数理解析研究所講究録
第1554巻 2007年 56-63

アメリカン・アジアンオプションの価格付けに対する計算幾何手法を用いた近似アルゴリズム

渋谷 彰信 (Akinobu Shibuya)　塩浦 昭義 (Akiyoshi Shioura)　徳山 豪 (Takeshi Tokuyama)
東北大学大学院 情報科学研究科
Graduate School of Information Sciences, Tohoku University

1 はじめに

オプションとは，金融市場で取引されている金融派生商品 (デリバティブ) の一つである．本論文では，アメリカン・アジアンオプションと呼ばれるオプションに対し，オプション価格の近似値を効率的に求めるアルゴリズムを提案する．

狭義には，オプションは株式などの原資産を，指定された時点もしくは期間に，決められた価格で売買する権利のことである．より広義には，オプションは原資産の価格の履歴に依存して決まる報酬 (ペイオフと呼ばれる) をオプションの所有者が得ることができる，という契約である．オプションは資産売買に伴うリスクのヘッジや，少ない投資で多額の利益を得るという目的のために使われている．様々なオプションが世界の金融市場で取引されている現在，オプションの適正な価格付けは重要な計算問題となっている．

本論文では，資産価格の変動を離散的に表現する確率モデルである2項モデル ([1] 参照) において，アメリカン・アジアンオプションの価格を計算する，という問題を扱う．オプションの価格は (利子を割り引いた) ペイオフの期待値として与えられるので，アメリカン・アジアンオプションのペイオフの期待値を計算する問題と言い換えることができる．この問題は非常に計算困難な問題であるが，その主な理由は次の2つである．

1つ目の理由は、このオプションがアジアン型であり，ペイオフの値が資産価格の過去の履歴に依存するので，厳密値の計算のためには資産価格の履歴の全ての可能性を列挙しなければならない，ということである．したがって，アジアンオプションの価格を厳密に計算するには指数時間が必要となる．現実的な問題を解くためにはオプション価格の近似値を高速に求めるアルゴリズムが求められる．

2つ目の理由は，アメリカンオプションでは任意の時期において権利行使が可能であるが，最適な行使時期の決定は計算困難である，ということである．例えば，モンテカルロ法はオプション価格の近似アルゴリズムとしてしばしば用いられ成功を収めているが，アメリカンオプションの場合にはその行使時期決定の問題からモンテカルロ法の適用には様々な工夫が必要とされる．

2項モデルにおけるアメリカン・アジアンオプションの近似アルゴリズムはHull–White [3], Neave [5], Chalasani ほか [4], Dai ほか [2] によって提案されている．しかし，いずれのアルゴリズムについても近似精度は実験的に検証されているだけであり，理論的な近似精度保証は与えられていない．アメリカン・アジアンオプションの価格付けに対して近似精度を理論的に保証した近似アルゴリズムは著者らの知る限りこれまでに提案されていない．

本論文では，アメリカン・アジアンオプションの価格付けに対し，近似精度を保証する多項式時間近似アルゴリズムを提案する．アメリカン・アジアンオプションの価格の厳密値は，各時点でのペイオフの期待値を表す区分線形関数を繰り返し計算することにより求めることが可能である．しかし，計算の過程で区分線形関数の複雑度は指数オーダーとなり，計算には指数時間が必要となる．これに対し，本論文では計算幾何学的な手法を用いて区分線形関数をより複雑度の小さい区分線形関数により近似する．これにより，近似精度を保証しつつアルゴリズムの高速化を計る．

