Huffman Algebras for Independent Random Variables

Cheng-Shang Chang Joy A. Thomas

IBM Research Division
T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

(June, 1992; revision: Jan., 1993)

Abstract

Based on a rearrangement inequality by Hardy, Littlewood and Polya, we define two-operator algebras for independent random variables. These algebras are called Huffman algebras since the Huffman algorithm on these algebras produces an optimal binary tree that minimizes the weighted lengths of leaves. Many examples of such algebras are given. For the case with random weights of the leaves, we prove the optimality of the tree constructed by the power of two rule, i.e., the Huffman algorithm assuming identical weights, when the weights of the leaves are independent and identically distributed.
Keywords: Huffman coding, (max,+)-algebras, stochastic ordering.

1 Introduction

The Huffman coding algorithm was proposed by Huffman[12] to build the optimum source code for a random variable with m symbols with probabilities $p_1, \ldots, p_m$, i.e., to construct a binary tree that minimizes the weighted lengths $\sum p_i l_i$, where p_i is the probability of symbol i and l_i is the depth of symbol i. The algorithm is a greedy tree building algorithm which constructs a tree from the bottom up recursively by merging the two lowest weights. Various generalizations of the algorithm have since appeared [11, 7, 8, 10, 14, 15, 20].

An important application of the Huffman algorithm is to parallel processing or circuit design. Consider the following problem:

Example 1.1 Assume that we are building a circuit out of two input gates of constant delay of 1 clock cycle. Assume that we have signals available at times 1,2,3 and 4 clock cycles respectively. We would like to build a circuit that produces the result at the earliest possible time.

It is not difficult to see in this simple example that the optimal circuit will combine the first two signals (available at $t = 1$ and $t = 2$) to produce a result at $t = 3$ (the maximum of the two input times plus the delay in the gate). This result can be combined with signal 3 to produce at result at time $t = 4$. This result can then be combined with the last signal to produce a result at time $t = 5$. The various stages of the construction are shown in Figure 1.

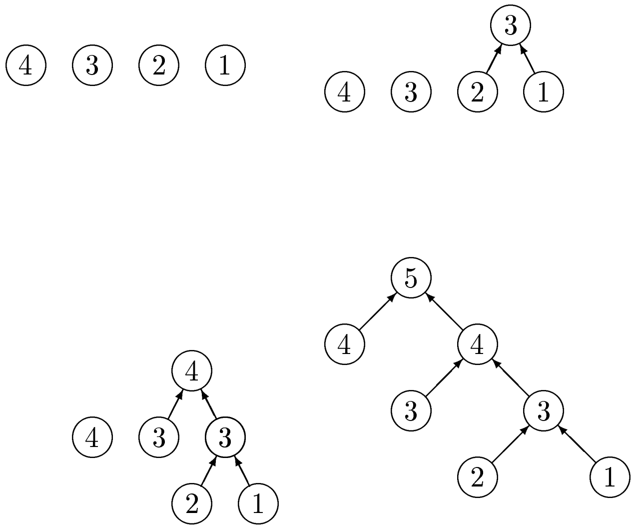

Figure 1: Application of the Huffman algorithm to circuit design

In general, the optimal circuit for n signals will combine the two earliest signals together to form a new signal, and then will use the optimal circuit for $n-1$ signals. This greedy bottom up approach to constructing trees is called the Huffman algorithm.

In the above example, the objective was to construct a binary tree that minimized $\max_i(t_i+l_i)$, where t_i is the time at which the i-th signal was available and the l_i was the number of gates that the i-th signal had to go through, i.e., the length of the path to the root of the tree. A similar objective function is also useful in scheduling tasks in parallel processing, where results of tasks executed in parallel have to be combined two at a time.

The generalized Huffman algorithm is a greedy bottom up tree construction procedure that is used to construct trees which optimize functionals of the form $\sum f(p_i, l_i)$. If we examine the proof of the optimality of the Huffman algorithm for source codes, we see that one of the key steps in the proof is to show that the lowest probabilities are associated with the longest codewords, i.e, the deepest leaves. The proof utilizes the following inequality: Consider two leaves at depths l_1 and l_2, and weights p_1 and p_2, with $p_1 > p_2$, and $l_1 > l_2$ then if we interchange the two leaves, the resulting tree will have a smaller weight. This is implied by the following rearrangement inequality

$$p_1 \cdot l_1 + p_2 \cdot l_2 \geq p_1 \cdot l_2 + p_2 \cdot l_1$$

if $l_1 \geq l_2$ and $p_1 \geq p_2$. This inequality is known as the Hardy-Littlewood-Polya inequality [9] and leads us to define in the next section a two-operator algebra $(S, \oplus, \otimes)$ that satisfies this rearrangement inequality with $+$ and $\cdot$ replaced by $\oplus$ and $\otimes$. These algebras are called arrangement-increasing Huffman algebras (AIHA), extending an earlier definition of Huffman algebras by Knuth[17] of one operator Huffman algebras. We also define arrangement-decreasing Huffman algebras (ADHA) if the Hardy-Littlewood-Polya inequality is reversed. We show that the Huffman algorithm on the AIHA (resp. ADHA) indeed produces an optimal tree that minimizes (resp. maximize) $\sum_{\oplus} p_i \otimes \prod_{\otimes_{l_i}} \gamma$, where γ is a constant not less than the identity element $\mathbf{1}$ for $\otimes$ if p_i's are deterministic. Several examples that satisfy our definition of Huffman algebras are given in Section 4.

In the case of the circuit design or parallel processing example given above, it would be interesting to ask what happens if the weights of the leaves were random? Does the Huffman algorithm construct the optimal tree (in expected value)? A major difference with random weights is that there does not necessarily exist a total ordering among the random variables. Recently, Chang and Yao [6] established the Hardy-Littlewood-Polya inequalities for random variables based on various partial orderings among these variables. Using these inequalities, we show in Section 6 that the power of two rule, which is the Huffman algorithm assuming identical weights, produces the optimal tree for an AIHA if the weights are independent and identically distributed. However, there is no corresponding result for an ADHA. A counterexample for an ADHA is given in Section 6. Finally, we conclude the paper by discussing possible connections between our Huffman algebra and various fundamental limits, including the entropy for source coding and a limit for a parallel processing problem.

2 Huffman algebras

As motivated in the introduction, we define Huffman algebras for *independent* random variables. Unlike the deterministic case, we do not have a total ordering of random variables. Based on comparison of *distributions* of random variables, various partial orderings of random variables have been introduced in the literature, e.g., likelihood ratio ordering, hazard rate ordering, stochastic ordering and increasing convex ordering (see Definition 4.7). Since the Hardy-Littlewood-Polya inequality (see Proposition 4.9) needed for the proof of the optimality of the Huffman algorithm uses three partial orderings on random variables, we define a two-operator Huffman algebra with three partial orderings as follows:

Definition 2.1 *Let S be a set of* independent *random variables (or probability distributions on $\mathcal{R}$) and $<_i$, $i = 1, 2$ and 3 be three partial orderings on S. Let $\oplus$ and $\otimes$ be binary operations defined on S. If $(S, \oplus, \otimes, <_1, <_2, <_3)$ has the following properties, then it is called a* two operator Huffman algebra *on S.*

Equivalence

- **S1** *For two elements a and b in S, $a = b$ means that a and b are identically distributed. If $a, b \in S$ are identically distributed, then $a \leq_i b$ and $b \leq_i a$, $i = 1$ and 2.*

Algebraic properties of the operators

- **S2** *$\oplus$ is commutative and associative, i.e., $a \oplus b = b \oplus a$, $(a \oplus b) \oplus c = a \oplus (b \oplus c)$, $\forall a, b, c \in S$.*
- **S3** *$\otimes$ is associative, i.e., $(a \otimes b) \otimes c = a \otimes (b \otimes c)$, $\forall a, b, c \in S$.*
- **S4** *$\otimes$ is distributive over $\oplus$, i.e., for $a, b, c \in S$,*

$$(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c). \tag{1}$$

- **S5** *There exists an identity element $\mathbf{1}$ for $\otimes$ so that for $a \in S$,*

$$\mathbf{1} \otimes a = a \otimes \mathbf{1} = a. \tag{2}$$

Monotonicity of partial orders

A binary operator, $\odot$, is said to be monotone with respect to the partial orderings $<_i$ and $<_j$ if $a \leq_i b$ and $c \leq_j d$, then $a \odot c \leq_i b \odot d$ for all $a, b, c, d \in S$. We denote this property as $(\odot, <_i, <_j)$-monotone.

