SBM

Matemática Contemporânea, Vol. 44, 1–10

On the minimum neighborhood of independent sets in the n-cube

Moysés da S. Sampaio Júnior **Fabiano de S. Oliveira**
Luérbio Faria **Paulo E. D. Pinto**

Abstract

Let n, ℓ be positive integers. This paper considers the problem of finding an independent set L having size ℓ of the n-cube graph Q_n such that the cardinality of the neighborhood of L is minimum, i.e., $|N(L)| = min\{|N(L')| : |L'| = \ell$ and L' is an independent set of $Q_n\}$, denoted by $\mathrm{Opt}(n, \ell)$. Such a problem is related to a more restricted version of that of constructing optimal Hamming-Huffman trees for t symbols uniformly distributed, an open problem since the eighties. We present a lower and an upper bound for $\mathrm{Opt}(n, \ell)$. These bounds are found to have the same values for a large number of instances, encouraging us to conjecture that either bound is actually $\mathrm{Opt}(n, \ell)$.

1 Introduction

In data transmission processes, two tasks are usually done separately from each other: data compression and preparation for error detection. Paradoxically, the two phases have conflicting goals: while data compression shrinks the message as much as possible, data preparation for error

2000 AMS Subject Classification: 68R10.

Key Words and Phrases: data compression, error detection, hypercube, independent set.

Supported by CAPES and FAPERJ.

detection adds redundancy to messages so that a receiver can detect (or even better, fix) corrupted ones. Data compression can be achieved with the use of Huffman trees [Huf52]. In 1980, Hamming proposed the union of both compression and error detection through the use of a structure called Hamming-Huffman tree [Ham80]. This data structure compresses data similarly to Huffman trees with the additional feature of enabling the detection of any 1-bit error due to error transmission.

A *Huffman tree* (HT) T is a rooted binary tree in which each edge (u, v), v being a left (resp. right) child of u, is labeled by 0 (resp. 1) and there is a one-to-one mapping between the set of leaves and Σ, the set of all distinct symbols from which a message M to be sent is consisted of. Given T, each symbol a of M is sequentially encoded into a binary string $c(a)$, namely, the sequence of 0's and 1's found on the edges of the directed path from the root of T to the leaf corresponding to a. An optimal HT for M is a tree T such that $\sum_{a\in\Sigma} p(a)|c(a)|$ is minimized, where $p(a)$ stands for the frequency of a (probability of occurrence) and $|c(a)|$ is the length of string $c(a)$.

A *Hamming-Huffman tree* (HHT) T is a HT except for the fact that, for each leaf labeled with $a \in \Sigma$, there must exist leaves $e_1, \ldots, e_k$ such that each $c(e_i)$, $1 \leq i \leq k$, differs from $c(a)$ for exactly one position. Therefore, $k = |c(a)|$. Leaves $e_1, \ldots, e_k$ are called error leaves of a. Under the assumption that when data is transmitted at most 1 bit can inadvertently be flipped, when $c(e)$ is identified in the decoding process, where e is an error leaf, a transmission error is detected. Optimal HHTs are defined exactly the same as optimal HTs.

Figure 1 depicts two HHTs associated with messages using symbols $\Sigma = \{a, b\}$. Error leaves are colored with black. In the tree on the right, leaves having encodings 000 and 111 are chosen as symbol leaves, resulting in 6 error leaves, whereas in the tree on the left, leaves having encodings 000 and 011 are the ones chosen as symbol leaves, resulting in 4 error nodes.

Although optimal HTs are possible to be built efficiently in a greedy

fashion, the construction of HHTs are open since defined by Hamming in the eighties [Ham80].

Since different HHTs may have different number of error leaves, the problem of minimizing the number of error leaves seems to be related to the problem of building an optimal HHT, since error leaves being numerous may force symbol leaves to be chosen from higher levels of the tree, increasing the size of encodings.

