Modeling Path-Dependent Options

Computational Finance Workshop

Yidong Liu
Department of Computer Science
University of Toronto

yidong@cs.toronto.edu

Thesis Supervisor
Ken Jackson
krj@cs.toronto.edu

Outline

Modeling Path-Dependent Options

1. Path-Dependent Options.

2. Binomial Tree Methods.

3. Implementation of Hull and White Tree Method for Asian Options.

4. Implementation of an Extended Tree Method for European-style Asian Options.

5. Implementation of an Extended Tree Method for American-style Asian Options.

6. Conclusion.

The PS-file of the thesis is available.

1. Path-Dependent Options

A **path-dependent option** is an option whose payoff at exercise or expire date depends, in some non-trivial way, on the past history of the underlying asset price as well as its price at exercise or expiry.

American option, **lookback option**, **Asian option** ...

Factors affecting the average:

1. **The period of averaging.**
2. **Arithmetic or geometric averaging.** $(\Sigma_{i=1}^{N} S_i/N)$ or $(\Pi_{i=1}^{N} S_i)^{1/N})$.
3. **Weighted or unweighted averaging.**
4. **Discrete or continuous sampling**

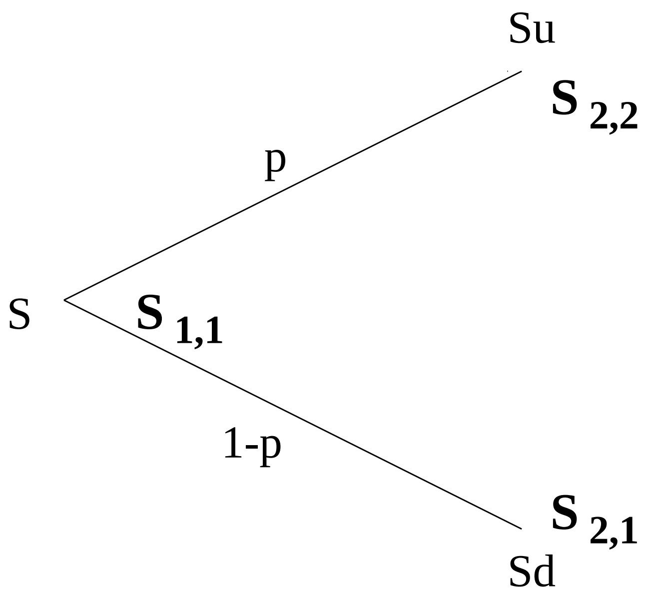
Issues:

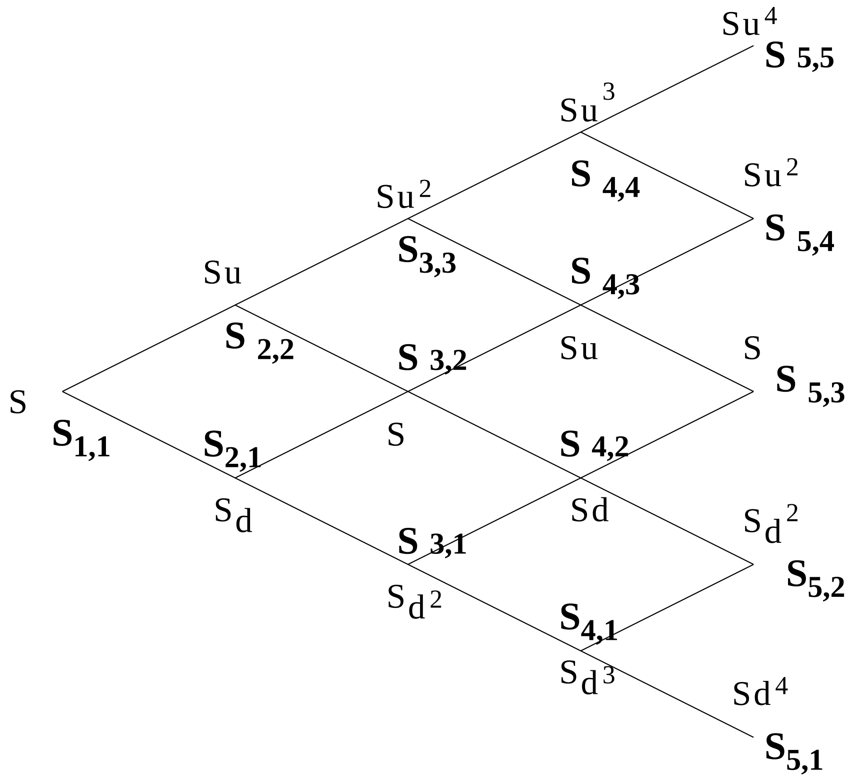
1. The Number of paths increases exponentially,
2. The average is not log-normally distributed.

