Information Processing Letters 125 (2017) 41–45

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Optimal depth-first algorithms and equilibria of independent distributions on multi-branching trees

Weiguang Peng [a], NingNing Peng [b,*], KengMeng Ng [c], Kazuyuki Tanaka [a], Yue Yang [d]

[a] Mathematical Institute, Tohoku University, Japan
[b] Department of Mathematics, Wuhan University of Technology, China
[c] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
[d] Department of Mathematics, National University of Singapore, Singapore

ARTICLE INFO

Article history:
Received 9 June 2016
Received in revised form 10 April 2017
Accepted 11 May 2017
Available online 17 May 2017
Communicated by R. Uehara

Keywords:
Multi-branching trees
Depth-first algorithms
Independent distribution
Computational complexity
Analysis of algorithms

ABSTRACT

The main purpose of this paper is to answer two questions about the distributional complexity of multi-branching trees. We first show that for any independent distribution d on assignments for a multi-branching tree, a certain directional algorithm DIR_d is optimal among all the depth-first algorithms (including non-directional ones) with respect to d. We next generalize Suzuki–Niida's result on binary trees to the case of multi-branching trees. By means of this result and our optimal algorithm, we show that for any balanced multi-branching AND–OR tree, the optimal distributional complexity among all the independent distributions (ID) is (under an assumption that the probability of the root having value 0 is neither 0 nor 1) actually achieved by an independent and identical distribution (IID).

1. Introduction

In this paper, we investigate the optimal depth-first algorithms and equilibria of independent distributions on multi-branching trees, in which every node may have different numbers of children. The height of a tree is the length of the longest path from the root to its leaves. Here, a balanced multi-branching tree means a tree such that all the non-terminal nodes at the same level have the same number of children and all paths from the root to the leaves are of the same length.

We first review some basic notions and results on game trees. An AND–OR tree (OR–AND tree, respectively) is a multi-branching tree such that the root is labeled AND (OR), and sequentially the internal nodes are level-by-level labeled by OR and AND (AND and OR) alternatively. An assignment for a tree is a mapping from the set of leaves to Boolean values $\{0, 1\}$. By evaluating a tree, we mean to compute the Boolean value of the root. For a given assignment, the cost of computation is defined to be the number of leaves that are queried to evaluate a tree. When we consider probability distributions on the set of assignments, the cost of computation is the expected cost under the given distribution.

An algorithm tells us a priority of searching leaves. An algorithm is called *alpha–beta pruning* if it checks only sufficiently many nodes to determine the value of the current subtree [1]. We assume that all the algorithms discussed here are deterministic alpha–beta pruning algorithms. A *directional* algorithm is one that queries the leaves on a given tree in a fixed order. *SOLVE* is a directional algorithm which evaluates a tree from left to right [3]. A *depth-first algorithm* is one that never jumps to another subtree until it completes evaluating the current one. For a given probabil-

* Corresponding author.
E-mail address: pengn@whut.edu.cn (N. Peng).

http://dx.doi.org/10.1016/j.ipl.2017.05.002
0020-0190/

ity distribution on the assignments, we are seeking for the optimal algorithms that can minimize the expected cost of computation under the given distribution.

The *deterministic complexity* is defined to be the minimum cost to compute the worst assignment for a tree, i.e., $\min_{A_D}\max_{\omega} C(A_D,\omega)$, where $C(A_D,\omega)$ is the cost of an algorithm A_D on an assignment ω, A_D ranges over all the deterministic algorithms and ω ranges over all the assignments. A *randomized algorithm* is a distribution over a family of deterministic algorithms. Then the *randomized complexity* to evaluate a tree is defined as $\min_{A_R}\max_{\omega} C(A_R,\omega)$, where $C(A_R,\omega)$ is the expected cost over the corresponding family of deterministic algorithms, A_R ranges over all the randomized algorithms and ω ranges over all the assignments. Obviously, the randomized complexity is not larger than the deterministic complexity.

Saks and Wigderson [5] calculated the randomized complexity for any balanced n-branching tree (each non-terminal node has n children) with height h to be $\Theta((\frac{n-1+\sqrt{n^2+14n+1}}{4})^h)$. Yao's principle [9] indicates that the randomized complexity is equivalent to the *distributional complexity*, $\max_{d}\min_{A_D} C(A_D,d)$ with A_D ranging over the deterministic algorithms and d over the distributions on assignments. Yao's principle provides a profound perspective to analyze randomized algorithms.

