NASA Contractor Report 191531

ICASE Report No. 93-64

ICASE

20 Years of Excellence

MULTIPHASE COMPLETE EXCHANGE: A THEORETICAL ANALYSIS

Shahid H. Bokhari

NASA Contract Nos. NAS1-19480 and NAS1-18605
August 1993

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23681-0001

Operated by the Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-0001

Multiphase Complete Exchange: A Theoretical Analysis*

Shahid H. Bokhari

Department of Electrical Engineering

University of Engineering & Technology, Lahore, Pakistan

Abstract

Complete Exchange requires each of N processors to send a unique message to each of the remaining $N-1$ processors. For a circuit switched hypercube with $N = 2^d$ processors, the Direct and Standard algorithms for Complete Exchange are optimal for very large and very small message sizes, respectively. For intermediate sizes, a hybrid Multiphase algorithm is better. This carries out Direct exchanges on a set of subcubes whose dimensions are a partition of the integer d. The best such algorithm for a given message size m could hitherto only be found by enumerating all partitions of d.

The Multiphase algorithm is analyzed assuming a high performance communication network. It is proved that only algorithms corresponding to *equipartitions* of d (partitions in which the maximum and minimum elements differ by at most 1) can possibly be optimal. The run times of these algorithms plotted against m form a hull of optimality. It is proved that, although there is an exponential number of partitions, (1) the number of faces on this hull is $\Theta(\sqrt{d})$, (2) the hull can be found in $\Theta(\sqrt{d})$ time, and (3) once it has been found, the optimal algorithm for any given m can be found in $\Theta(\log d)$ time.

These results provide a very fast technique for minimizing communication overhead in many important applications, such as matrix transpose, Fast Fourier transform and ADI.

*Research supported by the National Aeronautics and Space Administration under NASA contracts NAS1-19480 and NAS1-18605 while the author was in residence at the Institute for Computer Applications in Science & Engineering, Mail Stop 132C, NASA Langley Research Center, Hampton, VA 23681-0001.

1 Introduction

On a distributed memory parallel computer, the complete exchange or all-to-all personalized communication pattern requires each of N processors to send a unique m-byte message to each of the remaining $N-1$ processors. This pattern arises in many important algorithms, such as matrix transpose, vector-matrix multiply, Fast Fourier transforms, etc. It is also of importance in its own right since it is the densest communication requirement that can be imposed on an interconnection network. The time required to carry out the complete exchange is, thus, a useful measure of the power of a parallel computer system. Finally, in many applications that require a dense communication pattern that is a subset of the complete exchange, it is usually beneficial to use a highly tuned complete exchange routine rather than attempting to write specific code for the required communication.

On circuit switched hypercubes, such as the Intel iPSC-860 and the nCUBE-2, there are two basic algorithms for obtaining the complete exchange. For a hypercube with $N = 2^d$ processors, the Standard exchange algorithm attempts to minimize the impact of startup time of a message by combining several messages into one 'super' message and using only $d = \log N$ message transmissions[11]. After each transmission, a shuffle step serves to route messages towards their correct destinations. This algorithm suffers from substantial overhead of data permutation.

The Direct algorithm uses $N-1$ carefully scheduled 'direct' transmissions, relying on knowledge of the routing algorithm used by the hardware to avoid message contention[14, 16, 17]. This algorithm has no data permutation overhead but suffers from $N-1$ message startups. It is demonstrable that the Standard exchange algorithm is best for very small message sizes, while the Direct algorithm requires minimum time for very large messages[3].

Multiphase complete exchange is a hybrid algorithm that combines the features of the Standard exchange and Direct algorithms. It carries out the complete exchange as a series of 'partial' exchanges on a set of subcubes[2, 4, 9, 10]. It permits a compromise between the message transmission and permutation overhead of Standard exchange and the message startups of the Direct algorithm.