Methods: Monte Carlo methods, Partial Differential Equations and <u>Tree Methods</u>.

Continuous Asian options with simple arithmetic averages over different averaging periods.

2. Binomial Tree Methods





1. The underlying asset S log-normally distributed,

$$dS = Srdt + S\sigma dX \tag{1}$$

2. First Moment (expected return):

$$Se^{r\Delta t} = pSu + (1-p)Sd \tag{2}$$

3. Second Moment (variance):

$$\begin{aligned}(3)\quad & S^2 e^{2r\Delta t}(e^{\sigma^2\Delta t} - 1) \\ = \;& pS^2u^2 + (1-p)S^2d^2 - S^2[pu + (1-p)d]^2\end{aligned}$$

4. Another frequently used condition

$$u = \frac{1}{d} \quad \text{or} \quad ud = 1. \tag{4}$$

5. Solving (2), (3) and (4) approximately

$$u = e^{\sigma\sqrt{\Delta t}}, \tag{5}$$

$$d = e^{-\sigma\sqrt{\Delta t}}, \tag{6}$$

$$p = \frac{a-d}{u-d}, \tag{7}$$

where $a = e^{r\Delta t}$.

3. Implementation of a Tree Method for Asian Options (Hull and White Approach)

3.1 Payoff.

(8) $$\max\left[\frac{\Sigma_{i=1}^{N+1} S_{i,k_i}}{N+1} - K,\ 0\right],$$

at Maturity. The exercise price K, and path, $(k_1, k_2, ..., k_{N+1})$.

(9) $$\max\left[\frac{\Sigma_{i=1}^{I} S_{i,k_i}}{I} - K,\ 0\right],$$

3.2 Maximum and Minimum Averages at Each Level.

(10) $$A_{i,max} = \frac{\Sigma_{j=1}^{i} S_{j,j}}{i} = \frac{\Sigma_{j=1}^{i} Su^{j-1}}{i},$$

(11) $$A_{i,min} = \frac{\Sigma_{j=1}^{i} S_{j,1}}{i} = \frac{\Sigma_{j=1}^{i} Sd^{j-1}}{i}.$$