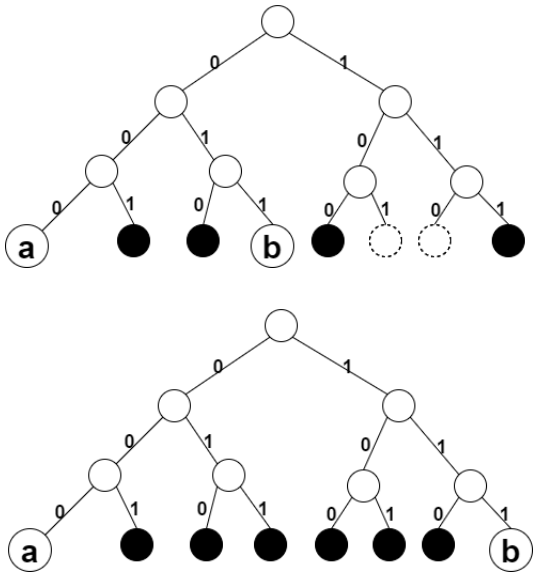

Figure 1: Examples of Hamming-Huffman trees.

The problem considered in this paper is the following: given natural numbers n and $\ell \leq 2^{n-1}$, determine the smallest number $f(n, \ell)$ of error leaves in a HHT having height n and in which all the ℓ symbol leaves are at level n.

To illustrate the relation between computing $f(n, \ell)$ and computing optimal HHTs, let us consider the constrained problem of building an optimal HHT for n symbols such that the leaves lie on at most two levels. The general idea of a dynamic programming that builds such a tree is as follows. A full binary tree having height $\lceil \lg n \rceil + 1$ is enough to build a

HHT in which all symbols correspond to leaves at the last level (choose any n-size subset from the set of leaves encoded with an even number of 1's). Therefore, for an optimal HHT having leaves only at levels h_1 and h_2, with $h_1 \leq h_2$, we have $h_1 \leq \lceil \lg n \rceil + 1$. For each $1 \leq h_1 \leq \lceil \lg n \rceil + 1$, $1 \leq \ell_1 \leq \min\{n, 2^{h_1-1}\}$, we consider mapping ℓ_1 leaves to symbols on height h_1. The least number possible of error leaves is $f(h_1, \ell_1)$ and thus $r = 2^{h_1} - (\ell_1 + f(h_1, \ell_1))$ are free leaves (neither mapped to a symbol nor error leaves). If $r > 0$ and $n > \ell_1$, allocate the remaining $n - \ell_1$ symbols in r trees (whose roots are the free leaves) and find a HHT for $\ell_2 = \lceil (n - \ell_1)/r \rceil$ symbols having the minimum height h_2 possible. This second problem can be reduced to that of minimizing the number of error leaves, i.e., such h_2 is the smallest natural for which $f(h_2, \ell_2)$ is defined. The cost of each HHT generated is therefore $h_1 \sum_{|c(a)|=h_1} p(a) + h_2 \sum_{|c(a)|=h_2} p(a)$, and an optimal HHT is any of those having the smallest computed cost.

2 Neighborhood of independent sets in Q_n

Let G be a simple graph. The *neighborhood* of $u \in V(G)$ is $N_G(u) = \{v \in V(G) \mid (u, v) \in E(G)\}$. Let $U \subseteq V(G)$ and define $N_G(U) = \bigcup_{u \in U} N_G(u)$. When G is clear in the context, it may be omitted from the notation.

A n-cube, denoted by Q_n, is the graph having as vertex set all the binary numbers with size n (2^n vertices therefore) where two distinct vertices are adjacent if their binary representations differ only in one position. We can make a one-to-one mapping between the leaves of a HHT which is a full binary tree having height n to the vertices of Q_n, and among those possible mappings, we shall assume the more natural one in which leaf a corresponds to $c(a) \in V(Q_n)$.

We define the *parity* of a binary representation v as the parity of the number of 1's in v. It is shown in [Föl77] that Q_n is a bipartite graph with partition $E \cup O$, where E and O are the subsets of vertices of $V(Q_n)$ with even parity and odd parity, respectively.

The problem of finding the minimum number of error leaves in a HHT which is a full binary tree T of height n with ℓ symbol leaves is equivalent to that of finding, over all independent sets L of size ℓ in n-cubes, one that minimizes $|N(L)|$. This is so because U can be the set of symbol leaves in T if and only if $c(u)$ and $c(v)$ differ at least in two positions for all distinct $u, v \in U$ if and only if $L = \{c(u) : u \in U\}$ is an independent set of Q_n. Thus, the set of errors leaves associated with U is precisely $N(L)$ in Q_n.

