Proceedings of the International MultiConference of Engineers and Computer Scientists 2012 Vol I,
IMECS 2012, March 14 - 16, 2012, Hong Kong

# Probability Distributions Achieving the Equilibrium of an AND-OR Tree under Directional Algorithms

Toshio Suzuki, *Member, IAENG,* and Ryota Nakamura

***Abstract*—Consider a probability distribution $d$ on the truth assignments to a perfect binary AND-OR tree. Liu and Tanaka (2007) extends the work of Saks and Wigderson (1986), and they characterize the eigen-distribution, the distribution achieving the equilibrium, as the uniform distribution on the 1-set (the set of all reluctant assignments for which the root has the value 1). We show that the uniqueness of the eigen-distribution fails provided that we restrict ourselves to directional algorithms. An alpha-beta pruning algorithm is said to be directional (Pearl, 1980) if for some linear ordering of the leaves (Boolean variables) it never selects for examination a leaf situated to the left of a previously examined leaf. We also show that the following weak version of the Liu-Tanaka result holds for the situation where only directional algorithms are considered; a distribution is eigen if and only if it is a distribution on the 1-set such that the cost does not depend on an associated deterministic algorithm.**

***Index Terms*—AND-OR tree; directional algorithm; computational complexity; randomized algorithms.**

## I. Introduction

THE concept of an AND-OR tree is interesting because of its two aspects, a Boolean function and a game tree. It is a tree whose internal nodes are labeled either AND ($\wedge$) or OR ($\vee$). In this paper, we use the terminology in more restricted sense. An *AND-OR tree* (an *OR-AND tree*, respectively) denotes a tree such that its root is an AND-gate (an OR-gate), layers of AND-gates and those of OR-gates alternate and each leaf is assigned Boolean value. 1 denotes true and 0 denotes false.

A perfect binary tree of this type with height $2k$ is denoted by $T_2^k$. Given a such tree, an algorithm answers the Boolean value of the root by probing Boolean values of some leaves. The cost of the computation is measured by the number of leaves probed. There are classical results showing that we may restrict ourselves to algorithms of a particular type. For example, see [9]. We consider alpha-beta pruning algorithms only. An alpha-beta pruning algorithm is characterized by the following two properties; whenever the algorithm knows a child node of an OR-gate has the value 1, the algorithm recognize that the OR-gate has the value 1 without probing the other child node (such a saving of cost is said to be a beta-cut), and whenever the algorithm knows a child node of an AND-gate has the value 0, the algorithm recognize that the AND-gate has the value 0 without probing the other child node (an alpha-cut). See [3] for more on alpha-beta pruning algorithms.

In this context, a randomized algorithm denotes a probability distribution on a set of deterministic algorithms. This is a kind of Las-Vegas algorithm. For a randomized algorithm, the cost is defined as to be the expected value of the cost. Then, the value

$$\min_{A_R} \max_{\omega} \mathrm{cost}(A_R, \omega),$$

where $A_R$ runs over randomized algorithms and $\omega$ runs over truth assignments, is the *randomized complexity*. A probability distribution on the truth assignments is easier to handle than a probability distribution on the algorithms. Yao's principle [11], [1] is a variant of Von-Neumann's Min-Max theorem. It says that the randomized complexity equals to the *distributional complexity*, that is,

$$\max_{d} \min_{A_D} \mathrm{cost}(A_D, d),$$

where $A_D$ runs over deterministic algorithms and $d$ runs over probability distributions on the truth assignments.

Saks and Wigderson [7] establish basic results on the randomized complexity. In particular, they show that the randomized complexity is $\Theta(\ ((1+\sqrt{33})/4)^h)$, where $h$ denotes the height of the tree, and they conjecture that a similar estimation holds for any Boolean function.

Liu and Tanaka [4] extends the work of Saks and Wigderson, and characterize a probability distribution achieving the equilibrium of $T_2^k$. In particular, they show that such a distribution is unique, and call it the eigen-distribution. A distribution $d_0$ on the truth assignments is the *eigen-distribution* if it achieves the distributional complexity, that is:

$$\min_{A_D} \mathrm{cost}(A_D, d_0) = \max_{d} \min_{A_D} \mathrm{cost}(A_D, d)$$

To be more precise, by extending the concept of a reluctant input in the paper of Saks and Wigderson, Liu and Tanaka define the concept of *i-set* (for $i \in \{0,1\}$) as the set of all assignments such that the root has the value $i$ and whenever an AND-node has the value 0 (and, whenever an OR node has the value 1), its one child node has the value 1 and the other child node has the value 0. They define an *$E^i$-distribution* as to be a distribution on the $i$-set such that all the deterministic algorithm has the same cost. They prove that, for a probability distribution $d$ on the truth assignments to the leaves of $T_2^k$, the followings (LT1)–(LT3) are equivalent.

(LT1) $d$ is the eigen-distribution;

Manuscript received November 7, 2011; revised November 24, 2011. This work was supported in part by Japan Society for the Promotion of Science (JSPS) KAKENHI (C) 22540146, (B) 19340019, and (B) 23340020.

