (K_1 - X) \right).$$

$$\text{If } t = 1, F(i) = e^{-rT} p^{n-i} (1-p)^i \binom{n}{h_1-i} (K_1 - X).$$

If $t = 0$, the payoff is 0, then $F(i) = 0$.

Thus, the price of two-rung ladder call is the sum of the option prices above calculated in case 1, 2 and 3, $c = \sum_{i=0}^n F(i)$.

3.2. m -rung ladder call

There are m -rungs K_1, \dots, K_m , and $K_j - K_{j-1} = l, 1 < j \leq m$; $K_m < S_0 u^n$. Let

$$h_i = \left\lfloor \frac{\ln(S_0 u^n / K_i)}{\ln u} \right\rfloor, i = 1, 2, \dots, m.$$

Accordingly, let $\widetilde{K}_i = S_0 u^{n-h_i}, i = 1, \dots, m$, then $K_1 \leq \dots \leq K_m$. Line \widetilde{K}_i is the price closest to K_i , and not exceeded by K_i in the lattice. Let

$$t = \min\left(\left\lfloor \frac{S_0 u^{n-i} - X}{l} \right\rfloor, m \right).$$

Next, we have to calculate the option price according to the three cases of node $N(n, i)$.

1. $2i \leq h_m$. Because all final nodes in the case are above \widetilde{K}_m (inclusive), each path ending at such nodes definitely hits \widetilde{K}_m . The option price in this part is

$$F(i) = e^{-rT} p^{n-i} (1-p)^i \binom{n}{i} (K_m - X).$$

2. $h_m < 2i \leq h_1$. All final nodes are located between \widetilde{K}_1 (inclusive) and \widetilde{K}_m . Let

$$s = \left\lfloor \frac{S_0 u^{n-2i} - X}{l} \right\rfloor.$$

When $t \neq s$, All paths can be divided into three parts: (1) the highest rung hit is \widetilde{K}_t , (2) the highest rungs hit are respectively $\widetilde{K}_{s+1}, \dots, \widetilde{K}_{t-1}$, and (3) the highest rung hit is \widetilde{K}_s . The number of paths in the first part is $\binom{n}{h_t - i}$, which is calculated according to the reflection principle. The option price in this part is

$$F(i, 1) = e^{-rT} p^{n-i} (1-p)^i \binom{n}{h_t - i} (K_t - X).$$

In the second part, since any path that hits \widetilde{K}_{j+1} definitely hits \widetilde{K}_j , the number of paths in which the highest rung hit is \widetilde{K}_j , $s+1 \leq j \leq t-1$, equals $\binom{n}{h_j - i} - \binom{n}{h_{j+1} - i}$. Then the option price in the second part is

$$F(i, 2) = \sum_{j=s+1}^{t-1} e^{-rT} p^{n-i} (1-p)^i \left(\binom{n}{h_j - i} - \binom{n}{h_{j+1} - i} \right) (K_j - X).$$

Then, we calculate the option price in the third part. Since the final node is located above \widetilde{K}_s (inclusive), the number of paths in this part is $\binom{n}{i} - \binom{n}{h_{s+1} - i}$. Then the option price in the third part is

$$F(i, 3) = e^{-rT} p^{n-i} (1-p)^i \left(\binom{n}{i} - \binom{n}{h_{s+1} - i} \right) (K_s - X).$$

So, if $t \neq s$, the option price is $F(i, 1) + F(i, 2) + F(i, 3)$. That is

$$\begin{aligned} F(i) &= e^{-rT} p^{n-i} (1-p)^i \left(\binom{n}{i} - \binom{n}{h_{s+1} - i} \right) (K_s - X) \\ &+ \sum_{j=s+1}^{t-1} e^{-rT} p^{n-i} (1-p)^i \left(\binom{n}{h_j - i} - \binom{n}{h_{j+1} - i} \right) (K_j - X) \\ &+ e^{-rT} p^{n-i} (1-p)^i \binom{n}{h_t - i} (K_t - X). \end{aligned}$$

If $s = t$,

$$F(i) = e^{-rT} p^{n-i} (1-p)^i \binom{n}{i} (K_s - X).$$

3. $2i > h_1$. Following the same process as in case 2. If $t \neq 0$,

$$\begin{aligned} F(i) &= \sum_{j=1}^{t-1} e^{-rT} p^{n-i} (1-p)^i \left(\binom{n}{h_j - i} - \binom{n}{h_{j+1} - i} \right) (K_j - X) \\ &+ e^{-rT} p^{n-i} (1-p)^i \binom{n}{h_t - i} (K_t - X). \end{aligned}$$

If $t = 0$, Thus $F(i) = 0$. The price of m -rung ladder call option is the sum of above calculated option prices. That is $c = \sum_{i=0}^n F(i)$.

4. Experimental Results

This section evaluates the performance of the combinatorial algorithm. All the running time measurements are obtained by running programs on a 512MB memory computer.

