Ch 10. Arithmetic Average Options and Asian Opitons

I. Asian Option and the Analytic Pricing Formula

II. Binomial Tree Model to Price Average Options

III. Combination of Arithmetic Average and Reset Options

- Asian options are path dependent derivatives whose payoffs depend on the average of the underlying asset prices during the option life. They were originally issued in 1987 by Bankers Trust Tokyo on crude oil contracts and hence with the name "Asian" option.

- The features or advantages of Asian options are as follows.

  1. Asian options are appropriate to meet the hedging needs of users of commodities, energies, or foreign currencies who will be exposed to the risk of average prices during a future period.

  2. Since the volatility for the average of the underlying asset prices is lower than the volatility for the underling asset prices, Asian options are less expensive than corresponding vanilla options and are therefore more attractive for some investors.

  3. Asian options are also useful in thinly-traded markets to prevent the manipulation of the underlying asset price.

- In this chapter, for the teaching purpose, average options and Asian options are classified dependent on either the price of the underlying asset at maturity or the strike price being replaced by the average price.

$$\text{Average options}\begin{cases}\text{average price call:}\quad \max(S_{\text{ave}}-K,0)\\[2ex]\text{average price put:}\quad \max(K-S_{\text{ave}},0)\end{cases}$$

$$\text{Asian option}\quad\begin{cases}\text{average strike call:}\quad \max(S_T-S_{\text{ave}},0)\\[2ex]\text{average strike put:}\quad \max(S_{\text{ave}}-S_T,0)\end{cases}$$

I. Asian Option and Its Analytic Pricing Formula

- If S_{ave} is defined as the geometric average of stock prices, since the product of lognormally distributed random variables also follows the lognormal distribution, S_{ave} is lognormally distributed. In the risk-neutral world, the process of S_{ave} over a certain period T is with the expected continuously compounding growth rate $\frac{1}{2}(r - q - \frac{\sigma^2}{6})T$ (i.e., $E[S_{\text{ave}}] = S_0 e^{\frac{1}{2}(r-q-\frac{\sigma^2}{6})T}$) and the volatility $\sigma\sqrt{T}/\sqrt{3}$.

- For geometric average options, because the role of S_{ave} is the same of S_T in the payoff function, based on the lognormal distribution of S_{ave} and the Black-Scholes formula, the price formula for geometric average option can be derived straightforward.

- For a geometric average call,

$$\begin{aligned}
\text{option value} &= S_0 e^{(a-r)T} N(d_1) - K e^{-rT} N(d_2) \\
&= e^{-rT}[S_0 e^{aT} N(d_1) - K N(d_2)] \\
&= e^{-rT}[E[\text{geometric average until } T] N(d_1) - K N(d_2)]
\end{aligned}$$

$$\begin{cases} d_1 = & \frac{\ln(S_0 e^{aT}/K) + (\frac{1}{2}\sigma_G^2)T}{\sigma_G\sqrt{T}} = \frac{\ln(S_0/K) + (a + \frac{1}{2}\sigma_G^2)T}{\sigma_G\sqrt{T}} \\ d_2 = & d_1 - \sigma_A\sqrt{T} \end{cases}$$

$$\begin{cases} a = & \frac{1}{2}(r - q - \frac{\sigma^2}{6}) \\ \sigma_G = & \frac{\sigma}{\sqrt{3}} \end{cases}$$

Kemna and Vorst (1990), "A Pricing Method for Option Based on Average Asset Values," *Journal of Banking & Finance* 14, pp. 113–129.

- If S_{ave} is defined as the arithmetic average of stock prices, it is more difficult to price the arithmetic average option. An approximation method is described as follows.

  First, calculate the first and the second moments of S_{ave} during the option life T.

  $M_1 = \frac{e^{(r-q)T}-1}{(r-q)T} S_0 = E[\text{arithmetic average until } T]$

  $M_2 = \frac{2e^{(2r-2q+\sigma^2)T} S_0^2}{(r-q+\sigma^2)(2r-2q+\sigma^2)T^2} + \frac{2S_0^2}{(r-q)T^2}\left(\frac{1}{2(r-q)+\sigma^2} - \frac{e^{(r-q)T}}{r-q+\sigma^2}\right)$

  Second, assume that S_{ave} is lognormally distributed with the first and second moments mentioned above.

