Lower Bounds on Merging Networks

ANDREW CHI-CHIH YAO AND FOONG FRANCES YAO

University of Illinois, Urbana, Illinois

ABSTRACT Let $M(m, n)$ be the minimum number of comparators needed in an $(m, n)$-merging network. It is shown that $M(m, n) \geq n(\log (m + 1))/2$, which implies that Batcher’s merging networks are optimal up to a factor of $2 + \epsilon$ for almost all values of $m$ and $n$. The limit $r_m = \lim_{n \to \infty} M(m, n)/n$ is determined to within 1. It is also proved that $M(2, n) = \lceil 3n/2 \rceil$.

KEY WORDS AND PHRASES comparator, network, merging, odd-even merge

CR CATEGORIES 525, 531, 631

1. Introduction

An $(m, n)$-merging network [4, p. 230] is a network in which the input consists of two sorted sets $\{x_1 \leq x_2 \leq \ldots \leq x_m\}$ and $\{y_1 \leq y_2 \leq \ldots \leq y_n\}$, and the output is the sorted set $\{z_1 \leq z_2 \leq \ldots \leq z_{m+n}\}$ with $\{z_i\}'s = \{x_i\}'s, y_k\}'s$. The network is built of comparators which are themselves $(1, 1)$-merging networks. A comparator is usually drawn as in Figure 1, and a merging network is shown in Figure 2. One can choose any input convention that specifies how the sequence $\{x_i\}'s$ are to be interspersed in the $\{y_k\}'s$. By standard technique [4, p. 236], it can be shown that a transformation exists between any two such conventions which preserves the number of comparators used.

Let $M(m, n)$ be the minimum number of comparators needed in an $(m, n)$-merging network. The famous "odd-even merge" by Batcher [1] readily gives the following upper bound for $M(m, n)$, which is also the best upper bound currently known [4, p. 226, eq. (6)]:

$$M(m, n) \leq m + m \left(\left\lfloor \log m \right\rfloor /2 + \log m\right) + 1 + R_m(n - m)$$

for $m \leq n$,

where

$$R_m(r) = r + \left[\sum_{j=1}^{r-1} (r + j)/2^{\left\lfloor \log (r + j) \right\rfloor + 1}\right].$$

By noting that $R_m(r) \leq r(\log m + m/2^{\log m}) + m - 1$, we get

$$M(m, n) \leq (n + m) (\log m + m/2^{\log m}).$$

On the other hand, Floyd [4, p. 230] proved $M(n, n) \geq \lceil \log n \rceil + O(n)$ for the case $m = n$. These efforts determined $M(n, n)$ asymptotically to within a factor of 2. The behavior of $M(m, n)$ for general $m$ and $n$, however, is not well understood.

In this paper we shall derive a lower bound on $M(m, n)$ by a different approach.

Copyright © 1976, Association for Computing Machinery, Inc. General permission to republish, but not for profit, all or part of this material is granted provided that ACM's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Association for Computing Machinery.

This work was partially supported by the National Science Foundation under Grant GJ-41538.

Authors' present address: A C -C Yao, Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139, F F Yao, Division of Engineering, Brown University, Providence, RI 02912

1 $\log$ is logarithm to the base 2.
Lower Bounds on Merging Networks

This bound, together with (1), determines $M(m, n)$ up to a factor of $2 + \epsilon$ for almost all $m$ and $n$. We shall also show that $M(2, n) = \lceil 3n/2 \rceil$. The main results are Theorems 1 and 2.

2. $(2, n)$-Merging Networks

It has been conjectured that Batcher's merging networks are optimal in terms of the number of comparators used. In this section we shall lend support to this conjecture by showing that the odd-even merge yields optimal $(2, n)$-merging networks. Indeed, Batcher's network uses $\lceil 3n/2 \rceil$ comparators (Figure 3), and we will prove the following theorem.

**Theorem 1.** $M(2, n) = \lceil 3n/2 \rceil$.

To prove the theorem, it suffices to show that $\lceil 3n/2 \rceil$ is a lower bound for $M(2, n)$. We shall need some notations and lemmas.

The lines in a network can be numbered from the top down as first, second, . . . , etc. A comparator is called an $(1, j)$-comparator if its two input lines are numbered the $i$th and the $j$th with $i < j$. In Figure 3, a is a $(3, 8)$-comparator. A comparator is said to be of the form $(i, *)$ if it is an $(i, j)$-comparator for some $j$. Comparator of the form $(*, j)$ is defined similarly.