S6 $(\oplus, <_1, <_1)$*-monotone.*

S7 $(\otimes, <_i, <_3)$*-monotone for $i = 1, 2$, and 3.*

The Hardy-Littlewood-Polya inequality

S8 *If $a, b, c, d \in S$ are independent and $a \leq_2 b$ and $c \leq_3 d$, then*

$$(a \otimes d) \oplus (b \otimes c) \leq_1 (a \otimes c) \oplus (b \otimes d). \tag{3}$$

We will also call the algebra $(S, \oplus, \otimes, <_1, <_2, <_3)$ defined above as an **arrangement-increasing Huffman algebra** (AIHA). An algebra satisfying all the above properties, but with inequality in the Hardy-Littlewood-Polya (HLP) inequality reversed, will be called an **arrangement-decreasing Huffman algebra** (ADHA). In the examples that we consider later, we will take $<_3$ to be the usual ordering $<$ on real numbers, although we permit it to be more general in the propositions and theorems.

Note that a sufficient condition for [S2-S5] is a ring with an identity element for $\otimes$. Neither the existence of an identity element for $\oplus$ nor the existence of an inverse for every element with respect to $\oplus$ is needed.

In the case of deterministic weights, the definition of the Huffman algebra is much simpler. In this case, there is only one ordering on the weights, which is the usual total ordering on real numbers. The conditions reduce to some algebraic conditions [S2-S5], monotonocity of the operators with respect to the total ordering (i.e., if $a \leq b$ and $c \leq d$, then $a \oplus c \leq b \oplus d$ and $a \otimes c \leq b \otimes d$), and the Hardy-Littlewood-Polya inequality. We will give some examples of these algebras in Section 4.1.

In order to gain some insight to our definition of an AIHA, we compare our algebra with the Huffman algebra introduced by Knuth [17], which is a Huffman algebra $(A, <, \circ)$ based on a *totally* ordered set A and a binary operator $\circ$. Consider the case that the set A in Knuth's algebra and the set S in our algebra are real numbers (degenerate random variables) and that the three partial orderings are the same as the usual ordering $<$ in real numbers. We map Knuth's operator with ours as follows:

$$a \circ b = (a \oplus b) \otimes \gamma, \tag{4}$$

where γ is a constant and $\mathbf{1} \leq \gamma$. It is easy to show that our algebra implies Knuth's algebra. However, Knuth also pointed out that the only continuous and strictly increasing operations that map the real numbers into his algebra are isomorphic to

$$a \circ b = (a + b) \times \gamma, \quad \gamma \geq 1.$$

Thus, we probably cannot find an example consisting of real numbers that is in Knuth's algebra, but not in our algebra. Our algebra is "almost" equivalent to Knuth's algebra in real numbers.

We define Huffman algebras using two operators instead of one because it provides a more intuitive approach to extensions to random variables. It allows us to directly use the Hardy-Littlewood-Polya inequality for partial orderings of random variables. Using the HLP inequality for partial orderings, $<_i, i = 1, 2$ and 3, on random variables, we are able to show, in Section 5, the optimality of the Huffman algorithm when the random variables considered are independent and identically distributed.

Another important issue is that the two-operator algebra allows us to define the notion of tree extensions, which are the analogs of block coding for source codes[5]. This idea leads to the asymptotic achievability of various fundamental limits, includes entropy in source coding and the response time of a parallel processing model.

3 The general Huffman algorithm

In the following, we will describe the optimization problems associated with the AIHA and ADHA and then describe the generalized Huffman algorithm which can be used to solve these optimization problems in special cases. First, we briefly review the definitions of binary trees and the weights of trees.

A **complete binary tree** T is a connected graph without cycles, one of whose vertices (the root) has degree 2, and remaining vertices have degree 1 or 3. The vertices of degree 1 are called the *leaves* of the tree. A complete binary tree is illustrated in Figure 2. Since we will be only considering complete binary trees, we will use a binary tree to mean a complete binary tree. In a binary tree, there is an unique path from the root to any of the other vertices. If vertex a is on the path from the root to vertex b, then we say that a is an ancestor of b and that b is a descendant of a. If a vertex c immediately precedes a vertex d on the path from the root to d, then c is the parent of d and d is the child of c. Vertices with the same parent are called siblings. The **depth** of a node in the tree is the length of the path from the root to the node. Thus the root is at depth 0, its children are at depth 1, etc.

For a complete binary tree, each internal node (all nodes that are not leaves) has two children. We will call one of the children the left child, and the other child the right child. The sub-tree

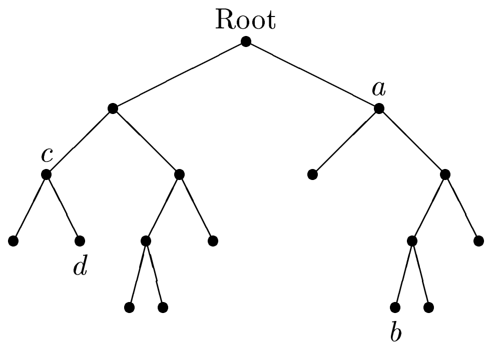

Figure 2: A complete binary tree

consisting of the left child and all its descendants will be called the left sub-tree, and similarly for the right child. Let $\mathcal{B}$ be the set of all binary trees.

For a Huffman algebra $(S, \oplus, \otimes, <_1, <_2, <_3)$, assume that we are given m random variables $P_1, P_2, \ldots, P_m \in S$ as the weights of the leaves of a binary tree. We define the weight of a tree as follows:

(i) a tree with depth 0 has the same weight as its only leaf;

(ii) the weight of a tree with depth larger than 0 is defined by the following recursive equation,

$$w(\mathcal{T}) = (w(\text{Left subtree of root}) \oplus w(\text{Right subtree of root})) \otimes \gamma \tag{5}$$

where $\gamma \in S$ is a degenerate r.v. (the unit cost), i.e., with probability 1, γ is a constant in S.

Using the distributive property of the algebra and applying the definition in (5) recursively, it follows that the weight of the tree is

$$w(\mathcal{T}) = \sum_{\oplus\, i=1}^{m} (P_i \otimes \underbrace{\gamma \otimes \gamma \otimes \cdots \otimes \gamma}_{l_i}) = \sum_{\oplus\, i=1}^{m} (P_i \otimes \prod_{\otimes\, l_i} \gamma), \tag{6}$$

where l_i is the length of the path from the root to leaf i. We note that assuming γ to be a degenerate r.v. is crucial in preserving independence of $P_i \otimes \prod_{l_i} \gamma$, $i = 1, 2, \ldots, m$ in (6).

