Journal of Universal Computer Science, vol. 3, no. 7 (1997), 813-820
submitted: 15/11/96, accepted: 8/5/97, appeared: 28/7/97 © Springer Pub. Co.

OPTIMUM HUFFMAN FORESTS*

Ioan Tomescu
(Faculty of Mathematics, University of Bucharest, Romania
ioan@inf.math.unibuc.ro)

Abstract: In this paper we solve the following problem: Given a positive integer f and L weights (real numbers), find a partition π with f classes of the multiset of weights such that the sum of the costs of the optimum m-ary Huffman trees built for every class of π is minimum. An application to the optimal extendibility problem for prefix codes is proposed.

Key words: Kraft's inequality, optimum m-ary Huffman tree, weight set partition, optimal extendibility problem.

1 Introduction

Let $M = \{0, 1, \ldots, m-1\}$ be an alphabet containing $m \geq 2$ letters. It is well known that the vertices of a positional m-ary directed tree (or shortly m-ary tree) have a natural correspondence to words over M[see Berstel and Perrin 1985, Cover and Thomas 1991, Even 1973]. The set of words that correspond to the terminal vertices of an m-ary tree forms a prefix (sometimes called instantaneous) code; that is, no word in the code is the beginning of another. Prefix codes are uniquely decodable, as the end of a codeword is immediately recognizable. The codewords of a prefix code $C = \{w_1, w_2, \ldots, w_L\}$ over M satisfy Kraft's inequality [Kraft 1949] :

$$\sum_{i=1}^{L} m^{-l_i} \leq 1, \tag{1}$$

where $l_i = |w_i|$ denotes the length of w_i. The sum in the left-hand side of (1) is called the characteristic sum of C. It is well known that if a vector of word lengths $(l_1, l_2, \ldots, l_L)$ satisfies the characteristic sum condition (1) then there exists a prefix code over the alphabet M with the given vector of code-word lengths. To every prefix code C over M we can associate an m-ary tree $\mathcal{T}(C)$ such that C is the set of words corresponding to the terminal vertices of $\mathcal{T}(C)$ and this tree has

*This work was done while the author has visited the Computer Science Department, University of Auckland, New Zealand.

height $h(\mathcal{T}(C)) = \max\{|w| | w \in C\}$. The inequality (1) holds true with equality if and only if $\mathcal{T}(C)$ is complete, i.e. in case when every non-terminal vertex of $\mathcal{T}(C)$ has exactly m sons, whose associated code words belong to C.
In what follows the degree of a vertex of an m-ary tree is equal, by definition, to the number of its children. If a tree T consists of one vertex, the root then we associate to T the empty word λ, such that $|\lambda| = 0$, so we shall consider that $C = \{\lambda\}$ is also a prefix code.

2 Kraft's inequality revisited

Now suppose that $C_1, C_2, \ldots, C_f$ are prefix codes over M. If we denote by $l_1, l_2, \ldots, l_L$ the code-word lengths in $C_1, C_2, \ldots, C_f$ (considered with their respective multiplicities), then we get

$$\sum_{i=1}^{L} m^{-l_i} = \sum_{i=1}^{f} \sum_{w \in C_i} m^{-l(w)} \leq f$$

since Kraft's inequality holds for every code C_i. Conversely, we can prove the following property:

Lemma 2.1 *If f is any positive integer and a vector of word lengths $(l_1, l_2, \ldots, l_L)$ where $l_i \geq 0$ for every $1 \leq i \leq L$, satisfies the condition*

$$\sum_{i=1}^{L} m^{-l_i} \leq f, \tag{2}$$

then there exists a partition with $s \leq f$ classes of the multiset $WL = \{l_1, l_2, \ldots, l_L\}$,

$$WL = A_1 \cup \ldots \cup A_s \tag{3}$$

such that

$$\sum_{l_i \in A_j} m^{-l_i} \leq 1 \; for \; every \; j = 1, \ldots, s \tag{4}$$

In other words, there exist prefix codes $C_1, \ldots, C_s$ over M such that the code-word lengths in C_i coincide with the numbers in the multiset A_i for any $i = 1, \ldots, s$.