Liu and Tanaka [2], subsequent to the study of Saks and Wigderson, investigated the uniform binary trees from the viewpoint of distributional complexity. A distribution δ is said to achieve the distributional complexity (or equilibrium) if and only if $\min_{A_D} C(A_D,\delta) = \max_{d}\min_{A_D} C(A_D,d)$. They assert that for any uniform binary AND–OR tree, if the equilibrium is achieved by an independent distribution (ID), then it is, in fact, an independent and identical distribution (IID). However, [2] does not include a proof of the assertion. Recently, Suzuki and Niida [7] gave a proof for the case where the probability of the root is constrained for uniform binary trees and showed Liu–Tanaka's assertion.

We treat probability distributions on multi-branching trees. In Section 2, for any ID d, we define a directional algorithm DIR_d, and show it is optimal among all the depth-first algorithms with respect to d for any multi-branching tree. Recall Tarsi's theorem [8] that SOLVE is optimal for IID. Our result is on ID among all the depth-first algorithms (with certain conditions) while Tarsi's theorem is on IID among all the algorithms not necessarily depth-first. In Section 3, we first extend the fundamental relationships between the minimum expected cost and the probability of the root of tree being 0 in [7] to balanced multi-branching trees. Based on this, we show that, for any ID d, there exists an IID d' such that the expected cost with d is not larger than that with d' following DIR_d. Then we establish Liu–Tanaka's assertion for any balanced multi-branching AND–OR tree (under an assumption that the probability of the root having value 0 is neither 0 nor 1).

2. DIR_d is optimal among all the depth-first algorithms

Let Ω be the set of assignments for a given tree. We say $d:\Omega\to[0,1]$ is an *independent distribution* (denote $d\in$ ID) if there exist p_i's (the probability of the i-th leaf being 0) such that for any $\omega\in\Omega$, $d(\omega)=\prod_{\{i:\ \omega(i)=0\}} p_i \prod_{\{i:\ \omega(i)=1\}} (1-p_i)$. We say $d\in$ IID if d is an ID satisfying $p_1=p_2=\cdots=p_n$. By $C(A,\omega)$, we denote the number of leaves checked by an algorithm A with an assignment ω.

Given $d\in$ ID and an algorithm A, for each node σ on T, we define $C_\sigma(A,d)$ and $p_\sigma(d)$ to be the evaluation cost of σ and the probability of σ being 0. Remark that if σ is a leaf, then $C_\sigma(A,d)=1$ and $p_\sigma(d)=p_i$. If σ is a non-terminal node and A is an algorithm on T_σ, $C_\sigma(A,d)$ is the expected cost of computing the value of σ following A, and $p_\sigma(d)$ is the probability of σ being 0.

For any non-terminal node σ in T, T_σ denotes the subtree of T rooted from σ. For a node σ with n children, $T_{\sigma*i}$ $(1\le i\le n)$ denotes the i-th subtree under σ from left to right, and particularly for the root λ, its subtree is simplified as T_i. For simplicity, we denote $C(A,d)=C_\lambda(A,d)$ at root λ, and $q_\sigma=p_\sigma(d)$ at any node σ. By q_i, we denote the probability of the root of T_i being 0 with respect to d.

Definition 1. For any uniform binary tree T and $d\in$ ID on T, the depth-first directional algorithm DIR_d is defined inductively as follows. The basic case is trivial. For the induction case, let $\sigma*i$ $(i=1,2)$ be a child of non-terminal node σ, and assume $\mathrm{DIR}_{d_{\sigma*i}}$ has been defined for each subtree $T_{\sigma*i}$.

(1) In the case that σ is labeled $\wedge$, DIR_{d_σ} is the concatenation of $\mathrm{DIR}_{d_{\sigma*1}}$ and $\mathrm{DIR}_{d_{\sigma*2}}$ (denote $\mathrm{DIR}_{d_\sigma}:=\mathrm{DIR}_{d_{\sigma*1}}\cdot\mathrm{DIR}_{d_{\sigma*2}}$) if $\frac{C_{\sigma*1}(\mathrm{DIR}_{d_{\sigma*1}},\,d_{\sigma*1})}{q_{\sigma*1}}\le\frac{C_{\sigma*2}(\mathrm{DIR}_{d_{\sigma*2}},\,d_{\sigma*2})}{q_{\sigma*2}}$, otherwise $\mathrm{DIR}_{d_\sigma}:=\mathrm{DIR}_{d_{\sigma*2}}\cdot\mathrm{DIR}_{d_{\sigma*1}}$.