T. Suzuki and R. Nakamura are with Department of Mathematics and Information Sciences, Tokyo Metropolitan University, Minami-Ohsawa, Hachioji, Tokyo 192-0397, Japan. e-mail: toshio-suzuki@tmu.ac.jp (see http://www.ac.auone-net.jp/∼bellp/index.html).

Current affiliation of R. Nakamura is Shimizu Corporation.
e-mail: r-nakamura@shimz.co.jp

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

(LT2) $d$ is an $E^1$-distribution;
(LT3) $d$ is the uniform distribution on the 1-set.

The current paper is motivated by an example that "contradicts to" the uniqueness of the eigen-distribution.

In general, an alpha-beta pruning algorithm can change its priority of scanning leaves throughout a computation. For example, an algorithm can move in such a way that, if a leaf $x$ is skipped due to a beta-cut then scan $y$ and then $z$, otherwise scan $x$, next $z$ and then $y$. An alpha-beta pruning algorithm is said to be *directional* (Pearl, [6]) if for some linear ordering of the leaves it never selects for examination a leaf situated to the left of a previously examined leaf.

We show that the uniqueness of the eigen-distribution fails when we restrict ourselves to directional algorithms (section IV). In particular, there are uncountably many eigen-distributions in the setting.

We also show that the following weak version of the result of Liu and Tanaka holds in the above setting; a distribution is eigen if and only if $E^1$ (section III). Our main method for showing this equivalence is the no-free-lunch theorem (NFLT, for short). Roughly speaking, NFLT says that, averaged over all cost functions, all search algorithms give the same performance. For an introduction to NFLT, see [10], [2]. For a family of algorithms closed under transposition of sub-trees, a variant of NFLT holds, and NFLT implies the equivalence. The current paper is an extension of the research report [8].

## II. Notation

We denote the empty string by $\lambda$. By $\{0,1\}^n$, we denote the set of all strings of length $n$. The cardinality of a set $X$ is denoted by $|X|$. For sets $X$ and $Y$, $X - Y$ denotes $\{x \in X : x \notin Y\}$. We let $\mathrm{prob}[E]$ denote the probability of an event $E$. *A $k$-round AND-OR tree* denotes a perfect binary AND-OR tree of height $2k$ ($k \geq 1$), and is denoted by $T_2^k$.

**Convention** Throughout the paper, unless specified, $h$ denotes a positive integer and $T$ denotes a perfect binary tree of height $h$ such that $T$ is either an AND-OR tree or an OR-AND tree. $\mathcal{A}_D$ denotes the family of all alpha-beta pruning algorithms calculating the root-value of $T$. For the definition of *an alpha-beta pruning algorithm*, see Introduction. $\mathcal{A}$ denotes a non-empty subset of $\mathcal{A}_D$. The height $h$ is fixed in the definition of $\mathcal{A}_D$. Thus, we should write, for example, $\mathcal{A}_D(h)$ in the precise manner, but we omit $h$. The same remark will apply to $\mathcal{A}_{\mathrm{dir}}$ in Definition 2. $\Omega$ is a non-empty family of assignment-codes, where we define assignment-codes in the following. We label each node of $T$ by a string as follows.

**Definition 1.**
1) A *node-code* is a binary string of length at most $h$. Each node of $T$ is assigned a node-code in such a way that the code of the root is the empty string, and nodes with codes of the form $u0, u1$ are child nodes of the node with code $u$.
2) A node-code is a *leaf-code* if its length is $h$. Otherwise, it is an *internal node-code*.
3) An *assignment-code* is a function of the leaf-codes to $\{0,1\}$.

By $\mathcal{A}_{\mathrm{dir}}$, we denote the family of all directional algorithms in $\mathcal{A}_D$. Formal definitions are as follows.

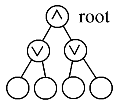


Fig. 1. $T_2^1$

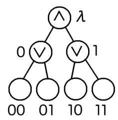


Fig. 2. Node-codes

**Definition 2.** Suppose that $A_D$ is a member of the $\mathcal{A}_D$. Let $\ell = 2^h$.

1) Suppose that $\langle u^{(1)}, u^{(2)}, \cdots, u^{(m)} \rangle$ is a sequence of strings, $\omega$ is an assignment-code and that the followings hold.
   a) $m (\leq \ell)$ is the total number of leaves queried during the computation of $A_D$ under the assignment $\omega$.
   b) For each $j (1 \leq j \leq m)$, the $j$-th query in the computation of $A_D$ under the assignment $\omega$ is the leaf of code $u^{(j)}$.