Now we consider a minimization problem on an AIHA, $(S, \oplus, \otimes, <_i, i = 1, 2, 3)$. The objective is to find the tree (among all the complete binary trees) that has the minimum weight under the partial ordering $<_1$, i.e.,

$$\min_{\mathcal{T} \in \mathcal{B}} w(\mathcal{T}) = \min_{\mathcal{T} \in \mathcal{B}} \sum_{\oplus\, i=1}^{m} (P_i \otimes \prod_{\otimes\, l_i} \gamma), \tag{7}$$

where $\mathcal{T}$ is a binary tree with m leaves with random weights $P_1, \ldots, P_m$ on the leaves. Similarly, one can consider a maximization problem on an ADHA. In general, these problems are difficult to solve. However, we will show in Theorems 5.1 and 6.1 that there are still some cases which can be solved by the Huffman algorithm. The Huffman algorithm is a greedy bottom up tree construction procedure which recursively merges the lowest two weights. For the general Huffman algebra with partial orders, the algorithm is given below:

The Huffman algorithm:

Step 1. `Arrange the weights in decreasing order under the partial ordering` $<_1$.

Step 2. `Form a subtree by combining the last two weights` P_{m-1} `and` P_m `to a single node of weight` $w'_{m-1} = (P_{m-1} \otimes \gamma) \oplus (P_m \otimes \gamma)$.

Step 3. `Solve the problem for` $m-1$ `leaves with the weights` $P_1, P_2, \ldots, P_{m-2}, w'_{m-1}$.

The same algorithm can be used to maximize the cost for an ADHA. An example of the use of this algorithm was given in the example in the introduction.

We will show in Section 5 that this algorithm produces the optimal tree that minimizes the cost for an AIHA (or maximizes the cost for an ADHA) under some additional conditions - essentially that the algebra be closed under the merge operation of the algorithm. But before we come to the proof of the optimality of the algorithm, we will give some examples of Huffman algebras, both for real numbers and for independent random variables.

4 Examples of Huffman algebras

In this section, we discuss various examples of Huffman algebras. We begin with a discussion of Huffman algebras for real numbers. For real numbers, we only need to verify that the two operators $\oplus$ and $\otimes$ are monotone and satisfy the HLP inequality. In these algebras, we can also identify fundamental limits to the cost of the optimal tree which are analogous to the entropy function for source coding.

4.1 Huffman algebras for real numbers

Example 4.1 $(\mathcal{R}^+, +, \cdot, <, <, <)$ is an AIHA. In this case, the optimization problem becomes

$$\min_{\mathcal{T}} \sum_{i=1}^{m} p_i \gamma^{l_i}. \tag{8}$$

When we set $\gamma = 1+\epsilon$, the first order term in the cost of the tree is $\sum p_i l_i$, which is the quantity minimized by the standard Huffman algorithm for source coding.

Just as the entropy of a distribution is the fundamental limit for the expected length of an prefix-free source code for the distribution, we can also identify a fundamental limit for the minimization in (8). It is shown in Campbell[3] and Parker[19] that

$$I(p) \leq \log_\gamma \left(\sum p_i \gamma^{l_i} \right) < I(p) + 1. \tag{9}$$

where

$$I(p) = \frac{1}{1-\alpha} \log_2 \left(\sum_{i=1}^{m} p_i^\alpha \right) \tag{10}$$

and $\alpha = \frac{1}{\log_2 \gamma + 1}$. The quantity on the right hand side of the equation is the Renyi-entropy of order α scaled by $\log_2 \gamma$ [21]. In the case that $\gamma \to 1$, these limits reduce to

$$-\sum_i p_i \log p_i \leq \sum_i p_i l_i < -\sum_i p_i \log p_i + 1 \tag{11}$$

which is one of the fundamental results of information theory.

Example 4.2 $(\mathcal{R}, \max, +, <, <, <)$ is an AIHA. The optimization problem in this case, is $\min_\mathcal{T} \max_i (p_i + \gamma l_i)$. This optimization is useful is designing circuits with binary gates that produce the result at the earliest possible time (see the example in the introduction). It is also useful in model of parallel processing where parallel subtasks produce results which have to be combined two at a time.

In this case too, we can identify the equivalent of the entropy function. It is shown in Golumbic[8] that

$$J(p) \leq \max_i (p_i + l_i) < J(p) + 1, \tag{12}$$

where

$$J(p) = \log_2 \left(\sum_{i=1}^{m} 2^{p_i} \right). \tag{13}$$

This result can also be derived by taking taking $p_i' = 2^{p_i \log \gamma}$ and letting $\gamma \to \infty$ in (9) to obtain the bounds of (12).

Example 4.3 $(\mathcal{R}^+, \max, \cdot, <, <, <)$ is an AIHA. The optimization problem in this case, is $\min_\mathcal{T} \max_i (p_i \cdot \gamma^{l_i})$. This can be mapped to the Example 4.2 by taking logarithms.

Example 4.4 $(\mathcal{R}, \max, \min, <, <, <)$ is an AIHA. The optimization problem in this case is to find the tree that minimizes

$$\max_i(\min(p_i, \underbrace{\gamma, \gamma, \ldots, \gamma}_{l_i})) = \max_i(\min(p_i, \gamma)),$$

which is a trivial problem since the value of the objective function is the same for all trees.

Example 4.5 $(\mathcal{R}, \min, +, <, <, <)$ is an ADHA. The optimization problem in this case is a maximization problem, namely $\max_{\mathcal{T}} \min_i(p_i + \gamma l_i)$. This is the dual of the problem in Example 4.2. In this case, we can show that

$$\max_{\mathcal{T}} \min_i(p_i + l_i) \leq J(p), \tag{14}$$

where $J(p)$ is as defined in (13).

Example 4.6 $(\mathcal{R}^+, \min, \cdot, <, <, <)$ is an ADHA. The optimization problem in this case is a maximization problem, namely $\max_{\mathcal{T}} \min_i(p_i \cdot \gamma^{l_i})$. Again we can map this to the previous example by taking logarithms.

4.2 Huffman algebras for random variables

In order to extend these Huffman algebras to nondegenerate random variables, we have to introduce the definitions for several commonly used partial orderings for random variables. Refer to Stoyan [24], Ross ([22], Chapter 8) and Barlow and Proschan [1].

Definition 4.7 *Suppose X and Y are two independent random variables with density functions $f(\cdot)$ and $g(\cdot)$, and distribution functions $F(\cdot)$ and $G(\cdot)$.*

(i) *Nonoverlapping ordering: $X \leq Y$ if their distributions are nonoverlapping, that is, $\sup\{x : f(x) > 0\} \leq \inf\{x : g(x) > 0\}$.*

(ii) *Likelihood ratio ordering: $X \leq_{lr} Y$ if their likelihood ratios are ordered, that is, for any given $a > 0$, $f(x)/f(x+a) \geq g(x)/g(x+a)$ for all x.*

(iii) *Hazard ratio ordering: $X \leq_{hr} Y$ if their hazard rates are ordered, that is, $f(x)/[1-F(x)] \geq g(x)/[1-G(x)]$ for all x.*

(iv) *Stochastic ordering: $X \leq_{st} Y$ if $Ef(X) \leq Ef(Y)$ for all $f : \mathcal{R} \mapsto \mathcal{R}$ that is increasing.*

(v) *Increasing convex (resp. concave) ordering:* $X \le_{ic} Y$ *(resp.* $X \le_{iv} Y$*) if* $Ef(X) \le Ef(Y)$ *for all* $f : \mathcal{R} \mapsto \mathcal{R}$ *that is increasing and convex (resp. concave).*

(vi) *mean ordering:* $X \le_{mean} Y$ *if* $E(X) \le E(Y)$.