  Finally, based on the Black-Scholes-like formula for geometric average options, the value of an arithmetic average call can be approximated as follows.

  $$c = e^{-rT}\left[E[\text{arithmetic average until } T]N(d_1) - KN(d_2)\right]$$
  $$d_1 = \frac{\ln(E[\text{arithmetic average until } T]/K) + \sigma_A^2 T/2}{\sigma_A\sqrt{T}}$$
  $$d_2 = d_1 - \sigma_A\sqrt{T}$$

  where $E[\text{arithmetic average until } T] = M_1$, $\sigma_A^2 = \frac{1}{T}\ln(\frac{M_2}{M_1^2})$

  Turnbull and Wakeman (1991), "A Quick Algorithm for Pricing European Average Option," *Journal of Financial and Quantitative Analysis* 26, pp. 377–389.

II. Binomial Tree Model to Price Average Options

⊙ The naive pricing method based on the tree-based model, which tracks all possible arithmetic average prices reaching each node, is able to derive exact option values for both arithmetic and geometric average options.

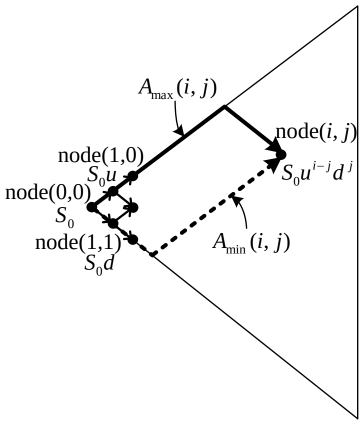
⊙ The naive pricing method only works for geometric average options. It is intractable to price arithmetic average options due to the exponential growth of the number of possible arithmetic average prices with respect to the number of time steps, n.

⊙ Instead of keeping track of all possible arithmetic average prices, Hull and White (1993) introduce representative average prices to be (logarithmically) equally-spaced placed between the maximum and minimum arithmetic average prices for each node. In addition, the piece-wise linear interpolation is employed to approximate the corresponding option values for nonexistent average prices during the backward induction.

⊙ The algorithm of Hull and White (1993):

(1) For any node(i, j), the maximum arithmetic average price is contributed by a price path starting with $i-j$ consecutive up movements followed by j consecutive down movements, and the minimum arithmetic average price can be calculated from a price path starting with j consecutive down movements followed by $i-j$ consecutive up movements.

Figure 10-1



$$A_{\max}(i,j) = S_0(1 + \overbrace{u + u^2 + \cdots + u^{i-j}}^{i-j \text{ up movements}} + \overbrace{u^{i-j}d + u^{i-j}d^2 + \cdots + u^{i-j}d^j}^{j \text{ down movements}}) / \, (i+1)$$
$$= (S_0 \tfrac{1-u^{i-j+1}}{1-u} + S_0 \cdot u^{i-j} \cdot d \cdot \tfrac{1-d^j}{1-d})/(i+1)$$

$$A_{\min}(i,j) = S_0(1 + \overbrace{d + d^2 + \cdots + d^j}^{j \text{ down movements}} + \overbrace{d^j u + d^j u^2 + \cdots + d^j u^{i-j}}^{i-j \text{ up movements}})/(i+1)$$
$$= (S_0 \tfrac{1-d^{j+1}}{1-d} + S_0 \cdot d^j \cdot u \cdot \tfrac{1-u^{i-j}}{1-u})/(i+1)$$