By the remark made in Section 1, without loss of generality we can consider only those $(2, n)$-merging networks where $x_1$ and $x_2$ are inputs to the top line and the bottom line, respectively. Furthermore, assume the input $(n + 2)$-tuple $(x_1, y_1, y_2, \ldots, y_n, x_2)$ to be a permutation of the integers $(1, 2, \ldots, n + 2)$. Now, if we fix $x_2 = n + 2$, then any value of $x_1$ between 1 and $n + 1$ determines a unique input $(n + 2)$-tuple and hence a unique path in the network—the path that takes $x_1$ to the appropriate output line (Figure 4 (a)). Note that in such a path $x_1$ only moves downward through comparators. That is, if an $(i, j)$-comparator $a$ is crossed by $x_1$ in such a path, $a$ must be used to take $x_1$ from line $i$ down to line $j$ (and the value of $x_1$ has to be no less than $j$). We define $A$ to be the set of all comparators crossed by $x_1$, $1 \leq x_1 \leq n + 1$, in these paths. Similarly, while fixing $x_1 = 1$, we consider all the possible paths traversed by $x_2$ and let $B$ be the set of all comparators crossed by $x_2$ in them. Here the comparators in $B$ are only used to move $x_2$ upward (Figure 4(b)).

**Lemma 1.** For each $j$, $2 \leq j \leq n + 1$, the following are true:

(a) There is a unique comparator in $A$ which is of the form $(*, j)$.

(b) There is a unique comparator in $B$ which is of the form $(j, *)$.

**Proof.** We will only prove (i), since (i) and (ii) are symmetrical. As we fix $x_2 = n + 2$ and let $x_1 = j$, a descending path will take $x_1$ to the $j$th output line. The last comparator crossed by $x_1$ in this path must be of the form $(*, j)$. We shall now show that for any fixed $j$, $2 \leq j \leq n + 1$, there is only one comparator in $A$ which is of the form $(*, j)$.

Let $w$ be the maximum value of $x_1$ that will cause $x_1$ to cross a comparator of the form $(*, j)$. Clearly $j \leq w \leq n + 2$. Denote by $a$ the $(i, j)$-comparator crossed by $x_1 = w$. The following two statements are true by the definition of $w$:

(1) All the values $u$ of $x_1$ where $j \leq u \leq w$ will cause $x_1$ to follow the same path until past comparator $a$. 
(2) When \( x_1 \) assumes any value \( v \) such that \( w < v \leq n + 2 \), \( x_1 \) does not cross any comparator of the form \((*, j)\).

As a consequence of these two statements, \( \alpha \) must be the only comparator of the form \((*, j)\) in \( A \). □

Because of Lemma 1, we can introduce the following mappings \( T \) and \( T' \).

**Definition 1.** For any \((j, k)\)-comparator \( \alpha \) in \( A \cap B \) (note that \( 2 \leq j < k \leq n + 1 \)), we will let \( T(\alpha) \) be the unique comparator in \( A \) of the form \((*, j)\), and let \( T'(\alpha) \) be the unique comparator in \( B \) of the form \((k, *)\).

It is easy to see that both \( T(\alpha) \) and \( T'(\alpha) \) must lie to the left of \( \alpha \) in the network.

**Lemma 2.** \( T \) is a mapping from \( A \cap B \) into \( A - B \).

**Proof.** We will show that for \( \alpha \in A \cap B \), \( T(\alpha) \notin B \). Suppose this is not true; then \( T(\alpha) \in A \cap B \). Assume \( \alpha \) is of the form \((j, *)\); then \( T(\alpha) \) is of the form \((*, j)\). Since \( T(\alpha) \in A \cap B \), by the definition of \( T' \) we have \( T'(T(\alpha)) = \alpha \). However, \( T(\alpha) \) must lie to the left of \( \alpha \), and \( T'(T(\alpha)) \), or \( \alpha \), must lie to the left of \( T(\alpha) \). This is a contradiction. □

**Lemma 3.** The mapping \( T \) is one-to-one.

**Proof.** Let \( \alpha \) and \( \alpha' \) be two separate comparators in \( A \cap B \). Assume \( \alpha \) is of the form \((j, *)\) and \( \alpha' \) is of the form \((j', *)\). By Lemma 1 we must have \( j \neq j' \). Therefore it is impossible to have \( T(\alpha) = T(\alpha') \). □

**Proof of Theorem 1.** Lemmas 2 and 3 together imply \( |A - B| \geq |A \cap B| \). It follows that \( |A - B| \geq \left| \frac{1}{2} |A| \right| \). However, it is easy to see that \( |A| = n \) and \( |B| = n \). (There are \( n \) internal nodes in a binary tree with \( n + 1 \) leaves.) Therefore \( |A - B| \geq \left| \frac{n}{2} \right| \). This leads to \( |A \cup B| = |A - B| + |B| \geq \left| \frac{n}{2} \right| + n = \left| \frac{3n}{2} \right| \). □