It is known that $\le \Rightarrow \le_{lr} \Rightarrow \le_{hr} \Rightarrow \le_{st} \Rightarrow \le_{ic} \Rightarrow \le_{mean}$. Throughout the paper, the notation $\le_i = \le_a$ means replacing the partial ordering $\le_i$ by the partial ordering $\le_a$ in the appropriate context.

We will use the following closure properties (see [22], Example 8.2.a, Proposition 8.5.4) for both stochastic and increasing convex (resp. concave) orderings in the proof of Examples 4.11-4.17.

Proposition 4.8 *Suppose that* X_1 *is independent of* X_2 *and that* Y_1 *is independent of* Y_2. *Then* $f(X_1, X_2) \le_i f(Y_1, Y_2)$ *if* $X_1 \le_i Y_1$ *and* $X_2 \le_i Y_2$ *for (i)* $\le_i = \le_{st}$ *and* f *increasing (ii)* $\le_i = \le_{ic}$ *and* f *increasing and convex in each variable (iii)* $\le_i = \le_{iv}$ *and* f *increasing and concave in each variable.*

Based on these partial orderings, Chang and Yao [6], Theorem 3.2 derived stochastic versions of HLP inequalities which cover the following cases (see also [2, 22, 25, 23, 4] for related results).

Proposition 4.9 *(HLP inequalities for r.v.'s) Let* X_1, X_2, Y_1 *and* Y_2 *be four independent r.v.'s. Suppose* $X_1 \le_2 X_2$ *and* $Y_1 \le_3 Y_2$. *Then,*

- *given* $X_1, X_2, Y_1, Y_2 \ge 0$, $X_1 \cdot Y_2 + X_2 \cdot Y_1 \le_1 X_1 \cdot Y_1 + X_2 \cdot Y_2$, *if (i)* $\le_2 = \le_{lr}, \le_3 = \le_{lr}, \le_1 = \le_{st}$ *(ii)* $\le_2 = \le_{hr}, \le_3 = \le_{lr}, \le_1 = \le_{ic}$ *and (iii)* $\le_2 = \le_{mean}, \le_3 = \le_{mean}, \le_1 = \le_{mean}$.
- $\max(X_1 + Y_2, X_2 + Y_1) \le_1 \max(X_1 + Y_1, X_2 + Y_2)$, *if (i)* $\le_2 = \le_{lr}, \le_3 = \le_{lr}, \le_1 = \le_{st}$ *and (ii)* $\le_2 = \le_{st}, \le_3 = \le_{st}, \le_1 = \le_{ic}$. *Similar results hold for* $\max(X_1 \cdot Y_2, X_2 \cdot Y_1) \le_1 \max(X_1 \cdot Y_1, X_2 \cdot Y_2)$, *given* $X_1, X_2, Y_1, Y_2 \ge 0$.
- $\min(X_1 + Y_2, X_2 + Y_1) \ge_1 \min(X_1 + Y_1, X_2 + Y_2)$, *if (i)* $\le_2 = \le_{lr}, \le_3 = \le_{lr}, \le_1 = \le_{st}$ *and (ii)* $\le_2 = \le_{st}, \le_3 = \le_{st}, \le_1 = \le_{iv}$. *Similar results hold for* $\min(X_1 \cdot Y_2, X_2 \cdot Y_1) \ge_1 \min(X_1 \cdot Y_1, X_2 \cdot Y_2)$, *given* $X_1, X_2, Y_1, Y_2 \ge 0$.

We consider the following sets of random variables.

- $S_{IFR} \stackrel{\text{def}}{=} \{$independent r.v.'s with Increasing Failure Rate (IFR) distributions$\}$, i.e., $g(x)/(1-G(x))$ is increasing in x.
- $S_{logIFR} \stackrel{\text{def}}{=} \{$independent r.v.'s with $\log(X)$ being IFR$\}$.

- $S_{ILR} \stackrel{\text{def}}{=}$ {independent r.v.'s with Increasing Likelihood Ratio (ILR) distributions}, i.e., $g(x)/g(x+a)$ is increasing in x for all $a \geq 0$.

- $S_{logILR} \stackrel{\text{def}}{=}$ {independent r.v.'s with $\log(X)$ being ILR}.

- In addition, $S^+ \stackrel{\text{def}}{=} S \cap$ {nonnegative r.v.'s}.

We note that the ILR property is also known as the PF2 property in the literature [1]. Also, ILR $\Rightarrow$ IFR $\Rightarrow$ logIFR and logILR $\Rightarrow$ logIFR. Examples of ILR r.v.'s are normal and Erlang. Examples of logILR r.v.'s are lognormal and Weibull. For more examples of these distributions, we refer to [1, 16]. The following proposition regarding the property of these sets of r.v.'s is shown in Keilson and Sumita [16], pp. 189.

Proposition 4.10 *(i) A r.v. X is IFR $\Leftrightarrow$ $X \leq_{hr} X + a$ for all $a \geq 0$ (ii) X is ILR $\Leftrightarrow$ $X \leq_{lr} X + a$ for all $a \geq 0$ (iii) X is logIFR $\Leftrightarrow$ $X \leq_{hr} aX$ for all $a \geq 1$ (iv) X is logILR $\Leftrightarrow$ $X \leq_{lr} aX$ for all $a \geq 1$.*

Example 4.11 The algebra $(S^+, +, \cdot, <_{mean}, <_{mean}, <)$ is an AIHA. Using the facts that $EX_1X_2 = EX_1EX_2$ and $E(X_1 + X_2) = EX_1 + EX_2$ for any two independent random variables, the conditions in Definition 2.1 can be verified easily (cf. Example 4.1). In this example, the mean ordering is a total order on S and the objective is $\min_{\mathcal{T}} E(\sum_{i=1}^m P_i \gamma^{l_i})$, which is equivalent to the deterministic case $\min_{\mathcal{T}} \sum_{i=1}^m EP_i \gamma^{l_i}$.

Example 4.12 The algebras $(S^+_{logIFR}, +, \cdot, <_{ic}, <_{hr}, <)$ is an AIHA. We need to verify (S6-8). First, note that the , $(\cdot, <, <)$-monotone property in (S7) are trivially satisfied. That the algebra satisfies the $(+, <_{ic}, <_{ic})$-monotone property in (S6) and the $(\cdot, <_{ic}, <)$-monotone property in (S7) follows from Proposition 4.8. Since the distributions are logIFR, we have from Proposition 4.10 that $aX_1 \leq_{hr} bX_1 \leq_{hr} bX_2$ if $0 \leq a \leq b$ and $0 \leq X_1 \leq_{hr} X_2$. Thus, the algebra satisfies $(\cdot, <_{hr}, <)$-monotone property in (S7). The HLP inequality in (S8), i.e., $X_1 \cdot a + X_2 \cdot b \leq_{ic} X_1 \cdot b + X_2 \cdot a$, given $a \geq b$ and $X_1 \leq_{hr} X_2$, follows from Proposition 4.9. The objective is $\min_{\mathcal{T}} (\sum_{i=1}^m P_i \gamma^{l_i})$ in the sense of increasing convex ordering.

The proofs of the following examples are similar to that in Example 4.12 using Propositions 4.8-4.10.

Example 4.13 The algebra $(S^+_{logILR}, +, \cdot, <_{st}, <_{lr}, <)$ is an AIHA. The objective is $\min_{\mathcal{T}} (\sum_{i=1}^m P_i \gamma^{l_i})$ in the sense of stochastic ordering.