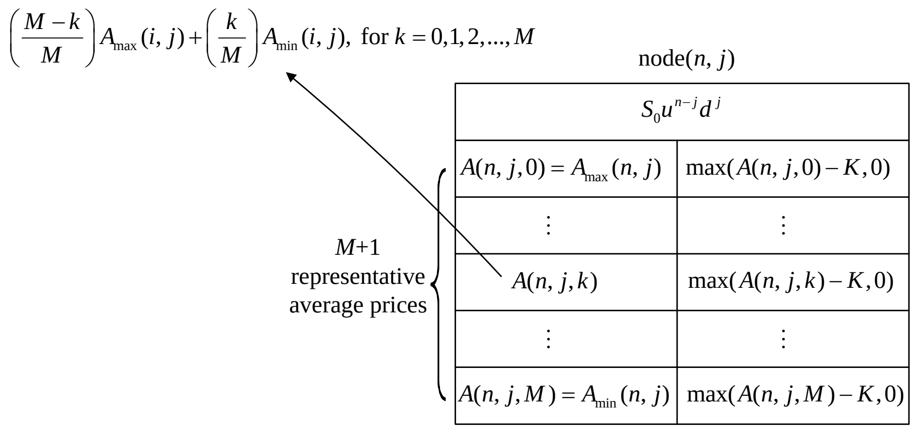
(2) For each node, representative average prices are arrayed (logarithmically) equally-spaced from the maximum to the minimum arithmetic average prices for each node via the following formula.

$$A(i,j,k) = \frac{M-k}{M}A_{\max}(i,j) + \frac{k}{M}A_{\min}(i,j), \text{ for } k = 0, ..., M.$$

$$\left(A(i,j,k) = \exp\left(\frac{M-k}{M}\ln(A_{\max}(i,j)) + \frac{k}{M}\ln(A_{\min}(i,j)) \right), \text{ for } k = 0, ..., M. \right)$$

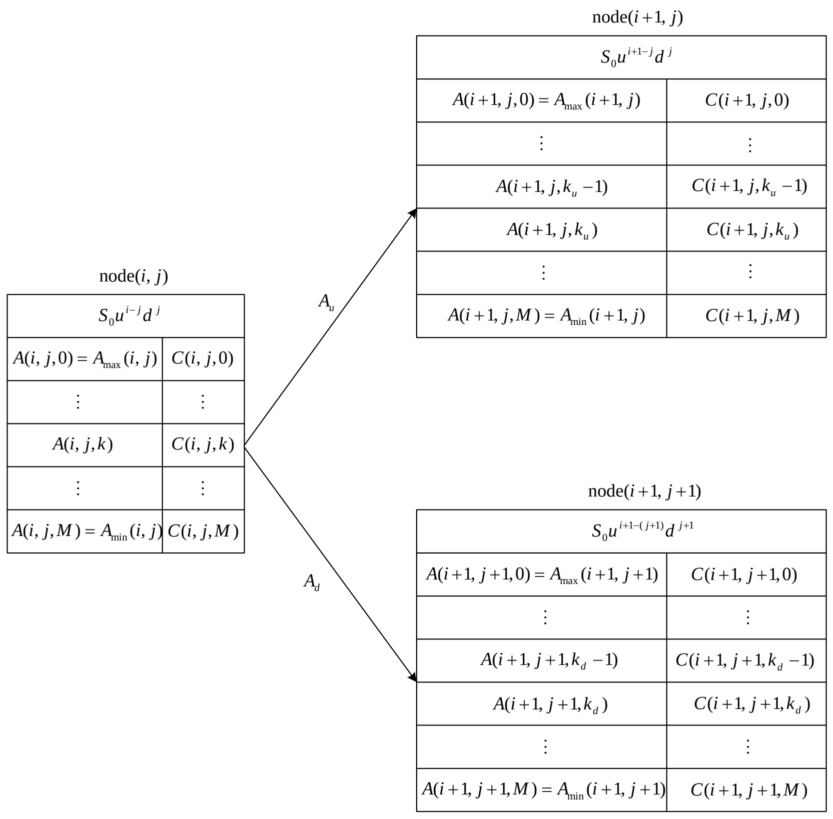
(3) For each terminal node(n,j), decide the payoff for each representative average price $A(n,j,k)$.

Figure 10-2



(4) Backward induction

Figure 10-3



For $A(i,j,k)$, $0 \le j \le i \le n$, and $k=$ 0, 1,. . ., M,

$\Rightarrow A_u = \frac{(i+1)A(i,j,k)+S_0u^{i+1-j}d^j}{i+2}$

Suppose A_u is inside the range $[A(i+1,j,k_u), A(i+1,j,k_u-1)]$. The corresponding option value C_u for A_u can be approximated by the linear interpolation, i.e.,

$$C_u = w_uC(i+1,j,k_u) + (1-w_u)C(i+1,j,k_u-1),$$

where

$$w_u = \frac{A(i+1,j,k_u-1) - A_u}{A(i+1,j,k_u-1) - A(i+1,j,k_u)}.$$

$\Rightarrow A_d = \frac{(i+1)A(i,j,k)+S_0u^{i+1-(j+1)}d^{(j+1)}}{i+2}$

Similarly, if A_d is inside the range $[A(i+1,j+1,k_d), A(i+1,j+1,k_d-1)]$. The corresponding option value C_d for A_d can be approximated by the linear interpolation following the same logic as above.