3. Lower Bound for \( M(m, n) \)

The upper bound for \( M(m, n) \) as given by formula (1) implies

\[
M(m, n) \leq (n + m)(\log(m + 1) + \text{const})/2 \quad \text{for } m \leq n
\]  

(2)

In this section we shall derive a lower bound

\[
M(m, n) \leq n(\log(m + 1))/2 \quad \text{for } m \leq n.
\]  

(3)

By comparing (2) and (3), we see that for any \( \varepsilon > 0 \), \( M(m, n) \) is determined to within a factor of \( 2 + \varepsilon \) for all sufficiently large \( m \) and \( n \). Our proof of (3) in the following will be
based on an entropy argument, which is inspired by a technique first employed by Floyd [2] in the study of matrix transposition.

In an \((m, n)\)-merging network, we can look at the input as a column vector with \(m + n\) components, and each comparator as a function which, given a vector, either interchanges two of its components or does not change it. We shall assume that the input vector is a permutation of \((1, 2, \ldots, m + n)\) and that the \(x_i\)'s are inputs to the first \(m\) lines. Thus, if a network contains \(l\) comparators appearing in sequence from left to right as \(\alpha_1, \alpha_2, \ldots, \alpha_l\), then any input vector \(V_0 = (x_1, \ldots, x_m, y_1, \ldots, y_n)^T\) is transformed into \((1, 2, \ldots, m + n)^T\) through a chain of vectors:

\[
V_0 \xrightarrow{\alpha_1} V_1 \xrightarrow{\alpha_2} V_2 \xrightarrow{\alpha_3} \cdots \xrightarrow{\alpha_l} V_l,
\]

where \(V_l = (1, 2, \ldots, m + n)^T\).

Now a set of \(r\) different input vectors can be viewed as forming an \((m + n) \times r\) matrix. Each \(\alpha_i\) can then be regarded, by its columnwise action, as a transformation on \((m + n) \times r\) matrices. We will choose \(r = m + 1\) and consider the effect of the \(\alpha_i\)'s when the following input matrix \(A_0\) is given:

\[
A_0 = \begin{bmatrix}
1 & 2 & 3 & \cdots & n + 1 & n + 2 & n + 3 & \cdots & m + 1 \\
1 & 2 & 3 & \cdots & n + 2 & n + 3 & n + 4 & \cdots & m + 2 \\
1 & 2 & 3 & \cdots & n + 3 & n + 4 & n + 5 & \cdots & m + 3 \\
1 & 2 & 3 & \cdots & n + m & n + m + 1 & n + m + 2 & \cdots & m + m \\
1 & 2 & 3 & \cdots & n + 1 & n + 2 & n + 3 & \cdots & m + n
\end{bmatrix}
\]

In the \(i\)th column of \(A_0\), the upper part is the ordered list of length \(m\), \((1, 2, \ldots, \nu - 1, n + \nu, n + \nu + 1, \ldots, n + m)\), and the lower part is the ordered list of length \(n\), \((\nu, \nu + 1, \ldots, n + \nu - 1)\).

Let

\[
A_0 \xrightarrow{\alpha_1} A_1 \xrightarrow{\alpha_2} A_2 \xrightarrow{\alpha_3} \cdots \xrightarrow{\alpha_l} A_l
\]

be the sequence of transformations that \(A_0\) undergoes in an \((m, n)\)-merging network, where

\[
A_i = \begin{bmatrix}
1 & 2 \\
2 & 2 \\
m + n & m + n
\end{bmatrix}
\]

To derive a lower bound on \(l\), the number of comparators in the network, we define an entropy function:

**Definition 2.** Given a vector \(v = (a_1, a_2, \ldots, a_{m+1})\), where \(1 \leq a_k \leq m + n\) for all \(k\), we define \(p_i = |\{k : a_k = i\}|\) for \(1 \leq i \leq m + n\), and \(E(v) = \sum_{i=1}^{m+n} p_i \log p_i\). (If \(p_i = 0\), then \(p_i \log p_i\) is taken to be 0.)

**Definition 3.** For the matrices \(A_h, 0 \leq h \leq l\), defined in (4), let \(E(A_h) = \sum_{j=1}^{m+n} E(v_j)\), where \(v_j\) is the \(j\)th row vector of \(A_h\).

**Lemma 4.**

\[
E(A_0) = 2 \sum_{j=1}^{m} j \log j
\]

and

\[
E(A_l) = (m + n) \cdot (m + 1) \log(m + 1).
\]
PROOF. In $A_o$, only the first $m$ row vectors have non-zero entropies. For $1 \leq j \leq m$, the $j$th row vector of $A_o$ has entropy $j \log j + (m + 1 - j) \log (m + 1 - j)$; hence (5) follows. Equation (6) is true because every row vector of $A_i$ has entropy $(m + 1) \log (m + 1)$.