3.3 Approximate Averages at Each Level.

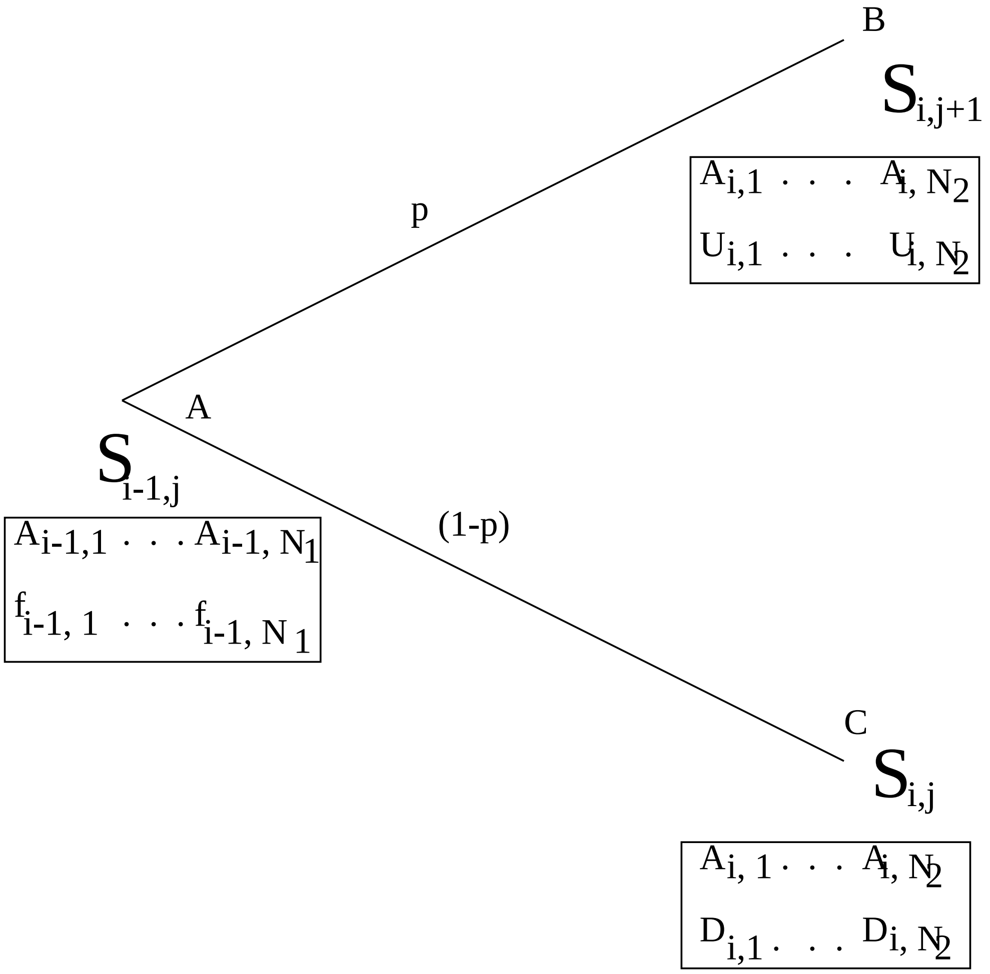
$[A_{i,min},\ A_{i,max}]$,

(12) $$Se^{kh},\ k = M_{min}, ..., M_{max},$$

The current price S, a partition parameter h, and $[Se^{M_{min}h},\ Se^{M_{max}h}] \supseteq [A_{i,min},\ A_{i,max}]$.

3.4 Compute Maximum and Minimum Averages at Each Level Efficiently.

3.5 Compute Option Prices Backward.



One Step of the Backward Computation in a Binomial Tree.

3.5.1 Interpolate Payoff for Each Approximate Average.

(13) $A^*_{i,k} = \dfrac{(i-1)A_{i-1,k} + S_{i,j+1}}{i}, \; k = 1, ..., N_1$

where $A_{i,k_1} \le A^*_{i,k} \le A_{i,k_2}$. Using linear interpolation produces $f^u_{i,k}$ from u_{i,k_1} and u_{i,k_2}.

(14) payoff of $A_{i,k} = \begin{cases} f^u_{i,k}, & S_{i-1,j} \text{ moves up} \\ f^d_{i,k}, & S_{i-1,j} \text{ moves down} \end{cases}$

3.5.2 Discount Backward. For a European-style Asian option

$$(15) \qquad f_{i-1,k} = e^{-r\Delta t}(pf^u_{i,k} + (1-p)f^d_{i,k}), \quad k = 1, ..., N_1,$$

where r is the risk-free interest rate.

For an American-style Asian call option,

$$(16) \quad f_{i-1,k} = \max[A_{i-1,k} - K, \; e^{-r\Delta t}(pf^u_{i,k} + (1-p)f^d_{i,k})], \quad k = 1, ..., N_1.$$

Similarly, for an American-style Asian put option.

4. An Extended Tree Method for Asian Options

4.1 Payoff of the European-style Asian Call Option.

$$\max\left[\frac{\Sigma_{l=I}^{N+1} S_{l,k_l}}{N_{avg}+1} - K,\ 0\right], \tag{17}$$

The exercise price K and path $(k_I, k_{I+1}, ..., k_{N+1})$.

$$e^{-rT} E\left[\max\left\{\frac{\Sigma_{l=I}^{N+1} S_{l,k_l}}{N_{avg}+1} - K, 0\right\}\right], \tag{18}$$

4.2 Direct Recursion. A European-style Asian option with exercise price K, maturity T_{avg}, current underlying asset price $S_{I,j}$ and averaging period T_{avg}.

It results in much duplication of computation since the common approximate averages of different nodes are not used.