The multiphase algorithm has been implemented and shown be useful on the iPSC-2 and iPSC-860 hypercubes. For a given hypercube dimension d, the number of possible multiphase algorithms equals the number of partitions

of the integer d. This is an exponential (though slowly growing) number and hitherto the only way to find the best multiphase algorithm for a given message size was to enumerate all these partitions.

In this paper we carry out a detailed analysis of the hull of optimality of all such multiphase algorithms. We make the assumption that the time to transmit a message from one processor to another is independent of the number of communication links traversed. This assumption is valid for most high-performance circuit-switched machines.

Our analysis reveals that only algorithms corresponding to *equipartitions* of d (partitions in which the largest and smallest elements differ by at most 1) can ever be optimal. Furthermore, the number of potentially optimal algorithms is always between $2\sqrt{d} - 1$ and $3\sqrt{d}$. We show that the hull of optimality can be found in $\Theta(\sqrt{d})$ time. Once the hull has been obtained, the optimal algorithm for a specific value of message size m can be found in $\Theta(\log d)$ time.

This result provides a very fast method of finding the optimal algorithm for a given message size and thus helps in reducing the communication overhead in a variety of important parallel applications. The $\Theta(\log d)$ time for finding the optimal algorithm is so fast that it may well be feasible to choose the algorithm during the course of program execution, based on the dimension of the hypercube and the size of the message currently being transmitted.

In Section 2 of this paper we discuss the complete exchange communication pattern and present the three algorithms. Section 3 contains our main analysis in which we present our notation, properties of equipartitions, main theorems, and obtain bounds on the number of faces on the hull. We conclude with a discussion of the ramifications of our results and suggestions for future research directions.

2 The Complete Exchange

Complete Exchange requires each of N processors of a parallel machine to send a different message to each of the remaining $N - 1$ processors. This pattern arises, for example, when transposing a matrix of $N \times N$ blocks that has been distributed over N processors, with one column per processor. The transpose requires each processor to send a different block to each of the remaining processors. The resulting communication pattern is equivalent to

the complete directed graph of N nodes.

The matrix mapping described above is required when using the Alternating Directions Implicit (ADI) method for solving partial differential equations [6, 13]. This method requires access to the matrix by rows and columns in successive phases, necessitating heavy use of a transpose. Matrix-matrix and matrix-vector multiplies have similar requirements. Complete exchanges are also required in many implementations of the parallel FFT.

The complete exchange, being equivalent to the complete directed graph, is the densest communication requirement that can be imposed on a network. The time required by the complete exchange is an upper bound on the time required by any other pattern and thus provides a useful measure of the power of a distributed memory parallel system.

2.1 Standard Exchange

The Standard exchange algorithm was presented by Johnsson & Ho[11] and uses $\log N$ transmissions of size $N/2$ blocks each. All communications are over single links, therefore no attention needs to be paid to the routing algorithm (in effect, the algorithm does the routing itself). The overheads in this algorithm are due to shuffling and the long message sizes that need to be transmitted. Despite this, the algorithm is competitive for small block sizes, since the total number of messages it transmits is $\log N$ as opposed to $N - 1$ for the Direct algorithm.

```
procedure Standard_Exchange;
begin
   for j = d − 1 downto 0 do
      begin
         if (bit j of mynumber = 0) then
            message = blocks n/2 to n − 1
         else
            message = blocks 0 to n/2 − 1;
         send_message_to_processor((mynumber) ⊕ (2^j));
         shuffle blocks;
      end;
end;
```

2.2 Direct Algorithm

The Direct algorithm was first reported (in Japanese) by Take [17] and later by Seidel et al.[14, 16]. In this algorithm each processor sends out $N-1$ messages, one to each of the remaining processors. The issue is to schedule the transmissions such that no edge contention takes place. Assuming the almost universal 'e-cube' routing algorithm, the exclusive-OR schedule described below achieves contention-free transmission. This algorithm always outperforms Standard Exchange for large message sizes.

```
procedure Direct;
begin
    for i = 1 to n − 1 do
        send_block_to_processor((mynumber) ⊕ (i));
end;
```