(2) In the case that σ is labeled $\vee$, $\mathrm{DIR}_{d_\sigma}:=\mathrm{DIR}_{d_{\sigma*1}}\cdot\mathrm{DIR}_{d_{\sigma*2}}$ if $\frac{C_{\sigma*1}(\mathrm{DIR}_{d_{\sigma*1}},\,d_{\sigma*1})}{1-q_{\sigma*1}}\le\frac{C_{\sigma*2}(\mathrm{DIR}_{d_{\sigma*2}},\,d_{\sigma*2})}{1-q_{\sigma*2}}$, otherwise $\mathrm{DIR}_{d_\sigma}:=\mathrm{DIR}_{d_{\sigma*2}}\cdot\mathrm{DIR}_{d_{\sigma*1}}$.

Theorem 1. *For any uniform binary tree T and $d\in$ ID, if A is any depth-first algorithm, then $C(A,d)\ge C(\mathrm{DIR}_d,d)$, i.e., DIR_d is optimal among all the depth-first algorithms.*

Proof. We prove this by induction on height h. The base case is trivial. For the induction step, let T be a uniform binary tree with height $h+1$, where the root λ is labeled $\wedge$. The other case can be shown similarly.

Suppose that DIR_{d_i} is optimal for each subtree T_i with height h. Let Ω_{h+1} be the set of assignments for T, Ω_h and Ω'_h the set of assignments for T_1 and T_2. For any $d\in$ ID on T, there exist d_i for $T_i (i=1,2)$ such that $d(\omega)=d_1(\omega_1)\times d_2(\omega_2)$, where $\omega=\omega_1\omega_2$, $\omega_1\in\Omega_h$ and $\omega_2\in\Omega'_h$. For any depth-first algorithm A and $d\in$ ID, if A evaluates the subtree T_1 first, then $C(A,d)=\sum_{\omega\in\Omega_{h+1}} C(A,\omega)\cdot d(\omega)=\sum_{\omega\in\Omega^0} C(A,\omega)\cdot d(\omega)+\sum_{\omega\in\Omega^1} C(A,\omega)\cdot d(\omega)$, where $\Omega^i:=\{\omega\in\Omega_{h+1}\mid$ the root of T_1 has value i with $\omega\}$.

Assume A is a depth-first non-directional algorithm. By A_1, we denote the algorithm of A for T_1, and by A_{ω_1}

the algorithm of A for T_2 depending on the assignment ω_1 in T_1. Thus, $C(A,d) = \sum_{\omega_1\in\Omega_h^0} C_{\lambda*1}(A_1,\omega_1)\cdot d_1(\omega_1) + \sum_{\omega_1\in\Omega_h^1}\sum_{\omega_2\in\Omega_h'} (C_{\lambda*1}(A_1,\omega_1) + C_{\lambda*2}(A_{\omega_1},\omega_2))\cdot d_2(\omega_2)d_1(\omega_1)$, where $\Omega_h^i := \{\omega_1 \in \Omega_h \mid \text{the root of } T_1 \text{ has value } i \text{ with } \omega_1\}$. We calculate that $C(A,d) = C_{\lambda*1}(A_1,d_1) + \sum_{\omega_1\in\Omega_h^1} d_1(\omega_1)\cdot C_{\lambda*2}(A_{\omega_1},d_2)$. By induction hypothesis, we can take algorithms DIR_{d_1} and DIR_{d_2} such that $C(A,d) \geq C_{\lambda*1}(\mathrm{DIR}_{d_1},d_1) + (1-q_1)C_{\lambda*2}(\mathrm{DIR}_{d_2},d_2)$. Let $A_1' = \mathrm{DIR}_{d_1}\cdot \mathrm{DIR}_{d_2}$, then clearly $C(A,d) \geq C(A_1',d)$.

Using the similar arguments as above, if A evaluates T_2 first, we can get algorithm $A_2' = \mathrm{DIR}_{d_2}\cdot\mathrm{DIR}_{d_1}$. Thus, $C(A,d) \geq C(A_2',d)$. If $\frac{C_{\lambda*1}(\mathrm{DIR}_{d_1},d_1)}{q_1} \leq \frac{C_{\lambda*2}(\mathrm{DIR}_{d_2},d_2)}{q_2}$, then $C(A_1',d) = C(A_2',d) - (q_1C_{\lambda*2}(\mathrm{DIR}_{d_2},d_2) - q_2C_{\lambda*1}(\mathrm{DIR}_{d_1},d_1)) \leq C(A_2',d)$. By Definition 1, DIR_d is A_1' if $\frac{C_{\lambda*1}(A_{d_1},d_1)}{q_1} \leq \frac{C_{\lambda*2}(A_{d_2},d_2)}{q_2}$, otherwise A_2'. Clearly, DIR_d is optimal among all the depth-first algorithms. $\square$

Now we define DIR_d for multi-branching trees.