4.3 Extended Tree Method.

4.3.1 Compute Maximum and Minimum Averages at Each Level.

4.3.2 Partition the Interval $[A_{k,min},\ A_{k,max}]$ of Each Level Using Se^{mh}.

4.3.3 Determine Option Prices at the Last Level.

4.3.4 Discount Backward in the Same Way as Hull and White Tree Method.

4.3.5 Compute a Single Option Price at Each Node of Level I.

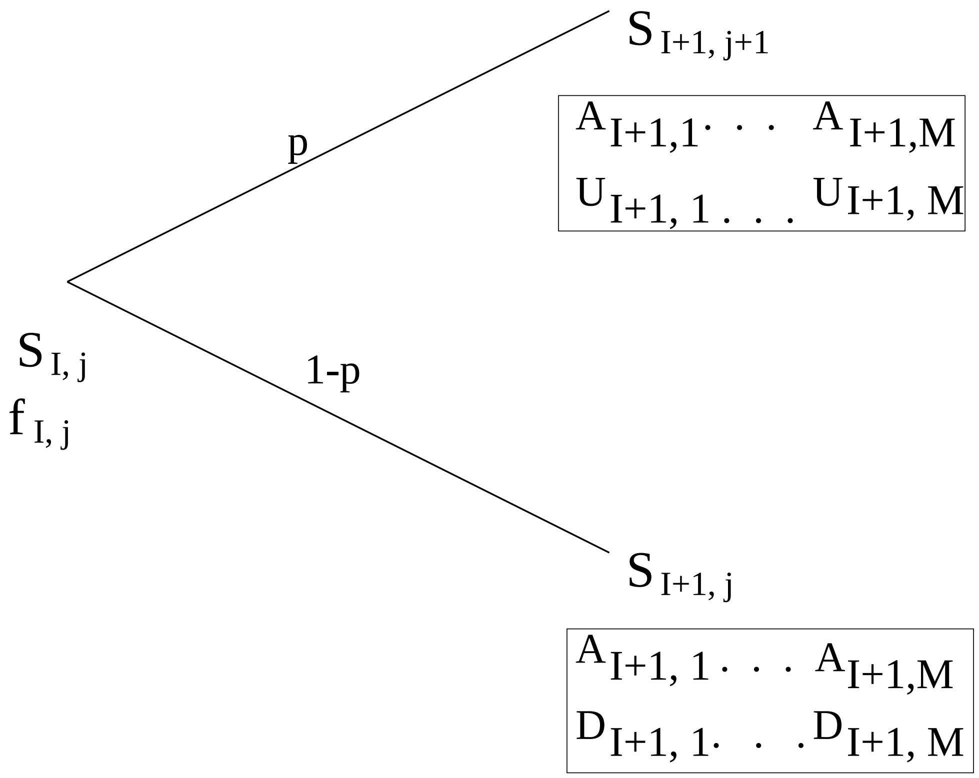


FIG. 1. *Computing Option Price at Each Node of Level I.*

$S_{I,j}$ may either move down to $S_{I+1,j}$ or up to $S_{I+1,j+1}$. Average $(S_{I,j} + S_{I+1,j})/2$ or $(S_{I,j} + S_{I+1,j+1})/2$ will determine the payoff and the option price.

4.3.6 Discount Backward in the First Part.

5. American-Style Asian Call Options

First Kind. Can not be exercised for $t < T_{avg}$ and its payoff at any time $t \geq T_{avg}$ is

$$\max\left[\frac{\Sigma_{j=J_1}^{J_2} S_{j,k_j}}{J_2 - J_1 + 1} - K,\ 0\right], \tag{19}$$

where $(k_{J_1}, ..., k_{J_2})$ is the path of the underlying asset process, and J_1 and J_2 satisfy

$$(J_2 - 1)\Delta t = t, \tag{20}$$

$$(J_1 - 1)\Delta t = t - T_{avg}. \tag{21}$$

Second Kind. Can be exercised any time. If the exercise time $t \geq T_{avg}$, then its payoff is (19). But, if the exercise time $t < T_{avg}$, then its payoff is

$$\max\left[\frac{\Sigma_{j=1}^{J} S_{j,k_j}}{J} - K,\ 0\right], \tag{22}$$

where $(k_1, k_2, ..., k_J)$ is the path of the underlying asset process and J satisfies

$$(J - 1)\Delta t = t. \tag{23}$$

5.1 Build the Tree.

5.1.1 Build a Regular Binomial Tree

5.1.2 Compute the Maximum and Minimum Averages at Each Node in the First Part.

5.1.3 Compute the Maximum and Minimum Averages at Each Node in the Second Part.

5.1.4 Approximate the Averages between the Maximum and Minimum Averages at Each Node with Se^{mh}.

5.2 Determine Beginning Nodes.

5.2.1 Beginning Nodes for the Maximum and Minimum Averages.

5.2.2 Possible Beginning Nodes for an Approximate Average.

5.2.3 Determine the Beginning Node for Each Approximate Average.

Middle node $S_{k-I,mid}$

$$mid = \left[\frac{max + min}{2}\right]. \tag{24}$$

$$Se^{M_1 h} < \cdots < Se^{mh} < \cdots < Se^{M_2 h} \tag{25}$$

$Se^{mh} \in [A^{min}_{k-I,mid},\ A^{max}_{k-I,mid}]$ begins with $S_{k-I,mid}$.

$Se^{mh} \in [A^{min}_{k-I,mid-1},\ A^{min}_{k-I,mid})$ begins with $S_{k-I,mid-1}$.

$Se^{mh} \in (A^{max}_{k-I,mid},\ A^{max}_{k-I,mid+1}]$ begins with $S_{k-I,mid+1}$