2.3 Multiphase Complete Exchange

The multiphase algorithm combines the Standard exchange and the Direct algorithms into one unified algorithm. It carries out the complete exchange as a sequence of two or more 'partial' exchanges. This algorithm has been implemented on the iPSC-2 and iPSC-860 [2, 4, 9, 10]. A complete exchange on a hypercube of dimension d with $n = 2^d$ processors and block size m is done using a set of partial exchanges $\mathcal{D} = \{d_1, d_2, \cdots, d_k\}$, on k subcubes, where each d_i specifies the dimension of the kth. subcube. Obviously $|\mathcal{D}| = k$, $1 \leq k$, and $\Sigma_{i=1}^{k} d_i = d$. Each partial exchange is called a *phase*.

The jth partial exchange is done on the set of subcubes determined by bits $\Sigma_{i=1}^{j} d_i - d_j$ to $\Sigma_{i=1}^{j} d_i$ of the hypercube node labels. In the partial exchange for the ith phase, 2^{d-d_i} blocks of m bytes each are transmitted, to each of $2^{d_i} - 1$ processors. The *effective block size* is thus $m2^{d-d_i}$.

```
procedure Multiphase;
{  d:     dimension of the hypercube
   n:     number of phases (subcubes) in partition D
   d_i:   dimension of the ith subcube in partition D
   start: starting bit of subcube label
   stop:  ending bit of subcube label }
```

```
begin
    start = d − 1;
    for i = 1 to n do
        {Partial exchange}
          begin
            stop = start − d_i + 1;
            compute effective blocksize;
            for j = 1 to (2^(start−stop+1) − 1) do
              send_effective_block_to_processor((mynumber) ⊕ (j2^stop));
            shuffle blocks d_i times;
            start = stop − 1;
          end;
end;
```

In the above algorithm, when $k = d$, all d_is are 1. In this case the outer i loop is executed k times with $start = stop = d-1, d-2, \cdots, 1, 0$. The inner j loop is executed only once for each i. In this case Multiphase degenerates into Standard exchange. When $k = 1$ and therefore $d_1 = d$, the outer loop is executed only once. *stop* always equals 0 and, in the inner loop, j takes on the values $1, 2, \cdots, 2^d - 1$ and thus Multiphase becomes Direct.

In our analysis, we have assumed that the complete exchange corresponds exactly to a transpose. Thus not only do blocks have to be transmitted among processors but each block needs to be placed in memory in the destination processor in its 'correct' transposed position. This accounts for the shuffle at the end of the last partial exchange. When there is only one phase, i.e. the algorithm corresponds to Direct exchange, the last set of d shuffles is equivalent to the identity permutation and is redundant. In the interest of simplicity, this has not been excluded from our analysis.

2.4 Implementation

A detailed evaluation of the performance of the Standard Exchange and Direct algorithms appears in [3]. The multiphase algorithm has been evaluated in [2, 4], wherein it has been shown that this approach can improve performance by as much as a factor of 2.

3 Analysis of the Multiphase Algorithm

The performance parameters characterizing a typical hypercube architecture are given in Table 1. τ is the time to transmit one byte while ρ the time to move a byte from one memory location to another, on the same processor. λ is the startup time, the time that elapses from issuance of a transmit request to initiation of transmission of the first byte. δ is the distance impact, that is the time required for a message to travel across the communication network of the processor. We assume this to be independent of the number of communication links traversed.

We omit the overhead of processor synchronization from our analysis. Each phase of our algorithm takes a precise amount of time. If all processors keep their clocks synchronized, there is no need for a global synchronization operation between phases, as the time to start a new phase can be computed by each processor independently. The issue of clock synchronization on hypercubes is discussed in [7].