   By "*query-history of* $\langle A_D, \omega \rangle$" and "*answer-history of* $\langle A_D, \omega \rangle$", we denote $\langle u^{(1)}, u^{(2)}, \cdots, u^{(m)} \rangle$ and $\langle \omega(u^{(1)}), \cdots, \omega(u^{(m)}) \rangle$, respectively.
2) $\mathcal{A}_{\mathrm{dir}}$ is a family of algorithms defined as follows. A deterministic algorithm $A_D \in \mathcal{A}_D$ belongs to $\mathcal{A}_{\mathrm{dir}}$ if there exists a permutation $\langle v^{(1)}, v^{(2)}, \cdots, v^{(\ell)} \rangle$ of the leaves such that for every assignment code $\omega$, the query-history of $\langle A_D, \omega \rangle$ is consistent with the permutation. More precisely, the query history is either equal to the permutation, or a sequence (say, $\langle v^{(1)}, v^{(3)}, v^{(4)} \rangle$) given by omitting some leaf-codes from the permutation.

For $A_D \in \mathcal{A}_D$ and an assignment-code $\omega$, we let $C(A_D, \omega)$ denote the number of leaves scanned in the computation of $A_D$ under $\omega$. By the phrase "the cost of $A_D$ with respect to $\omega$", we denote (not time-complexity but) $C(A_D, \omega)$ . If $d$ is a probability distribution on the assignments then $C(A_D, d)$ denotes the expected value of the cost with respect to $d$. For the definitions of the 0-*set* and the 1-*set*, see Introduction.

**Definition 3.** Suppose that $\mathcal{A}$ is a non-empty subset of $\mathcal{A}_D$, and that $\Omega$ is a non-empty set of assignment-codes.

1) A distribution $d$ on $\Omega$ is called an *eigen-distribution with respect to* $\langle \mathcal{A}, \Omega \rangle$ if the following holds.
$$\min_{A_D \in \mathcal{A}} C(A_D, d) = \max_{d'} \min_{A_D \in \mathcal{A}} C(A_D, d'),$$
   where $d'$ runs over all probability distributions on $\Omega$. In the case where $\Omega$ is the set of all assignment-codes, we say "$d$ is an eigen-distribution with respect to $\mathcal{A}$".
2) Let $i \in \{0,1\}$. A distribution $d$ on the $i$-set is called an *$E^i$-distribution with respect to* $\mathcal{A}$ if there exists a real number $c$ such that for every $A_D \in \mathcal{A}$, it holds that $C(A_D, d) = c$.
3) [4] A distribution $d$ on the truth assignments is eigen (respectively, $E^0$, $E^1$) if it is so with respect to $\mathcal{A}_D$.

## III. The equivalence of eigen and $E^1$

We show that the equivalence of an eigen-distribution and an $E^1$-distribution holds even when we restrict ourselves to

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

directional algorithms. We develop a framework including both the directional case and the usual case.

**Definition 4.** Suppose that $u$ is an internal node-code.

1) Suppose that $v$ and $v'$ are node-codes of the same length. We say "*$v'$ is the $u$-transposition of $v$*" (in symbol, $v' = \mathrm{tp}_u(v)$) if one of the followings holds.
   a) There exist $i \in \{0,1\}$ and a string $w$ such that $v = uiw$ (concatenation) and $v' = u(1-i)w$.
   b) $u$ is not a prefix of $v$ and it holds that $v = v'$.
2) Suppose that $\omega$, $\omega'$ are assignment-codes. We say "*$\omega'$ is the $u$-transposition of $\omega$*" (in symbol, $\omega' = \mathrm{tp}_u(\omega)$) if the following holds for each leaf-code $v$.
$$\omega'(v) = \omega(\mathrm{tp}_u(v))$$
3) Suppose that $A_D$ and $A'_D$ are deterministic algorithms. We say "*$A'_D$ is the $u$-transposition of $A_D$*" (in symbol, $A'_D = \mathrm{tp}_u(A_D)$) if the following holds: "For each assignment-code $\omega$, the query-history of $\langle A'_D, \omega\rangle$ is given by applying component-wise $\mathrm{tp}_u$ operation to the query-history of $\langle A_D, \mathrm{tp}_u(\omega)\rangle$." To be more precise, denote the query-history of $\langle A_D, \mathrm{tp}_u(\omega)\rangle$ and that of $\langle A'_D, \omega\rangle$ by $\langle x^{(1)}, \cdots, x^{(m)}\rangle$ and $\langle y^{(1)}, \cdots, y^{(m)}\rangle$, respectively. Then, the following holds.
$$\forall j \le m \;\; y^{(j)} = \mathrm{tp}_u(x^{(j)})$$
And, the answer history of $\langle A_D, \mathrm{tp}_u(\omega)\rangle$ is the same as that of $\langle A'_D, \omega\rangle$.

**Example 1.** We consider the case where $h = 2$. Then the followings hold. $\mathrm{tp}_\lambda(abcd) = cdab$, $\mathrm{tp}_0(abcd) = bacd$ and $\mathrm{tp}_1(abcd) = abdc$, where we denote a truth assignment $\omega$ by a string $\omega(00)\omega(01)\omega(10)\omega(11)$.