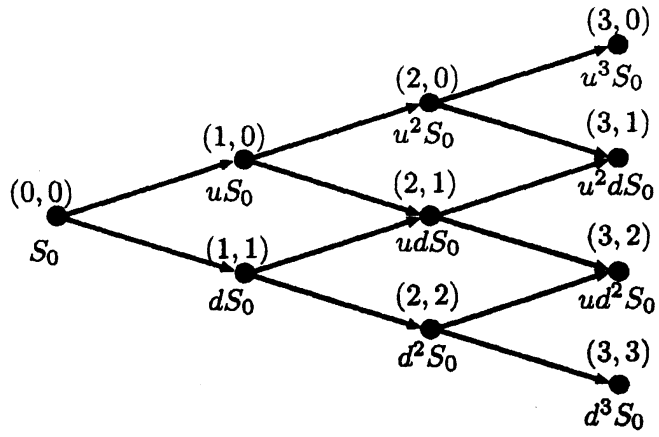

図 1: 2 項木と 2 項モデル．各ノードの上にはそのノードのラベル，下にはそのノードでの資産価格が書かれている．

2 準備

2 項モデル 深さ n の 2 項木とは，以下のように定義される閉路を持たない有向グラフのことである (図 1 参照)．深さ n の 2 項木は $n+1$ 個のレベルをもつ．第 i 番目 $(0 \leq i \leq n)$ のレベルには $i+1$ 個のノードがあり，各ノードは (i,j) $(0 \leq j \leq i)$ のようなラベルをもつ．第 i レベル $(i < n)$ の各ノード (i,j) は 2 つのノード $(i+1,j)$, $(i+1,j+1)$ への有向枝をもつ．

2 項モデルは，資産価格の変動を離散的に表現した確率モデルであり，資産価格の変動は 2 項木を使って表現される (図 1 参照)．2 項モデルでは，現時点からオプションの満期までの期間を n 期間に分割し，現時点を第 0 期，分割された各期間の終了時を順に第 1 期，第 2 期，...，第 n 期と呼ぶことにする．第 n 期はオプションの満期に対応する．既知である現時点の資産価格を S_0 とおき，各 $i = 1,2,\ldots,n$ に対し第 i 期の資産価格を S_i という確率変数により表す．2 項モデルでは，各期において資産価格は確率 p で uS に上がる，もしくは確率 $1-p$ で dS に下がる，という 2 通りのみを考える．ここで u と d は定数であり，$u > d$ および $u = 1/d$ を満たす．2 項モデルでは，2 項木の第 i レベルの頂点により第 i 期の資産価格を表す．第 i 期 $(i < n)$ のノード (i,j) における資産価格が S であるとすると，資産価格が uS に増加するときはノード $(i+1,j)$ に移動し，資産価格が uS に減少するときはノード $(i+1,j+1)$ に移動する．したがって，ノード (i,j) における資産価格は $S_{i,j} = u^{i-j}d^jS_0$ と表せる．

オプション オプションは，株式などの原資産を，指定された時点もしくは期間に，決められた価格で売買する権利のことである．原資産を指定価格で買う権利のことをコールオプション，指定価格で売る権利のことをプットオプションと呼ぶ．本論文ではコールオプションに限定して話を進めるが，本論文の主結果はプットオプションについても同様に成り立つ．

もっとも基本的なオプションであるヨーロピアンオプションは，満期と呼ばれる将来の決められた時点において，行使価格と呼ばれる価格で原資産を買う権利のことである．行使価格が X のとき，満期における資産価格 S_n が X より大きい場合には，オプションを行使して資産を価格 X で購入し，直ちに市場価格 S_n で資産を売ることにより $S_n - X$ の利益を得る．したがって，ヨーロッパオプションにより得られるペイオフは $\max\{S_n - X, 0\}$ と表せる．

一方，アメリカンオプションでは満期までの任意の時点において権利行使が可能である．ヨーロピアンオプションと同様の考え方により，第 i 期においてアメリカンオプションを行使すると $\max\{S_i - X, 0\}$ というペイオフが得られる．満期以外の期においてオプションを行使することを

早期行使というが，アメリカンオプションでは各期において権利を早期行使するか否かの決定をする必要がある．そのためには，権利を早期行使した場合のペイオフと早期行使しなかった場合のペイオフの期待値を計算し，比較する必要があるが，ペイオフの期待値の計算は本質的にオプション価格の計算と等価な問題であり，非常に困難である．

次にアジアンオプションについて説明する．オプションの行使価格が X のとき，先に紹介した2つのオプションのペイオフは権利行使時点での資産価格 S_i に依存していたが，アジアンオプションのペイオフ値は過去の資産価格の平均値 $A_i = (\sum_{k=0}^{i} S_k)/(i+1)$ に依存する．例えば，ヨーロピアン・アジアンオプションの場合には権利は満期においてのみ行使可能であり，そのペイオフは $\max\{A_n - X, 0\}$ となる．また，アメリカン・アジアンオプションの場合には任意の時点で権利を行使可能であり，第 i 期で行使した場合にはペイオフが $\max\{A_i - X, 0\}$ となる．