Example 4.14 The algebra $(S, \max, +, <_{ic}, <_{st}, <)$ is an AIHA. The objective is $\min_{\mathcal{T}}(\max_i(P_i + \gamma l_i))$ in the sense of increasing convex ordering.

Example 4.15 The algebra $(S_{ILR}, \max, +, <_{st}, <_{lr}, <)$ is an AIHA. The objective is $\min_{\mathcal{T}}(\max_i(P_i + \gamma l_i))$ in the sense of stochastic ordering.

Similarly, one can show that $(S^+_{logILR}, \max, \cdot, <_{st}, <_{lr}, <)$ and $(S^+, \max, \cdot, <_{ic}, <_{st}, <)$ are AIHA. In the following, we provide some examples of ADHA's.

Example 4.16 The algebra $(S, \min, +, <_{iv}, <_{st}, <)$ is an ADHA. The objective is $\max_{\mathcal{T}}(\min_i(P_i + \gamma l_i))$ in the sense of increasing concave ordering.

Example 4.17 The algebra $(S_{ILR}, \min, +, <_{st}, <_{lr}, <)$ is an ADHA. The objective is $\max_{\mathcal{T}}(\min_i(P_i + \gamma l_i))$ in the sense of stochastic ordering.

Similarly, one can show that $(S^+, \min, \cdot, <_{iv}, <_{st}, <)$ and $(S^+_{logILR}, \min, \cdot, <_{st}, <_{lr}, <)$ are ADHA's.

We summarize these examples in Table 1.

Table 1. Examples of Huffman algebras

Algebra	type	objective	Ref.
$(\mathcal{R}^+, +, \cdot, <, <, <)$	AIHA	$\min \sum_{i=1}^m P_i \gamma^{l_i}$ under $<$	Ex. 4.1
$(\mathcal{R}, \max, +, <, <, <)$	AIHA	$\min \max_i (P_i + \gamma l_i)$ under $<$	Ex. 4.2
$(\mathcal{R}^+, \max, \cdot, <, <, <)$	AIHA	$\min \max_i (P_i \cdot \gamma^{l_i})$ under $<$	Ex. 4.3
$(\mathcal{R}, \max, \min, <, <, <)$	AIHA	$\min \max_i (\min(P_i, \gamma))$ under $<$	Ex. 4.4
$(\mathcal{R}, \min, +, <, <, <)$	ADHA	$\max \min_i (P_i + \gamma l_i)$ under $<$	Ex. 4.5
$(\mathcal{R}^+, \min, \cdot, <, <, <)$	ADHA	$\max \min_i (P_i \cdot \gamma^{l_i})$ under $<$	Ex. 4.6
$(S^+, +, \cdot, <_{mean}, <_{mean}, <)$	AIHA	$\min \sum_{i=1}^m P_i \gamma^{l_i}$ under $<_{mean}$	Ex. 4.11
$(S^+_{logIFR}, +, \cdot, <_{ic}, <_{hr}, <)$	AIHA	$\min \sum_{i=1}^m P_i \gamma^{l_i}$ under $<_{ic}$	Ex. 4.12
$(S^+_{logILR}, +, \cdot, <_{st}, <_{lr}, <)$	AIHA	$\min \sum_{i=1}^m P_i \gamma^{l_i}$ under $<_{st}$	Ex. 4.13
$(S, \max, +, <_{ic}, <_{st}, <)$	AIHA	$\min \max_i (P_i + \gamma l_i)$ under $<_{ic}$	Ex. 4.14
$(S_{ILR}, \max, +, <_{st}, <_{lr}, <)$	AIHA	$\min \max_i (P_i + \gamma l_i)$ under $<_{st}$	Ex. 4.15
$(S^+, \max, \cdot, <_{ic}, <_{st}, <)$	AIHA	$\min \max_i (P_i \cdot \gamma^{l_i})$ under $<_{ic}$	Ex. 4.15
$(S^+_{logILR}, \max, \cdot, <_{st}, <_{lr}, <)$	AIHA	$\min \max_i (P_i \cdot \gamma^{l_i})$ under $<_{st}$	Ex. 4.15
$(S, \min, +, <_{iv}, <_{st}, <)$	ADHA	$\max \min_i (P_i + \gamma l_i)$ under $<_{iv}$	Ex. 4.16
$(S_{ILR}, \min, +, <_{st}, <_{lr}, <)$	ADHA	$\max \min_i (P_i + \gamma l_i)$ under $<_{st}$	Ex. 4.17
$(S^+, \min, \cdot, <_{iv}, <_{st}, <)$	ADHA	$\max \min_i (P_i \cdot \gamma^{l_i})$ under $<_{iv}$	Ex. 4.17
$(S^+_{logILR}, \min, \cdot, <_{st}, <_{lr}, <)$	ADHA	$\max \min_i (P_i \cdot \gamma^{l_i})$ under $<_{st}$	Ex. 4.17

5 Optimality of the Huffman algorithm

We now prove the optimality of the Huffman algorithm for finding the tree that solves the optimization problems described earlier. It is clear that we need a *total* order among P_i's in order to arrange them in decreasing order in Step 1 of the algorithm. Similarly, we also require w'_{m-1} to be compatible with P_i's under $<_1$ in Step 3. This implies that the partial ordering $<_1$ should be closed under the merging operation. To use the HLP inequality, we also need $<_1 \Rightarrow <_2$. These conditions are summarized in Theorem 5.1.

Theorem 5.1 *Let $\tilde{S}$ be a subset of an AIHA (resp. ADHA) $(S, \oplus, \otimes, <_i, i = 1, 2, 3)$. If*

(i) $P_i \in \tilde{S}, i = 1, 2, \dots m$, *and the ordering $<_1$ is a total order in $\tilde{S}$,*

(ii) $\tilde{S}$ *is closed under $\oplus$ and $\otimes$, i.e., $a \oplus b \in \tilde{S}$ if $a, b \in \tilde{S}$ and $a \otimes \gamma \in \tilde{S}$ if $a \in \tilde{S}$ and γ is a degenerate r.v.,*

(iii) $<_1 \Rightarrow <_2$ *in $\tilde{S}$, i.e., $a <_1 b \Rightarrow a <_2 b$ for all $a, b \in \tilde{S}$,*

(iv) $\mathbf{1} \leq_3 \gamma$, *where $\mathbf{1}$ is the identity element for $\otimes$,*

then the weight of the tree constructed by the Huffman algorithm minimizes (resp. maximizes) $w(\mathcal{T})$ in the sense of the partial ordering $<_1$.

Note: There may be more than one tree that has the minimal value of $w(\mathcal{T})$. The Huffman algorithm will produce one of them. Also, if the algebra has an element $\mathbf{0}$ such that $a \otimes \mathbf{0} = \mathbf{0} \leq_1 \mathbf{1}$, then the condition $<_1 \Rightarrow <_2$ implies $<_1 \Leftrightarrow <_2$ from the HLP inequality.