**Definition 5.**
1) $\mathcal{A}$ is *closed (under transposition)* if for each $A_D \in \mathcal{A}$ and for each internal node-code $u$, we have $\mathrm{tp}_u(A_D) \in \mathcal{A}$.
2) $\Omega$ is *closed (under transposition)* if for each $\omega \in \Omega$ and for each internal node-code $u$, we have $\mathrm{tp}_u(\omega) \in \Omega$.
3) $\Omega$ is *connected (with respect to transposition)* if for every distinct members $\omega, \omega' \in \Omega$, there exists a finite sequence $\langle \omega_i\rangle_{i=1,\cdots,N}$ in $\Omega$ and a finite sequence $\langle u^{(i)}\rangle_{i=1,\cdots,N-1}$ of strings such that $\omega_1 = \omega, \omega_N = \omega'$ and for each $i < N$, $\omega_{i+1}$ is the $u^{(i)}$-transposition of $\omega_i$.

**Convention** Throughout the rest of the section, $\mathcal{A}$ denotes a non-empty closed subset of $\mathcal{A}_D$.

**Definition 6.** Suppose that $p_1, \cdots, p_n$ are non-negative real numbers such that their sum makes 1. And, suppose that $\Omega_1, \cdots, \Omega_n$ are mutually disjoint non-empty families of assignment-codes. In addition, suppose that $d_1, \cdots, d_n$ are distributions such that each $d_j$ is a distribution on $\Omega_j$.

1) $p_1 d_1 + \cdots + p_n d_n$ denotes the distribution $d$ on $\Omega_1 \cup \cdots \cup \Omega_n$defined as follows. For each $j$ ($1 \le j \le n$) and each truth assignment $\omega \in \Omega_j$, we have $\mathrm{prob}[\ d \text{ is } \omega\ ] = p_j \times \mathrm{prob}[\ d_j \text{ is } \omega\ ]$.
2) Given a distribution $d$, we say "$d$ is a distribution on $p_1\Omega_1 + \cdots + p_n\Omega_n$" if there exist distributions $d'_j$ on $\Omega_j$ ($1 \le j \le n$) such that $d = p_1 d'_1 + \cdots + p_n d'_n$.

The following is a variant of the no-free-lunch theorem.

**Lemma 1.** *Suppose that $p_1, \cdots, p_n$ and $\Omega_1, \cdots, \Omega_n$ satisfy the requirements in Definition 6, and that each $\Omega_j$ is connected. Then, there exits a real number $c$ such that for every distribution $d$ on $p_1\Omega_1 + \cdots + p_n\Omega_n$, the following holds.*
$$\sum_{A_D \in \mathcal{A}} C(A_D, d) = c \tag{1}$$

*Proof:* We show the case of $n = 1$. For every assignment-code $\omega$ and for every internal node-code $u$, the mapping of $A_D \in \mathcal{A}$ to $\mathrm{tp}_u(A_D)$ is a permutation of $\mathcal{A}$. And, we have $C(\mathrm{tp}_u(A_D), \omega) = C(A_D, \mathrm{tp}_u(\omega))$. Hence, the sum of $C(A_D, \omega)$ over all $A_D \in \mathcal{A}$ is the following.
$$\sum_{A_D \in \mathcal{A}} C(\mathrm{tp}_u(A_D), \omega) = \sum_{A_D \in \mathcal{A}} C(A_D, \mathrm{tp}_u(\omega)) \tag{2}$$

Therefore, there exists a real number $c$ such that for every $\omega \in \Omega$, $\sum_{A_D \in \mathcal{A}} C(A_D, \omega) = c$. Hence, the left-hand side of (1) is equal to the following.
$$\begin{aligned} &\sum_{A_D \in \mathcal{A}} \sum_{\omega \in \Omega} \mathrm{prob}[d = \omega] C(A_D, \omega) \\ &= \sum_{\omega \in \Omega} \big(\mathrm{prob}[d = \omega] \sum_{A_D \in \mathcal{A}} C(A_D, \omega)\big) = c \end{aligned} \tag{3}$$

Thus, the case of $n = 1$ is proved. The general case is immediately shown by this case. ∎

**Lemma 2.** *Suppose that $p_1, \cdots, p_n$ and $\Omega_1, \cdots, \Omega_n$ satisfy the requirements in Definition 6. And, suppose that each $\Omega_j$ is closed.*

1) *Let $d_{\mathrm{unif.}}(p_1\Omega_1 + \cdots + p_n\Omega_n)$ denote the distribution $p_1 d_1 + \cdots + p_n d_n$, where each $d_j$ is the uniform distribution on $\Omega_j$. Then, there exits a real number $c$ such that for every deterministic algorithm $A_D \in \mathcal{A}_D$, it holds that $C(A_D, d_{\mathrm{unif.}}(p_1\Omega_1 + \cdots + p_n\Omega_n)) = c$.*
2) *Suppose that each $\Omega_j$ is not only closed but also connected and that $d$ is a distribution on $p_1\Omega_1 + \cdots + p_n\Omega_n$. Then, the following (a), (b) and (c) are equivalent, where $B_j$ are any deterministic algorithms (not necessarily a member of $\mathcal{A}$), and $d_{\mathrm{unif.}}(\Omega_j)$ is the uniform distribution on $\Omega_j$.*
   a) *The following holds, where $d'$ runs over distributions on $p_1\Omega_1 + \cdots + p_n\Omega_n$.*
   $$\min_{A_D \in \mathcal{A}} C(A_D, d) = \max_{d'} \min_{A_D \in \mathcal{A}} C(A_D, d') \tag{4}$$
   b) *There exits a real number $c$ such that for every $A_D \in \mathcal{A}$, it holds that $C(A_D, d) = c$.*
   c)
   $$\min_{A_D \in \mathcal{A}} C(A_D, d) = \sum_{j=1}^{n} p_j C(B_j, d_{\mathrm{unif.}}(\Omega_j)) \tag{5}$$