Shifting to this perspective, given two integers n and ℓ, the problem considered in this paper is alternatively to find an independent set L of Q_n with $|L| = \ell$ having the least neighborhood cardinality over all the possible independent sets with the same size. Such optimal cardinality will be denoted by $\mathrm{Opt}(n, \ell)$.

Lemma 2.1. *If the distance between $u, v \in V(Q_n)$ is 2, then $|N(u) \cap N(v)| = 2$.*

Lemma 2.2. *If L is an independent set of Q_n with $|L| = \ell$ such that $|N(L)| = \mathrm{Opt}(n, \ell)$, then all vertices of L belong to a same part of the partition $E \cup O$ of $V(Q_n)$, where E (resp. O) consists of vertices with even (resp. odd) parity.*

3 Lower bound for $\mathrm{Opt}(n, \ell)$

This section presents a lower bound for $\mathrm{Opt}(n, \ell)$. Given a vertex $v \in Q_n$, let $v(i)$ denote the i-th binary digit of v and $0v$ (resp. $1v$) be the binary number obtained from v by prefixing a 0 (resp. 1).

Let (V_0, V_1) be a bipartition of $V(Q_n)$ such that $V_0 = \{v \in V(Q_n) \mid v(1) = 0\}$ and $V_1 = \{v \in V(Q_n) \mid v(1) = 1\}$. Let $Q^0 = Q_n[V_0]$ and $Q^1 = Q_n[V_1]$. Clearly, Q^0 and Q^1 are isomorphic to Q_{n-1}, with $v \in V(Q_{n-1})$ mapped to vertices $0v \in V_0$ and $1v \in V_1$, respectively. Furthermore, for $E^{01} = \{(0v, 1v) : v \in V(Q_{n-1})\}$, $E(Q_n) = E(Q^0) \cup E(Q^1) \cup E^{01}$ holds.

Theorem 1. $\mathrm{Opt}(n, \ell) \geq \min\{\max\{\ell_1, \mathrm{Opt}(n-1, \ell_0)\} + \max\{\ell_0, \mathrm{Opt}(n-1, \ell_1)\} : \ell_0 + \ell_1 = \ell\}$.

Proof. Let $L \subset V(Q_n)$, $|L| = \ell$, be an independent set of Q_n such that $|N(L)| = \mathrm{Opt}(n, \ell)$. As $V_0 \cup V_1$ is a bipartition of $V(Q_n)$, then $(L \cap V_0) \cup (L \cap V_1)$ is a bipartition of L and $(N(L) \cap V_0) \cup (N(L) \cap V_1)$ is a bipartition of $N(L)$. Let $\overline{\ell_0} = |L \cap V_0|$ and $\overline{\ell_1} = |L \cap V_1|$. Since $N(L) \cap V_0 \supseteq N(L \cap V_0) \cap V_0$ whose cardinality is at least $\mathrm{Opt}(n-1, \overline{\ell_0})$, and $N(L) \cap V_0 \supseteq N(L \cap V_1) \cap V_0$ whose cardinality is at least $\overline{\ell_1}$ because $|N(v) \cap V_0| = 1$ for all $v \in V_1$, then $|N(L) \cap V_0| \geq \max\{\overline{\ell_1}, \mathrm{Opt}(n-1, \overline{\ell_0})\}$ and, by symmetric argument, $|N(L) \cap V_1| \geq \max\{\overline{\ell_0}, \mathrm{Opt}(n-1, \overline{\ell_1})\}$. So, we have that $|N(L)| \geq \max\{\overline{\ell_1}, \mathrm{Opt}(n-1, \overline{\ell_0})\} + \max\{\overline{\ell_0}, \mathrm{Opt}(n-1, \overline{\ell_1})\} \geq \min\{\max\{\ell_1, \mathrm{Opt}(n-1, \ell_0)\} + \max\{\ell_0, \mathrm{Opt}(n-1, \ell_1)\} : \ell_0 + \ell_1 = \ell\}$. ■

4 Upper bound for $\mathrm{Opt}(n, \ell)$

We define an extension of the Breadth-First Search (BFS) in Q_n, called *Unrestricted Breadth-First Search* and show some properties related to this search useful in helping to find an upper limit for $\mathrm{Opt}(n, \ell)$. In UBFS, each vertex becomes available to be visited again after its exploration is completed. A similar variation of DFS, called Unrestricted Depth-First Search, is described in [Szw88].