3 オプション価格の厳密値の計算

アメリカン・アジアンオプションの価格の厳密値 U を計算するアルゴリズムについて説明する．

各ノード (i,j) に対し，根 $(0,0)$ から (i,j) に至るパスのノードにおける資産価格の合計値を $T = T_{i,j}$ とおく．ノード (i,j) でのペイオフの期待値は資産価格の合計値 T の関数として表せるが，これを $f_{i,j}(T)$ とおき，期待ペイオフ関数と呼ぶことにする．すると，オプション価格の厳密値 U は，ノード $(0,0)$ での期待ペイオフ関数の初期資産価格 $T = S_0$ における値 $f_{0,0}(S_0)$ として与えられる．したがって，オプション価格の厳密値を計算するためには，ノード $(0,0)$ における期待ペイオフ関数 $f_{0,0}(T)$ を計算すれば良い．本節で紹介するアルゴリズムは，各ノードの期待ペイオフ関数 $f_{i,j}(T)$ を第 n 期のノード，第 $n-1$ 期のノード，...，第0期のノード，という順に後進的に求めていき，最終的に $f_{0,0}(T)$ を求める．

まず，満期である第 n 期のノード (n,j) における期待ペイオフ関数 $f_{n,j}$ について考える．第 n 期では，権利行使するかしないかの2つの選択肢があるため，期待ペイオフ関数 $f_{n,j}$ は次のように与えられる．

$$f_{n,j}(T) = \max\left(0, \frac{T}{n+1} - X\right). \tag{1}$$

次に，第 i 期 $(i < n)$ のノード (i,j) における期待ペイオフ関数 $f_{i,j}$ について考える．第 i 期では，権利を早期行使するかしないかの2つの選択肢がある．権利を早期行使する場合のペイオフの期待値を $f^E_{i,j}(T)$，早期行使しない場合のペイオフの期待値を $f^P_{i,j}(T)$ と表す．すると，

$$f_{i,j}(T) = \max(f^E_{i,j}(T), f^P_{i,j}(T)) \tag{2}$$

が成り立つ．以下では $f^E_{i,j}(T)$ および $f^P_{i,j}(T)$ の計算方法を説明する．

権利を早期行使する場合のペイオフの期待値 $f^E_{i,j}$ については，第 n 期のノードの場合と同様の考え方により，次の式により与えられる．

$$f^E_{i,j}(T) = \max\left(0, \frac{T}{i+1} - X\right). \tag{3}$$

次に，権利を早期行使しない場合のペイオフの期待値 $f^P_{i,j}$ の計算方法を説明する．ノード (i,j) から確率 p でノード $(i+1,j)$ へ移動し，確率 $1-p$ でノード $(i+1,j+1)$ へ移動するので，$f^P_{i,j}$ の値は次のようになる．

$$f^P_{i,j}(T) = e^{-r\Delta t}\left[p f_{i+1,j}(T + uS_{i,j}) + (1-p) f_{i+1,j+1}(T + dS_{i,j})\right]. \tag{4}$$

ここで右辺の値は1期間分の利子 $e^{-r\Delta t}$ を割り引いたものになっている．

以上の式 (1), (2), (3), (4) を用いて再帰的に計算することにより，各期におけるノードの期待ペイオフ関数が得られる．

最後にこのアルゴリズムの時間および空間計算量について議論する．その定義により，期待ペイオフ関数 $f_{i,j}(T)$ は区分線形な凸関数であるが，アルゴリズムの計算量はその区分線形関数の線分の数に比例する．式 (2), (3), (4) を使って関数 $f_{i+1,j}(T), f_{i+1,j+1}(T)$ から関数 $f_{i,j}(T)$ を求める際，線分の数は最悪の場合2倍に増えてしまう．このことより，本節で紹介したアルゴリズムの時間計算量は $O(n2^n)$，空間計算量は $O(2^n)$ となる．

4 オプション価格の近似アルゴリズム

4.1 アルゴリズムの概要

本節では，オプション価格の近似値を求めるアルゴリズムを提案する．提案するアルゴリズムは，与えられた近似誤差パラメータ ε (>0) に対し，$U \leq \Phi \leq (1+\varepsilon)U$ を満たす近似値 Φ を求める．