*Proof:* 1. The case of $n \ge 2$ is immediate from the case of $n = 1$. In the following, we prove the case of $n = 1$ by induction on $h$. The case of $h = 1$ is immediate. At the the induction step, let $T_0$ ($T_1$, respectively) be the left (right) sub-tree just under the root. Since $\Omega_1$ is closed, the assertion is equivalent to its weaker form: "The costs are the same for all algorithms which probe $T_0$ before $T_1$." We call such algorithms "*left-first algorithms*" in this proof.

Now, $\Omega_1$ is partitioned into sets such that each component $\Omega'$ is of the following form. There exist strings $\alpha_0$ and $\alpha_1$ (depending on $\Omega'$) such that $\alpha_i$ is an assignment-code for $T_i$ for each $i$, and the component $\Omega'$ is the direct product of the closure of $\alpha_0$ and that of $\alpha_1$. Here, the closure of $\alpha_i$ denotes the following set.

$$\{\mathrm{tp}_u(\alpha_i) : u \text{ is an internal node-code}\} \tag{6}$$

By the induction hypothesis, the cost of a left-first algorithm depends only on $\Omega'$, and does not depend on an algorithm. Hence, the same holds with respect to $\Omega_1$.

2. By the assertion 1 of the current lemma and Lemma 1, each of the assertions (a) and (b) is equivalent to (7).

$$\min_{A_D \in \mathcal{A}} C(A_D, d) = \frac{1}{|\mathcal{A}|} \sum_{A_D \in \mathcal{A}} C(A_D, d) \tag{7}$$

Therefore, by the assertion 1, (a) is equivalent to the following.

$$\min_{A_D \in \mathcal{A}} C(A_D, d) = \min_{A_D \in \mathcal{A}} C(A_D, d_{\mathrm{unif.}}(p_1\Omega_1 + \cdots + p_n\Omega_n)) \tag{8}$$

And, the right-hand side of (8) equals to the right-hand side of (5). Hence, (a) is equivalent to (c). ∎

**Lemma 3.** *Assume that $T$ is an AND-OR tree. Suppose that $d$ is an eigen-distribution with respect to $\mathcal{A}$ (see Definition 3). Then $d$ is a distribution on the 1-set.*

*Proof:* For each positive integer $h$ and each $i \in \{0, 1\}$, we let $c_i^{\wedge,h}$ ($c_i^{\vee,h}$, respectively) denote $C(A_D, d_{\mathrm{unif.}}(i\text{-set}))$ for the perfect binary AND-OR tree (OR-AND tree, respectively) of height $h$.

**Claim 1.** Suppose that $\Omega$ is closed. If a given tree is an AND-OR tree and $\Omega$ is not the 1-set (a given tree is an OR-AND tree and $\Omega$ is not the 0-set, respectively), then for any deterministic algorithm $A_D$, $C(A_D, d_{\mathrm{unif.}}(\Omega))$ is less than $c_1^{\wedge,h}$ ($c_0^{\vee,h}$, respectively).

Proof of Claim 1: Note that the followings hold, where the inequalities are proved by induction on $h$.

$$c_0^{\wedge,h} = c_1^{\vee,h} < c_1^{\wedge,h} = c_0^{\vee,h} \leq \frac{4}{3} c_0^{\wedge,h} \tag{9}$$

By induction on $h$, the following holds. "Suppose that $\Omega'$ is a non-empty family of truth assignments such that $\Omega'$ is closed. Let $i \in \{0, 1\}$ be such that for all elements of $\Omega'$, the root has the value $i$. In addition, suppose that $\Omega'$ is not the $i$-set. If a given tree is an AND-OR tree (an OR-AND tree, respectively), then for any deterministic algorithm $A_D$, $C(A_D, d_{\mathrm{unif.}}(\Omega'))$ is less than $c_i^{\wedge,h}$ ($c_i^{\vee,h}$, respectively)."