In the programming, we store $\binom{n}{i}$ and $p^{n-i}(1-p)^i$ into two arrays. And both and run in $O(n)$ since $\binom{n}{i} = \binom{n}{i-1} \frac{n-i+1}{i}$, $p^{n-i}(1-p)^i = p^{n-(i-1)}(1-p)^{i-1} \times (1-p)/p$. Then it can be concluded that the time complexity of m -rung ladder is $O(mn)$. When m is a finite value, our algorithm is linear. In contrast, the time complexity of Monte Carlo simulation is $O(mn^2)$. Chart 1, Figures 4, 5, 6 and 7 reveal the result. Chart 1 shows the comparison of results of Monte Carlo simulation and the combinatorial algorithm for the 2-rung ladder option and the 1000-rung one. It is easy to conclude that the option price converges quickly when it is priced by the combinatorics, and the time for 2-rung ladder and 1000-rung one is nearly the same. For example, when the time steps are 500, the running time of Monte Carlo simulation for the 2-rung ladder amounts to 1283 ms, while that of the combinatorial algorithm is 1 ms.

When the time steps are the same, the running time of these two methods does not increase significantly.

Chart 1. Values and running-time comparison of results of Monte Carlo simulation and the combinatorial algorithm for the 2-rung ladder option and the 1000-rung one.

n	2 ladder				1000 ladder			
	Monte Carlo		Combinatorics		Monte Carlo		Combinatorics	
	Value	Time	Value	Time	Value	Time	Value	Time
100	12.06872	157	12.29178	< 1	19.7915	499	20.20252	< 1
200	12.55462	513	12.78082	< 1	20.367	847	20.42476	< 1
300	12.22073	770	12.32154	< 1	20.6538	1209	20.58611	1
500	12.89665	1283	12.56564	1	20.8022	1916	20.67503	3
800	12.58991	2056	12.80727	4	20.8239	3003	21.01756	8
1000	12.89936	2567	12.67048	6	21.0429	3675	20.7911	12

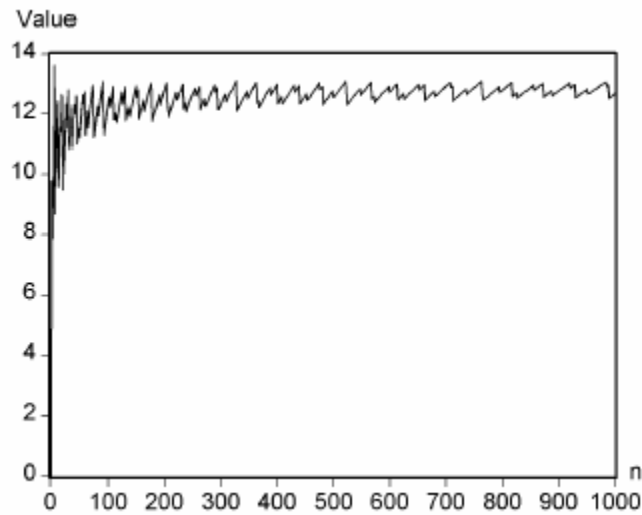


Figure 4. Pricing a 2-rung ladder option. The initial stock price $S_0 = 100$, the strike price $X = 100$, risk-free rate per annual $r = 10\%$, the volatility $\sigma = 0.25$, the maturity time $T = 1$, rungs are set at 130 and 160. X-axis denotes the number of time steps, Y-axis denotes the corresponding option prices.

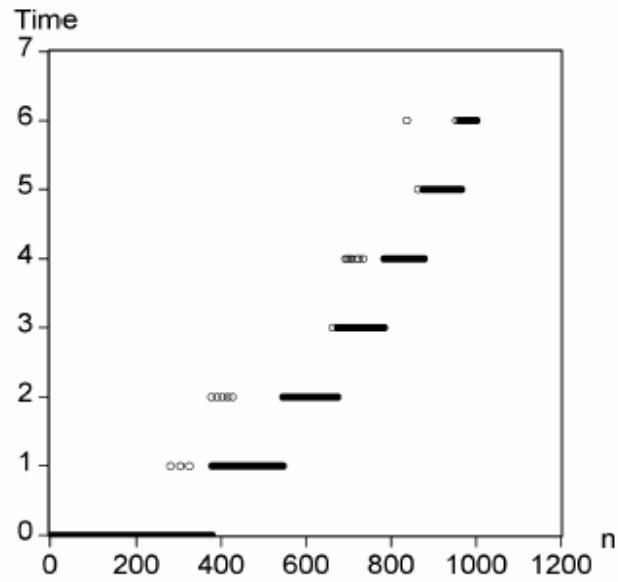


Figure 5. The running time of the 2-rung ladder option. The parameters of the option are the same as in Figure 4. X -axis denotes the number of time steps, Y -axis denote the corresponding time the unit is ms.