Proof: We closely follow the construction described in Lemma 7.2 of [Even 1973]: Let us assume that $l_1 \leq l_2 \leq \ldots \leq l_L$ and let μ be the number of word lengths equal to l_L. By multiplying (2) by m^{l_L} we get

$$\sum_{i=1}^{L-\mu} m^{l_L - l_i} + \mu \leq f m^{l_L}$$

By a divisibility argument we find that

$$\sum_{i=1}^{L-\mu} m^{l_L - l_i} + \mu + \pi \leq f m^{l_L},$$

where $\mu = \sigma m + \varrho$; σ and ϱ are nonnegative integers; $0 \leq \varrho < m$ and $\pi = 0$ if $\varrho = 0$ and $\pi = m - \varrho$ if $\varrho > 0$. Now $\mu + \pi = \tau m$, where $\tau = \sigma$ if $\varrho = 0$ and $\tau = \sigma + 1$ if $\varrho > 0$. We have

$$\sum_{i=1}^{L-\mu} m^{l_L - l_i} + \tau m \leq f m^{l_L},$$

or equivalently

$$\sum_{i=1}^{L-\mu} m^{-l_i} + \tau m^{-(l_L - 1)} \leq f$$

and this sum corresponds to a multiset of word lengths with $L - \mu + \tau$ words. This multiset still satisfies (2) and its largest word length is smaller by one than the largest word length of the original multiset. We shall proceed by induction on the size of the largest word length. If $l_L = 0$, then $l_i = 0$ for all i and (2) implies that $L \leq f$. Then partition (3) consists of $s = L$ classes and each of them contains a single length equal to zero. Assume that the lemma holds if the largest word length is less than l. By the method just described we can find a multiset WL_1 whose largest length is $l - 1$, and therefore, by the inductive hypothesis, there exists a partition with $s_1 \leq f$ classes $WL_1 = B_1 \cup \ldots \cup B_{s_1}$ such that $\sum_{l_i \in B_j} m^{-l_i} \leq 1$ for every $j = 1, \ldots, s_1$. Since $\tau m \geq \mu$, we can find a partition of WL with $s = s_1$ classes satisfying (4) in the following way: Every occurrence of the length $l_i = l - 1$ in a class B_j will be replaced by q_i occurrences of the length l, such that $0 \leq q_i \leq m$ and $\sum_{j=1}^{s_1} \sum_{\substack{l_i \in B_j \\ l_i = l-1}} q_i = \mu$. The contribution of $l - 1$ to the characteristic sum is $m^{-(l-1)}$, and since $q_i m^{-l} \leq m^{-(l-1)}$, the contribution of the new q_i lengths does not exceed that of $l - 1$, hence (4) is verified for each new class obtained in this way. □

3 Algorithm for optimum Huffman forests

Huffman's algorithm [Huffman 1952] solves the problem of finding a prefix code over M with the minimum weighted length: Given the weights $p_1 \geq p_2 \ldots \geq p_L \geq 0$, construct a prefix code $C = \{w_1, w_2, \ldots, w_L\}$ over M such that if $l_i = |w_i|$, for all $1 \leq i \leq L$, then the cost

$$cost(C) = \sum_{i=1}^{L} p_i l_i \tag{5}$$

is minimum among all prefix codes of cardinality L, or equivalently, find a multiset of word lengths $\{l_1, l_2, \ldots, l_L\}$ which has a minimum cost among all multisets of word lengths satisfying the inequality (1). Huffman's algorithm starts with $L' = L$ and the weights $p_1 \geq p_2 \geq \ldots \geq p_L$. At every step we add the least, i.e. the last, d numbers in the ordered sequence, put the result in the proper place, and decrease the length $L' = L - d + 1$; here $d = 2$ if $m = 2$; otherwise,

$$(6) \qquad d = \begin{cases} m & \text{if } L' \equiv 1 \ (\text{mod } m-1), \\ m-1 & \text{if } L' \equiv 0 \ (\text{mod } m-1), \\ \varrho & \text{if } L' \equiv \varrho \ (\text{mod } m-1), \text{ and } 2 \leq \varrho \leq m-2. \end{cases}$$

After the first step the number of weights, L', satisfies $L' \equiv 1$ (mod $m-1$) and will be equal to one modulo $m-1$, and d becomes equal to m from there on. The operation is repeated until we end up with $L' \equiv m$ weights, each to be assigned length one. Then we start working our way back up: we assign the same length to weights in the previous step, and we increase by one the length of each of the last d weights. The resulting prefix code C is optimal; $cost(C) = cost(\mathcal{T}(C))$, where in (5) l_i is now the level number of the terminal vertex of $\mathcal{T}(C)$ which is associated with weight p_i, the root having level number equal to zero.
In this section we solve the following problem:
Optimum Huffman Forest Problem (OHFP).