By this fact and (9), the claim holds. Q.E.D.(Claim 1)

Now, suppose that $T$ is $T_2^k$ for some positive integer $k$. Suppose that $d$ is an eigen-distribution with respect to $\mathcal{A}$. Let $\langle \Omega_j : j = 1, \cdots, n \rangle$ be a partition of the set of all truth assignments to connected closed sets. Without loss of generality, $\Omega_1$ is the 1-set. For each $j$, let $p_j$ be the probability of $d$ being a member of $\Omega_j$. By Lemma 2, (5) holds. Hence, by Claim 1, there are positive real numbers $c_2, \cdots, c_n$ such that the following holds.

$$\forall j \geq 2 \;\; c_j < c_1^{\wedge,h} \tag{10}$$

$$\min_{A_D \in \mathcal{A}} C(A_D, d) = p_1 c_1^{\wedge,h} + \sum_{j=2}^{n} p_j c_j \tag{11}$$

Since $d$ is eigen with respect to $\mathcal{A}$, $d$ achieves the maximum value of (11). Hence, it holds that $p_1 = 1$ and $p_j = 0$ for all $j \geq 2$. Thus, $d$ is a distribution on the 1-set. ∎

**Theorem 4.** *Assume that a given tree $T$ is $T_2^k$ for some positive integer $k$. Suppose that a family $\mathcal{A}$ of algorithms is closed under transposition and that $d$ is a probability distribution on the assignment-codes. Then, the followings are equivalent (see Definition 3).*

*(LT1$^{\mathcal{A}}$) $d$ is an eigen-distribution with respect to $\mathcal{A}$.*

*(LT2$^{\mathcal{A}}$) $d$ is an $E^1$-distribution with respect to $\mathcal{A}$.*

*Proof:* By Lemma 3, (LT1$^{\mathcal{A}}$) is equivalent to "$d$ is an eigen-distribution with respect to $\langle \mathcal{A}, (1\text{-set}) \rangle$". By Lemma 2, this is equivalent to (LT2$^{\mathcal{A}}$). ∎

## IV. A CASE WHERE THE UNIQUENESS FAILS

A direct corollary to Lemma 2 is the following.

**Corollary 5.** *Assume that a given tree $T$ is $T_2^k$ for some positive integer $k$. Then, (LT3) implies (LT2$^{\mathcal{A}}$):*

*(LT3) $d$ is the uniform distribution on the 1-set.*

*(LT2$^{\mathcal{A}}$) $d$ is an $E^1$-distribution with respect to $\mathcal{A}$.*

We show that the uniqueness of the eigen-distribution fails in the directional case. This is shown by proving that (LT2$^{\mathcal{A}}$) does not imply (LT3) with respect to $\mathcal{A} = \mathcal{A}_{\mathrm{dir}}^k$ (see below).

**Convention** $\mathcal{A}_{\mathrm{dir}}^k$ denotes $\mathcal{A}_{\mathrm{dir}}$ (see Definition 2) in the case where $h = 2k$ and $T = T_2^k$.

TABLE I
$C(A_D, \omega)$ FOR $k = 1$ ($\omega \in$ THE 1-SET).

| | | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|---|---|---|---|---|---|
| | | 1234 | 4312 | 3421 | 2143 |
| $\omega_1$ | 1010 | 2 | 3 | 3 | 4 |
| $\omega_2$ | 1001 | 3 | 2 | 4 | 3 |
| $\omega_3$ | 0110 | 3 | 4 | 2 | 3 |
| $\omega_4$ | 0101 | 4 | 3 | 3 | 2 |
| | | $A_5$ | $A_6$ | $A_7$ | $A_8$ |
| | | 3412 | 1243 | 2134 | 4321 |
| $\omega_1$ | 1010 | 2 | 3 | 3 | 4 |
| $\omega_2$ | 1001 | 3 | 2 | 4 | 3 |
| $\omega_3$ | 0110 | 3 | 4 | 2 | 3 |
| $\omega_4$ | 0101 | 4 | 3 | 3 | 2 |

For the time being, we investigate the case where $k = 1$. Table I shows the values of $C(A_D, \omega)$ in the case where $\omega$ is an element of the 1-set. In the table, each $\omega_i$ is the name of an assignment. We denote an assignment-code $\omega$ by a string $\omega(00)\omega(01)\omega(10)\omega(11)$. And, each $A_j$ is the name of an element of $\mathcal{A}_{\mathrm{dir}}^1$. Recall Definition 2. Each $A_j$ is determined by a permutation $xyzw$ of $\{0, 1\}^2$ that shows priority of scanning leaves. In the table, a string such as 1234 denotes a permutation of the above property, where we denote leaf-codes 00, 01, 10 and 11 by numerals 1, 2, 3 and 4, respectively. Since we consider alpha-beta pruning algorithms, only the eight permutations are considered.