$\Rightarrow C(i,j,k) = (P \cdot C_u + (1-P) \cdot C_d) \cdot e^{-r\Delta t}$

$*$ If American arithmetic average options are considered, the option value $C(i,j,k)$ $= \max(A(i,j,k) - K, (P \cdot C_u + (1-P) \cdot C_d) \cdot e^{-r\Delta t})$.

- As a consequence, the interpolation error emerges and pricing results might not converge to exact option values unless the number of representative average prices for each node, M, is sufficiently large and well collocated with the number of time steps, n, in the tree model. Generally speaking, with the increase of the number of time steps in the tree model, more representative average prices are needed for each node to derive convergent results.

III. Combination of Arithmetic Average and Reset Options

- This section introduces a financial innovation to combine two attrative features, the Arithmetic Average and Reset Options, to form a new options. The pricing model of this new option is first proposed by Kim, Chang, and Byun (2003), "Valuation of Arithmetic Average Reset Options," *Journal of Derivatives* 11, pp. 70–80.

- The payoff of a standard reset call: $\max(S_T - K_T, 0)$.

  Since the strike price is reset downward for calls, $K_T = \min(K_0, S_{t_1}, S_{t_2}, \cdots, S_{t_I})$, where $t_1, t_2, \ldots, t_I$ are reset dates.

- Arithmetic average reset calls: the same payoff function as that for standard reset calls, except that $K_T = \min(K_0, A_{t_1}, A_{t_2}, \cdots, A_{t_I})$.

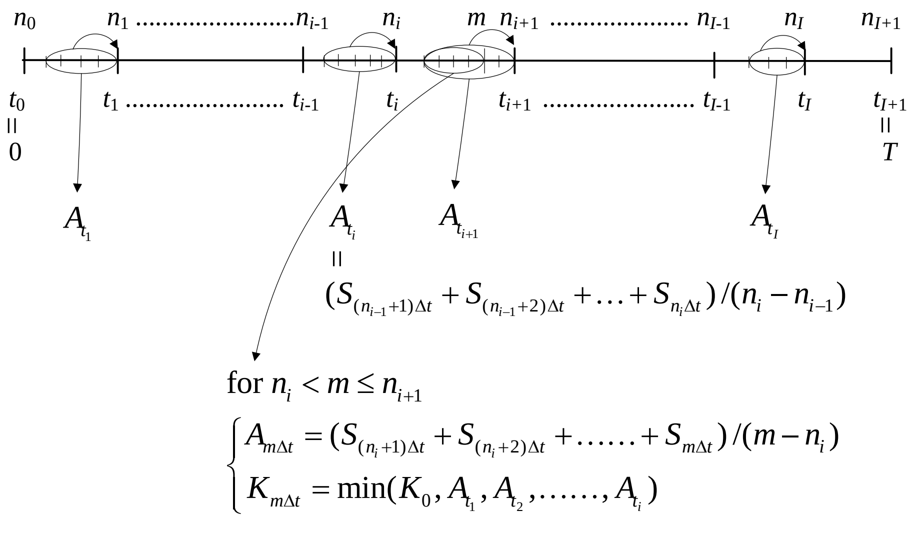
- The advantages of the arithmetic average reset options:

  ⊙ Avoid manipulation on (or near) the reset date.

  ⊙ The arithmetic average feature can reduce the option premium.

