Kazuyuki Tanaka’s work on AND-OR trees and subsequent development

Toshio Suzuki

Department of Math. and Information Sciences, Tokyo Metropolitan University,

CTFM 2015, Tokyo Institute of Technology

September 7–11, 2015
Abstract

Searching a game tree is an important subject of artificial intelligence. In the case where the evaluation function is bi-valued, the subject is interesting for logicians, because a game tree in this case is a Boolean function.

Kazuyuki Tanaka has a wide range of research interests which include complexity issues on AND-OR trees. In the joint paper with C.-G. Liu (2007) he studies distributional complexity of AND-OR trees. We overview this work and subsequent development.
Outline

1. Abstract
2. Background
3. Tanaka’s work (1)
4. (2)
5. Extension (1)
6. (2)
7. (3)
8. References
9. Appendix: Graphs
Min-max search on a game tree

VALUES of the EVALUATION FUNCTION

Time of computing $\simeq \#$ of times of calling the evaluation function
Our setting

A uniform binary AND-OR tree $T_2^k$

- $\land = \text{AND} = \text{Min.}$
- $\lor = \text{OR} = \text{Max.}$
- $T_2^{k+1}$ is defined by replacing each leaf of $T_2^k$ with $T_2^1$.
- Find: root = 1 (TRUE) or 0 (FALSE)?
- Each leaf is hidden.
- Cost := # of leaves probed
- Allowed to skip a leaf ($\alpha$-$\beta$ pruning).
Alpha-beta pruning algorithm

Definition.

- Depth-first.
- A child of an AND-gate has the value 0

\[\downarrow\]

Recognize that the AND-gate has the value 0 without probing the other child (an alpha-cut).

- Similar rule applies to an OR-gate (a beta-cut).

The optimality of alpha-beta pruning algo. for IID

- ID = independent distribution
- IID = independent and identical distribution
- CD = correlated distribution

In the case of IID:

The optimality of alpha-beta pruning algorithms is studied by Baudet (1978) and Pearl (1980), and the optimality is shown by Pearl (1982) and Tarsi (1983).

A variant of von Neumann’s min-max theorem

Yao’s Principle (1977)

\[
\min_{A_R} \max_{\omega} \text{cost}(A_R, \omega) = \max_{d} \min_{A_D} \text{cost}(A_D, d),
\]

\(\omega\) : truth assignment  \(A_D\) : deterministic algorithm

\(A_R\) : randomized algo.  \(d\) : prob. distribution on the truth assignments

Yao, A.C.-C.: Probabilistic computations: towards a unified measure of complexity.
Estimation of the equilibrium value

Saks and Wigderson (1986)

For a perfect binary AND-OR tree,

\[(\text{Randomized Complexity}) \approx (\text{Constant}) \times \left(\frac{1 + \sqrt{33}}{4}\right)^h,\]

where \(h\) is the height of the tree.

Saks, M. and Wigderson, A.:
Probabilistic Boolean decision threes and the complexity of evaluating game trees.
Kazuyuki Tanaka’s work with C.-G. Liu (1)

Liu, C.-G. and Tanaka, K.:

Preliminary versions:

In: *SAC ’07* pp.78–79 (2007).

The eigen-distribution

Def.

"$d$ is the eigen-distribution (for ▲▲▲)"
⇔ $d$ has the property ▲▲▲ and

$$\min_{A_D} \text{cost}(A_D, d) = \max_{\delta} \min_{A_D} \text{cost}(A_D, \delta)$$

Here,

$A_D$ runs over all deterministic alpha-beta pruning algorithms.
$\delta$ runs over all prob. distributions s.t. ▲▲▲.

▲▲▲ is e.g., ID.

They study the eigen-distributions in the two cases:
the ID-case and the CD-case.
The ID case

Theorem 4 (Liu and Tanaka, IPL (2007))

If $d$ is the eigen-distribution for IDs then $d$ is an IID.

Given $k$ (i.e., height of $T_2^k = 2k$), define $\varrho$ as follows. The IID in which prob[the value is 0] = $\varrho$ at every leaf is the eigen-distribution among IID.

Theorem 5 (Liu and Tanaka, IPL (2007)) For $T_2^k$ and IID:

\[
\frac{\sqrt{7} - 1}{3} \leq \varrho \leq \frac{\sqrt{5} - 1}{2}
\]

- $\varrho$ is strictly increasing function of $k$. 
The CD case

Def. (Saks-Wigderson) The reluctant inputs

- When assigning 0 to an $\wedge$, assign 0 to exactly one child.
- When assigning 1 to an $\vee$, assign 1 to exactly one child.

Liu and Tanaka extend the above concept.

RAT (the reverse assignment technique), $i = 0, 1$

- $i$-set is the set of all reluctant inputs s.t. the root has value $i$.
- $E_i$-distribution is the dist. on $i$-set s.t. all the deterministic algorithm have the same complexity.
The CD case (continued)

Theorem 8 (Liu and Tanaka, IPL (2007)) For $T_k^2$:

$E_i$-distribution is the uniform distribution on $i$-set.