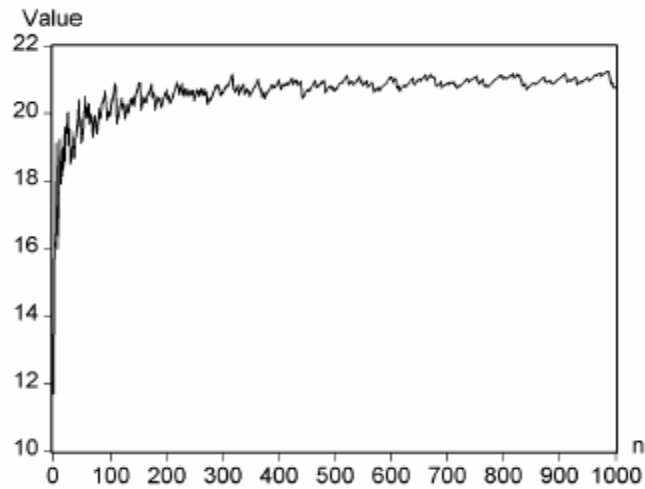


Figure 6. Pricing a 1000-rung ladder option whose rung distance is 10. The other parameters in the 1000-rung ladder are set the same as the 2-rung ladder.

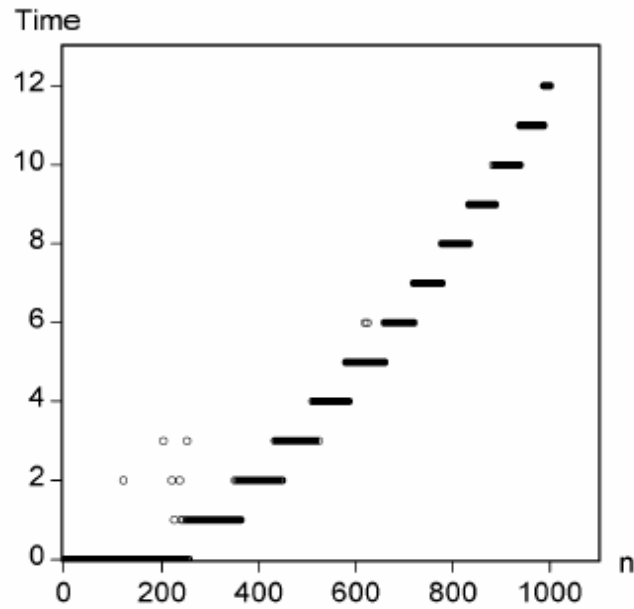


Figure 7. The running time of a 1000-rung ladder option whose rung distance is 10. The other parameters in the 1000-rung ladder are set the same as the 2-rung ladder.

5. Conclusion

The combinatorial algorithm has been widely applied in many fields. In the paper, we apply the combinatorial algorithm to price the ladder option. The result indicates that the combinatorial formula costs less time, gain higher accuracy and lowers the time complexity to $O(mn)$. Moreover, the algorithm can also be used to price barrier options that are usually path-dependent. We are expecting that the application of combinatorial algorithm to more complex options' pricing would achieve more improvement.

References

- [1] F. Black and M. Scholes, The pricing of options and corporate liabilities, *J. Polit. Econ.* 81 (1973), 637-659.
- [2] P. Bolye and S. Lau, Bumping up against the barrier with the binomial method, *J. Deriv.* 1 (1994), 6-14.
- [3] P. Chalasani, S. Jha and I. Saias, Approximate option pricing, *Algorithmica* 25 (1999), 2-21.
- [4] J. Cox, S. Ross and M. Rubinstein, Option pricing: A simplified approach, *J. Finance Econ.* 7 (1979), 229-264.
- [5] T. S. Dai and Y. D. Lyuu, Linear-time option pricing algorithms by combinatorics, *Computers and Mathematics with Applications* 55 (2008), 2142-2157.
- [6] T. S. Dai and Y. D. Lyuu, The Bino-trinomial Tree Model-A Simple Model for Efficient and Accurate Option Pricing, Announced in the Asian FA/FMA Meeting, Auckland, New Zealand. 2006.
- [7] J. Hull, *Options, Futures, and Other Derivatives*, 5th ed., Prentice-Hall Press, Englewood Cliffs, New Jersey, 2003.
- [8] J. Hull and A. White, Efficient procedures for valuing European and American path-dependent options, *J. Deriv.* 1 (1993), 21-31.
- [9] Y. D. Lyuu, *Financial Engineering and Computation: Principles, Mathematics, Algorithms*, Cambridge University Press, Cambridge, UK, 2002.
- [10] Y. D. Lyuu, Very fast algorithms for barrier option pricing and the ballot problem, *J. Deriv.* 5 (1998), 68-79.
- [11] Sri Gopal Mohanty, *Lattice Path Counting and Applications*, Academic Press, New York, 1979.
- [12] P. Ritchken, On pricing barrier options, *J. Deriv.* 3 (1995), 19-28.
- [13] Pavel V. Shevchenko, *Ladder Options*, Working Paper, CSIRO Mathematical and Information Sciences, 2004.
- [14] P. G. Zhang, *Exotic Options, A Guide to the Second Generation Options* (2nd Edition), World Scientific, New York, 1998.

