数理解析研究所講究録
第 1729 巻 2011 年 95-110

ランダム性をもったゲーム木を読み切るコストの期待値

首都大学東京 理工学研究科 数理情報科学専攻
大学院生（M２） 中村亮太 nakamura-ryota@ed.tmu.ac.jp
准教授 鈴木登志雄 toshio-suzuki@tmu.ac.jp

平成 22 年 12 月 13 日

概 要

ランダムな真理値割り当てを与えられた AND-OR 木のゲーム均衡値は固有分布と呼ばれる。Liu-Tanaka(2007) は分布が固有分布であるための必要十分条件を与え、固有分布の一意性を示した。しかし、アルゴリズムの形に少し制限を与えると固有分布の一意性が成り立たないことがわかった。すなわち、問い合わせ履歴の途中経過によらず問い合わせる葉の順番を変えないアルゴリズム（FIX なアルゴリズムと呼ぼう）だけを考えるとき、固有分布一意性は破れる (本号の鈴木の報告参照)。本稿では、葉の置換について閉じたアルゴリズム族に関する固有分布の一意性について考察する。とくに、FIX でないアルゴリズムだけを考えるとき、固有分布一意性が成り立つことを示す。

1 序

2 人ゼロサムゲームで双方が混合戦略を用いる場合、必ず均衡が存在する。これがフォン・ノイマンのミニマックス定理であり、その応用の一つが Yao の principle である。AND-OR 木における Yao の principle は以下の通りである。アルゴリズムがいくつかの葉に真理値を問い合わせて根の真理値を求める状況を考える。どの葉に問い合わせるか、その順番はアルゴリズムが決める。このとき、真理値を問い合わせた回数を計算コストと考える。プレイヤー I はアルゴリズムの混合戦略を選び、プレイヤー II は葉への真理値割り当ての単純戦略を選ぶと min max 値が定まり、これを randomized complexity という。一方、プレイヤー I は真理値割り当ての混合戦略、すなわち真理値割り当ての確率分布を選び、プレイヤー II はアルゴリズムの単純戦略を選ぶと max min 値が定まり、これを distributional complexity という。このとき randomized complexity と distributional complexity が一致するというのが Yao の principle である (詳細は [4, 3] 参照)。この max min = min max という均衡値を与える分布を固有分布と呼ぶ。

具体的にどういう場合に均衡値が実現されるかを解決したのが Liu-Tanaka [2] である。彼らは次を示した。

Assertion2.[2, p76,Theorem9]
任意の木 T_2^k に対し、E^1-*distribution* は、全分布において唯一の固有分布である。

Liu-Tanaka[2] は根を i（ただし $i \in \{0,1\}$）とする真理値割り当てのうち、計算に手間がかかるものを集めて i-set を定義した。i-set 上の確率分布のうち、いかなるアルゴリズムに対しても計算コストが等しくなるものを E^i-distribution というのである。

しかしながら、アルゴリズムの族に制限を加えることによって、一意性が成立しないことが示された [3]。問い合わせ履歴に関わらず次に問い合わせする葉を変えないアルゴリズム全体の族を $\mathcal{A}^k_{fix}$ としよう。ここで、添え字 k は AND-OR の交代が k 回ある木を対象にしたアルゴリズムを考えていることを表す。アルゴリズムを $\mathcal{A}^k_{fix}$ の要素に限るとき、固有分布一意性は成り立たないのである。その証明の本質は、上記のアルゴリズム族に対して E^i-distribution の一意性が成り立たないことにある。

本論文では、葉の置換について閉じたアルゴリズムの族において、固有分布一意性が成り立つための条件（および一意性が破れるための条件）を研究する。この目的のため、アルゴリズム全体の族を、葉の置換について閉じた部分族に分割し、これら部分族がつくる構造を研究する。特に、次が成立することを示す。ただし、$\mathcal{A}^k - \mathcal{A}^k_{fix}$ とは、葉に問い合わせを行う順番が固定されていないアルゴリズム全体の族を表す。

Theorem4.
$i \in \{0,1\}$ とする。d を *i-set* 上の分布で、$\mathcal{A}^k - \mathcal{A}^k_{fix}$ について、E^i-*distribution* であるとする。
このとき、d は *i-set* 上の一様分布である。

本論文の構成は以下の通りである。§2 では Definition、Notation に関する説明を行い、§3 では過去の結果について述べ、最後に §4 で本題を証明する。

2 定義

まず、この論文で扱うゲーム木について定義する。本論文での表記は、基本的に [1] に則して記述している。

Definition1.
図 1 は、1-*round* のときの 2 分岐 AND-OR 木 T^1_2、図 2 は、2-*round* のときの木 T^2_2 を表している。このように *round* がひとつ増えるごとに、葉の部分に木 T^1_2 がぶらさがった形になる。一般に、任意の整数 $k \geq 2$ に対し、*k-round* の木 T^k_2 は木 T^{k-1}_2 の各葉の部分を木 T^1_2 に代えたものである。

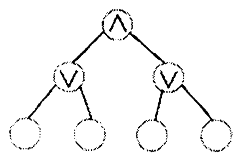

図 1: ゲーム木（k=1 の場合）

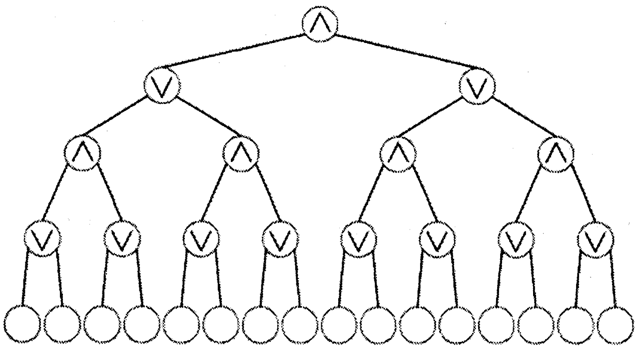

図 2: ゲーム木（k=2 の場合）

次に、アルゴリズムの表記について定義する。前提として、この論文では、Liu-Tanaka 同様、無駄の省略の為、決定性のアルゴリズムとしては *α-β pruning algorithm* のみに制限する（*α-β pruning algorithm* については後で簡単に説明する。正確な式は [2] 参照のこと）。