提案する近似アルゴリズムは，前節で紹介した厳密値を求めるアルゴリズムを修正したものになっている．厳密値を求めるアルゴリズムでは，区分線形関数 $f_{i,j}(T)$ の線分の数が最悪の場合に指数オーダーとなっていた．提案する近似アルゴリズムでは，関数 $f_{i,j}(T)$ を求める代わりに，多項式個の線分をもつ区分線形関数 $g_{i,j}(T)$ により関数 $f_{i,j}(T)$ を近似する．これにより，多項式時間アルゴリズムを実現するとともに，求められた近似値の精度を保証する．

提案する近似アルゴリズムの流れは以下のようになる．まず，満期である第 n 期のノード (n,j) においては，$g_{n,j}(T) = f_{n,j}(T)$ とおく．次に，第 i 期 $(i<n)$ のノード (i,j) については，関数 $g^E_{i,j}(T), g^P_{i,j}(T)$ を

$$g^E_{i,j}(T) = f^E_{i,j}(T), \qquad g^P_{i,j}(T) = e^{-\frac{rn}{n}}\left[pg_{i+1,j}(T+uS_{i,j}) + (1-p)g_{i+1,j+1}(T+dS_{i,j})\right]$$

により定義し，

$$\widetilde{g}_{i,j}(T) = \max(g^E_{i,j}(T), g^P_{i,j}(T))$$

とおく．区分線形凸関数 $\widetilde{g}_{i,j}(T)$ を，線分の数の少ない区分線形関数で近似したものを $g_{i,j}(T)$ と定める．ここで用いる区分線形凸関数の近似方法については第4.2節で詳しく説明する．

このようにして後進的に $f_{i,j}(T)$ の近似関数 $g_{i,j}(T)$ を求めたら，関数値 $\Phi = g_{0,0}(S_0)$ をオプション価格の近似値として出力する．

4.2 計算幾何学手法を用いた区分線形凸関数の近似

以下では，区間 $[0,+\infty)$ 上で定義された，単調非減少かつ非負の値をとる区分線形凸関数 $\widetilde{g}(T)$ および誤差パラメータ δ $(0<\delta\leq 1)$ が与えられたとき，

$$\widetilde{g}(T) \leq g(T) \leq (1+\delta)\widetilde{g}(T) \qquad (0 \leq T < +\infty) \tag{5}$$

を満たす区分線形凸関数 $g(T)$ を求める2つの方法について説明する．

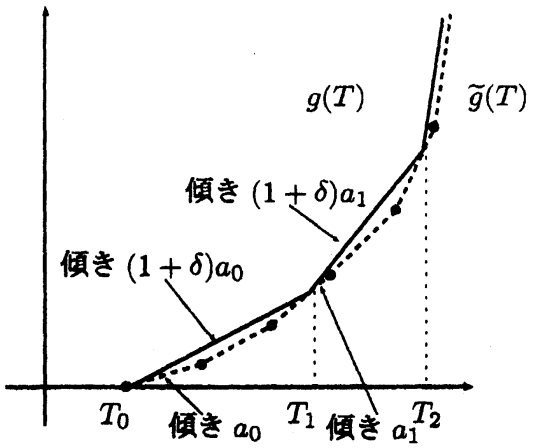

図 2: 近似方法その 1

4.2.1 近似方法その 1

まず，$\tilde{g}(T) > 0$ $(\forall T > T_0)$ を満たす最小の値 T_0 (≥ 0)，および $T = T_0$ における関数 $\tilde{g}(T)$ の右側微分係数 a_0 を求める (図 2 参照)．そして，点 $(T_0, \tilde{g}(T_0))$ を通過する傾き $(1+\delta)a_0$ の直線を描き，その直線と関数 $\tilde{g}(T)$ との交点を $(T_1, \tilde{g}(T_1))$ とする．

次に，$T = T_1$ おける関数 $\tilde{g}(T)$ の右側微分係数 a_1 を求める．そして，点 $(T_1, \tilde{g}(T_1))$ を通過する傾き $(1+\delta)a_1$ の直線を描き，その直線と関数 $\tilde{g}(T)$ との交点を $(T_2, \tilde{g}(T_2))$ とする．