Table 1: Performance parameters of a hypercube

	Description	Units
τ	transmission	time per byte
ρ	data permutation	time per byte
λ	startup (latency)	time per message
δ	distance impact	time per message

The time taken for a message of size m bytes is $\tau m + \lambda + \delta$. The Standard Exchange algorithm requires d transmissions of $m2^{d-1}$ bytes each, and d shuffles on 2^d blocks of m bytes. This leads to

$$\begin{aligned} t_{\text{standard}} &= d(\tau m 2^{d-1} + \lambda + \delta) + d\rho 2^d m \\ &= d\left[(\frac{\tau}{2} + \rho)2^d m + (\lambda + \delta)\right]. \end{aligned}$$

The Direct algorithm needs $2^d - 1$ transmissions of m bytes each, giving us

$$t_{\text{direct}} = (2^d - 1)\left[\tau m + (\lambda + \delta)\right].$$

A Multiphase algorithm, with n phases of size d_i each, requires for the ith phase $2^{d_i} - 1$ transmissions of $m2^{d-d_i}$ bytes each, followed by a permutation of 2^d bytes. Thus

$$\begin{aligned} t_{d,d_i} &= (2^{d_i} - 1)(\lambda + \tau m 2^{d-d_i} + \delta) + \rho m 2^d \\ &= \{(1 - \frac{1}{2^{d_i}})\tau + \rho\} 2^d m + (2^{d_i} - 1)(\lambda + \delta). \end{aligned} \quad (1)$$

Since $\sum_{i=1}^{n} d_i = d$, the total time required by the Multiphase algorithm is

$$\begin{aligned} t_{\text{multiphase}} &= \sum_{i=1}^{n} t_{d,d_i} \\ &= \sum_{i=1}^{n} \{(1 - \frac{1}{2^{d_i}})\tau + \rho\} 2^d m + (2^{d_i} - 1)(\lambda + \delta). \end{aligned} \quad (2)$$

3.1 Finding the best Multiphase algorithm

In our presentation of the multiphase algorithm, we have not stated which of the many possible partitions of the integer d is best in terms of total time. The total number of partitions of the integer d is approximated by: [1, 8]

$$p(d) \sim \frac{1}{4\sqrt{3}\,d} e^{\pi\sqrt{2/3}\sqrt{d}},$$

which is a slowly growing exponential, with $p(20) = 627$. It is feasible, though neither efficient nor elegant, to enumerate all partitions of d to find the best algorithm using the expression for $t_{\text{multiphase}}$ (2). Furthermore, $t_{\text{multiphase}}$ is not convex for $n = 2$. It is therefore not possible to find the best partition by recursively halving d.

The objective of this paper is to carry out a detailed investigation of the multiphase algorithm. We shall be concerned with the hull of optimality formed by the straight lines that describe the run times of all possible multiphase algorithms on a hypercube of dimension d plotted against the message size m. We shall show that a large class of algorithms can never be optimal. Of the remaining algorithms, a large fraction are optimal only at vertices of the hull of optimality and can be ignored. These results permit us to obtain a bound of $\Theta(\sqrt{d})$ on the number of optimal algorithms.

3.2 Notation

Let $[n_1, a_1][n_2, a_2][n_3, a_3]\cdots$ denote the sequence made up of n_1 a_1's, followed by n_2 a_2's, etc. Thus $[3,2][4,3][2,5]$ denotes the sequence $\{222333355\}$. The elements of a sequence shall always be enumerated in non-decreasing order.

Let the calligraphic letter $\mathcal{A}_{d,n}$ denote an arbitrary partition of the integer d with cardinality n. The elements of this partition are denoted by the lowercase letters a_i. We shall omit subscripts when they are irrelevant to the discussion. Example: two of many possible cardinality 6 partitions of the integer 30 are $\{224679\}$ and $\{115788\}$. Table 2 shows the partitions of $d = 5$. Define an *equipartition* of the integer d to be a partition in which

Table 2: Partitions of the integer 5.