Definition2.
ゲーム木の葉に対し、左から順に 1、2、3、4、… と番号を付けていく。アルゴリズムは、探索する順に葉の番号を書き連ねることで表すことにする。例えば、木 T_2^1 の葉を左から優先的に探索するアルゴリズムは「1234」と記述する (この論文の中では、左から葉を優先的に探索するアルゴリズムを「基本形」と呼ぶ)。
また、「if(0)1234else1⟨2⟩43」と書けば、始めに決めた探索順序は 1234 であるが、もし最初にめくった(探索した) 葉の値が 1 のときは、それ以降の探索順序を 43 に変更するということを表している (⟨2⟩ と括られているのは、2 と番号付けられた葉の読み飛ばしを表している)。このように、同じ節の子にあたる 2 枚の葉をめくる順序を入れ替えることを「*flip* する」と呼ぶことにする。

次に、コスト計算に関する表記について記述する。
まず、コストの数え方であるが計算の始まりの際、各葉には 0 または 1 が記されているが、どちらの値が書き込まれているかは、分からないように覆われている状態にある。根の値が決まるまでに葉をめくった枚数をコスト (計算量) とする。min や max の判定等、その他の全ての計算はコスト 0 として扱う。

Definition3.[2, p74]
k を正の整数とする。A_D を木 T_2^k の根の値を計算する決定性アルゴリズム、ω を木 T_2^k の葉の割り当てとするとき、$C(A_D, \omega)$ と書くことで、ω が割り当てられた木 T_2^k をアルゴリズム A_D により計算した際にめくられた葉の枚数、つまりコストを表す。
$\mathcal{W}$ を木 T_2^k の葉の割り当ての集合とし、p_ω^d を分布 d での $\mathcal{W}$ の中で ω が割り当てられる確率とする。分布 d の割り当てにおける決定性アルゴリズム A_D の平均計算量 $C(A_D, d)$ は次のように定義される。

$$C(A_D, d) = \sum_{\omega \in \mathcal{W}} p_\omega^d C(A_D, \omega)$$

Definition4.[2, p75]
また、$\mathcal{D}$ を分布の集合、$\mathcal{A}_D(T_2^k)$ を木 T_2^k の根の値を計算する決定性アルゴリズムの集合とする。木 T_2^k を計算する *distributional complexity* $P(T_2^k)$ は次で定義される。

$$P(T_2^k) = \max_{d\in\mathcal{D}} \min_{A_D\in\mathcal{A}_D(T_2^k)} C(A_D, d)$$

集合 $\mathcal{D}$ において、$P(T_2^k)$ が $\min_{A_D\in\mathcal{A}_D(T_2^k)} C(A_D, \delta) = P(T_2^k)$ を成り立たせるような分布 δ を、木 T_2^k に対する割り当ての固有分布と言う。

ここで、*α-β pruning algorithm* について説明をする。
この *α-β pruning algorithm* は、多くのゲーム木アルゴリズムが長年に渡って研究された中でも、非常に巧みに証明され、実用的かつ幅広く使用されているアルゴリズムである。このアルゴリズムの特徴を知るために例を挙げる。

Example1.
決定性アルゴリズムを 1234 とする（つまり左から優先的に探索するアルゴリズム)。葉には図 3 のように値が割り当てられたとする。まず最初は 0 を読み取る。そして、右の葉に移って 0 を読み取る。これにより、上の∨の節には 0 が入ることが分かる。すると、明らかになっていない残る 2 枚の葉をめくらなくとも、根の∧の節には 0 が入ることが分かり、計算は終了する。このような読み飛ばしを *α-cut* と呼ぶ。ちなみに、コストは根の値が決まるまでにめくった葉の枚数が 2 枚であることから 2 となる。

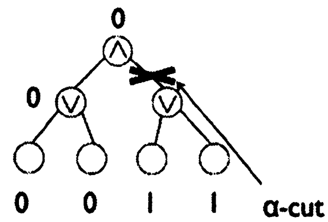

図 3: *α-cut* の例

Example2.
Example1 と同様にして、決定性アルゴリズムを 1234 とする。ただし、葉には図 4 のように先ほどとは違う値が割り当てられたとしよう。まずは 1 を読み取る。すると、2 番目の葉をめくらなくても、上にある∨の節には 1 が入ることが分かるので読み飛ばして 3 番目の葉に移る。このような葉の読み飛ばしを *β-cut* と呼ぶ。ちなみに、コストは 3 となる。

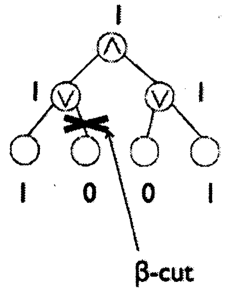

図 4: β-*cut* の例

次に、アルゴリズムの族について見ていく。

Definition5.
k を正の整数とする。木 T_2^k を探索する決定性アルゴリズムを考える。このとき、以下のように $\mathcal{A}_{fix}^k$、$\mathcal{A}^k$、${\mathcal{A}_{fix}^k}^+$、$\mathcal{A}^{k+}$ を定義する。
$\mathcal{A}_{fix}^k$ ={ A : A は良い決定性アルゴリズム（深さ優先アルゴリズム）でかつ、探査順序固定 }
$\mathcal{A}^k$ ={ A : A は良い決定性アルゴリズム（深さ優先アルゴリズム）}
${\mathcal{A}_{fix}^k}^+$ ={ A : A は決定性アルゴリズムでかつ、探査順序固定 }
$\mathcal{A}^{k+}$ ={ A : A は決定性アルゴリズム }