**Theorem 6.** *[8]*

1) *There are uncountably many $E^1$-distributions with respect to $\mathcal{A}_{\mathrm{dir}}^1$. Hence, (LT2$^{\mathcal{A}}$) does not imply (LT3) with respect to $\mathcal{A} = \mathcal{A}_{\mathrm{dir}}^1$.*
2) *There are uncountably many eigen-distributions with respect to $\mathcal{A}_{\mathrm{dir}}^1$.*

ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

*Proof:* 1. Suppose that $0 \le \varepsilon \le 1/2$. By $d_\varepsilon$, we denote the distribution $d$ on the 1-set such that the probabilities of $d$ being $\omega_1, \omega_2, \omega_3$ and $\omega_4$ are $\varepsilon, 1/2-\varepsilon, 1/2-\varepsilon$ and $\varepsilon$, respectively. Let $j \in \{1, 2, \cdots, 8\}$. Note that $\varepsilon(2+4)+(1/2-\varepsilon)(3+3) = \varepsilon(3+3)+(1/2-\varepsilon)(2+4) = 3$. Therefore, by Table I, it holds that $C(A_j, d_\varepsilon) = 3$. Hence, for every $\varepsilon$ such that $0 \le \varepsilon \le 1/2$, $d_\varepsilon$ is a distribution on the 1-set, and the value $C(A_j, d_\varepsilon)$ does not depend on $j$. And, $d_\varepsilon$ is not the uniform distribution on the 1-set unless $\varepsilon = 1/4$.

The assertion 2 of the theorem is immediate from the assertion 1 and Theorem 4. ■

TABLE II
$C(A_D, \omega)$ FOR $k = 1$ ($\omega \in$ THE 0-SET).

| | | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|---|---|---|---|---|---|
| | | 1234 | 4312 | 3421 | 2143 |
| $\omega_5$ | 1000 | 3 | 2 | 2 | 4 |
| $\omega_6$ | 0100 | 4 | 2 | 2 | 3 |
| $\omega_7$ | 0010 | 2 | 4 | 3 | 2 |
| $\omega_8$ | 0001 | 2 | 3 | 4 | 2 |
| | | $A_5$ | $A_6$ | $A_7$ | $A_8$ |
| | | 3412 | 1243 | 2134 | 4321 |
| $\omega_5$ | 1000 | 2 | 3 | 4 | 2 |
| $\omega_6$ | 0100 | 2 | 4 | 3 | 2 |
| $\omega_7$ | 0010 | 3 | 2 | 2 | 4 |
| $\omega_8$ | 0001 | 4 | 2 | 2 | 3 |

On the other hand, $E^0$-distribution with respect to $\mathcal{A}^1_{\text{dir}}$ is unique. Sketch of the proof is as follows. Table II shows the values of $C(A_D, \omega)$ in the case where $\omega$ is an element of the 0-set. Now, under the assumption that a distribution $d$ on the 0-set has the same cost for all $A_j$ ($1 \le j \le 8$), set up equations on probabilities of $d$ being equal to $\omega_i$ ($5 \le i \le 8$). Then it is easy to see that the equations have a unique solution of $\text{prob}[d = \omega_i] = 1/4$ ($5 \le i \le 8$), in other words, $d$ is the uniform distribution on the 0-set.

By means of Theorem 6 and induction on $k$, we can show the following.

**Theorem 7.** *[8] For each positive integer $k$, the statements of Theorem 6 hold for $\mathcal{A}^k_{\text{dir}}$ in place of $\mathcal{A}^1_{\text{dir}}$.*

## V. A CASE WHERE THE UNIQUENESS HOLDS

In this section, we give an alternative proof for the characterization of the eigen-distribution (in the usual case) as the uniform distribution on the 1-set [4]. To be more precise, we show that (LT2$^{\mathcal{A}}$) implies (LT3) with respect to the non-directional algorithms (see Corollary 5).

An example of an element of $\mathcal{A}^1_D - \mathcal{A}^1_{\text{dir}}$ is the following algorithm. It begins with scanning the leaf of code 00. If a beta-cut does not happen there, the query-history is $\langle 00, 01, 10, 11\rangle$. Otherwise, the query-history is $\langle 00, 11, 10\rangle$, where the leaf-code 01 is skipped due to the beta-cut. By taking transpositions of this algorithm, we know that $\mathcal{A}^1_D - \mathcal{A}^1_{\text{dir}}$ consists of 8 algorithms.

**Theorem 8.** *[5] Suppose that $h \ge 2$, where $h$ is the height of $T$. Then, (LT2$^{\mathcal{A}}$) implies (LT3) with respect to $\mathcal{A} = \mathcal{A}_D - \mathcal{A}_{\text{dir}}$.*

*Proof:* For each $i \in \{0, 1\}$ and a positive integer $g$, let ($i$-set)$^g$ denote the $i$-set in the case of $h = g$. By induction on $h \ge 2$, we shall show the following requirement $R_h$.

$R_h$: "Suppose that $i \in \{0, 1\}$ and that a distribution $d$ on ($i$-set)$^h$ is an $E^i$-distribution with respect to $\mathcal{A} = \mathcal{A}_D - \mathcal{A}_{\text{dir}}$. Then, $d$ is the uniform distribution on ($i$-set)$^h$."

The base case $R_2$ is shown by solving equations; it is in the same way as our proof of the uniqueness of $E^0$-distribution with respect to $\mathcal{A}^1_{\text{dir}}$.

Suppose that $R_n$ holds. In the rest of the proof, let $h = n+1$, $i \in \{0, 1\}$ and assume that $d$ is a distribution on ($i$-set)$^{n+1}$ and that $d$ is an $E^i$-distribution with respect to $\mathcal{A}_D - \mathcal{A}_{\text{dir}}$. We investigate the case where the root is an AND-gate and $i = 1$. The other cases are shown in the same way.