5.3 Computing Option Prices Backward.

5.3.1 Option Price at the Last Level.

5.3.2 Discount Backward at the Second Part of the Tree.

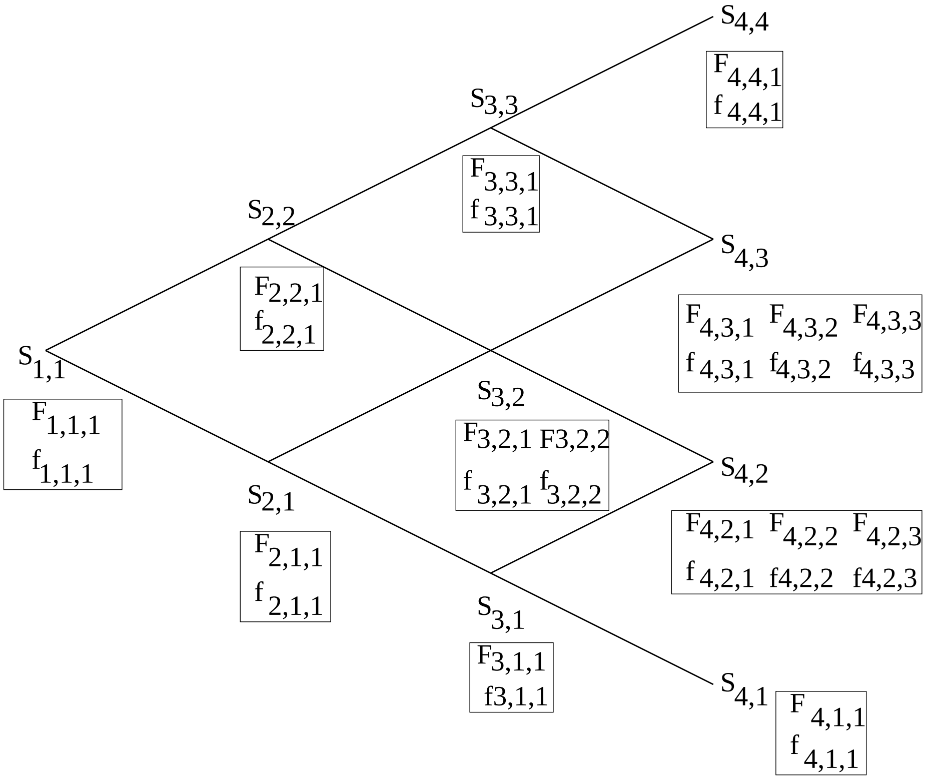
$$(26)\ A^*_{k-1,l} = \frac{IA_{k-1,l} - S^{begin}_{k-1-I,l} + S_{k,j+1}}{I},$$
$$l = 1, ..., N_1,$$

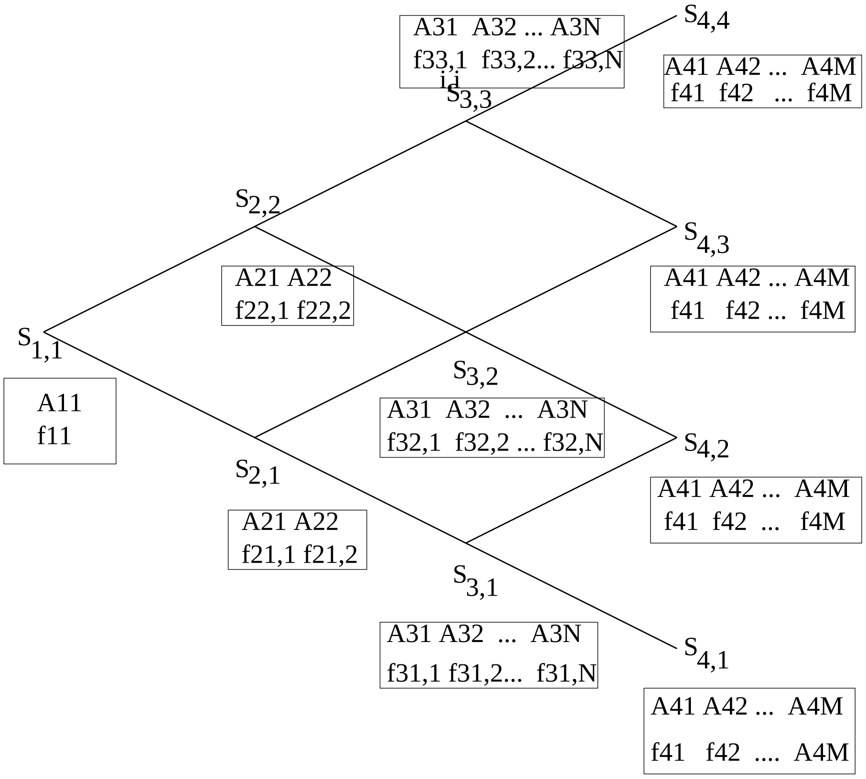
5.3.3 Discount Backward at the First Part of the Tree.