Definition 2. For any multi-branching tree T, given $d \in \mathrm{ID}$, the depth-first directional algorithm DIR_d is defined inductively as follows. The basic case is trivial. For the induction case, assume T_σ has n subtrees $T_{\sigma*i}$ $(i = 1,\cdots,n)$ and $\mathrm{DIR}_{d_{\sigma*i}}$ has been defined for each subtree $T_{\sigma*i}$.

(1) In the case that σ is labeled $\wedge$, for the lexicographically minimal permutation f of $\{1,\cdots,n\}$ such that $\frac{C_{\sigma*f(1)}(\mathrm{DIR}_{d_{\sigma*f(1)}},d_{\sigma*f(1)})}{q_{\sigma*f(1)}} \leq \cdots \leq \frac{C_{\sigma*f(n)}(\mathrm{DIR}_{d_{\sigma*f(n)}},d_{\sigma*f(n)})}{q_{\sigma*f(n)}}$, $\mathrm{DIR}_d := \mathrm{DIR}_{d_{\sigma*f(1)}}\cdot\cdots\cdot\mathrm{DIR}_{d_{\sigma*f(n)}}$.
(2) In the case that σ is labeled $\vee$, for the lexicographically minimal permutation f of $\{1,\cdots,n\}$ such that $\frac{C_{\sigma*f(1)}(\mathrm{DIR}_{d_{\sigma*f(1)}},d_{\sigma*f(1)})}{1-q_{\sigma*f(1)}} \leq \cdots \leq \frac{C_{\sigma*f(n)}(\mathrm{DIR}_{d_{\sigma*f(n)}},d_{\sigma*f(n)})}{1-q_{\sigma*f(n)}}$, $\mathrm{DIR}_d := \mathrm{DIR}_{d_{\sigma*f(1)}}\cdot\cdots\cdot\mathrm{DIR}_{d_{\sigma*f(n)}}$.

Theorem 2. *For any multi-branching tree T, if $d \in \mathrm{ID}$ and A is any depth-first algorithm, then $C(A,d) \geq C(\mathrm{DIR}_d,d)$.*

Proof. We prove this by induction on the height of T and the number of children of the root, that is, by induction on the sum of the height and the number of children. Suppose that T is a tree with height $h+1$ where the root λ is labeled $\wedge$. The case where λ is labeled $\vee$ can be shown in the similar way. Without loss of generality, we assume T has n subtrees denoted by T_i $(i\in\{1,\cdots,n\})$.

At first, we look at the case that A evaluates T_1 first. The remaining parts of T is denoted by T_1'. For any distribution $d \in \mathrm{ID}$ and depth-first algorithm A that evaluates T_1 first, by induction hypothesis, we can find depth-first directional algorithms DIR_{d_1} and $\mathrm{DIR}_{d_1'}$ for T_1 and T_1' respectively, where d_1 (d_1') is an ID distribution of d restricted to T_1 (T_1'). Let $A_1' = \mathrm{DIR}_{d_1}\cdot\mathrm{DIR}_{d_1'}$. We have $C(A,d) \geq C(A_1',d)$.

Similarly, for all the subtrees T_i's under the root of T, we can get n directional algorithms $A_1',\cdots,A_n'$ such that A_i' evaluates subtree T_i first. Then our desired algorithm DIR_d is at least one of A_i''s.

Let f be a permutation of $\{1,\cdots,n\}$ such that $\frac{C_{\lambda*f(1)}(\mathrm{DIR}_{d_{\lambda*f(1)}},d_{\lambda*f(1)})}{q_{f(1)}} \leq \cdots \leq \frac{C_{\lambda*f(n)}(\mathrm{DIR}_{d_{\lambda*f(n)}},d_{\lambda*f(n)})}{q_{f(n)}}$. We can show that $A'_{f(1)} = \mathrm{DIR}_{d_{\lambda*f(1)}}\cdot\cdots\cdot\mathrm{DIR}_{d_{\lambda*f(n)}} = \mathrm{DIR}_d$, and compute $C(A'_{f(1)},d) = \sum_{i=1}^{n}\prod_{j=1}^{i-1}(1-q_{f(j)})C_{\lambda*f(i)}(\mathrm{DIR}_{d_{\lambda*f(i)}},d_{\lambda*f(i)})$. Then we have $C(A'_{f(1)},d) \leq C(A'_{f(i)},d)$ for all i. Thus DIR_d is optimal among all the depth-first algorithms. $\square$

3. Distributional complexity for independent distributions on balanced multi-branching trees

For any balanced multi-branching tree T, we consider an IID on T such that each leaf is assigned 0 with probability x. Notice that when the distribution on T is an IID, any depth-first directional algorithm produces the same cost as SOLVE. Furthermore, since T is a balanced multi-branching tree, $C_\sigma(A,d)$ and $p_\sigma(d)$ are actually functions independent from the algorithms. Thus, we may denote the expected cost as $C_\sigma(x)$ or $C_{\wedge,h}(x)$, where h is the height of σ on T, and $p_\sigma(d)$ as $p_\wedge(x)$ or $p(x)$ etc.