Figure 10-4

Suppose $\Delta t = T/n$, and the reset dates $t_i = n_i \Delta t$.


n_0
n_1 .... $n_{i\text{-}1}$
n_i
m n_{i+1} .... $n_{I\text{-}1}$
n_I
n_{I+1}
$t_0 = 0$
t_1 .... $t_{i\text{-}1}$
t_i
t_{i+1} .... $t_{I\text{-}1}$
t_I
$t_{I+1} = T$
A_{t_1}
A_{t_i}
$A_{t_{i+1}}$
A_{t_I}

$$A_{t_i} = (S_{(n_{i-1}+1)\Delta t} + S_{(n_{i-1}+2)\Delta t} + \ldots + S_{n_i \Delta t})/(n_i - n_{i-1})$$

for $n_i < m \le n_{i+1}$

$$\begin{cases} A_{m\Delta t} = (S_{(n_i+1)\Delta t} + S_{(n_i+2)\Delta t} + \ldots\ldots + S_{m\Delta t})/(m - n_i) \\ K_{m\Delta t} = \min(K_0, A_{t_1}, A_{t_2}, \ldots\ldots, A_{t_i}) \end{cases}$$

- The evolution rule of state variables (K_t, A_t):

  (i) For the root and the reset time points, the state variables at the next time point is $(K_{t+\Delta t},\ A_{t+\Delta t}) = (K_t,\ S_{t+\Delta t})$ ($A_{t+\Delta t} = S_{t+\Delta t}$ indicates the start (or restart) of calculating the arithmetic average price at the next time point).

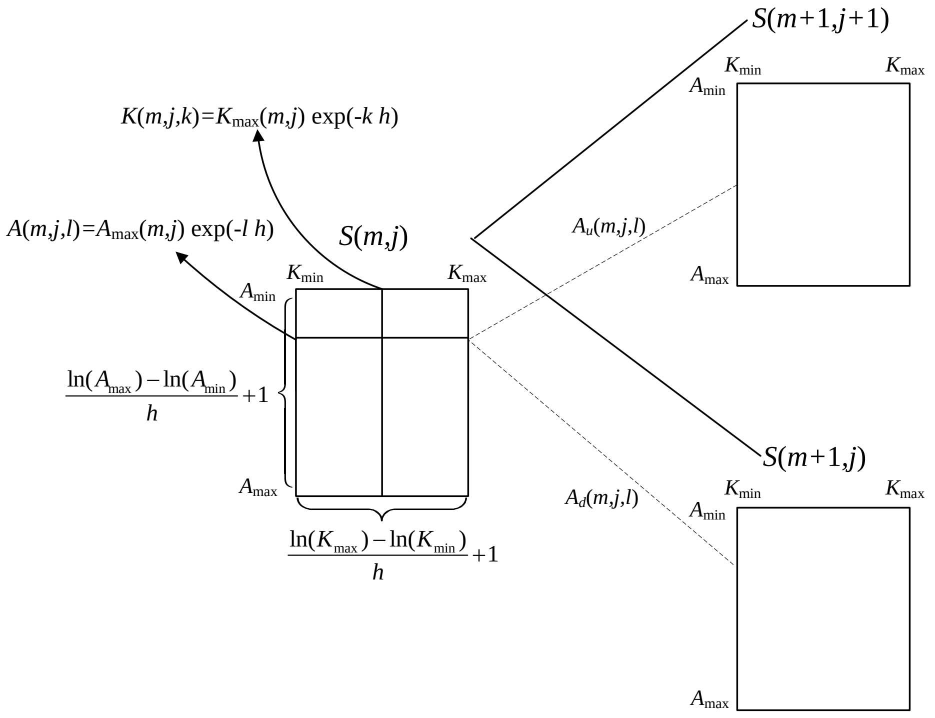
  (ii) For time points just before the reset time points, i.e., $(n_i - 1)\Delta t$, the state variables at the next time point is $(K_{t+\Delta t},\ A_{t+\Delta t}) = (\min(K_t,\ G(A_t,\ S_{t+\Delta t})),\ G(A_t,\ S_{t+\Delta t}))$, where $G(A_t, S_{t+\Delta t})$ is an updating function for the arithmetic average price, which returns $A_{t+\Delta t}$ given A_t and $S_{t+\Delta t}$.

  (iii) For time points other than those in (i) and (ii), only update the arithmetic average price such that the state variables at the next time point is $(K_{t+\Delta t},\ A_{t+\Delta t}) = (K_t, G(A_t,\ S_{t+\Delta t}))$.