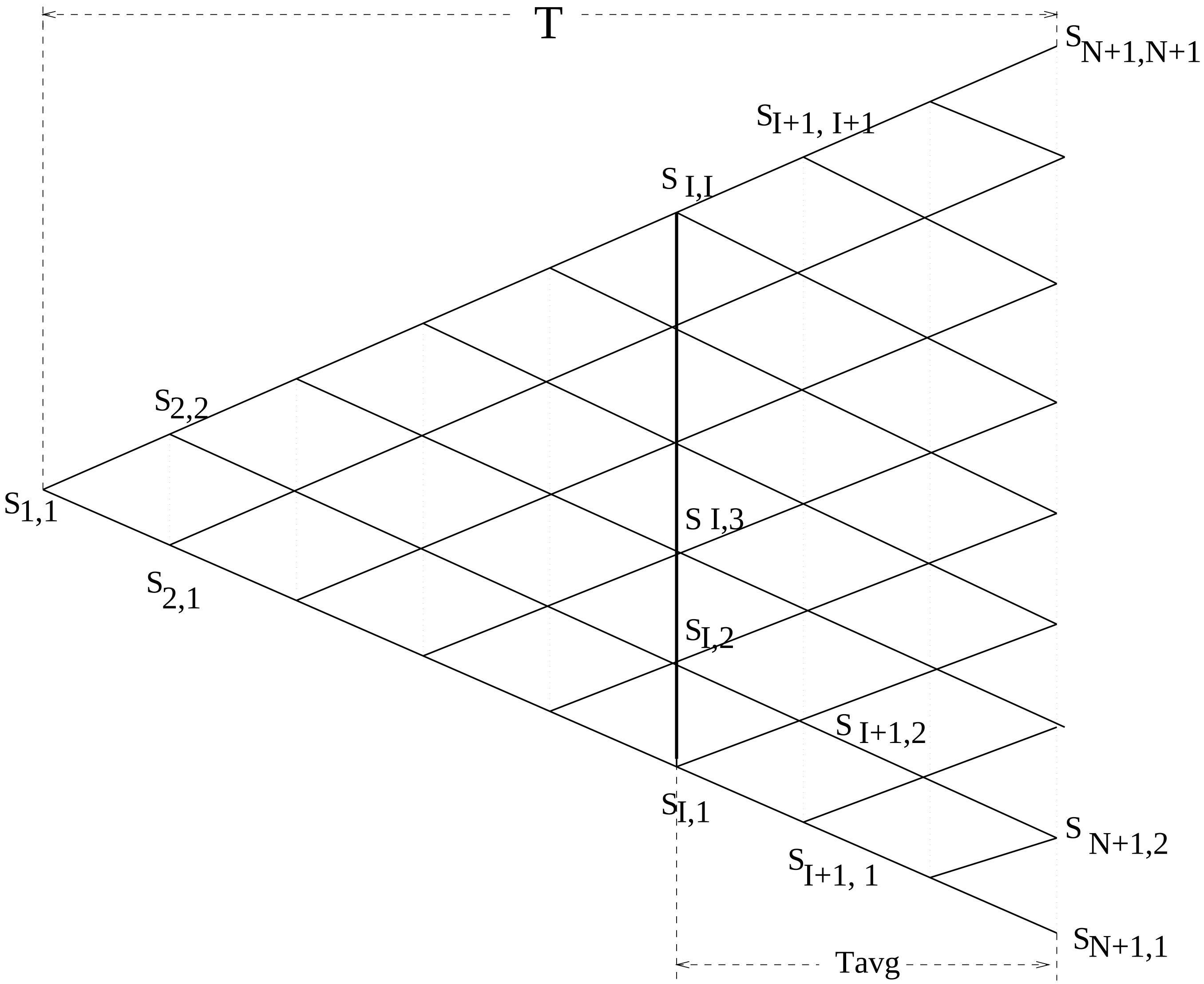
$$(27)\quad A^*_{k-1,l} = \frac{(k-1)A_{k-1,l} + S_{k,j+1}}{k},$$
$$l = 1, ..., N_2,$$