The UBFS tree differs from the classical BFS tree by the fact that some vertices appear more than once. Contrary to UDFS, in which even allowing multiple visits to a same vertex the search comes to an end in finite time, UBFS must impose some termination condition to avoid infinite search. In the context of this paper, the UBFS algorithm stops when the corresponding tree has n levels. For this section, we consider the levels numbered from 0 to $n-1$.

Figure 2 shows a search tree for Q_5 starting with the vertex 0 (vertices are named in their decimal representation). The tree edges are represented by solid lines and cross edges are represented by dashed lines. In the case of UBFS in the Q_n, cross edges always connect two nodes in consecutive levels. For the sake of clearness, the figure only depicts the cross edges incident to the second node of each level of the tree.

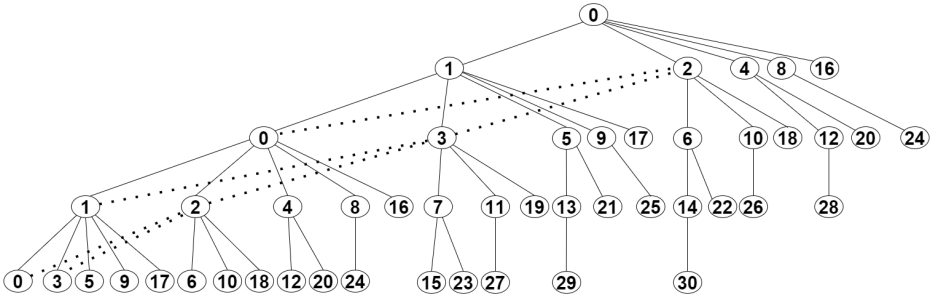

Figure 2: UBFS tree for the Q_5.

It is easy to see that, in any UBFS tree for the Q_n, all the nodes at even levels have the same parity of the root while nodes at odd levels have opposite parity.

Theorem 2. The set of vertices at the level $(n-1)$ of an UBFS tree for Q_n is precisely either E or O, the parts of the partition $E \cup O$ of $V(Q_n)$, where E (resp. O) consists of vertices with even (resp. odd) parity.

The previous theorem justifies the termination condition of stopping the search at level $n-1$: the search beyond the $(n-1)$-th level will alternate the vertices of E and O for each new level.

Lemma 4.1. *Let $S = \langle s_1, s_2, s_3, \cdots, s_n \rangle$ be an ordering of the n vertices of $N(u), u \in Q_n$. Then $|N(s_i) \cap N(\{s_1, \cdots, s_{i-1}\})| = i$, for $2 \leq i \leq n$.*

Let $R(i,j)$ be the number of leaves which are descendants of the node $v = (i,j)$, which is the j-th node of i-the level of a particular UBFS tree, for all $1 \leq j \leq n$, $0 \leq i \leq n-1$. The value $R(i,j)$ is said to be the number of leaves represented by v. We define $R(i,j) = 0$ for all $1 < j \leq n$ when $i = 0$. Theorem 3 helps to partially evaluate the function R for the first n nodes of each level of the tree. In the sequence it is shown how to extend the function to all nodes.

Theorem 3. For all $0 \leq i \leq n-1$ and $1 \leq j \leq n$, we have:

$$R(i,j) = \begin{cases} 0, & \text{if} \quad i = 0 \text{ and } j > 1 \\ 1, & \text{if} \quad i = n-1 \\ \sum_{1 \leq k \leq n} R(i+1,k), & \text{if} \quad i < n-1,\ j = 1 \\ \sum_{j+1 \leq k \leq n} R(i+1,k), & \text{if} \quad i < n-1,\ j > 1 \end{cases} \tag{1}$$

For each node (i,j) in the tree, $a(i,j)$ is its *affiliation order*, defined in terms of the position of (i,j) relatively to its siblings and the number of cross edges of its parent node. If (i,j) is the k-th child node and its parent has c cross edges, than $a(i,j) = k+c$. It can be shown that we can extend the function R to all nodes (i,j), $j > n$, of the tree, by considering $R(i,j) = R(i,a(i,j))$.