- Given two positive integers L and f such that $L \geq f$ and L positive weights $p_1 \geq p_2 \geq \ldots \geq p_L$ find a multiset consisting of non-negative integers $WL = \{l_1, l_2, \ldots, l_L\}$, such that (2) is fulfilled and

$$\sum_{i=1}^{L} p_i l_i$$

is minimum over all positive integers $l_1, l_2, \ldots, l_L$ such that (2) is satisfied.

An equivalent graphical formulation is the following one:

- Given positive integers L,f and L positive weights $p_1 \geq p_2 \geq \ldots \geq p_L$, such that $L \geq f$, find a partition with f classes of the multiset

$$\{p_1, p_2, \ldots, p_L\} = P_1 \cup P_2 \cup \ldots \cup P_f$$

such that if T_i denotes the optimum m-ary Huffman tree built for the weights in P_i for $i = 1, \ldots, f$, then

$$\sum_{i=1}^{f} cost(T_i)$$

is minimum.

Indeed, by Lemma 2.1 if (2) is fulfilled then there exists $s \leq f$ prefix codes $C_1, \ldots, C_s$ over M such that

$$\sum_{i=1}^{L} p_i l_i = \sum_{i=1}^{s} cost(\mathcal{T}(C_i))$$

Since $\sum_{i=1}^{L} p_i l_i$ is minimum it follows that each $\mathcal{T}(C_i)$ is a minimum Huffman tree built for the weights in P_i. Also if $L \geq f$ it follows that $s = f$ since otherwise $s < f$ and there exists at least a tree T among $\mathcal{T}(C_i)$ that has at least a terminal

vertex x different from the root on a level $l \geq 1$ having associated a weight q. By deleting x from T and creating a new tree consisting only of the root having associated the weight q, the cost of the new family of trees decreases strictly, which contradicts the minimality of $\sum_{i=1}^{L} p_i l_i$.
An optimum Huffman forest will be denoted by $OHF(f; p_1, p_2, \ldots, p_L)$.
In building $OHF(f; p_1, p_2, \ldots, p_L)$ we again rely on Huffman's algorithm and change the construction of d in (6) as follows: $d = 2$ if $m = 2$; otherwise,

$$d = \begin{cases} \varrho + 1 & \text{if } \varrho \neq 0, \\ m & \text{if } \varrho = 0 \text{ and } L' > f, \end{cases} \tag{7}$$

where $L' - f \equiv \varrho \pmod{m-1}$ and $0 \leq \varrho \leq m-2$.
We can suppose that $L > f$ since otherwise the construction of $OHF(f; p_1, p_2, \ldots, p_L)$ is obvious: it consists of L trees that are reduced each to the root and its cost is equal to zero. After the first step the length of the vector of word lengths, L', satisfies the relation $L' \equiv f \pmod{m-1}$ and will be equal to f modulo $m-1$; d becomes equal to m from there on. The rule (7) ensures that we end up with exactly f weights.
Each weight q in the current ordered sequence of weights corresponds to a root of an m-ary tree, having the sum of the weights associated to its terminal vertices equal to q. By adding the last d weights these d trees are unified into a single tree, their roots becoming sons of a new root that is associated with a weight equal to the sum of the last d weights. The algorithm terminates when the forest consists of exactly f trees.

Lemma 3.1 *There exists an optimum Huffman forest F consisting of f Huffman trees for the problem $OHFP$ satisfying the following two properties:*
(i) there exists an internal vertex x of F such that all internal vertices different from x have degree equal to m;
(ii) x has exactly d sons on the level with the greatest number of F, where d comes from the formula (7) replacing L' by L, and these d terminal vertices have assigned the smallest weights $p_{L-d+1}, p_{L-d+2}, \ldots, p_L$.

Proof: Let $(l_1, l_2, \ldots, l_L)$ be an optimal word length vector for the weight vector $(p_1, p_2, \ldots, p_L)$ where $p_1 \geq p_2 \geq \ldots \geq p_L \geq 0$. Exactly as for the case of optimum m-ary Huffman trees one can prove that if $p_i > p_j$, then $l_i \leq l_j$, so we may assume that $l_1 \leq l_2 \leq \ldots \leq l_L$. It is clear that F cannot contain internal vertices on levels less than $h(F) - 1$ having their degrees less than m. If y is such a vertex we can take a terminal vertex z on the level $h(F)$ and make it son of y and by this operation the cost of F decreases strictly, which contradicts its optimality.
Now we assume that there exist two vertices x and y on the level $h(F) - 1$ such that x has $n_1 \leq m-1$ sons, and y has $n_2 \leq m-1$ sons on the level $h(F)$. If $n_1 + n_2 \leq m$, then we take $n_1 - 1$ sons of x and make them sons of y; the unique remaining son of x is deleted and its weight is assigned to x (which becomes a terminal vertex). A new forest F_1 is obtained and $cost(F_1) < cost(F)$, a contradiction. If $n_1 + n_2 \geq m + 1$, then we move some sons of x and make them