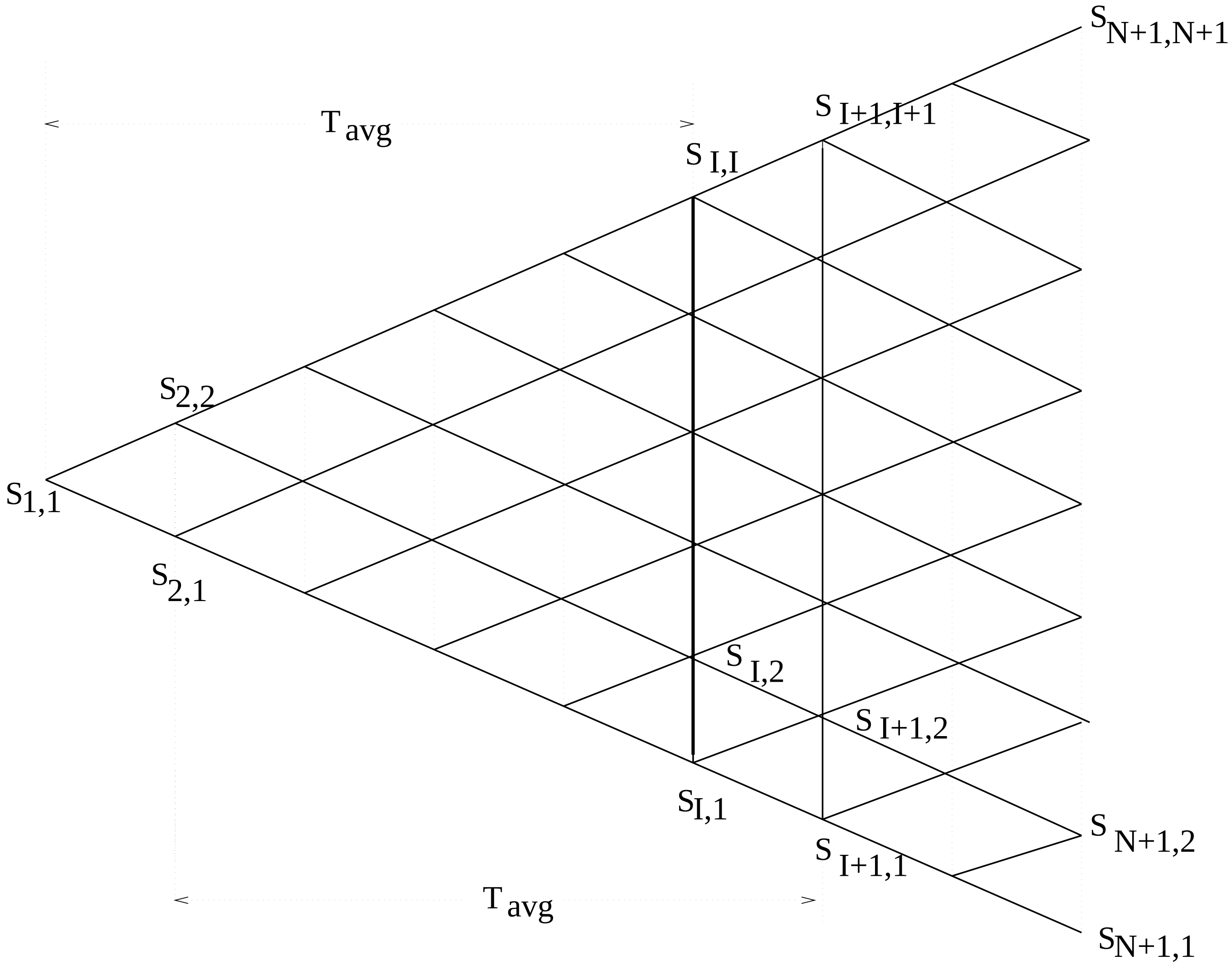
6. Conclusion

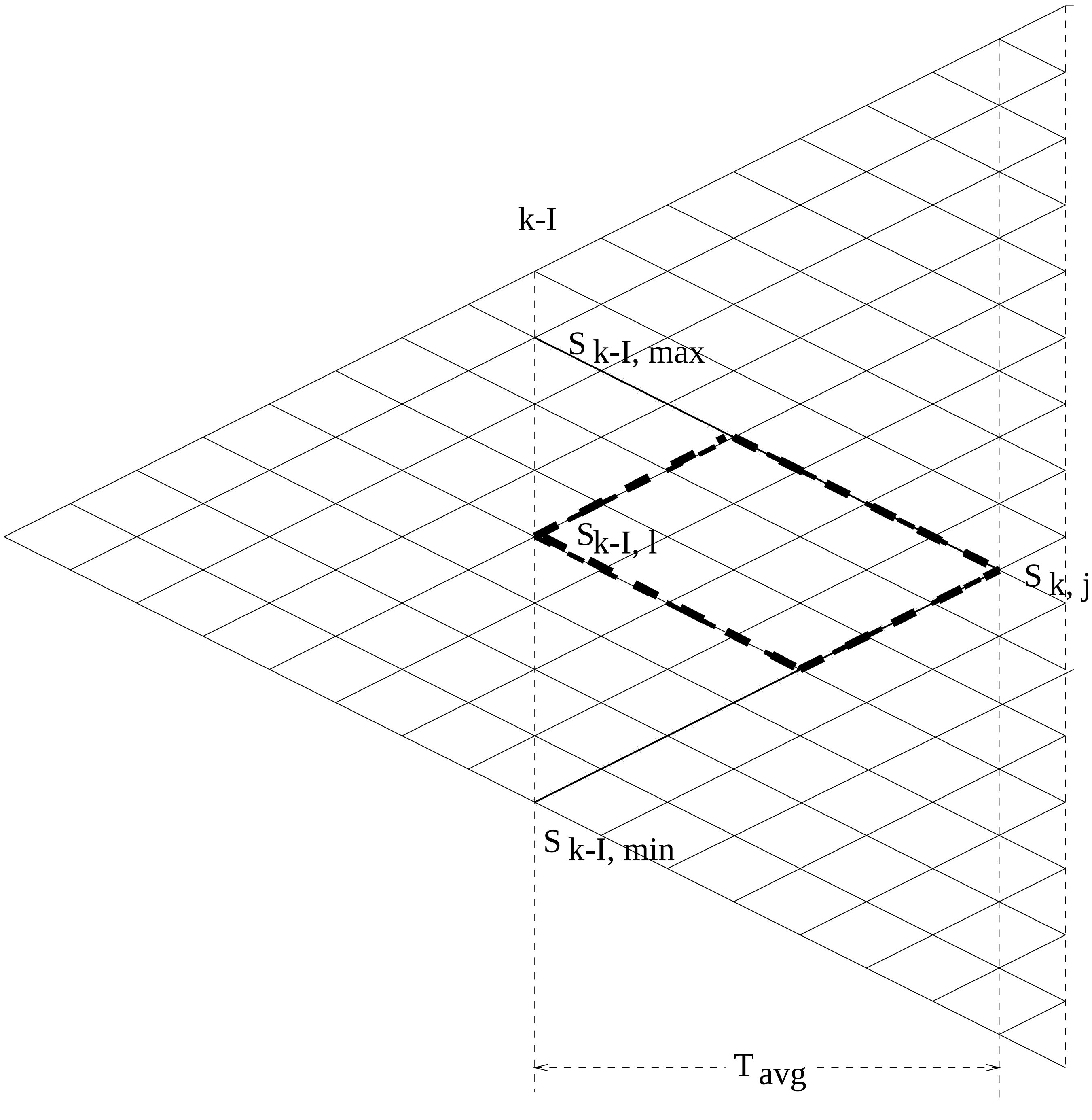
1. Our method can price general Asian options (both European-style and American-style).
2. Numerical testing confirms that Hull and White tree method is a special case of our method.
3. Our results for European-style Asian options are consistent with the results of Monte Carlo simulations and our method is more efficient.
4. Since we use different data structures in implementing our tree method, our method is more efficient than the straight-forward Hull and White tree method.
5. For American-style Asian options whose averages are taken over the last period, we have not yet found an efficient method with which to compare the results from our method.
6. Further work should be done in determining the beginning nodes.
7. We plan to extend our method to other path-dependent options.
8. We hope to analyze the method.