この操作を繰り返し，k 回目 $(k \geq 0)$ の反復に入ったとき，点 $(T_k, \tilde{g}(T_k))$ を通過する傾き $(1+\delta)a_k$ の直線と関数 $\tilde{g}(T)$ が交点をもたなかったとする．このとき，関数 $g(T)$ を次の式により定義する．

$$g(T) = \begin{cases} 0 & (0 \leq T < T_0), \\ (1+\delta)a_0(T - T_0) + \tilde{g}(T_0) & (T_0 \leq T \leq T_1), \\ \vdots & \\ (1+\delta)a_{k-1}(T - T_{k-1}) + \tilde{g}(T_{k-1}) & (T_{k-1} < T \leq T_k), \\ a_{\max}(T - T_k) + \tilde{g}(T_k) & (T_k < T < +\infty). \end{cases}$$

ここで，関数 $\tilde{g}(T)$ の線分の傾きの最大値を $a_{\max}$ とおいた．この関数 $g(T)$ が条件 (5) を満たすことは，その定義より明らかである．

ここで，区分線形関数 $g(T)$ の線分の数について解析を行う．傾き $a_0, a_i, \ldots, a_k$ に対して

$$a_i > (1+\delta)a_{i-1} \qquad (i = 1, 2, \ldots, k)$$

が成り立つ．これより，$a_{\max} \geq a_k > (1+\delta)^k a_0$ が導かれる．ゆえに，$\log(1+\delta) > \delta/2$ $(0 \leq \forall\delta \leq 1)$ という事実を使うと，

$$k < \frac{2\log(a_{\max}/a_0)}{\delta}$$

という不等式が得られる．なお，a_0 は関数 $\tilde{g}(T)$ の線分の傾きのうち，非零のものの中での最小値である．

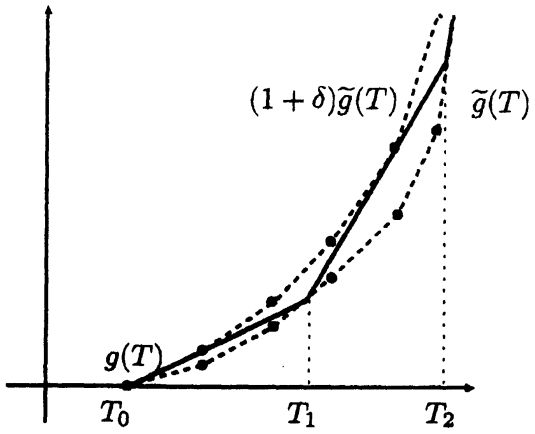

図 3: 近似方法その 2

4.2.2 近似方法その 2

この方法では，2 つの関数 $\tilde{g}(T)$ と $(1+\delta)\tilde{g}(T)$ に対し，この 2 つに挟まれるような区分線形関数を描く (図 3 参照)．そのとき，一つ一つの線分がなるべく長くなるように「貪欲」に関数を描いていく．

より具体的には以下の通りである．第 4.2.1 節と同様に T_0 を定める．点 $(T_0, f(T_0))$ から右側に線分を描くが，その傾き b_0 が最大になるようにする．すなわち，

$$b_0(T-T_0)+f(T_0) \le (1+\delta)\tilde{g}(T) \qquad (\forall T \ge T_0)$$

の条件のもとで最大の b_0 を求める．そして，直線 $b_0(T-T_0)+f(T_0)$ と $\tilde{g}(T)$ の (T_0 の右側にある) 交点を $(T_1, f(T_1))$ とする．

次に，点 $(T_1, f(T_1))$ から右側に線分を描くが，その傾き b_1 が

$$b_1(T-T_1)+f(T_1) \le (1+\delta)\tilde{g}(T) \qquad (\forall T \ge T_1)$$

の条件のもとで最大になるようにする．そして，直線 $b_1(T-T_1)+f(T_1)$ と $\tilde{g}(T)$ の (T_1 の右側にある) 交点を $(T_2, f(T_2))$ とする．