sons of y such that y has now exactly m sons on the level $h(F)$. The forest F_2 thus obtained has the same cost as F. By repeating this procedure we find an optimum forest F_3 having a unique vertex x on the level $h(F_3) - 1$ such that x has eventually less than m sons. x cannot have a single son since in this case F_3 would not be optimal. If x has a sons $(2 \le a \le m)$ then by repeatedly deleting all terminal sons of any father we arrive at f trees consisting each of the root, hence

$$L \equiv f + a - 1 \pmod{m-1} \tag{8}$$

¿From (8) it follows that $a = d$. The smallest d weights $p_{L-d+1}, p_{L-d+2}, \ldots, p_L$ are assigned to terminal vertices of F_3 lying on the level $h(F_3)$. Eventually, by permuting these vertices and some sons of x between them, we find on optimal forest F fulfilling (i) and (ii). □

Theorem 3.2 *The forest built by the proposed algorithm using rule (7) is an* $OHF(f; p_1, p_2, \ldots, p_L)$.

Proof: For a fixed $f \ge 2$ the proof is by induction on $L \ge f + 1$ and follows the proof for the case of optimum binary Huffman trees [Mehlhorn 1984]:
For every L such that $f + 1 \le L \le m + f - 1$ it is clear that the optimum forest consists of $f - 1$ trees that are reduced each to the root and they have weights $p_1, \ldots, p_{f-1}$ and a tree consisting of a root with $L - f + 1$ sons having weights $p_f, p_{f+1}, \ldots, p_L$. Its cost is $\sum_{i=f}^{L} p_i$ and the optimal length vector is $(\underbrace{0, \ldots, 0}_{f}, \underbrace{1, \ldots, 1}_{L-f+1})$. Let us assume that $L \ge m + f$ and let F_{alg} be the forest constructed by our algorithm for weights $p_1 \ge p_2 \ge \ldots \ge p_L$. The algorithm combines weights $p_{L-d+1}, p_{L-d+2}, \ldots, p_L$ first and construct a tree consisting of a root of weight $\sum_{i=L-d+1}^{L} p_i$ having d sons on the level one. Let F'_{alg} be the forest constructed by our algorithm for the multiset of weights $\{p_1, p_2, \ldots, p_{L-d}, \sum_{i=L-d+1}^{L} p_i\}$. Then

$$cost(F_{alg}) = cost(F'_{alg}) + \sum_{i=L-d+1}^{L} p_i$$

since F_{alg} can be obtained from F'_{alg} by replacing a terminal vertex of weight $\sum_{i=L-d+1}^{L} p_i$ by an internal vertex with d sons that are terminal vertices of weights $p_{L-d+1}, p_{L-d+2}, \ldots, p_L$. Also F'_{alg} is optimal for the problem $OHFP$, hence $F'_{alg} = OHF(f; p_1, p_2, \ldots, p_{L-d}, \sum_{i=L-d+1}^{L} p_i)$ by the induction hypothesis.
Let F_{opt} be an optimal forest satisfying Lemma 3.1, i.e. the terminal vertices with weights $p_{L-d+1}, p_{L-d+2}, \ldots, p_L$ are brothers in F_{opt}. Let F' be the forest obtained from F_{opt} by replacing these d terminal vertices and their father by a

single terminal vertex of weight $\sum_{i=L-d+1}^{L} p_i$. Then

$$cost(F_{opt}) = cost(F^{'}) + \sum_{i=L-d+1}^{L} p_i \geq cost(F_{alg}^{'}) + \sum_{i=L-d+1}^{L} p_i = cost(F_{alg})$$

since $cost(F_{alg}^{'}) \leq cost(F^{'})$ by the induction hypothesis. It follows that $cost(F_{alg}) = cost(F_{opt})$.