Proof. Note that (i) the total order in $\tilde{S}$ and (ii) the closure property guarantees that all the trees are compatible under $<_1$. We will prove this theorem by induction. Assume that the cost of a binary tree with m leaves is not less than the cost of the tree constructed by the Huffman algorithm in the sense of $<_1$ ordering as our induction hypothesis. Since there is only one complete binary tree with two leaves, the hypothesis holds for $m = 2$. Suppose this hypothesis hold for $m = k$. Consider a tree with $k + 1$ leaves. Without loss of generality, we assume that $P_1 \geq_1 P_2 \geq_1 \dots \geq_1 P_{k+1}$. Let l' be the largest depth of any leaf in the tree. If the two leaves, P_{i_1} and P_{i_2} ($P_{i_1} \leq_1 P_{i_2}$), with depth l' are not the two lowest weights, then there is P_{i_3} such that $P_{i_3} \leq_1 P_{i_2}$ and P_{i_3} has depth $l'' \leq l'$. We will show that interchanging P_{i_2} and P_{i_3} does not increase the cost. From the $(\otimes, <_3, <_3)$-monotone property and (iv), we have $\prod_{\otimes_{l''}} \gamma \leq_3 \prod_{\otimes_{l'}} \gamma$. Since $<_1 \Rightarrow <_2$ in $\tilde{S}$, we have from the HLP inequality that

$$\Big(P_{i_2} \otimes \prod_{\otimes_{l''}} \gamma\Big) \oplus \Big(P_{i_3} \otimes \prod_{\otimes_{l'}} \gamma\Big) \leq_1 \Big(P_{i_2} \otimes \prod_{\otimes_{l'}} \gamma\Big) \oplus \Big(P_{i_3} \otimes \prod_{\otimes_{l''}} \gamma\Big). \tag{15}$$

Applying both the $(\oplus, <_1, <_1)$-monotone property and the $(\otimes, <_1, <_3)$-monotone property to (6), we conclude that interchanging P_{i_2} and P_{i_3} does not increase the cost. Thus, for a binary tree $\mathcal{T}_0$ with $k+1$ leaves, we can construct a tree $\mathcal{T}_1$ with $k+1$ leaves such that the two leaves with the longest depth are the two lowest weights P_k and P_{k+1}, and that the cost of $\mathcal{T}_1$ is not larger than that of $\mathcal{T}_0$. From the cost reduction rule, we note that the cost of the tree $\mathcal{T}_1$ with $k+1$ leaves is the same as the cost of the new binary tree with k leaves, formed by combining P_k and P_{k+1} into a leaf of weight $(P_k \otimes \gamma) \oplus (P_{k+1} \otimes \gamma)$. Since this is also the first step of the Huffman algorithm, it then follows from the induction hypothesis for $m = k$ that the cost of $\mathcal{T}_1$ is not less than the cost of the tree constructed by the Huffman algorithm. This completes the argument for $m = k+1$. The proof for an ADHA is similar. $\Box$

Example 5.2 [Huffman algebra on real numbers] Since there is only one total order $<$ in Examples 4.1-4.6, it is easy to verify that the conditions in Theorem 5.1 are satisfied and the tree constructed by the Huffman algorithm is the optimal one. Some of these examples are reported in [20, 10, 13].

Example 5.3 Consider the AIHA $(S, +, \cdot, <_{mean}, <_{mean}, <)$ in Example 4.11. Since the ordering $<_{mean}$ is a total order among all random variables, the conditions in Theorem 5.1 are satisfied. Thus, the Huffman algorithm by merging two smallest random variables in the mean ordering minimizes $E\sum_{i=1}^{m} P_i\gamma^{l_i}$ for $\gamma \geq 1$.

Example 5.4 Let $\tilde{S}$ be the set of exponential r.v.'s. Then $\tilde{S}$ is a subset of the ADHA $(S^+, min, \cdot, <_{st}, <_{lr}, <)$. From the fact that the minimum of two exponential r.v. is still an exponential r.v. and the fact that the constant multiple of an exponential r.v. is also an exponential r.v., the closure property is satisfied. Moreover, the family of exponential r.v. is totally ordered by the mean ordering which implies both the stochastic ordering and the likelihood ratio ordering. Thus, conditions (i) and (iii) are also satisfied. We then conclude that the Huffman algorithm by merging two smallest element in the stochastic ordering maximizes $\min(P_i \cdot \gamma^{l_i})$ in the stochastic ordering for P_i being exponentially distributed and $\gamma \geq 1$.

6 The Huffman algorithm for i.i.d. random variables and the "power of two" rule

In most of the Huffman algebras, it is difficult to establish a total ordering. We therefore consider a special case, the case when the random variables are independent and identically distributed. In this case, intuition tells us that the optimal tree is the "most balanced" tree.

The most balanced tree is one where all the leaves are as close to the same depth as possible, i.e., all the leaves have depth $\lfloor \log m \rfloor$ or $\lfloor \log m \rfloor + 1$, where m is the number of leaves.

In the following, we will show that the tree constructed by the "power of 2" rule (the "most balanced" tree) minimizes the weight if the weights of leaves are independent and identically distributed (i.i.d) random variables for an AIHA. However, for an ADHA, the same intuition does not hold, and we will give a counterexample to show that the "most balanced" tree is not optimal for an arrangement decreasing Huffman algebra.

The "most balanced" tree can be constructed by a top down procedure called the "power of 2" rule (Glassey and Karp[7]). If the weights of leaves are equal, the cost of an optimum tree is parameterized by the number of leaves. The choice of constructing the left subtree and the right subtree is then reduced to the choice of the numbers of leaves in the left subtree and the right subtree. The "power of 2" rule proceeds as follows: Choose the number t such that t is a power of 2 and $m/3 \leq t < 2m/3$ (It is easy to see that t is unique). Assign the left subtree t leaves and the right subtree $m - t$ leaves. Construct the subtrees recursively using the same rule. The tree constructed using the power of 2 rule is the same as the tree constructed using the Huffman algorithm by assuming every leaf has *identical* weight. The following theorem proves the optimality of such trees.

Theorem 6.1 *If (i) P_i, $i = 1, 2, \ldots, m$, are i.i.d. weights of leaves on an AIHA $(S, \oplus, \otimes, <_i, i = 1, 2, 3)$ and (ii) $\mathbf{1} \leq_3 \gamma$, where $\mathbf{1}$ is the identity element for $\otimes$, then the power of 2 rule minimizes $w(\mathcal{T})$ in the sense of the partial ordering $<_1$.*

The intuition behind Theorem 6.1 is as follows: Initially, all the weights are equivalent and one can choose any arbitrary two weights to merge. After merging these two weights, the new weight is "larger" than the others due to the monotonic property of an AIHA. Thus, this new weight would not be merged until all the other weights are merged. Repeating the process generates the most balanced tree which is the tree constructed by the power of 2 rule.

Applying Theorem 6.1 to the AIHA in Examples 4.11-4.13, we derive the following corollary.

Corollary 6.2 *If $P_i, i = 1, 2, \ldots, m$ are nonnegative i.i.d. r.v.'s, then the power of 2 rule minimizes $\sum_{i=1}^{m} P_i \gamma^{l_i}$ $(\gamma \geq 1)$ (i) in the mean ordering (ii) in the increasing convex ordering if $\log(P_i)$'s have increasing failure rate distributions and (iii) in the stochastic ordering if $\log(P_i)$'s have increasing likelihood ratio distributions.*

Similarly, applying Theorem 6.1 to the AIHA in Examples 4.14 and 4.15, we derive the following corollaries.

Corollary 6.3 *If $P_i, i = 1, 2, \ldots, m$ are i.i.d. r.v., then the power of 2 rule minimizes $max(P_i + \gamma l_i)$ ($\gamma \geq 0$) (i) in the increasing convex ordering (ii) in the stochastic ordering if P_i's have increasing likelihood ratio distributions.*

Corollary 6.4 *If $P_i, i = 1, 2, \ldots, m$ are* nonnegative *i.i.d. r.v., then the power of 2 rule minimizes $max(P_i \gamma^{l_i})$ ($\gamma \geq 1$) (i) in the increasing convex ordering (ii) in the stochastic ordering if $\log(P_i)$'s have increasing likelihood ratio distributions.*

Note that the intuitive argument used for explaining Theorem 6.1 cannot be applied to an ADHA since the new weight (after merging two weights) might not be "larger" than the others in an ADHA. The following is a counterexample showing that there is no corresponding result for an ADHA.