この操作を繰り返し，k 回目 $(k \ge 0)$ の反復に入ったとき，直線 $b_k(T-T_k)+f(T_k)$ と $\tilde{g}(T)$ が，T_k の右側において交わりをもたなかったとする．すると，$a_{\max} \le b_k \le (1+\delta)a_{\max}$ が成り立つ．このとき，関数 $g(T)$ を次の式により定義する．

$$g(T) = \begin{cases} 0 & (0 \le T \le T_0), \\ (1+\delta)b_0(T-T_0)+\tilde{g}(T_0) & (T_0 < T \le T_1), \\ \vdots & \\ (1+\delta)b_{k-1}(T-T_{k-1})+\tilde{g}(T_{k-1}) & (T_{k-1} < T \le T_k), \\ a_{\max}(T-T_k)+\tilde{g}(T_k) & (T_k < T < +\infty). \end{cases}$$

以下では，この近似方法で得られた区分線形関数 $g(T)$ の線分の数の最悪値オーダーが，第 1 の近似方法のオーダー $O(\log(a_{\max}/a_0)/\delta)$ に等しいことを示す．そのためには，

$$b_i > (1+\delta)b_{i-1} \qquad (i = 1, 2, \ldots, k-1) \tag{6}$$

を示せば十分である.

関数 $\tilde{g}(T)$ の $T=T_i$ での右側微分係数を α_i とおく. すると $\alpha_i > b_{i-1}$ が成り立つ. また, 線分 $(1+\delta)b_i(T-T_i)+\tilde{g}(T_i)$ $(T_i < T < T_{i+1})$ は, $T_i < T_* < T_{i+1}$ を満たすある T_* において関数 $(1+\delta)\tilde{g}(T)$ に接しているが, 関数 $(1+\delta)\tilde{g}(T)$ の $T=T_*$ での左側微分係数を $(1+\delta)\alpha_i'$ とおくと, $(1+\delta)\alpha_i' \le b_i$ が成り立つ. さらに, $T_i < T_*$ なので $\alpha_i \le \alpha_i'$ が成り立つ. 以上の不等式をまとめると, (6) が導かれる.

4.3 アルゴリズムの解析

第 4.1 節で説明した近似アルゴリズムに, 第 4.2 節で提案した区分線形関数の近似手法を組み合わせたときの, アルゴリズムの近似精度と時間計算量を解析する.

4.3.1 近似精度

第 4.2 節で述べたように, 近似アルゴリズムで求められた関数 $g_{i,j}(T)$ は $i<n$ のときに条件

$$\tilde{g}_{i,j}(T) \le g_{i,j}(T) \le (1+\delta)\tilde{g}_{i,j}(T) \qquad (0 \le T < +\infty)$$

を満たす. また, その定義により $g_{n,j}(T)=f_{n,j}(T)$ が成り立つ. したがって, 帰納法により

$$f_{i,j}(T) \le g_{i,j}(T) \le (1+\delta)^{n-i} f_{i,j}(T) \qquad (0 \le T < +\infty)$$

が示せる. とくに, アルゴリズムの出力 $\Phi = g_{0,0}(S_0)$ に対して

$$U = f_{0,0}(S_0) \le \Phi \le (1+\delta)^n f_{0,0}(S_0) = (1+\delta)^n U$$

が成り立つ. よって, 与えられた近似誤差パラメータ ε に対して $\delta = \varepsilon/2n$ とおけば

$$U \le \Phi \le (1+\varepsilon)U$$

が導かれる.

4.3.2 時間計算量

区分線形関数 $g_{i,j}(T)$ $(0 \le i \le n,\ 0 \le j \le i)$ の線分の数の最大値を m とする. 各ノード (i,j) での関数 $g_{i,j}(T)$ の計算は $O(m)$ 時間で出来るので, アルゴリズム全体での計算時間は $O(n^2 m)$ 時間となる.

次に, m のオーダーを求めるために, 関数 $g_{i,j}(T)$ の線分の数を解析する. 第 4.2 節の結果より, 線分の数のオーダーは $O(\log(a_{\max}/a_0)/\delta)$ である. ここで, $a_{\max}$ は 関数 $\tilde{g}_{i,j}(T)$ の線分の傾きの最大値, a_0 は関数 $\tilde{g}_{i,j}(T)$ の線分の傾きのうち, 非零のものの中での最小値である.