この定義より、$\mathcal{A}_{fix}^k \subset \mathcal{A}^k$、${\mathcal{A}_{fix}^k}^+ \subset \mathcal{A}^{k+}$ であることは明らか。
本論文は、この良い決定性アルゴリズム $\mathcal{A}^k$ に焦点を当て、考察していく。

ここで、この論文の中で、重要な概念である兄弟互換、*closure property* の定義について説明する。

Definition6.
同じ節 (根) の葉と葉同士、節と節同士の交換を施すことを「兄弟互換」と呼ぶ。

Definition7.
$\beta \subseteq \mathcal{A}^k$ に対し、$\{A : \exists B \in \beta\ \exists\sigma$:兄弟互換の有限個の積 $A = \sigma B\}$ を $cl(\beta)$ と書く (β の *closure*(閉包))。
また、β=$cl(\beta)$ が成立している β に対して、β は *closure property* を持つ、もしくは β は兄弟互換について閉じていると呼ぶ。

次に、Liu-Tanaka が提案した *RAT* について説明する。
RAT は簡単に言うと、アルゴリズムの側からして、なるべく手間のかかる葉への割り当て集合を形成する手法である。以下にその Methodology を記述する。

Methodology8. 逆割り当て技法 (Reverse Assigning Technique: *RAT*)[2, p75-76,Methodology6]
逆割り当て技法とは、木 T_2^k の 1-*set*(0-*set*) を形成する技法である。形成ステップは以下の通り。
(1) 木 T_2^k の根に先に 1(0) を割り当てる。
(2) 根からスタートして、葉に行き着くまで、次のように各節の子に、0 か 1 を割り当てていく。
・$\wedge$に 1 とラベルされた節の子には、どちらも 1 を割り当てる。

・$\vee$に 0 とラベルされた節の子には、どちらも 0 を割り当てる。
・$\wedge$に 0 とラベルされた節の子には、ランダムに片方の子に 0 を割り当てて、もう片方の子には 1 を割り当ててやる。
・$\vee$に 1 とラベルされた節の子には、ランダムに片方の子に 1 を割り当てて、もう片方の子には 0 を割り当ててやる。
(3) 葉の部分に形成される割り当ての、起こりうる全てのものを集めたものが 1-*set*(0-*set*) である。
これにより、1-*set* (0-*set*) が生成される。

Definition9.[2, p.76]
すべての決定性のアルゴリズムが同じコスト期待値であるような 1-*set* (0-*set*) 上の分布を E^1-*distribution* (E^0-*distribution*) と呼ぶ。

3 過去の結果 (背景)

Liu-Tanaka

Assertion1.[2, p76,Theorem8]
任意の木 T_2^k に対し、E^1-*distribution*(E^0-*distribution*) における 1-*set*(0-*set*) の各割り当ての確率は、$\frac{1}{4^{(4^k-1)/3}}$ である。

Assertion2.[2, p76,Theorem9]
任意の木 T_2^k に対し、E^1-*distribution* は、全分布において唯一の固有分布である。

Suzuki

Assertion3.[3, Suzuki,Theorem1]
いま、アルゴリズムを $\mathcal{A}_{fix}^k$ のみに限定して考える (つまり葉をめくる優先順位を探索 (計算) の途中で変えないもののみを考える)。
このとき k=1 に対して、E^1-*distribution* は非可算個 (連続 (体) 濃度) 存在する。
よって、この設定下では、*Assertion*1 は成立しない。
(*proof*) [3] 参照。

Assertion4.[3, Suzuki,Corollary1]
同様に、アルゴリズムを $\mathcal{A}_{fix}^k$ のみに限定して考える (つまり葉をめくる優先順位を探索 (計算) の途中で変えないもののみを考える)。
k=1 のとき、固有分布は非可算個 (連続 (体) 濃度) 存在する。
(*proof*) [3] 参照。

Assertion5.[3, Suzuki,Theorem2]
アルゴリズムを $\mathcal{A}_{fix}^k$ のみに限定して考える (つまり葉をめくる優先順位を探索 (計算) の途中で変えないもののみを考える)。

任意の正整数 k に対し、固有分布は非可算個 (連続 (体) 濃度) 存在する。
$(proof)$ [3] 参照。

4 本題

さて、Liu-Tanaka の定理 (2007) の「反例」のように見える結果を示した Suzuki(2009) の理論を一般化させたい。つまり、具体的には、固有分布の一意性が破れるのは、アルゴリズムがどのような性質を持つ場合であるかを知りたい。以下、*closure property* を持つアルゴリズム族について、考察していくことにする。

まず、$\mathcal{A}^k_{fix}$、$\mathcal{A}^k$(k は正の整数) が持つ性質について触れる。

k=1 のとき、

$$
\begin{aligned}
\mathcal{A}^1_{fix} &= \{1234, 4312, 3421, 2143, 3412, 1243, 2134, 4321\} \\
&= cl(\{1234\}) \\
\mathcal{A}^1 &= \{1234, 4312, 3421, 2143, 3412, 1243, 2134, 4321, \\
&\quad if(0)\ 1234\ else1\langle 2\rangle 43, if(0)\ 4312\ else4\langle 3\rangle 21, if(0)\ 3421\ else3\langle 4\rangle 12, if(0)\ 2143\ else2\langle 1\rangle 34, \\
&\quad if(0)\ 3412\ else3\langle 4\rangle 21, if(0)\ 1243\ else1\langle 2\rangle 34, if(0)\ 2134\ else2\langle 1\rangle 43, if(0)\ 4321\ else4\langle 3\rangle 12\} \\
&= cl(\{1234,\ if(0)1234\ else1\langle 2\rangle 43\}) \\
&= cl(\{1234\}) \oplus cl(\{if(0)1234\ else1\langle 2\rangle 43\})
\end{aligned}
$$