3.1. Suzuki–Niida lemma for multi-branching trees

We now prove a technical lemma which is a generalization of "fundamental relationships between costs and probabilities" due to Suzuki and Niida in [7].

Lemma 1. *Suppose that the distribution on T is an IID with all leaves assigned probability x and we follow the algorithm SOLVE (or equivalently any depth-first directional algorithm A). Then*

(1) $p_\sigma(x)$ is a strictly increasing function of x.
(2) $\frac{C_\sigma(x)}{p_\sigma(x)}$ is strictly decreasing.
(3) $\frac{C_\sigma'(x)}{p_\sigma'(x)}$ is strictly decreasing if σ is not a leaf; and at leaves $\frac{C_\sigma'(x)}{p_\sigma'(x)} = 0$ is non-increasing, where the primes denote differentiation with respect to x.

Proof. Statement (1) is easy to prove. When σ is a leaf, $p_\sigma(x) = x$, then the statement holds. Suppose $p_{\vee,h}(x)$ and $p_{\wedge,h}(x)$ are strictly increasing and each node at height $h+1$ has n children. Then, $p_{\wedge,h+1}(x) = 1-(1-p_{\vee,h}(x))^n$ and $p_{\vee,h+1}(x) = p^n_{\wedge,h}(x)$ are both strictly increasing.

Next we prove (2) and (3) by simultaneous induction. The base case is trivial. For inductive step, fix a node σ at height $h+1$ having n children. Suppose that (2) and (3) hold for all children of σ with height h.

Case 1. σ is labeled $\vee$. Using the similar arguments as Theorem 1, we have the following recursive equations (we drop x for simplicity): $C_{\vee,h+1} = C_{\wedge,h}(1+p_{\wedge,h}+\cdots+p^{n-1}_{\wedge,h})$, and $p_{\vee,h+1} = p^n_{\wedge,h}$.

From now on, we further drop the subscripts $\wedge$ and h on the right hand side. For statement (2), we have $\frac{C_{\vee,h+1}}{p_{\vee,h+1}} = \frac{C}{p^n}(1+p+\cdots+p^{n-1}) = \frac{C}{p}\left(\frac{1}{p^{n-1}}+\cdots+\frac{1}{p}+1\right)$. By induction hypothesis and (1), the terms $\frac{C}{p}, \frac{1}{p^{n-1}}, \ldots, \frac{1}{p}$

are strictly decreasing and positive; hence $\frac{C_{\vee,h+1}}{p_{\vee,h+1}}$ is also strictly decreasing, i.e., (2) holds.

Next we look at (3). By differentiating $C_{\vee,h+1}(x)$ and $p_{\vee,h+1}(x)$ with respect to x, we get

$$\frac{C'_{\vee,h+1}}{p'_{\vee,h+1}} = \frac{C'}{p'}\frac{1}{n}(\frac{1}{p^{n-1}} + \cdots + \frac{1}{p} + 1) + \frac{1}{n}\frac{C}{p}(\frac{1}{p^{n-2}} + \frac{2}{p^{n-3}} + \cdots + (n-1)).$$

By induction hypothesis and similar arguments as above, $\frac{C'_{\vee,h+1}}{p'_{\vee,h+1}}$ is strictly decreasing. Moreover, the second summand makes it strictly decreasing. That establishes (3) for this case.

Case 2. σ is labeled $\wedge$. Then we have the following recursive equations (we drop x for simplicity):

$$C_{\wedge,h+1} = C_{\vee,h}(1 + (1-p_{\vee,h}) + \cdots + (1-p_{\vee,h})^{n-1}),$$
$$p_{\wedge,h+1} = 1 - (1-p_{\vee,h})^n = p_{\vee,h}(1 + (1-p_{\vee,h}) + \cdots + (1-p_{\vee,h})^{n-1}).$$

Hence, $\frac{C_{\wedge,h+1}}{p_{\wedge,h+1}} = \frac{C_{\vee,h}}{p_{\vee,h}}$ is strictly decreasing by induction hypothesis, which establishes (2) for case 2.