- The data structure of each node:

  Representative values for A (and K) are logarithmically equally-spaced placed with the difference h between the maximum and minimum arithmetic average prices (and the maximum and minimum strike prices) for each node.

Figure 10-5



- The updating function for the arithmetic average price, $G(A_t,\ S_{t+\Delta t})$:

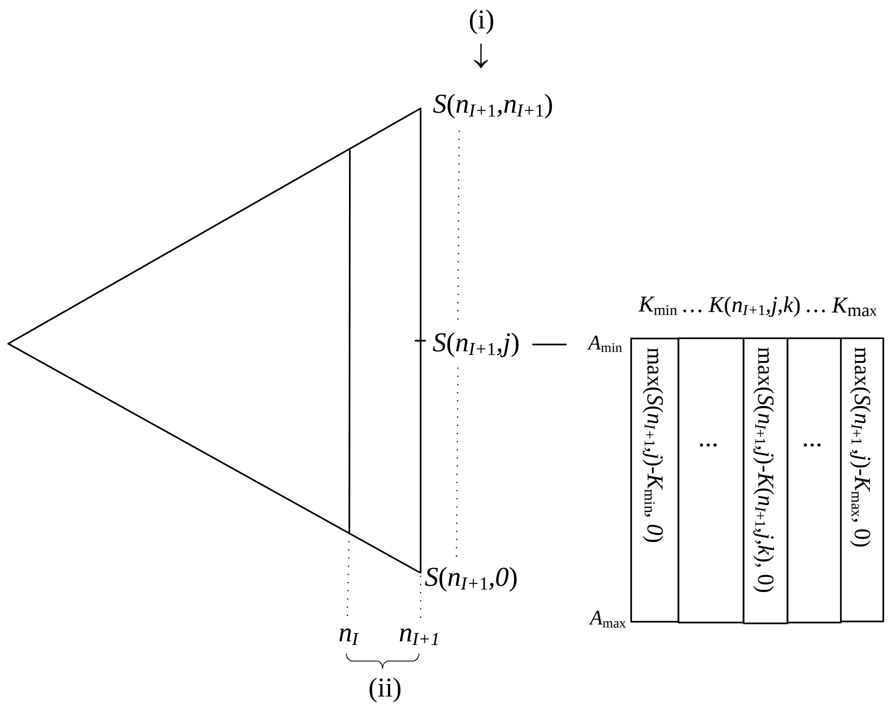
  For $A(m,j,l)$ and $n_i < m < n_{i+1}$

$$\begin{cases} A_u(m,j,l) = [(m-n_i)A(m,j,l) + S(m+1,j+1)]/(m-n_i+1) \\ A_d(m,j,l) = [(m-n_i)A(m,j,l) + S(m+1,j)]/(m-n_i+1) \end{cases}$$

- Backward induction

  (i) Decide the payoff for each pair of (K,A) on terminal nodes. The payoff is $\max(S_T - K_T, 0)$, which is independent of the average variable A, so for each column with the same representative values of K, the payoff is the same (see Figure 10-6).

Figure 10-6



(i)
S(n_{I+1},n_{I+1})
S(n_{I+1},j)
S(n_{I+1},0)
n_I
n_{I+1}
(ii)
K_min … K(n_{I+1},j,k) … K_max
A_min
A_max
max(S(n_{I+1},j)-K_min, 0)
…
max(S(n_{I+1},j)-K(n_{I+1},j,k), 0)
…
max(S(n_{I+1},j)-K_max, 0)

(ii) For $m = n_I, n_I+1, n_I+2, \ldots, n_{I+1}-1$,

$$V(m,j,K(m,j,k),A(m,j,l)) = [P_u \cdot V(m+1,j+1;K(m,j,k),A(m,j,l)) + P_d \cdot V(m+1,j;K(m,j,k),A(m,j,l)]e^{-r\Delta t}$$

(For the time period between $(n_I+1)\Delta t$ and $(n_{I+1}-1)\Delta t$, the strike price K will not be reset, and the arithmetic average A will not change either at the next time point. Therefore, it is only necessary to find option values at the next time point with state variable (K,A) identical to the values of $K(m,j,k)$ and $A(m,j,l)$.)