傾き $a_{\max}$ については, 帰納的により $1/(i+1)$ に等しいことを証明できる. また, 関数 $\tilde{g}_{i,j}(T)$ は $\tilde{g}_{i,j}(T) \ge f_{i,j}(T)$ を満たし, かつ関数 $f_{i,j}(T)$ の線分の傾きのうち非零のものは $\{\min(p,1-p)\}^n/(n+1)$ 以上であることが示せるので, $a_0 \ge \{\min(p,1-p)\}^n/(n+1)$ が成り立つ.

以上の事実より, 関数 $g_{i,j}(T)$ の線分の数のオーダーは, 確率 p を定数と見なすと,

$$O\left(\frac{1}{\delta}\log\frac{a_{\max}}{a_0}\right) = O\left(\frac{1}{\delta}\log\frac{n+1}{(i+1)\{\min(p,1-p)\}^n}\right) = O\left(\frac{n}{\delta}\right)$$

となる.

したがって，第4.3.1節で求めた δ の決め方と合わせると，値 m のオーダーは $O(n^2/\varepsilon)$ となり，次の結果を得る.

定理 4.1. 提案した近似アルゴリズムは $O(n^4/\varepsilon)$ の計算時間で $U \leq \Phi \leq (1+\varepsilon)U$ を満たす U の近似値 Φ を求める.

なお，第 4.2 節で提案した 2 つの区分線形関数の近似手法のどちらを用いても，理論的な結果は同じであるが，計算機実験をしたところ性能に大きな違いが現れた.

どちらの近似手法を用いた場合でも，実際に得られた近似値の精度は理論値 ε よりかなり小さかったが，第 1 の近似方法を適用した場合には極めて良い精度が得られた．例えば，実験で $\varepsilon = 0.1$ と設定したときに，第 2 の方法の誤差は最大で 0.03 であったのに対し，第 1 の方法では 0.005 程度であった．これは，第 1 の近似方法により得られる関数 $g_{i,j}$ が元の関数 $f_{i,j}$ を非常に良く近似していることを示している.

次に計算時間であるが，どちらの方法でもほぼ理論値通りのオーダーで計算時間は増加しているが，第 2 の方法の計算時間は第 1 の方法に比べ 4 分の 1 程度である．これは，第 2 の近似方法により得られる関数 $g_{i,j}$ の線分の数が第 1 の方法に比べ非常に少ないことを示している.

4.4 アルゴリズム 1 の計算時間の改良

前節で提案した近似アルゴリズムの計算時間は $O(n^4/\varepsilon)$ であったが，近似の際に加法的誤差 (additive error) を許すことにより，計算時間を $O(n^3 \log n/\varepsilon)$ に減らすことができる.

前節で提案した近似アルゴリズムで求められた近似関数 $g_{i,j}$ の最小の傾きの値は $\{\min(p, 1-p)\}^n/(n+1)$ まで小さくなる可能性があった．これにより，近似関数の線分の数のオーダーが $O(n^2/\varepsilon)$ と大きくなっていた．これに対し，近似関数の最小の傾きの値が与えられたしきい値より常に大きくなるようにアルゴリズムを改良することで，近似関数の線分の数の増加を防ぐ．この改良により，近似関数の線分の数のオーダーを $O(n \log n/\varepsilon)$ と削減することができる．詳細については省略する.

参考文献

[1] J. C. Cox, S. A. Ross, and M. Rubenstein, Option pricing: a simplified approach, J. Financial Econ. 7 (1979), 229–263.

[2] T.-S. Dai, G.-S. Huang, and Y.-D. Lyuu, Extremely accurate and efficient tree algorithms for Asian options with range bounds, 2002 NTU Int'l Conf. on Finance, National Taiwan University, Taiwan, May 2002.

[3] J. Hull and A. White, Efficient procedures for valuing European and American path-dependent options, J. Derivatives 1 (1993), 21–31.

[4] P. Chalasani, S. Jha, F. Egriboyun, and A. Varikooty, A refined binominal lattice for pricing American Asian options, Review of Derivatives Research 3 (1999), 85–105.

[5] E. H. Neave, A frequency distribution method for valuing average options, ASTIN Bulletin 27 (1997), 173–205.