To prove statement (3) for case 2, we can first show the following statements by induction on height h:

$$\forall h,\ C_{\vee,h}(x) = C_{\wedge,h}(1-x) \text{ and } p_{\vee,h}(x) = 1 - p_{\wedge,h}(1-x). \qquad (*)$$

Consequently, by substituting x by $1-x$, we have the dual equations $C_{\wedge,h}(x) = C_{\vee,h}(1-x)$ and $p_{\wedge,h}(x) = 1 - p_{\vee,h}(1-x)$. Then we have $\frac{C'_{\wedge,h+1}(x)}{p'_{\wedge,h+1}(x)} = -\frac{C'_{\vee,h+1}(1-x)}{p'_{\vee,h+1}(1-x)}$. By the results on the $\vee$-case, we know that $\frac{C'_{\wedge,h+1}(x)}{p'_{\wedge,h+1}(x)}$ is also strictly decreasing, which completes the proof of Lemma 1. □

3.2. Getting more uniformity while increasing the cost and keeping the probability

The next lemma illustrates how to make one step advance towards uniformity. For this, we focus on a single node σ on T. Let $\tau_1, \ldots, \tau_n$ be all children of σ from left to right. We assume that the distribution on T_σ is a segment-wise IID, in the sense that for every child τ_i of σ, the distribution restricted to T_{τ_i} is an IID with probability x_i. We may assume that $x_1 \le \cdots \le x_n$.

For the distribution d as above, we define the algorithm DIR_d as follows: if σ is labeled $\vee$, then DIR_d evaluates the value of τ_i's following the increasing order of their probability, i.e., evaluates τ_1 first and τ_n last; if σ is labeled $\wedge$, then DIR_d evaluates τ_n first and τ_1 last. The evaluation of each τ_i is done by SOLVE as the distribution restricted to T_{τ_i} is an IID. The DIR_d defined in Definition 2 and that defined here are the same up to isomorphism of trees, in the following sense. For DIR_d defined here, there is a permutation f of child nodes, though f is not necessarily lexicographically minimal, such that the inequality of Definition 2 holds.

Lemma 2. *Let d be as above. Suppose that $p_\sigma(d) \neq 0, 1$. If there is an i such that $x_i < x_{i+1}$ then we can find another distribution d' with the following properties: d' restricted to T_σ is a segment-wise IID; $C(\mathrm{DIR}_{d'}, d') = C(\mathrm{DIR}_d, d') > C(\mathrm{DIR}_d, d)$; and $p_\sigma(d) = p_\sigma(d')$.*

Proof. Let $x_1 \le x_2 \le \cdots \le x_i = x < y = x_{i+1} \le \cdots \le x_n$. We show that we can adjust the values of x and y so that the cost strictly increases. We have two cases:

Case 1. σ is labeled $\vee$. Our new distribution d' agrees with d at x_j for all $j \notin \{i, i+1\}$; and the adjusted values of x and y still satisfy the constraint $x_{i-1} \le x < y \le x_{i+2}$ (here we assume $x_0 = 0$ and $x_{n+1} = 1$). Since all parameters other than x and y are fixed during the discussion, we may view the cost and probability functions as functions of arguments x and y, and denote them by $C(x, y)$ and $p(x, y)$ respectively.

Following the algorithm DIR_d, which goes from x_1 to x_n, we have $C(x, y) = \gamma + \beta(C_\wedge(x) + p_\wedge(x)(C_\wedge(y) + p_\wedge(y)\alpha))$ and $p(x, y) = \delta p_\wedge(x)p_\wedge(y)$, where $C_\wedge, p_\wedge$ are the cost and probability functions associated with any child of σ; and α, β, γ and δ are constants with $\beta > 0$ if $p_\sigma(d) \neq 0$. We may further drop γ, β as it does not affect the optimization. Since we want to keep $p_\sigma(d') = p_\sigma(d)$, we may assume $p_\wedge(x)p_\wedge(y)$ is some constant e. We view x as a function of y which is implicitly defined by $p_\wedge(x)p_\wedge(y) = e$, and claim that $f(y) = C(x(y), y) = C_\wedge(x) + p_\wedge(x)C_\wedge(y) + \alpha p_\wedge(x)p_\wedge(y)$ is strictly decreasing with respect to y.

To show this, it suffices to show $f'(y) < 0$. We drop the subscript $\wedge$. $f'(y) = C'(x)\frac{dx}{dy} + p'(x)\frac{dx}{dy}C(y) + p(x)C'(y)$. Since $p(x)p(y) = e$, we have $\frac{dx}{dy} = -\frac{p(x)p'(y)}{p'(x)p(y)}$. Hence $f'(y) = -\frac{C'(x)p(x)p'(y)}{p'(x)p(y)} - \frac{p(x)p'(y)C(y)}{p(y)} + p(x)C'(y)$. Since $-\frac{p(x)p'(y)C(y)}{p(y)} < 0$, to show $f'(y) < 0$ it suffices to show that $p(x)C'(y) - \frac{C'(x)p(x)p'(y)}{p'(x)p(y)} < 0$.