Let $T_0$ ($T_1$, respectively) be the left (right) sub-tree just under the root. For each assignment $\alpha$ on $T_0$ (such that $\alpha \in$ (1-set)$^n$ and the denominator of (12) is positive), consider the distribution $d_\alpha$ on $T_1$ as follows. For each assignment $\beta$ on $T_1$, we let $\text{prob}[\ d_\alpha \text{ is } \beta\ ]$ as to be the following conditional probability, where $\alpha\beta$ denotes the concatenation of $\alpha$ and $\beta$.

$$\text{prob}[\ d \text{ is } \alpha\beta \mid \exists x\ d \text{ is } \alpha x\ ] \tag{12}$$

By the induction hypothesis $R_n$, for all $\alpha$ such that $\alpha \in$ (1-set)$^n$ and the denominator of (12) is positive, $d_\alpha$ is the uniform distribution on (0-set)$^n$. The same holds for the case where the roles of $T_0$ and $T_1$ are exchanged.

Now, by the induction hypothesis $R_n$, it is not hard to see that the requirement $R_{n+1}$ is satisfied. ■

## VI. CONCLUSIVE REMARKS

By extending the work of Tarsi [9], it is shown by Saks and Wigderson that the randomized complexity of an AND-OR tree is the same as that for directional algorithms; for more pricise, see [7, Theorem 5.2]. In the case of $T^k_2$, by using our results in § III, the above result is extended as follows.

**Proposition 9.** *Suppose that $\mathcal{A}$ is a non-empty subset of $\mathcal{A}_D$ and $\mathcal{A}$ is closed under transposition. Then, the following holds, where $d$ runs over all distributions.*

$$\max_d \min_{A_D \in \mathcal{A}_D} C(A_D, d) = \max_d \min_{A \in \mathcal{A}} C(A_D, d) \tag{13}$$

*Proof:* Let $d_{\text{unif.}}$ be the uniform distribution on the 1-set. By Lemma 2, the both sides of (13) are equal to the following.

$$\min_{A_D \in \mathcal{A}^k_D} C(A_D, d_{\text{unif.}}) = \min_{A_D \in \mathcal{A}} C(A_D, d_{\text{unif.}})$$

■

Hence, the $\mathcal{A}_D$ (the class of all deterministic algorithms for the tree) and $\mathcal{A}_{\text{dir}}$ (that of all directional algorithms) have the same distributional complexity.

In contrast, they do not agree on the question of "Which distribution achieves the equilibrium?" A variant of the no-free-lunch theorem implies the equivalence of "eigen" and "$E^1$", but it does not imply the uniqueness of the eigen-distribution. The set of all *non*-directional algorithms plays an important role to show the uniqueness.

## ACKNOWLEDGMENT

The authors would like to thank M. Kumabe, C.-G. Liu, Y. Niida, K. Tanaka and T. Yamazaki for helpful discussions.

## References


[1] Arora, S. and Barak, B.: *Comput. Complex.*, Cambridge university press, New York, 2009.

[2] Ho, Y.C. and Pepyne, D.L.: Simple explanation of the no-free-lunch theorem and its implications. *J. Optimiz. Theory App.*, **115** pp.549–570 (2002).

[3] Knuth, D.E. and Moore, R.W.: An analysis of alpha-beta pruning. *Artif. Intell.*, **6** pp. 293–326.

[4] Liu, C.-G. and Tanaka, K.: Eigen-distribution on random assignments for game trees. *Inform. Process. Lett.*, **104** pp.73–77 (2007).

[5] Nakamura, R.: Randomized complexity of AND-OR game trees. Master thesis, Department of Mathematics and Information Sciences, Tokyo Metropolitan University, Tokyo (2011).

[6] Pearl, J.: Asymptotic properties of minimax trees and game-searching procedures. *Artif. Intell.*, **14** pp.113–138 (1980).

[7] Saks, M. and Wigderson, A.: Probabilistic Boolean decision threes and the complexity of evaluating game trees. In: Proc. 27th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp.29–38 (1986).

[8] Suzuki, T.: Failure of the uniqueness of eigen-distribution on random assignments for game trees. *Sūrikaisekikenkyūsho-Kokyuroku*, **1729** pp.111–116, Research Institute for Mathematical Sciences, Kyoto University.

[9] Tarsi, M.: Optimal search on some game trees. *J. ACM*, **30** pp. 389–396 (1983).

[10] Walpert, D.H. and MacReady, W.G.: No-free-lunch theorems for search, *Technical report SFI-TR-95-02-010*, Santa Fe Institute, Santa Fe, New Mexico (1995).

[11] Yao, A.C.-C.: Probabilistic computations: towards a unified measure of complexity, in: *Proc. 18th annual IEEE symposium on foundations of computer science (FOCS)*, pp.222–227 (1977).


ISBN: 978-988-19251-1-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)