上記のように、$\mathcal{A}^1_{fix}$、$\mathcal{A}^1$ は、*closure property* を持つアルゴリズム族の直和によって表せることができる。この結果より、k=1 のとき次のことが成り立つことが分かる。

Theorem1.
β を基本形を含む $\mathcal{A}^1$ の部分集合 (つまり $1234 \in \beta \subseteq \mathcal{A}^1$)、かつ β は兄弟互換について閉じているものとする。
このとき、β=$\mathcal{A}^1_{fix}$ または $\beta = \mathcal{A}^1$ となる。

上記のことが $k \geq 2$ についても成り立つだろうか。つまり、以下のことが成り立つかどうかをこれより調べていく。

Conjecture2.
β を基本形を含む $\mathcal{A}^k$ の部分集合、かつ β は兄弟互換について閉じているものとする。
このとき、β=$\mathcal{A}^k_{fix}$ または $\beta = \mathcal{A}^k$ となる。

いきなり k=2 の場合について考えると、飛躍的に取り扱うアルゴリズムの数が多くなってしまうので、前段階として、2 つの k=1 のゲーム木を $\vee$ の節でつないだゲーム木を考えてみる (図 5 参照)。このゲーム木を k=1.5 と呼ぶことにしよう。

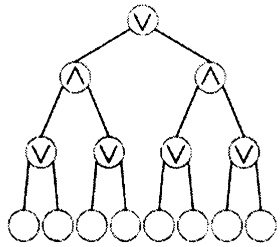

図 5: ゲーム木（k=1.5 の場合）

k=1 の場合は、葉の枚数が 4 枚で、読む順番が途中で変わる機会は β-*cut* の発生時の 1 回のみだったが、k=1.5 の場合になると、葉の枚数は 2 倍の 8 枚、読む順番が途中で変わる機会は β-*cut* だけでなく α-*cut* の発生時にも読む順番を変えることができるので、k=1 の場合と比べて非常に煩雑なアルゴリズムになる。確かに煩雑ではあるが、β-*cut* 発生直後に *flip* するか否か、β-*cut* 発生後は読む順番をどのように変えるか否か、同様にして、α-*cut* が発生後が起きた時はどんな順番で読むかについて注意すれば、$\mathcal{A}^{1.5}_{fix}$、$\mathcal{A}^{1.5}$ を閉包を用いて次のように表すことができる。

$$
\begin{aligned}
\mathcal{A}^{1.5}_{fix} &= cl(A(12345678;5678;-;-;5678))\\
\mathcal{A}^{1.5} &= cl(A(12345678;5678;-;-;5678))\\
&\quad \oplus cl(A(12345678;5678;-;-;if(0)5678\ else5\langle 6\rangle 87))\\
&\quad \oplus cl(A(12345678;if(0)5678\ else5\langle 6\rangle 87;-;-;5678))\\
&\quad \oplus cl(A(12345678;if(0)5678\ else5\langle 6\rangle 87;-;-;if(0)5678\ else5\langle 6\rangle 87))\\
&\quad \vdots\\
&\quad \oplus cl(A(12345678;if(0)8765\ else8\langle 7\rangle 56;-;-;if(0)8765\ else8\langle 7\rangle 56))
\end{aligned}
$$

この A で括られた部分は $A^{1.5}$ の決定性アルゴリズムを表している。この決定性アルゴリズムの読み方ついて説明する。

$A(A_{base};A_\alpha;f_1;f_2;A_\beta)$

$A_{base} \in \mathcal{A}^{1.5}_{fix}$ ··· 最初に決めた木 $T_2^{1.5}$ の葉をめくる順番を表す探査順序固定の決定性アルゴリズム。これを基にして、α-*cut*、β-*cut* が起きたときにどのようにアルゴリズムを変化させるか、あるいは変えないままにするかを場合分けしていく。結局 *closure* をとるので、$\mathcal{A}^{1.5}_{fix}$ の元なら何でも良い。簡単の為、基本形の 12345678 で全て統一している。

$A_\alpha \in \mathcal{A}^1$ ··· α-*cut* が起こった後、隣の木 T_2^1 の葉をめくる順番を表す決定性アルゴリズム。

f_1、$f_2 \in \{+,-\}$ ··· f_1 は 1 枚目で β-*cut* が起こった時、1 枚目の葉のいとこにあたる葉をめくる順番を *flip* させるか否かを決める記号。*flip* させるなら +、*flip* させず、そのままならば − と表す。

f_2 は、葉をめくって出た値が 1 枚目から順に 0、1、0、0 となり (α-*cut* も β-*cut* も起こらず、計算も終了せず)、探索場所が隣の木 T_2^k の葉に移り、5 枚目で β-*cut* が

起こった時、5 枚目の葉のいとこにあたる葉をめくる順番を $flip$ させるか否かを決める記号。f_1 と同様に、$flip$ させるなら $+$、$flip$ させず、そのままならば $-$ と表す。

$A_\beta \in \mathcal{A}^1$ $\cdots$ 1 枚目で β-cut が起こり、2 枚目で 0、3 枚目で 0 となり (計算が終了せず)、隣の木 T_2^1 の葉をめくる順番を表す決定性アルゴリズム。

$\mathcal{A}^{1.5}$ が 2^{10} 個の $closure$ の直和で構成されていることから、$\mathcal{A}^{1.5}_{fix}$ 以外のコンポーネントのどれをとるかを考えることで、次が成り立つことが分かる。