Now let us consider the cardinality of the neighborhood of the set L composed by the first ℓ leaves of the tree. This value can be obtained by expanding the search one level ahead and counting the descendants of those nodes. However, this expansion is not necessary. If L corresponds exactly to the set of leaves descendants of the first node of a level $i > 0$ of the tree, that is, node $(i,1)$, then such a value is the number of vertices with distance $n-i$ to $(i,1)$, which is $R(i-1,1)$. If the set L is fully represented by the first $j < n$ nodes of level i, a similar argument shows that $|N(L)| = \sum_{1 \leq k \leq j} R(i-1,k)$.

In general, the cardinality of the neighborhood for the leaves represented by node (i,j) that is not common to other leaves is $R(i-1,a(i,j))$.

Now we describe a method to evaluate the cardinality of the neighborhood L. We search the tree looking for a minimal set S of nodes (i,j) whose union of its representation is L. For each node (i,j) of S, we accumulate $R(i-1,a(i,j))$. This evaluation can be made in time $O(n)$ in a top-down fashion using the matrix R. Such a process is as follows.

We start by searching the lowest line i of R having $R(i,1) \leq \ell$. Mark the cell $(i,1)$ and continue sequentially right in this line marking cells, while the sum of $R(i,j)$ of marked cells (i,j) does not exceed ℓ. When this happens (in column $k+1$), continue the process in the following line,

column $k+1$. Repeat the process in the next lines until the sum of $R(i,j)$ of all marked cells (i,j) equals ℓ. Then the sum of $R(i-1,j)$ related to all marked cells is $f(n,\ell) = |N(L)|$.

Clearly, $f(n,\ell)$ is an upper bound for $\mathrm{Opt}(b,\ell)$.

5 Conclusion

This work was motivated by the problem of determining optimal Hamming-Huffman trees, an open problem for 30 years now. A particular case of the problem, namely when Hamming-Huffman trees allow symbol leaves lying at most at two levels, reduces to the problem of finding an independent set having a given size ℓ of a n-cube with the smallest neighborhood cardinality over all such independent sets. We provide lower and upper bounds for the problem. Experimental results show that the lower and upper bounds are equal for all $1 \leq n \leq 22$, $1 \leq \ell \leq 2^{n-1}$. This fact may suggest that both bounds are the actual $\mathrm{Opt}(n,\ell)$ values in general.
Acknowledgment: the authors are thankful to the referees for their thorough reports which improved substantially the final version.

References

[Föl77] S. Földes, *A characterization of hypercubes*, Discrete Math. **17** (1977), no. 2, 155–159. MR 0450136

[Ham80] Richard Wesley Hamming, *Coding and information theory*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1980. MR 555735

[Huf52] D. A. Huffman, *A method for the construction of minimum-redundancy codes*, Proceedings of the IRE **40** (1952), no. 9, 1098–1101.

[Szw88] J. L. Szwarcfiter, *Grafos e algoritmos computacionais*, 2 ed., Ed. Campus Ltda, 1988.

Moysés da S. Sampaio Júnior
Universidade do Estado do Rio de Janeiro (UERJ)
Rio de Janeiro, Brazil
moyses.sampaio@gmail.com

Fabiano de S. Oliveira
Universidade do Estado do Rio de Janeiro (UERJ)
Rio de Janeiro, Brazil
fabiano.oliveira@ime.uerj.br

Luérbio Faria
Universidade do Estado do Rio de Janeiro (UERJ)
Rio de Janeiro, Brazil
luerbio@cos.ufrj.br

Paulo E. D. Pinto
Universidade do Estado do Rio de Janeiro (UERJ)
Rio de Janeiro, Brazil
pauloedp@ime.uerj.br