By Lemma 1, $\frac{C'}{p'}$ is strictly decreasing. Since $x < y$, we have $\frac{C'(y)}{p'(y)} < \frac{C'(x)}{p'(x)}$. Furthermore the latter is $\le \frac{1}{p(y)}\frac{C'(x)}{p'(x)}$ as $p(y) \le 1$. So $f'(y) < 0$. Hence f is decreasing with respect to y.

Thus if we set y' to be $y - \epsilon$ where $\epsilon > 0$ is chosen so that the unique x' satisfying $p(x')p(y') = e$ is strictly less than y', we can still keep the constraint $x_{i-1} < x' < y' < x_{i+2}$, but increase the cost $f(y')$. In addition, by the proof of Theorem 1, $C(\mathrm{DIR}_{d'}, d') = C(\mathrm{DIR}_d, d')$.

Case 2. σ is labeled $\wedge$. In this case, DIR_d goes from right to left. Thus the cost at σ is $C_\vee(x_n) + (1 - p_\vee(x_n))[C_\vee(x_{n-1}) + (1 - p_\vee(x_{n-1}))(C_\vee(x_{n-2}) + \cdots + (1 - p_\vee(x_2))C_\vee(x_1))]$. Here $C_\vee$ and $p_\vee$ stand for $C_{\vee,|\sigma|}$ and $p_{\vee,|\sigma|}$ respectively, and we drop the index $|\sigma|$ for simplicity. By $(*)$ in Lemma 1, this cost is $C_\wedge(1-x_n) + p_\wedge(1-x_n)[C_\wedge(1-x_{n-1}) + p_\wedge(1-x_{n-1})(C_\wedge(1-x_{n-2}) + \cdots + p_\wedge(1-x_2)C_\wedge(1-x_1))]$. But this is just the cost at a $\vee$-node with probability $1-x_n \le \cdots \le 1-x_1$ with respect to algorithm DIR_d. Together with the fact that keeping $1-(1-p_\vee(x_n))\cdots(1-p_\vee(x_1))$ to be a constant is the same as keeping $p_\wedge(1-x_n)\cdots p_\wedge(1-x_1)$ to be a constant, our argument for $\vee$-case can be transferred to $\wedge$-case also. □

Corollary 1. *Given a node σ with a segment-wise IID distribution $\bar{d} = x_1, \cdots, x_n$, there exists $d \in ID$ such that $C_\sigma(DIR_{\bar{d}}, d) = \max_{e \in D} C_\sigma(DIR_{\bar{d}}, e)$ where $D = \{(x_1, x_2, \cdots, x_n) : 0 \le x_1 \le \cdots \le x_n \le 1$, and $p_\sigma(e) = p_\sigma(\bar{d})\}$. Moreover, when $p_\sigma(\bar{d}) \neq 0, 1$, the maximal value is achieved if and only if $d \in IID$.*

Proof. Since $C_\sigma(\mathrm{DIR}_{\bar{d}}, d)$ is a continuous function of d on $[0,1]^n$ and D a compact subset of $[0,1]^n$, there is some $d_0 \in D$ at which the maximal value is achieved. If $p_\sigma(d) \neq 0, 1$, by Lemma 2, d_0 has to be an IID. □

3.3. Analysis of cost for independent distributions with respect to DIR_d

Lemma 3. *Given $d \in ID$ on T, there exists $d' \in IID$ such that $C(DIR_d, d) \le C(DIR_d, d')$, and the probability at the root λ, $p_\lambda(d') = p_\lambda(d)$. Furthermore the equation holds if and only if $d \in IID$ or $(p_\lambda(d) = 0$ or $1)$.*

Proof. First observe that once d is given, we can calculate the parameter $p_\sigma(d)$ on each node $\sigma \in T$. We define the algorithm DIR_d as follows. For any two nodes σ_1 and σ_2 of T, we have the following two cases: case 1, one node is a descendant of the other, say, σ_2 is a descendant of σ_1, then evaluate σ_2 first; case 2, σ_1 and σ_2 are incomparable. Let σ_0 be the node on T such that τ_1 and τ_2 are children of σ_0 satisfying that σ_i is a descendant of τ_i for $i = 1, 2$. Suppose that $p_{\tau_1}(d) \le p_{\tau_2}(d)$. If σ_0 is labeled $\vee$, then evaluate τ_1 first; if σ is labeled $\wedge$, then evaluate τ_2 first.