Theorem3.
β を基本形を含む $\mathcal{A}^{1.5}$ の部分集合 (つまり $12345678 \in \beta \subseteq \mathcal{A}^{1.5}$)、かつ β は兄弟互換について閉じているものとする。
このとき、上記を満たす β は $\mathcal{A}^{1.5}_{fix}$、$\mathcal{A}^{1.5}$ 含め $2^{2^{10}-1}$ 個存在する。
よって、k=1.5 のとき、Conjecture2 は成立しない。

また、$k \geq 2$ について、$\mathcal{A}^k$ は、$\mathcal{A}^{1.5}$ を含むので、k=1.5 よりも更に多くの $closure$ の直和の形で構成されていることが分かる。それら $closure$ すべての直和が $\mathcal{A}^k$ であり、α-cut、β-cut が発生しても葉のめくり方を変化させないアルゴリズムの $closure$ が $\mathcal{A}^k_{fix}$ である。しかし、他にも α-cut もしくは β-cut 発生時に葉のめくり方に変化を許した $closure$ が数多く存在する。
したがって、任意の整数 $k \geq 2$ について、Conjecture2 は成立しないことが示された。

E^0-*distribution*、E^1-*distribution* についての条件を課すことで場合の数を絞れるだろうか。

これより β-cut が起きたときに必ず $flip$ するようなアルゴリズムに焦点を置いて考察する。

k=1 のとき、β-cut が起きたときに必ず $flip$ するようなアルゴリズム全体のクラス A^1_{flip} について考える。
A_9、A_{10}、$\cdots$、A_{16} を以下のように定義する。

A_9 : $if(0)$ 1234 $else1\langle 2\rangle 43$
A_{10} : $if(0)$ 4312 $else4\langle 3\rangle 21$
A_{11} : $if(0)$ 3421 $else3\langle 4\rangle 12$
A_{12} : $if(0)$ 2143 $else2\langle 1\rangle 34$
A_{13} : $if(0)$ 3412 $else3\langle 4\rangle 21$
A_{14} : $if(0)$ 1243 $else1\langle 2\rangle 34$
A_{15} : $if(0)$ 2134 $else2\langle 1\rangle 43$
A_{16} : $if(0)$ 4321 $else4\langle 3\rangle 12$

それでは、実際に E^1-*distribution* がどれほど存在するのか確かめてみる。
k=1 のときの 1-*set* は $\{1010, 1001, 0110, 0101\}$ である。この 1-*set* の元に ω_1=1010、ω_2=1001、ω_3=0110、

ω_4=0101 と記号を付ける。また、$\omega_i (i \in \{1,2,3,4\})$ の割り当て確率を p_i とし、$\sum_{i=1}^{4} p_i = 1$ を満たすとする。ここで、各割り当てをアルゴリズム A_9、$\cdots$、A_{16} により計算してみると、次の表のようになる。

		A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}
ω_1	1010	3	3	2	4	3	2	3	4
ω_2	1001	2	3	4	3	3	3	4	2
ω_3	0110	3	4	3	2	2	4	3	3
ω_4	0101	4	2	3	3	4	3	2	3

各アルゴリズムのコスト期待値を求め、E^1-*distribution* という条件を課すと、

$3p_1 + 2p_2 + 3p_3 + 4p_4$
$=3p_1 + 3p_2 + 4p_3 + 2p_4$
$=2p_1 + 4p_2 + 3p_3 + 3p_4$
$=4p_1 + 3p_2 + 2p_3 + 3p_4$
$=3p_1 + 3p_2 + 2p_3 + 4p_4$
$=2p_1 + 3p_2 + 4p_3 + 3p_4$
$=3p_1 + 4p_2 + 3p_3 + 2p_4$
$=4p_1 + 2p_2 + 3p_3 + 3p_4$

$\therefore p_1 = p_2 = p_3 = p_4$

つまり、E^1-*distribution* は一意的である。

コスト期待値３

次に E^0-*distribution* がどれほど存在するのか確かめてみる。

k=1 のときの 0-*set* は $\{1000, 0100, 0010, 0001\}$ である。この 0-*set* の元に ω_5=1000、ω_6=0100、ω_7=0010、ω_8=0001 と記号を付ける。また、$\omega_j (i \in \{5,6,7,8\})$ の割り当て確率を p_j とし、$\sum_{j=5}^{8} p_j = 1$ を満たすとする。ここで、各割り当てをアルゴリズム A_9、$\cdots$、A_{16} により計算してみると、次の表のようになる。

		A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}
ω_5	1000	3	2	2	4	2	3	4	2
ω_6	0100	4	2	2	3	2	4	3	2
ω_7	0010	2	4	3	2	3	2	2	4
ω_8	0001	2	3	4	2	4	2	2	3

各アルゴリズムのコスト期待値を求め、E^0-*distribution* という条件を課すと、

$3p_5 + 4p_6 + 2p_7 + 2p_8$

$=2p_5+2p_6+4p_7+3p_8$
$=2p_5+2p_6+3p_7+4p_8$
$=4p_5+3p_6+2p_7+2p_8$
$=2p_5+2p_6+3p_7+4p_8$
$=3p_5+4p_6+2p_7+2p_8$
$=4p_5+3p_6+2p_7+2p_8$
$=2p_5+2p_6+4p_7+3p_8$

$\therefore p_5=p_6=p_7=p_8$

つまり、E^0-*distribution* は一意的である。

コスト期待値 $\dfrac{11}{4}$

したがって、k=1 の場合は、E^0-*distribution*、E^1-*distribution* ともに一意的であることが示された。
また、この一意に定まった E^1-*distribution* が固有分布でもあることも示された。
さらに、この結果を用いて、次が成り立つことを示すことができる。