				5
			1	4
			2	3
		1	1	3
		1	2	2
	1	1	1	2
1	1	1	1	1

the largest and smallest elements differ by at most 1. An equipartition of d with cardinality n is denoted $\mathcal{E}_{d,n}$. By definition, $\mathcal{E}_{d,1} = d$. In Table 2 the cardinality 3 equipartition is $\{122\}$.

It is straightforward to verify that

$$\mathcal{E}_{d,n} = \left[n - d \bmod n, \lfloor \frac{d}{n} \rfloor\right] \left[d \bmod n, \lceil \frac{d}{n} \rceil\right] \tag{3}$$

For example $\mathcal{E}_{19,8} = \{22222333\} = [5,2][3,3]$. Since the cardinality n equipartition of an integer d is unique, there are d unique equipartitions of the integer d.

The time taken by a set of partial exchanges corresponding to a partition of the integer $e \leq d$, $\mathcal{M}_{e,n} = \{m_1, m_2, \cdots, m_n\}$ on a dimension d hypercube

is

$$t_{d,\mathcal{M}_{e,n}} = \sum_{i=1}^{n} t_{d,m_i}. \tag{4}$$

In the case $e < d$, $\mathcal{M}_{e,n}$ is not a partition of d and the resultant data movement is not a complete exchange. Nevertheless this definition is important for subsequent analysis. When $e = d$, $\mathcal{M}_{e,n}$ is a partition of d, and the set of partial exchanges corresponding to $\mathcal{M}_{e,n}$ constitutes a multiphase algorithm for complete exchange, as described above. We shall use the terms 'algorithm' and 'partition' interchangeably, so that when we say 'time required by a partition', we mean the 'time required by a set of partial exchanges corresponding to that partition'.

Of particular interest to the ensuing discussion is the time required by an equipartition, which is obtained by combining (3) and (4):

$$t_{d,\mathcal{E}_{d,n}} = (n - d \bmod n)t_{d,\lfloor \frac{d}{n} \rfloor} + (d \bmod n)t_{d,\lceil \frac{d}{n} \rceil} \tag{5}$$

For a partition $\mathcal{A}_{a,n} = \{a_1, a_2, \cdots, a_n\}$, we have

$$\begin{aligned}
t_{d,\mathcal{A}_{a,n}} &= \sum_{i=1}^{n} t_{d,a_i} \\
&= \left[(1 - 2^{-a_1})\tau + \rho\right] 2^d m + (2^{a_1} - 1)(\lambda + \delta) + \\
&\quad \left[(1 - 2^{-a_2})\tau + \rho\right] 2^d m + (2^{a_2} - 1)(\lambda + \delta) + \\
&\quad \cdots \\
&\quad \left[(1 - 2^{-a_n})\tau + \rho\right] 2^d m + (2^{a_n} - 1)(\lambda + \delta) \\
&= \left[(n - 2^{-a_1} - 2^{-a_2} - \cdots - 2^{-a_n})\tau + n\rho\right] 2^d m + \\
&\quad (2^{a_1} + 2^{a_2} + \cdots + 2^{a_n})(\lambda + \delta)
\end{aligned}$$

This prompts us to define, for the partition $\mathcal{A}_{a,n}$

$$2^{\mathcal{A}_{a,n}} = 2^{a_1} + 2^{a_2} + \cdots + 2^{a_n}$$

and

$$2^{-\mathcal{A}_{a,n}} = 2^{-a_1} + 2^{-a_2} + \cdots + 2^{-a_n}$$

which then leads to the compact expression

$$t_{d,\mathcal{A}_{a,n}} = \left[(n - 2^{-\mathcal{A}_{a,n}})\tau + n\rho\right] 2^d m + (2^{\mathcal{A}_{a,n}} - n)(\lambda + \delta). \tag{6}$$

Since every element of $\mathcal{A}$ is at least 1, the coefficient of m in the above expression is > 0 as is the coefficient of $(\lambda + \delta)$. Thus when $t_{d,\mathcal{A}}$ is plotted against m we obtain a line with positive slope and intercept.