For each $h$, $A_h$ and $A_{h-1}$ differ in at most two rows; thus the difference in their entropies is bounded as implied by the following lemma.

LEMA 5. Let $v$ and $v'$ be two vectors satisfying the conditions of Definition 2, and let $w$ and $w'$ be obtained from $v$ and $v'$ by exchanging certain corresponding components. Then

$$E(w) + E(w') \leq E(v) + E(v') + 2(m + 1).$$

PROOF. Let $p_i, q_i, q_i', p_i, q_i', p_i', q_i'$ denote the number of i's appearing in $v, v', w, w'$, respectively. Then $p_i + p_i' = q_i + q_i', i = 1, 2, \ldots, m + n$. Hence

$$E(w) + E(w') - E(v) - E(v') = \sum_i q_i \log q_i + \sum_i q_i' \log q_i' - \sum_i p_i \log p_i - \sum_i p_i' \log p_i'$$

$$\leq \sum_i (q_i + q_i') \log(q_i + q_i') - \sum_i p_i \log p_i - \sum_i p_i' \log p_i'$$

$$= \sum_i (p_i + p_i') \log(p_i + p_i') - \sum_i p_i \log p_i - \sum_i p_i' \log p_i'. \tag{7}$$

Since it is true that $p_i \log(p_i + p_i')/p_i + p_i' \log(p_i + p_i')/p_i'$ assumes its maximum value $p_i + p_i'$ when $p_i = p_i'$, it follows from (7) that

$$E(w) + E(w') = E(v) - E(v') \leq \sum_i (p_i + p_i') = \sum_i p_i + \sum_i p_i'$$

$$= (m + 1) + (m + 1) = 2(m + 1).$$

This proves Lemma 5. □

THEOREM 2. $M(m, n) \leq n (\log(m + 1))/2$

PROOF. By Lemma 4, $E(A_o) = 2 \sum_{j=1}^{m} \log j \leq m(m + 1) \log m$. Thus

$$E(A_i) - E(A_o) \geq (m + n)(m + 1) \log(m + 1) - m(m + 1) \log m$$

$$\geq n(m + 1) \log(m + 1).$$

On the other hand, Lemma 5 implies that

$$E(A_h) - E(A_{h-1}) \leq 2(m + 1) \quad \text{for } 1 \leq h \leq l.$$

It follows that

$$l \geq n(m + l) \log(m + 1)/2(m + 1) = n \log(m + 1)/2,$$

where $l$ is the number of comparators in any $(m, n)$-merging network. □

Remark on $h_m \sim M(m, n)/n$: Observe that an $(m, n_1 + n_2)$-merging network can be obtained by cascading an $(m, n_1)$-merging network and an $(m, n_2)$-merging network. Therefore, for fixed $m$, $M(m, n)$ is a subadditive function of $n$, i.e. $M(m, n_1 + n_2) \leq M(m, n_1) + M(m, n_2)$. This fact implies (see [3, p. 605, Ans. to Exer. 39]) that the limit $r_m = \lim_{n \to \infty} M(m, n)/n$ exists. The exact values of $r_m$ are not known for $m$ greater than 2; the following bounds, which determine $r_m$ to within 1, are immediate consequences of (1) and (3): $(\log(m + 1))/2 \leq r_m \leq \log m)/2 + m/2^{\log m}$. 

4. Conclusion

We have shown that Batcher's $(m, n)$-merging network is in general optimal up to a constant factor. And at least in one nontrivial case ($m = 2$), we have shown that Batcher's merging network is in fact the best possible. It will be interesting to see
whether Batcher's merging network is optimal for more cases—in particular, when \( m = 3 \).

ACKNOWLEDGMENT. The authors are grateful for the valuable suggestions made by the referees.

REFERENCES

1 Batcher, K E. Sorting networks and their applications. Proc AFIPS 1968 SJCC, Vol 32, AFIPS Press, Montvale, N J, pp 307-314
2 Floyd, R W. Permuting information in idealized two-level storage. In Complexity of Computer Computations, R E Miller and J W Thatcher, Eds, Plenum Press, 1972, pp 105-110
3 Knuth, D E. The Art of Computer Programming, Vol 1, 2nd ed Addison-Wesley, Reading, Mass, 1973
4 Knuth, D E. The Art of Computer Programming, Vol 3 Addison-Wesley, Reading, Mass, 1973

RECEIVED OCTOBER 1974; REVISED JANUARY 1976