We inductively define a sequence of distributions $d_0, \ldots, d_h$ on T, where h is the height of T, such that

- $d_0 = d$ which is the given distribution, d_h is an IID, and d_{i+1} is *one step more uniform* than d_i, i.e., if d_i restricted to T_σ is a segment-wise IID, then d_{i+1} restricted to T_σ is an IID and
- for all σ of height $\ge i$, $p_\sigma(d_i) = p_\sigma(d)$ and $C_\sigma(\mathrm{DIR}_d, d_i) \ge C_\sigma(\mathrm{DIR}_d, d)$.

Let $d_0 = d$. Suppose that d_i has been defined and for each node σ of height $i + 1$, d_i restricted to T_σ is a segment-wise IID. To be more precise, let $\tau_1, \ldots, \tau_n$ be the children of σ, and d_i restricted to T_{τ_j} is an IID with x_j. Then let d_{i+1} be the distribution d' such that when restricted to T_σ, d' is an IID with leaves having probability x such that $p_\sigma(x) = p_{\tau_1}(x_1) \cdots p_{\tau_n}(x_n)$ if σ is labeled $\vee$; and $p_\sigma(x) = 1 - (1 - p_{\tau_1}(x_1)) \cdots (1 - p_{\tau_n}(x_n))$ if σ is labeled $\wedge$.

By induction hypothesis, d_{i+1} satisfies that for each node σ of height $i + 1$, $p_\sigma(d_{i+1}) = p_\sigma(d_i) = p_\sigma(d_0)$. By Lemma 2, $C_\sigma(\mathrm{DIR}_d, d_{i+1}) \ge C_\sigma(\mathrm{DIR}_d, d_i) \ge C_\sigma(\mathrm{DIR}_d, d_0)$. Then we can take d' to be d_h. □

By Lemma 3, we have the following theorem.

Theorem 3. *For any balanced multi-branching AND–OR tree T, suppose that $d \in ID$ and $p_\lambda(d) \neq 0$ or 1. If d achieves the distributional complexity, then, $d \in IID$.*

4. Future works

Until now, our results on game trees are all restricted to depth-first algorithms. Using the method of transpositions at some nodes in [6,4], we can show that for any ID, the minimum cost among the depth-first directional algorithms is not larger than that among the non-depth-first directional algorithms for any uniform binary tree with height 2.

We would like to construct the optimal non-depth-first algorithms. Another challenge is to consider the distributional complexity with respect to certain classes of non-depth-first algorithms. In [5], Saks and Wigderson proposed the notion of leaf cost function. Instead of assuming unit cost for each leaf, they consider the cost as a function of leaf and its value. We would also like to investigate the eigen-distribution for game trees with a leaf cost function.

Acknowledgements

The first author would like to thank professor Toshio Suzuki and Shohei Okisaka for their valuable comments and discussions. This work was supported by the Grants-in-Aid for Science Research (Japan): No. 2654001 and No. 15H03634, JSPS-NUS Bilateral Joint Research Projects J150000618 (PI's: K. Tanaka, C.T. Chong).

References

[1] D.E. Knuth, R.W. Moore, An analysis of alpha–beta pruning, Artif. Intell. 6 (4) (1975) 293–326.
[2] C.G. Liu, K. Tanaka, Eigen-distribution on random assignments for game trees, Inf. Process. Lett. 104 (2) (2007) 73–77.
[3] J. Pearl, Asymptotic properties of minimax trees and game-searching procedures, Artif. Intell. 14 (2) (1980) 113–138.
[4] W. Peng, S. Okisaka, W. Li, K. Tanaka, The eigen-distribution for multi-branching trees, in: Proc. 24th International MultiConference of Engineers and Computer Scientists, 2016, pp. 88–93.
[5] M. Saks, A. Wigderson, Probabilistic Boolean decision trees and the complexity of evaluating game trees, in: Proc. 27th Annual IEEE Symposium on Foundations of Computer Science, 1986, pp. 29–38.
[6] T. Suzuki, R. Nakamura, The eigen distribution of an AND–OR tree under directional algorithms, IAENG Int. J. Appl. Math. 42 (2) (2012) 122–128.
[7] T. Suzuki, Y. Niida, Equilibrium points of an AND–OR tree: under constraints on probability, Ann. Pure Appl. Log. 166 (11) (2015) 1150–1164.
[8] M. Tarsi, Optimal search on some game trees, J. ACM 30 (3) (1983) 389–396.
[9] A.C.C. Yao, Probabilistic computations: toward a unified measure of complexity, in: Proc. 18th Annual IEEE Symposium on Foundations of Computer Science, 1977, pp. 222–227.