Theorem4.
$i\in\{0,1\}$ とする。d を *i-set* 上の分布で、$\mathcal{A}^k-\mathcal{A}^k_{fix}$ について、E^i-*distribution* であるとする。
このとき、d は *i-set* 上の一様分布である。
($proof$)
帰納法を用いて証明していく。
i)k=1 のとき、$\mathcal{A}^1-\mathcal{A}^1_{fix}=\mathcal{A}^1_{flip}$ より、すでに d が *i-set* 上の一様分布であることは示している。
よって、k=1 のとき成立。

ii)$k=n(n\geq1$、n は整数) まで、d は *i-set* 上の一様分布であることが成立すると仮定する。
この仮定の下、k=n+1 のときを考える。
準備として、木 T^n_2 から生成される *i-set*($i\in\{0,1\}$) を (i-set)^n と表すと定義する。また、木 T^{n+1}_2 の根から 1 つ下の *round* にあたる 4 つの節∧に左から I、II 、III 、IV と記号を付ける (図 6 参照)。

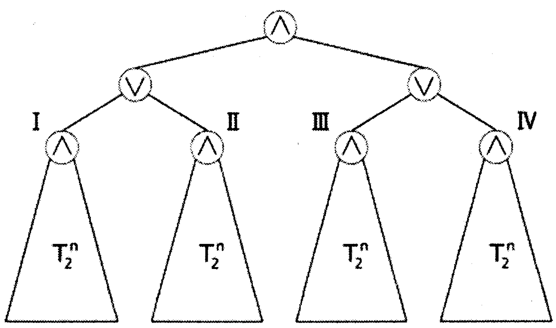

図 6: ゲーム木 (k=n+1 の場合)

さて、$(i\text{-}set)^{n+1}$ 上の分布 d について考えてみる。
$j \in \{0,1\}$ に対し、分布 d において、I$=j$ となる確率を q_j とする。また、$m \in \{$I,II ,III ,IV$\}$ に対し、分布 d の中でも m を根とする木 T_2^n の葉の箇所に表れる分布を $d|m$ と表し、更に m=j という条件下での分布 $d|m$ を $(d|m)_{m=j}$ と表すことにする。

次の 3 つの Claim を証明することで、Theorem4 は示される。
Claim1.
$i, j \in \{0,1\}$ とし、d は $(i\text{-}set)^{n+1}$ 上の分布とする。このとき、I を根とする木 T_2^k を計算する任意の決定性アルゴリズム X について $C(X,(d|\mathrm{I})_{I=j})$ の値はすべて等しい。
また、II、III、IV についても同様のことが言える。

Claim2.
各 $i, j \in \{0,1\}$ に対し、$(d|\mathrm{I})_{I=j}$ は $(i\text{-}set)^n$ 上の一様分布である。
また、II、III、IV についても同様のことが言える。

Claim3.
$i \in \{0,1\}$ とする。各 $\omega \in (i\text{-}set)^1$ に対し、$prob[\omega]$ を、$(i\text{-}set)^{n+1}$ 上の分布 d において、I II III IV $= \omega$ となる確率とする。
このとき、各 $\omega \in (i\text{-}set)^1$ に対し、$prob[\omega]$ はすべて等確率となる。

見ればわかるように、Claim2 と Claim3 だけで Theorem4 は証明できるのだが、Claim1 は Claim2 を導くための重要な主張である。まずこの Claim1 を示す。

Claim1.
$i, j \in \{0,1\}$ とし、d は $(i\text{-}set)^{n+1}$ 上の分布とする。このとき、I を根とする木 T_2^k を計算する任意の決定性アルゴリズム X について $C(X,(d|\mathrm{I})_{I=j})$ の値はすべて等しい。
また、II、III、IV についても同様のことが言える。
(proof of Claim1)
Case1(i=1 のとき)
次のようなアルゴリズムを考える。
A : D で IV、C で III、B で II、X で I の値を求める決定性アルゴリズム
ここで、B、C、D を固定し、X だけ自由に変えられるとする。
このとき、次のことが成り立つ。

$$
\begin{aligned}
C(A,d) &= (\text{定数}) + q_0(B,(d|\mathrm{II})_{II=1}) + q_1\{C(B,(d|\mathrm{II})_{II=0}) + C(X,(d|\mathrm{I})_{I=1})\} \\
&= (\text{定数}) + q_1 C(X,(d|\mathrm{I})_{I=1})
\end{aligned}
$$

また、次のようなアルゴリズムを考える。
A' : X で I、B で II、C で III、D で IV の値を求める決定性アルゴリズム
A と同様にして、B、C、D を固定し、X だけ自由に変えられるとする。
このとき、次が成り立つ。

$$C(A',d) = q_0 C(X,(d|\mathrm{I})_{I=0}) + q_1 C(X,(d|\mathrm{I})_{I=1}) + (\text{定数})$$

Case1.1($q_1 > 0$ のとき)
d は $\mathcal{A}^{n+1} - \mathcal{A}^{n+1}_{fix}$ に関して E^1-*distribution* なので、$\mathcal{A}^{n+1} - \mathcal{A}^{n+1}_{fix}$ のどんな決定性アルゴリズムを割り当ててても $C(A, d)$ は同じ値をとることから、$C(A, d)$ は X によらない値だと分かる。
したがって、$q_1 > 0$ より、$C(X, (d|\mathrm{I})_{I=1})$ も X によらずコスト期待値一定となる。
Case1.1.1($0 < q_1 < 1$ のとき)
$C(X, (d|\mathrm{I})_{I=1})$ が X によらず、かつ $q_0 \neq 0$ であるので、$C(A', d)$ の式より、
$C(X, (d|\mathrm{I})_{I=0})$ は X によらずコスト期待値一定となる。
Case1.1.2($q_1 = 1$ のとき)
$q_0 = 1 - q_1 = 0$ となる。
つまり、$(d|\mathrm{I})_{I=0}$ なる分布は存在せず、$(d|\mathrm{I})_{I=1}$=$d|\mathrm{I}$ であると分かる。
よって、この場合について、Claim1 は示された。