For an equipartition we have

$$t_{d,\mathcal{E}_{e,n}} = \left[(n - 2^{-\mathcal{E}_{e,n}})\tau + n\rho\right] 2^d m + (2^{\mathcal{E}_{e,n}} - n)(\lambda + \delta). \tag{7}$$

3.3 Properties of Equipartitions

Several properties of $2^{\mathcal{E}_{e,n}}$ and $2^{-\mathcal{E}_{e,n}}$ shall be useful in the ensuing discussion and are presented in this Section. In understanding these properties, it is useful to refer to Table 3 which lists $\mathcal{A}_{e,n}$, $2^{\mathcal{A}_{e,n}}$ and $2^{-\mathcal{A}_{e,n}}$ for all partitions of $e = 7$. The last column of this table indicates if an entry is an equipartition.

Table 3: Partitions of the integer $e = 7$.

n	$2^{\mathcal{A}_{7,n}}$	$2^{-\mathcal{A}_{7,n}}$	$\mathcal{A}_{7,n}$							
1	128	0.007812	7							$\mathcal{E}_{7,1}$
2	66	0.515625	1	6						
2	36	0.281250	2	5						
2	24	0.187500	3	4						$\mathcal{E}_{7,2}$
3	36	1.031250	1	1	5					
3	22	0.812500	1	2	4					
3	18	0.750000	1	3	3					
3	16	0.625000	2	2	3					$\mathcal{E}_{7,3}$
4	22	1.562500	1	1	1	4				
4	16	1.375000	1	1	2	3				
4	14	1.250000	1	2	2	2				$\mathcal{E}_{7,4}$
5	16	2.125000	1	1	1	1	3			
5	14	2.000000	1	1	1	2	2			$\mathcal{E}_{7,5}$
6	14	2.750000	1	1	1	1	1	2		$\mathcal{E}_{7,6}$
7	14	3.500000	1	1	1	1	1	1	1	$\mathcal{E}_{7,7}$

The first three properties arise from the theory of Schur-convexity[12] which we summarize as follows.

1. Given $\mathsf{X}, \mathsf{Y} \in \mathbb{R}^n$, with $\sum_{i=1}^n x_i = \sum_{i=1}^n y_i$. Let $x_{[i]}, y_{[i]}$ be the ith largest component of X, Y, respectively.

$$\text{We say } \mathsf{X} \prec \mathsf{Y} \text{ if } \sum_{i=1}^{j} x_{[i]} = \sum_{i=1}^{j} y_{[i]} \text{ for all } j = 1, 2, \cdots n.$$

2. $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$ is called Schur-convex if, whenever $\mathsf{X} \prec \mathsf{Y}$, then $\Phi(\mathsf{X}) \leq \Phi(\mathsf{Y})$.

3. If $g : \mathbb{R} \rightarrow \mathbb{R}$ is convex then $\Phi(\mathsf{X}) = \sum_{i=1}^n g(x_i)$ is Schur-convex. Examples of such functions are $g_1(x) = 2^x, g_2(x) = 1/2^x$.

Property 1 *For any* $1 \leq n < e$

(a) $2^{\mathcal{E}_{e,e}} \leq 2^{\mathcal{A}_{e,n}}$

(b) $2^{-e} = 2^{-\mathcal{E}_{e,1}} \leq 2^{-\mathcal{A}_{e,n}} \leq \frac{n}{2}$.

Property 2

(a) $2^{-\mathcal{E}_{d,2}} < 2^{-\mathcal{A}_{d,2}}$

(b) $2^{\mathcal{E}_{d,2}} < 2^{\mathcal{A}_{d,2}}$.

Property 3 $2^{\mathcal{E}_{e,n}} \leq 2^{\mathcal{E}_{e,n-1}}$.

Property 4 $2^{-\mathcal{E}_{e,n}} - 2^{-\mathcal{E}_{e,n-1}} \leq 3/4$.