Example 6.5 Consider the maximization problem $\max_{\mathcal{T}}(E \min_i(P_i + l_i))$ with P_i, i=1,2,3 and 4, being i.i.d. exponential r.v.'s with mean $1/\lambda$. Note that there are only two complete binary trees with four leaves (see Figure 3). One is $l_i = 2$, $i = 1, 2, 3, 4$. The other is $l_1 = 1$, $l_2 = 2$, $l_3 = 3$ and $l_4 = 3$. Clearly, the first tree is constructed by the power of 2. However, it can be shown that for $\lambda < 1/81$ (see the full report [5]),

$$E \min(P_1 + 2, P_2 + 2, P_3 + 2, P_4 + 2) < E \min(P_1 + 1, P_2 + 2, P_3 + 3, P_4 + 3). \tag{16}$$

The intuitive reason for this property is that for exponentials with low rate (high mean value), the quantities on both sides of (16) are essentially determined by $\min\{P_1, P_2, P_3, P_4\}$. All four random values are equally likely to be the minimum value, and on the right hand side of (16), about half the time, this value occurs at depth 3 and one-fourth of the time at depth 1. Since the minimum value is likely to occur at a larger depth than 2, the expected value of the left hand side is larger than that of the right hand side in (16). This provides a counterexample for an ADHA.

Now we prove Theorem 6.1 using the notion of majorization [18]. A m-vector $\underline{l}^1$ is **majorized** by $\underline{l}^2$ ($\underline{l}^1 \prec \underline{l}^2$) if (i) $\sum_{i=1}^{j} l^1_{[i]} \leq \sum_{i=1}^{j} l^2_{[i]}$, $j = 1, \ldots, m-1$ and (ii) $\sum_{i=1}^{m} l^1_{[i]} = \sum_{i=1}^{m} l^2_{[i]}$, where $l^1_{[i]}$ and $l^2_{[i]}$ are the i-th largest components in $\underline{l}^1$ and $\underline{l}^2$. If $\sum_{i=1}^{j} l^1_{[i]} \leq \sum_{i=1}^{j} l^2_{[i]}$, $j = 1, \ldots, m$, we say that $\underline{l}^1$ is **weakly submajorized** by $\underline{l}^2$ ($\underline{l}^1 \prec_w \underline{l}^2$). A function $f : \mathcal{R}^m \to \mathcal{R}$ is (resp. increasing) **Schur convex** if $f(\underline{l}^1) \leq f(\underline{l}^2)$ for all $\underline{l}^1 \prec \underline{l}^2$. ($resp. \underline{l}^1 \prec_w \underline{l}^2$).

Since P_i, $i = 1, 2, \ldots, m$ are i.i.d. and

$$w(\mathcal{T}) = \sum_{\oplus\, i=1}^{m} (P_i \otimes \prod_{\otimes\, l_i} \gamma), \tag{17}$$

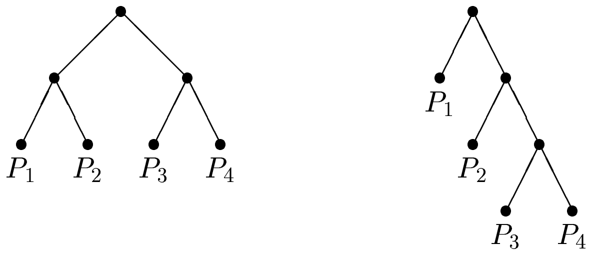

Figure 3: Binary trees with 4 leaves

the distribution of $w(\mathcal{T})$ is characterized by the length vector $\underline{l} = (l_1, l_2, \ldots, l_n)$. Now consider a family of r.v.'s $w(\underline{l})$ whose distributions are characterized by (17).

Lemma 6.6 *Under the assumptions in Theorem 6.1, the family of random variables $\{w(\underline{l})\}$ is increasing and integer Schur convex in the vector $\underline{l}$ in the sense of the partial ordering $<_1$. That is, given two vectors $\underline{l}^1 \prec_w \underline{l}^2$, $w(\underline{l}^1) \leq_1 w(\underline{l}^2)$.*

Proof. Since $\underline{l}^1 \prec_w \underline{l}^2$, there exists a vector $\underline{l}^3$ such that $\underline{l}^1 \prec \underline{l}^3$ and $\underline{l}^3 \leq \underline{l}^2$. To show $w(\underline{l}^1) \leq_1 w(\underline{l}^3)$, we apply a lemma by Muirhead (e.g., see [18], Lemma 5.D.1). If two integer vectors $\underline{l}^1 \prec \underline{l}^3$ for two integer vectors $\underline{l}^1$ and $\underline{l}^3$, then l^1 can be derived from l^3 by successive applications of a finite number of "transfers". A transfer of an integer-valued vector $\underline{l}$ and $l_i > l_j$ is another integer-valued vector $\underline{l}'$ with $l'_i = l_i - 1$, $l'_j = l_j + 1$ and $l'_k = l_k$ for $k \neq i, j$. Clearly, $\underline{l}' \prec \underline{l}$. Thus, it suffices to show the inequality between the original vector and the transferred vector, i.e., $w(\underline{l}') \leq_1 w(\underline{l})$. It follows from the associative property of $\otimes$ that

$$\begin{aligned}
&(P_i \otimes \prod_{\otimes_{l_i - 1}} \gamma) \oplus (P_j \otimes \prod_{\otimes_{l_j + 1}} \gamma) \\
&= ((P_i \otimes \prod_{\otimes_{l_i - 2}} \gamma) \otimes \gamma) \oplus ((P_j \otimes \prod_{\otimes_{l_j - 1}} \gamma) \otimes (\gamma \otimes \gamma)).
\end{aligned}$$

Since $l_i > l_j$ and $\mathbf{1} \leq_3 \gamma$, we have from the $(\otimes, <_3, <_3)$-monotone property in (S7) that $\gamma \leq_3 \gamma \otimes \gamma$ and that $\prod_{\otimes_{l_j - 1}} \gamma \leq_3 \prod_{\otimes_{l_i - 2}} \gamma$. Since P_i and P_j are i.i.d., it follows from the equivalence property that $P_i \leq_2 P_j$. Thus, we have from $(\otimes, <_2, <_3)$-monotone property in (S7) that

$P_j \otimes \prod_{\otimes_{l_j-1}} \gamma \leq_2 P_i \otimes \prod_{\otimes_{l_i-2}} \gamma$. Applying the HLP inequality for an AIHA yields

$$
\begin{aligned}
&(P_i \otimes \prod_{\otimes_{l_i-1}} \gamma) \oplus (P_j \otimes \prod_{\otimes_{l_j+1}} \gamma) \\
&\leq_1 ((P_j \otimes \prod_{\otimes_{l_j-1}} \gamma) \otimes \gamma) \oplus ((P_i \otimes \prod_{\otimes_{l_i-2}} \gamma) \otimes (\gamma \otimes \gamma)) \\
&= (P_j \otimes \prod_{\otimes_{l_j}} \gamma) \oplus (P_i \otimes \prod_{\otimes_{l_i}} \gamma).
\end{aligned}
$$

That $w(\underline{l}') \leq_1 w(\underline{l})$ follows from the commutative property for $\oplus$, the $(\oplus, <_1, <_1)$-monotone property and the $(\otimes, <_1, <_3)$-monotone property. This in turns implies that $w(\underline{l}^1) \leq_1 w(\underline{l}^3)$.