Case1.2($q_1 = 0$ のとき)
$q_1 = 0$ より、$q_0 = 1 - q_1 = 1$ となる。
つまり、$(d|\mathrm{I})_{I=1}$ なる分布は存在せず、$(d|\mathrm{I})_{I=0}$=$d|\mathrm{I}$ であると分かる。
このとき、次が成り立つ。
$C(A', d) = C(X, d|\mathrm{I}) + (\text{定数})$
したがって、$C(X, (d|\mathrm{I})_{I=0})(=C(X, d|\mathrm{I}))$ は X によらずコスト期待値一定となる。

Case2(i=0 のとき)
分布 d において、Ⅲ=Ⅳ=0 となる確率を s とする (定義より $q_1 \leq s$ が成り立つことは明らか)。
このとき、次が成り立つ。
$$\begin{aligned} C(A, d) &= s(\text{定数}) + (1 - s)\{(\text{定数}) + C(X, (d|\mathrm{I})_{I=0})\} \\ &= (\text{定数}) + (1 - s)C(X, (d|\mathrm{I})_{I=0}) \end{aligned}$$

また、i=0 の場合についても、次が成り立つ。
$C(A', d) = q_0 C(X, (d|\mathrm{I})_{I=0}) + q_1 C(X, (d|\mathrm{I})_{I=1}) + (\text{定数})$

さらに、A'' を次のように定義する。
A'' : B でⅡ、X でⅠ、C でⅢ、D でⅣの値を求める決定性アルゴリズム
同様にして、B、C、D を固定し、X だけ自由に変えられるとする。
このとき、次が成り立つ。
$$\begin{aligned} C(A'', d) = &\ prob[\mathrm{I}{=}0, \mathrm{II}{=}0]\{C(X, (d|\mathrm{I})_{I=0}) + (\text{定数})\} \\ &+ prob[\mathrm{I}{=}0, \mathrm{II}{=}1](\text{定数}) \\ &+ prob[\mathrm{I}{=}1, \mathrm{II}{=}0]\{C(X, (d|\mathrm{I})_{I=1}) + (\text{定数})\} \\ = &\ (1 - s)C(X, (d|\mathrm{I})_{I=0}) + (\text{定数}) + q_1 C(X, (d|\mathrm{I})_{I=1}) \end{aligned}$$

Case2.1($s < 1$ のとき)
$s < 1$ なので、$C(A, d)$ の式より、$C(X, (d|\mathrm{I})_{I=0})$ は X によらずコスト期待値一定となる。
Case2.1.1($q_1 > 0$ のとき)
$C(X, (d|\mathrm{I})_{I=0})$ が X によらないので、$C(A', d)$ の式より、

$C(X,(d|\mathrm{I})_{I=1})$ は X によらずコスト期待値一定となる。
Case2.1.2($q_1 = 0$ のとき)
$q_1 = 0$ より、$q_0 = 1 - q_1 = 1$ となる。
つまり、$(d|\mathrm{I})_{I=1}$ なる分布は存在せず、$(d|\mathrm{I})_{I=0}$=$d|\mathrm{I}$ であると分かる。
よって、$C(X,(d|\mathrm{I})_{I=0})$=$C(X,d|\mathrm{I})$ より、この場合について、Claim1 は示された。

Case2.2($s = 1$ のとき)
Case2.2.1($q_1 > 0$ のとき)
$C(A'',d)$ の式より、$C(X,(d|\mathrm{I})_{I=1})$ は X によらずコスト期待値一定となる。
Case2.2.1.1($0 < q_1 < 1$ のとき)
$C(X,(d|\mathrm{I})_{I=1})$ が X によらず、かつ $q_0 \neq 0$ であるので、$C(A',d)$ の式より、
$C(X,(d|\mathrm{I})_{I=0})$ は X によらずコスト期待値一定となる。
Case2.2.1.2($q_1 = 1$ のとき)
$q_0 = 1 - q_1 = 0$ となる。
つまり、$(d|\mathrm{I})_{I=0}$ なる分布は存在せず、$(d|\mathrm{I})_{I=1}$=$d|I$ であると分かる。
よって、この場合について、Claim1 は示された。
Case2.2.2($q_1 = 0$ のとき)
$q_1 = 0$ より、$q_0 = 1 - q_1 = 1$ となる。
つまり、$(d|\mathrm{I})_{I=1}$ なる分布は存在せず、$(d|\mathrm{I})_{I=0}$=$d|\mathrm{I}$ であると分かる。
このとき、次が成り立つ。
$C(A',d) = C(X,d|\mathrm{I})$ + (定数)
したがって、$C(X,(d|\mathrm{I})_{I=0})$(=$C(X,d|\mathrm{I})$) は X によらずコスト期待値一定となる。

上記の結果より、I を根とする木 T_2^k を計算する任意の決定性アルゴリズム X について、
$C(X,(d|\mathrm{I})_{I=j})(j \in \{0,1\})$ の値はすべて等しい。II, III, IV についても同様の議論で示すことができる。
よって、Claim1 は示された。

Claim2.
各 $i,j \in \{0,1\}$ に対し、$(d|\mathrm{I})_{I=j}$ は $(i\text{-}set)^n$ 上の一様分布である。
また、II、III、IV についても同様のことが言える。
(proof of Claim2)
$m \in \{\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{IV}\}$ とする。Claim1 より、各 $(d|m)_{I=j}$ のコスト期待値は一定ということが分かったので、帰納法の仮定により、$(d|m)_{I=j}$ は $i\text{-}set$ 上の一様分布となる。
これについても、 II, III, IV も同様の議論で示すことができる。
よって、Claim2 は示された。