(iii) If $m\Delta t$ is one of the reset dates for $m = n_1, n_2, \ldots, n_{I-1}$,

$$V_{reset}(m,j;K(m,j,k),A(m,j,l)) = [P_u \cdot V(m+1,j+1;K(m,j,k),S(m+1,j+1))+ \\ P_d \cdot V(m+1,j;K(m,j,k),S(m+1,j))]e^{-r\Delta t}$$

(Since $K(m,j,k)$ represents the strike price after the reset, the strike price K will not change at the next time point. Therefore, find option values with the state variable K which is identical to the value of $K(m,j,k)$. As to the average state variable A, because the calculation of the arithmetic average price will restart at the next time point, find option values with the state variable A which is equal to the stock prices of the following child nodes.)

(iv) If m is the time point just before the reset date,

$$\begin{aligned} &V(m,j;K(m,j,k),A(m,j,l)) \\ &\quad =[P_u \cdot V_{reset}(m+1,j+1;\min(K(m,j,k),A_u(m,j,l)),A_u(m,j,l)) \\ &\qquad + P_d \cdot V_{reset}(m+1,j;\min(K(m,j,k),A_d(m,j,l)),A_d(m,j,l))]e^{-r\Delta t} \end{aligned}$$

(First, the arithmetic average price will be updated to be $A_u(m,j,l)$ for the upper child node and $A_d(m,j,l)$ for the lower child node. Second, the both strike prices are reset to be the minimums between $K(m,j,k)$ and $A_u(m,j,l)$ for the upper child node and $K(m,j,k)$ and $A_d(m,j,l)$ for the lower child node.)

(v) For values of m other than those in cases (i), (ii), (iii), and (iv),

$$V(m,j;K(m,j,k),A(m,j,l)) = [P_u \cdot V(m+1,j+1;K(m,j,k),A_u(m,j,l)) \\ +P_d \cdot V(m+1,j;K(m,j,k),A_d(m,j,l))]e^{-r\Delta t}$$

(Since the strike price will not be reset at the next time point, it is only necessary to take the update of the arithmetic average price into account. So, find option values with the state variable (K,A) to be $(K(m,j,k),A_u(m,j,l))$ for the upper child node and $(K(m,j,k),A_d(m,j,l))$ for the lower child node.)

- During the backward induction process, if there are no matched representative arithmetic average price and strike price, find the adjacent representative arithmetic average prices and adjacent representative strike prices to contain the target arithmetic average price and strike price. Then apply the two-dimensional linear interpolation to derive the corresponding option price.

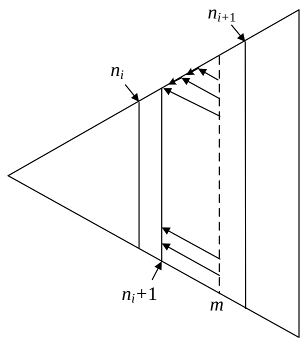
- In addition to the above algorithm of the backward induction, it is also important to decide $K_{\min}$, $K_{\max}$, $A_{\min}$, and $A_{\max}$ for each node. In fact, it is necessary to derive $A_{\min}$ and $A_{\max}$ for each node first, then to determine $K_{\min}$ and $K_{\max}$ for the nodes at the time points just before the reset dates, and finally to derive $K_{\min}$ and $K_{\max}$ for other nodes following a backward inheritance process.

$\odot$ For $n_i + 1 \leq m \leq n_{i+1}$, and $i = 0, 1, \ldots, I-1$,

$$A_{\max}(m,j) = \begin{cases} [S(m,j) + S(m-1,j) + \cdots + S(n_i+1,j)]/(m-n_i) & \text{if } j \leq n_i + 1 \\ \\ \{[S(m,j) + S(m-1,j) + \cdots + S(j,j)]+ \\ \quad [S(j-1,j-1) + S(j-2,j-2) + \cdots + S(n_i+1,n_i+1)]\} \\ \quad /(m-n_i) & \text{if } j > n_i + 1 \end{cases}$$

(For the upper case, trace the upper parent node backward until $m = n_i + 1$. For the lower case, trace the upper parent node backward first. Once reaching the uppermost node of the tree, trace the lower parent node backward until $m = n_i + 1$.)