Theorem 9 (Liu and Tanaka, IPL (2007)) For $T_k^2$ and CD:

$E_1$-distribution is the unique eigen-distribution.
Given a Boolean function $f$, $\beta(f)$ denotes the distributional complexity (i.e. the max-min-cost achieved by the eigen-distribution) w.r.t. 1-set. $\alpha(f)$ denotes that w.r.t. 0-set.

Trees are not supposed to be binary in this paper.

Given a tree $T$, they study recurrences on $\beta(f_T)$ and $\alpha(f_T)$.

Liu, C.-G. and Tanaka, K.:
The computational complexity of game trees by eigen-distribution. 
Extension (1): CD-case of $T_2^k$ [1/5]

We consider classes of truth assignments (and, algorithms) closed under transpositions. The concept of “a distribution achieving the equilibrium w.r.t. the given classes” is naturally defined.

Extension (1): CD-case of $T_2^k$ [2/5]

of a node / a truth assignment / an algorithm
Extension (1): CD-case of $T_2^k$ [2/5]

of a node / a truth assignment / an algorithm
Definition. Directional Algorithms

An alpha-beta pruning algorithm is said to be *directional* if for some linear ordering of the leaves it never selects for examination a leaf situated to the left of a previously examined leaf.

Suppose $x$, $y$ and $z$ are leaves.

- Allowed: To skip $x$.
- Not allowed:
  - If ($x$ is skipped) \{ scan $y$ before $z$ \} else \{ scan $z$ before $y$\}

Extension (1): CD-case of $T_2^k$ [4/5]

S. and Nakamura (2012)

(a) The Failure of the Uniqueness

In the situation where only directional algorithms are considered, the uniqueness of $d$ achieving the equilibrium fails.

(b) A Counterpart of the Liu-Tanaka Theorem

In the situation where only directional algorithms are considered, A weak version of the Liu-Tanaka theorem holds. (1) is equivalent to (2).

1. $d$ achieves the equilibrium.
2. $d$ is on the 1-set and the cost does not depend on an algorithm.
Extension (1): CD-case of $T_2^k$ [5/5]

A key to the result (b) is the following.

**No-Free-Lunch Theorem (Wolpert and Macready, 1995)**

(Under certain assumptions)
Averaged over all cost functions,
all search algorithms give the same performance.

Wolpert, D.H. and MacReady, W.G.:
No-free-lunch theorems for search,
Extension (2): ID-case of $T^k_2$ [1/4]

Theorem 4 (Liu and Tanaka, 2007)

If $d$ achieves the equilibrium among IDs then $d$ is an IID.

Their proof: “It is not hard.”

Is it (↑) really easy to prove? No. A brutal induction does not work. We show a stronger form of Theorem 4 with clever tricks of induction.

S. and Niida, Y.: Equilibrium points of an AND-OR tree: Under constraints on probability,
Key to the solution

**Lemma 1 (S. and Niida, 2015)**

Consider an IID on an OR-AND tree.

- $x := \text{prob. of a leaf (having the value 0).}$
- $p(x) := \text{prob. of the root (having the value 0).}$
- $c(x) := \text{expected cost of the root.}$

Then, both of the followings are decreasing functions of $x$ ($0 < x < 1$).

\[
\frac{c(x)}{p(x)}, \quad \frac{c'(x)}{p'(x)}
\]
Extension (2): ID-case of $T_2^k$ [3/4]

Lemma 2 (S. and Niida, 2015)

A certain constraint extremum problem has a unique solution.

The proof highlight: By means of Lemma 1, the objective function is decreasing in a certain open interval.

Remark:
At the maximizer, the objective function is NOT differentiable.
Theorem (S. and Niida, 2015)

Fix an \( r \) (\( 0 < r < 1 \)). Let \( r\text{ID} \) denote an ID s.t. prob. of the root (having the value 0) is \( r \).

If \( d \) achieves the equilibrium among \( r\text{IDs} \) then \( d \) is an IID.

As a corollary

Theorem 4 (Liu and Tanaka, 2007)

If \( d \) achieves the equilibrium among IDs then \( d \) is an IID.
Extension (3): ID-case for more general trees

Recently, NingNing Peng, Yue Yang, Keng Meng Ng and Kazuyuki Tanaka extend the results of S. and Niida (2015) to trees not necessarily binary.
Thank you for your attention.

Happy 60th birthday.


Figure 0: $c_{\vee,1}(x)/p_{\vee,1}(x)$ ($0.1 < x < 0.9$)
Figure 1: $c_{\nu,2}(x)/p_{\nu,2}(x)$ \(0 < x < 1\)
Figure 2: $c_{\vee,3}(x)/p_{\vee,3}(x)$ ($0.1 < x < 0.9$)
Figure 3: $c_{6,4}(x)/p_{6,4}(x)$ ($0.1 < x < 0.9$)
Figure 4: $c'_{\vee,1}(x)/p'_{\vee,1}(x) \ (0.1 < x < 0.9)$
Figure 5: $c'_{\vee,2}(x)/p'_{\vee,2}(x)$ ($0 < x < 1$)
Figure 6: $c_{\vee,3}'(x)/p_{\vee,3}'(x)$ ($0.1 < x < 0.9$)
Figure 7: $c'_{\vee,4}(x)/p'_{\vee,4}(x)$ ($0.1 < x < 0.9$)