Claim3.
$i \in \{0,1\}$ とする。各 $\omega \in (i\text{-}set)^1$ に対し、$prob[\omega]$ を、$(i\text{-}set)^{n+1}$ 上の分布 d において、I II III IV $= \omega$ となる確率とする。
このとき、各 $\omega \in (i\text{-}set)^1$ に対し、$prob[\omega]$ はすべて等確率となる。
(proof of Claim3)
Case1(i=1 のとき)

各 $j \in \{0,1\}$ に対し、δ_j を $(j\text{-}set)^n$ 上の一様分布を表すとしよう。いま、アルゴリズム B を 1 つ固定する。ただし、B は $\mathcal{A}^n - \mathcal{A}^n_{fix}$ 上の決定性アルゴリズムとする。また、$C(B,\delta_j)$ を C_j と表すことにする。

A_1、A_1'、A_1'' を次のように定義する。
A_1 : B でⅠ、B でⅡ、B でⅢ、B でⅣの値を求める決定性アルゴリズム。
A_1' : B でⅡ、B でⅠ、B でⅢ、B でⅣの値を求める決定性アルゴリズム。
A_1'' : 基本 A_1 と同じように挙動するが、I=1 となり、β-cut が発生した時は、その後の値を求める順番を B でⅣ、B でⅢと変更を施す決定性アルゴリズム。

ここで、Claim2 より、次が成り立つ。
$C(A_1,d) = prob[1010] \times 2C_1 + prob[0110](C_0+2C_1) + prob[1001](C_0+2C_1) + prob[0101](2C_0+2C_1)$
$C(A_1',d) = prob[1010](C_0+2C_1) + prob[0110] \times 2C_1 + prob[1001](2C_0+2C_1) + prob[0101](C_0+2C_1)$
$C(A_1'',d) = prob[1010](C_0+2C_1) + prob[0110](C_0+2C_1) + prob[1001] \times 2C_1 + prob[0101](2C_0+2C_1)$

$C(A_1,d) - C(A_1',d) = 0$ であるので、
$-prob[1010] + prob[0110] - prob[1001] + prob[0101] = 0$
12 と 34 の役割交換をすることにより、
$-prob[1010] - prob[0110] + prob[1001] + prob[0101] = 0$
また、$C(A_1,d) - C(A_1'',d) = 0$ より、
$-prob[1010] + prob[1001] = 0$

上記 3 式より、
$prob[1010] = prob[0110] = prob[1001] = prob[0101] = \frac{1}{4}$

よって、Case1 は示された。

Case2(i=0 のとき)
Case1 と同様にして、アルゴリズム B を固定する。ただし、B は $\mathcal{A}^n - \mathcal{A}^n_{fix}$ 上の決定性アルゴリズムとし、$C(B,\delta_j)$ を C_j と表すことにする。

A_2、A_2'、A_2''、A_2''' を次のように定義する。
A_2 : B でⅠ、B でⅡ、B でⅢ、B でⅣの値を求める決定性アルゴリズム。
A_2' : B でⅡ、B でⅠ、B でⅢ、B でⅣの値を求める決定性アルゴリズム。
A_2'' : B でⅢ、B でⅣ、B でⅠ、B でⅡの値を求める決定性アルゴリズム。
A_2''' : B でⅣ、B でⅢ、B でⅡ、B でⅠの値を求める決定性アルゴリズム。

Claim2 より、次が成り立つ。
$C(A_2,d) = prob[1000](2C_0+C_1) + prob[0100](3C_0+C_1) + prob[0010] \times 2C_0 + prob[0001] \times 2C_0$
$C(A_2',d) = prob[1000](3C_0+C_1) + prob[0100](2C_0+C_1) + prob[0010] \times 2C_0 + prob[0001] \times 2C_0$
$C(A_2'',d) = prob[1000] \times 2C_0 + prob[0100] \times 2C_0 + prob[0010](2C_0+C_1) + prob[0001](3C_0+C_1)$
$C(A_2''',d) = prob[1000] \times 2C_0 + prob[0100] \times 2C_0 + prob[0010](3C_0+C_1) + prob[0001](2C_0+C_1)$

$C(A_2, d) - C(A_2', d) = 0$ であるので、
$-prob[1000] + prob[0100] = 0$
$C(A_2'', d) - C(A_2''', d) = 0$ であるので、
$-prob[0010] + prob[0001] = 0$
また、$C(A_2, d) - C(A_2''', d) = 0$ より、
$prob[1000]C_1 + prob[0100](C_0 + C_1) - prob[0010](C_0 + C_1) - prob[0001]C_1 = 0$

上記の 3 式より、
$prob[1000] = prob[0100] = prob[0010] = prob[0001] = \frac{1}{4}$

よって、Case2 も示された。
したがって、Case1、Case2 より、Claim3 が示された。

また、Claim2、Claim3 より、$\mathcal{A}^{n+1} - \mathcal{A}^{n+1}_{fix}$ について、E^1-*distribution* となっている分布 d でも *i-set*$(i \in \{0,1\})$ 上の一様分布である。よって、k=n+1 のときも成立。

i)、ii) より、任意の正の整数 k に対し、d を *i-set*$(i \in \{0,1\})$ 上の分布で、$\mathcal{A}^k - \mathcal{A}^k_{fix}$ について、E^1-*distribution* であるとしたとき、d は *i-set* 上の一様分布であることが示された。

参考文献

[1] D.E.Knuth and R.W.Moore: An analysis of alpha-beta pruning. *Artifical Intelligence* 6(4) (1975) 293-326.

[2] C.G.Liu and K.Tanaka: Eigen-distribution on random assignments for game trees. *Information Processing Letters* 104(2007)73-77.

[3] T. Suzuki: Failure of the uniqueness of eigen-distribution on random assignments for game trees. 数理解析研究所講究録（本号）.

[4] A.C.-C. Yao: Probabilistic computations:towards a unified measure of complexity. In: Proc. 18th Annual IEEE Symposium on Foundations of Computer Science(FOCS),1977,pp.222-227.