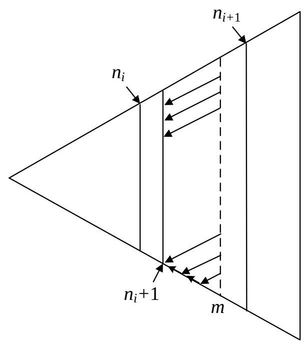
Figure 10-7



$$A_{\min}(m,j) = \begin{cases} [S(m,j) + S(m-1,j-1) + \cdots + \\ \quad S(n_i+1, j-m+n_i+1)]/(m-n_i) & \text{if } j \geq m - n_i - 1 \\ \\ \{[S(m,j) + S(m-1,j-1) + \cdots + S(m-j,0)]+ \\ \quad [S(m-j-1,0) + \cdots + S(n_i+1,0)]\} \\ \quad /(m-n_i) & \text{if } j < m - n_i - 1 \end{cases}$$

(For the upper case, trace the lower parent node backward until $m = n_i + 1$. For the lower case, trace the lower parent node backward first. Once reaching the lowermost node of the tree, trace the upper parent node backward until $m = n_i + 1$.)

Figure 10-8



⊙ For $m = n_{i+1} - 1$, and $i = 1, 2, \ldots, I$,

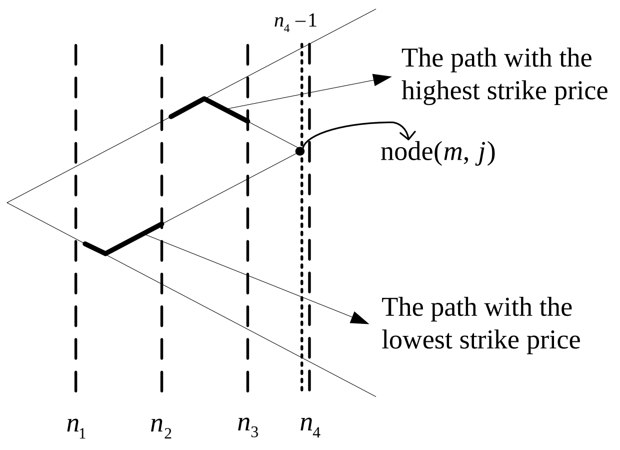
$K_{\max}(m, j) = \min(A_{\max}(n_i, \min(j, n_i)), K_0)$,

where the outside minimum operator is to ensure the possible strike price after resets must be smaller than K_0.

$$K_{\min}(m,j) = \begin{cases} \min(A_{\min}(n_{q-1}, 0), K_0) & \text{if } q < i+1 \\ \\ \min(A_{\min}(n_q, j - (m - n_q)), K_0) & \text{otherwise} \end{cases},$$

where q is chosen to satisfy $n_{q-1} \le m - j < n_q$.

Figure 10-9



$n_4 - 1$
The path with the highest strike price
node(m, j)
The path with the lowest strike price
n_1
n_2
n_3
n_4

⊙ For the time points $n_{i+1} - 2, n_{i+1} - 3, \ldots, n_i$, the $K_{\min}$ and $K_{\max}$ for each node at these time points can be determined backward given the $K_{\min}$ and $K_{\max}$ for each node at the time point of $n_{i+1} - 1$:

$$\begin{cases} K_{\min}(m,j) = K_{\min}(m+1, j+1) \text{ (inherit from the upper child node)} \\ K_{\max}(m,j) = K_{\max}(m+1, j) \text{ (inherit from the lower child node)} \end{cases}$$

⊙ The method proposed by Kim, Chang, and Byun (2003) to determine $K_{\min}$ and $K_{\max}$ for each node is complicated. In fact, $K_{\min}$ and $K_{\max}$ for each node can be set to be 0 and K_0, respectively. Because the strike price is reset downward, the maximum value for $K_{\max}$ of all nodes must be K_0. In addition, since the stock price cannot be negative, it is impossible that the minimum value for $K_{\min}$ becomes negative, and thus we can set $K_{\min}$ for each node to be 0.

⊙ The above alternative by setting $K_{\min}$ and $K_{\max}$ to be globally minimum and maximum for each node is much simpler. However, the larger difference between $K_{\min}$ and $K_{\max}$ will increase the number of representative strike prices for each node and in turn cause the heavier usage of the memory space and the CPU power to calculate the option value.