□

4 Application to optimal extensions of prefix codes

In [Calude and Tomescu] the following problem has been addressed:
Suppose that we have an extendible prefix code $C = \{w_1, w_2, \ldots, w_s\}$, i.e. a code satisfying strict inequality:

$$\sum_{i=1}^{s} m^{-l_i} < 1,$$

where $l_i = |w_i|$, for $1 \leq i \leq s$. In other words, its associated m-ary tree $\mathcal{T}(C)$ is not complete, i.e. there exists at least one non-terminal vertex of $\mathcal{T}(C)$ having less than m sons. Suppose that we want to extend C with L new words $w_{s+1}, w_{s+2}, \ldots, w_{s+L}$ having the weights $p_1, p_2, \ldots, p_L$, such that if $l_{s+i} = |w_{s+i}|$ for $1 \leq i \leq L$, then

$$\sum_{i=1}^{L} p_i l_{s+i}$$

is minimum over all positive integers $l_{s+1}, l_{s+2}, \ldots, l_{s+L}$ such that

$$\sum_{i=1}^{s+L} m^{-l_i} \leq 1.$$

Suppose that for all non-terminal vertices of $\mathcal{T}(C)$ having less than m sons we construct new sons $v_1, v_2, \ldots, v_f$ such that the resulting tree $\overline{\mathcal{T}(C)}$ is complete. For an optimal extension of $\mathcal{T}(C)$ we must find a partition with $r \leq f$ classes of the multiset of weights $\{p_1, p_2, \ldots, p_L\} = P_1 \cup P_2 \cup \ldots \cup P_r$ such that if T_i denotes the optimum m-ary Huffman tree built for the weights in P_i for $i = 1, \ldots, r$, by identifying, in an injective way, the roots of trees $T_1, T_2, \ldots, T_r$ with some terminal vertices $v_1, v_2, \ldots, v_f$ in $\overline{\mathcal{T}(C)}$, the resulting tree has a minimum cost.

Corollary 4.1 *If all extendible vertices $v_1, v_2, \ldots, v_f$ in $\overline{\mathcal{T}(C)}$ have the same depth, the optimum extension problem can be reduced to the problem $OHFP$.*

Proof: Suppose that all vertices $v_1, v_2, \ldots, v_f$ in $\overline{\mathcal{T}(C)}$ have the level number equal to t. If $L \leq f$ the optimum extension of C is produced by L m-ary words of length t associated to L terminal vertices among $v_1, v_2, \ldots, v_f$ in $\overline{\mathcal{T}(C)}$, so we

can suppose that $L > f$. Let $P_1 \cup P_2 \cup \ldots \cup P_f$ be a partition with f classes of the weight multiset $\{p_1, \ldots, p_L\}$; denote by T_i an m-ary tree whose terminal vertices are associated with the weights in P_i for every $1 \leq i \leq f$.
By identifying the roots of $T_1, \ldots, T_f$ in any order with extendible vertices $v_1, \ldots, v_f$ in $\overline{\mathcal{T}(C)}$ a new tree is obtained and

$$\sum_{i=1}^{L} p_i l_{s+i} = \sum_{i=1}^{f} cost(T_i) + t \sum_{i=1}^{L} p_i.$$

Now $\sum_{i=1}^{L} p_i l_{s+i}$ is minimum if and only if $\sum_{i=1}^{f} cost(T_i)$ is minimum, hence we must find an $OHF(f; p_1, p_2, \ldots, p_L)$. □

However, the general case of the problem addressed in [Calude and Tomescu] remains open.

References

[1] [Berstel and Perrin (85)] Berstel, J., Perrin, D.: *Theory of codes*, Academic Press, New York (1985).

[2] [Calude and Tomescu] Calude, C., Tomescu, I.: *Optimum extendible prefix codes* (submitted).

[3] [Cover and Thomas (91)] Cover, T. M., Thomas, J. A.: *Elements of information theory*, John Wiley, New York (1991).

[4] [Even (73)] Even, S.: *Algorithmic combinatorics*, Macmillan, New York (1973).

[5] [Huffman 52] Huffman, D. A.: A method for the construction of minimum-redundancy codes, *Proc. IRE* 40 (1952), 1098–1101.

[6] [Kraft 49] Kraft, L. G.: *A device for quantizing grouping and coding amplitude modulated pulses*, MS Thesis, Electrical Eng. Dept., MIT, Cambridge, MA. (1949).

[7] [Mehlhorn (84)] Mehlhorn, K.: *Data structures and algorithms. Vol. 1: Sorting and searching*, Springer-Verlag, Berlin (1984).