S4,4
F4,4,1
f 4,4,1
S3,3
F3,3,1
f 3,3,1
S2,2
F2,2,1
f2,2,1
S4,3
F4,3,1 F4,3,2 F4,3,3
f 4,3,1 f4,3,2 f4,3,3
S1,1
F1,1,1
f1,1,1
S3,2
F3,2,1 F3,2,2
f 3,2,1 f3,2,2
S4,2
F4,2,1 F4,2,2 F4,2,3
f 4,2,1 f4,2,2 f4,2,3
S2,1
F2,1,1
f 2,1,1
S3,1
F3,1,1
f3,1,1
S4,1
F 4,1,1
f 4,1,1


A31 A32 ... A3N
f33,1 f33,2... f33,N
S4,4
A41 A42 ... A4M
f41 f42 ... f4M
S3,3
S2,2
A21 A22
f22,1 f22,2
S4,3
A41 A42 ... A4M
f41 f42 ... f4M
S1,1
A11
f11
S3,2
A31 A32 ... A3N
f32,1 f32,2 ... f32,N
S4,2
A41 A42 ... A4M
f41 f42 ... f4M
S2,1
A21 A22
f21,1 f21,2
S3,1
A31 A32 ... A3N
f31,1 f31,2... f31,N
S4,1
A41 A42 ... A4M
f41 f42 .... A4M


T
S N+1,N+1
S I+1, I+1
S I,I
S 2,2
S 1,1
S I,3
S 2,1
S I,2
S I+1,2
S I,1
S N+1,2
S I+1, 1
S N+1,1
Tavg


$S_{N+1,N+1}$
$S_{I+1,I+1}$
T_{avg}
$S_{I,I}$
$S_{2,2}$
$S_{1,1}$
$S_{2,1}$
$S_{I,2}$
$S_{I+1,2}$
$S_{I,1}$
$S_{N+1,2}$
$S_{I+1,1}$
T_{avg}
$S_{N+1,1}$


k-I
S k-I, max
S k-I, l
S k, j
S k-I, min
T avg