For the case that $\underline{l}^3 \leq \underline{l}^2$, one can also use the $(\oplus, <_1, <_1)$-monotone property and the $(\otimes, <_1, <_3)$-monotone property to show that $w(\underline{l}^3) \leq_1 w(\underline{l}^2)$. □

Let $\underline{l}$ be the length vector of a binary tree with m leaves. Let $\underline{l}^o$ be the length vector of the binary tree with m leaves constructed by power of 2 rule.

Lemma 6.7 *The vector $\underline{l}^o$ is weakly submajorized by $\underline{l}$, i.e., $\underline{l}^o \prec_w \underline{l}$.*

Proof. Let $l^o_{\max}$ and $l^o_{\min}$ be the longest length and the smallest length in $\underline{l}^o$. Note that $l^o_{\max} - l^o_{\min} \leq 1$. Thus, $\underline{l}^o$ is weakly submajorized by any vector $\underline{l}$ as long as $\sum_{i=1}^m l_i^o \leq \sum_{i=1}^m l_i$. Since $\underline{l}^o$ is obtained by constructing a binary tree using the Huffman algorithm with identical weights, we have that $\sum_{i=1}^m l_i^o \leq \sum_{i=1}^m l_i$. □

Theorem 6.1 follows from Lemmas 6.6 and 6.7.

We also note that the result (i) in Corollary 6.2 can be strengthened to the increasing convex ordering. Using Proposition 11.B.5.a in [18], the function $f(\underline{l}) \stackrel{\text{def}}{=} Eg(\sum_{i=1}^m P_i \gamma^{l_i})$ is symmetric, increasing and convex if g is increasing and convex and P_i's are exchangeable r.v.'s. Thus, the function f is also Schur convex by Proposition 3.C.2 in [18]. Applying Lemma 6.7 completes the argument.

7 Conclusions and future research

In this paper, we defined Huffman algebras for independent random variables and showed that the Huffman algorithm produces the optimal tree under various assumptions in Theorems 5.1 and 6.1. This includes both the cases when the weights are deterministic and when the weights are i.i.d. random variables.

In the case of source coding in information theory, there is a natural notion of coding blocks of symbols of a sequence of random variables rather than a single symbol. For example, coding two symbols together corresponds to a probability distribution on the pair of symbols that is a product of the corresponding probabilities of each symbol. We can extend this notion to the other Huffman algebras and define a concept of a tree extension which corresponds to finding a tree for the outer product of the original weights. In the full report [5], we define this notion and show that the sequence of the weights of the optimal trees formed by tree extensions form a subadditive (resp. superadditive) process for an AIHA (resp. ADHA) if the weights are deterministic, and therefore the normalized cost has a limit. Based on this, we can show that the fundamental limit to the normalized cost of the tree is the equivalent of the entropy in the various examples of deterministic Huffman algebras. For the case of random variables, for the $(\max, +)$ algebra (Example 4.2), we can find upper and lower bounds on the normalized cost of of a tree extension if $P_i = p_i + X_i$ for some constants p_i and some independent and identically distributed random variables X_i. These two bounds coincide with each other if p_i's are identical, i.e., P_i's are independent and identically distributed. However, the physical interpretation for tree extensions with random weights for this algebra is not clear at present.

There are other open questions as well: we don't know if there exists a general algorithm for solving the problem $\min_{\mathcal{T}} E \max(P_i + l_i)$ given the knowledge of the distributions of the independent random variables P_i. The results of this paper show that in special cases the Huffman algorithm is optimal, but we do not know the answer for all cases.

This work is only a special case of a very general problem: How well do algorithms designed for deterministic quantities work for random variables. In many applications like scheduling, parallel processing, etc., the deterministic case often reduces to some NP-complete problem, whereas the random case sometimes yields interesting results. And though the intuition of the deterministic case often carries over to the random case, counterexamples like Example 6.5 show that one must always be careful in applying intuition for random variables.

Acknowledgements:

The authors would like to thank the referees for their valuable and constructive comments which improve the presentation of this paper.

References

[1] R.E. Barlow and F. Proschan. *Statistical Theory of Reliability and Life Testing Probabiltity Models.* Holt, Reinhart and Winston, Inc., 1981.

[2] M. Brown and H. Solomon. Optimal issuing policies under stochastic field lives. *J. Appl. Prob.*, 10:761–768, 1973.

[3] L.L. Campbell. Definition of entropy by means of a coding problem. *Z. Wahrscheinlichkeitstheories u. verw. Geb.*, 6:113-118, 1966.

[4] C.S. Chang. A new ordering for stochastic majorization: theory and applications. *Adv. Applied Prob.*, 24:604-634, 1992.

[5] C.S. Chang and J.A. Thomas. Huffman algebras for independent random variables. *IBM RC 17524*, 1992.

[6] C.S. Chang and D.D. Yao. Rearrangement, majorization and stochastic scheduling. *To appear in Math. of Oper. Res.*, 1993.

[7] C.R. Glassey and R.M. Karp. On the optimality of Huffman trees. *SIAM J. Appl. Math.*, 31:368–378, 1976.

[8] M.C. Golumbic. Combinatorial merging. *IEEE Trans. Comput.*, C-25:1164–1167, 1976.

[9] G.H. Hardy, J.E. Littlewood, and G. Polya. *Inequalities.* Cambridge Univ. Press, London and New York, 1952.

[10] T.C. Hu, D. Kleitman, and J.K. Tamaki. Binary trees optimum under various criteria. *SIAM J. Appl. Math.*, 37:246–256, 1979.

[11] T.C. Hu and A.C. Tucker. Optimal computer search trees and variable length alphabetic codes. *SIAM J. Appl. Math.*, 21:514–532, 1971.

[12] D.A. Huffman. A method for the construction of minimum redundancy codes. *Proc. IRE*, 40:1098–1101, 1952.

[13] P.A. Humblet. Generalization of Huffman coding to minimize the probability of buffer overflow. *IEEE Trans. Inform. Theory*, IT-27:230–232, 1981.

[14] F.K. Hwang. Generalized Huffman trees. *SIAM J. Appl. Math.*, 37:124–127, 1979.

[15] A. Itai. Optimal alphabetic trees. *SIAM J. Comput.*, 5:9–18, 1976.

[16] J. Keilson and U. Sumita. Uniform stochastic ordering and related inequalities. *The Canadian Journal of Statistics*, 10:181–198, 1982.

[17] D.E. Knuth. Huffman's algorithm via algebra. *Journal of Combinatorial Theory, Series A*, 32:216–224, 1982.

[18] A.W. Marshall and I. Olkin. *Inequalities: Theory of Majorization and Its Applications.* Academic Press, 1979.

[19] D.S. Parker. Combinatorial merging and Huffman's algorithm. *IEEE Trans. Comput.*, TC -28:365–367, 1979.

[20] D.S. Parker. Conditions for optimality of the Huffman algorithm. *SIAM J. Comput.*, 9:470–489, 1980.

[21] A. Renyi. On measures of entropy and information. *Proc. Fourth Berkeley Symp. Math. Statist. Probability I*, pages 547-561, 1961.

[22] S.M. Ross. *Stochastic Processes.* J. Wiley & Sons, 1983.

[23] J.G. Shanthikumar and D.D. Yao. Bivariate characterization of some stochastic order relations. *Adv. Appl. Prob.*, 23:642–659, 1991.

[24] D. Stoyan. *Comparison Methods for Queues and Other Stochastic Models.* J. Wiley & Sons, 1983.

[25] R.R. Weber, P. Varaiya, and J. Walrand. Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime. *J. Appl. Prob.*, 23:841–847, 1986.