Proof. $\mathcal{E}_{e,n-1}$ can always be obtained by deleting the smallest element of $\mathcal{E}_{e,n}$ and distributing it over the remaining elements of $\mathcal{E}_{e,n}$. Suppose

$$\mathcal{E}_{e,n} = \{e_1, e_2, \cdots, e_n\}$$

and that for some $k < n$,

$$e_1 = e_{1,1} + e_{1,2} + \cdots + e_{1,k}$$

all of which are greater than zero. Then

$$\begin{aligned} &2^{-\mathcal{E}_{e,n}} - 2^{-\mathcal{E}_{e,n-1}} = \\ &\quad 2^{-e_{1,1}-e_{1,2}-\cdots-e_{1,k}} + \\ &\quad 2^{-e_2}(1 - 2^{-e_{1,1}}) + 2^{-e_3}(1 - 2^{-e_{1,2}}) + \cdots 2^{-e_{k+1}}(1 - 2^{-e_{1,k}}) + \\ &\quad 2^{-e_{k+2}} + \cdots + 2^{-e_n}. \end{aligned}$$

This is a positive quantity that achieves a maximum when $k = 1$ and $e_{1,1} = 1$, in which case it is

$$2^{-1} + 2^{-1}(1 - 2^{-1}) = 3/4.$$

■

We have mentioned earlier that the time for a partition, when plotted against m, leads to a line with positive slope and intercept. The lines corresponding to the run times of equipartitions are of critical importance in this discussion.

Property 5 *Consider the straight lines corresponding to the two equipartitions $\mathcal{E}_{e,n}$ and $\mathcal{E}_{e,n-1}$, plotted against m. Then*
(1) $t_{d,\mathcal{E}_{e,n}}$ has greater slope than $t_{\mathcal{E}_{e,n-1}}$, and
(2) $t_{d,\mathcal{E}_{e,n}}$ has smaller intercept than $t_{\mathcal{E}_{e,n-1}}$.

Proof. We have from (7)

$$\begin{aligned} t_{d,\mathcal{E}_{e,n}} &= \left[(n - 2^{-\mathcal{E}_{e,n}})\tau + n\rho\right] 2^d m + (2^{\mathcal{E}_{e,n}} - n)(\lambda + \delta) \\ t_{d,\mathcal{E}_{e,n-1}} &= \left[(n - 1 - 2^{-\mathcal{E}_{e,n-1}})\tau + (n-1)\rho\right] 2^d m + (2^{\mathcal{E}_{e,n-1}} - n + 1)(\lambda + \delta) \end{aligned}$$

$$\begin{aligned} slope(t_{\mathcal{E}_{e,n}}) - slope(t_{\mathcal{E}_{e,n-1}}) &= 2^{-\mathcal{E}_{e,n-1}} - 2^{-\mathcal{E}_{e,n}} + 1 \\ &> 0 \quad \text{by Property 4} \\ intercept(t_{\mathcal{E}_{e,n}}) - intercept(t_{\mathcal{E}_{e,n-1}}) &= 2^{\mathcal{E}_{e,n}} - 2^{\mathcal{E}_{e,n-1}} - 1 \\ &< 0 \quad \text{by Property 3} \end{aligned}$$

■

The times taken by equipartitions thus form a hull in which the leftmost face corresponds to a partition with maximum cardinality, while the rightmost face corresponds to a partition of minimum cardinality. Faces of decreasing cardinality lie between these extreme faces. Figure 1(a) shows plots of the run times of *all* partitions (not necessarily equipartitions) of $d = 4$ on a hypercube of dimension 4. We can see that the hull of optimality is formed by equipartitions {1111},{22} and {4}. The non-equipartition {13} does not touch the hull. The equipartition {112} touches the hull but does not contribute a face (it passes through the point of intersection of {1111} and {22}). Figure 1(b) shows the times for all partitions of $d = 6$ on a hypercube of dimension 6. In this case the hull is formed by the equipartitions

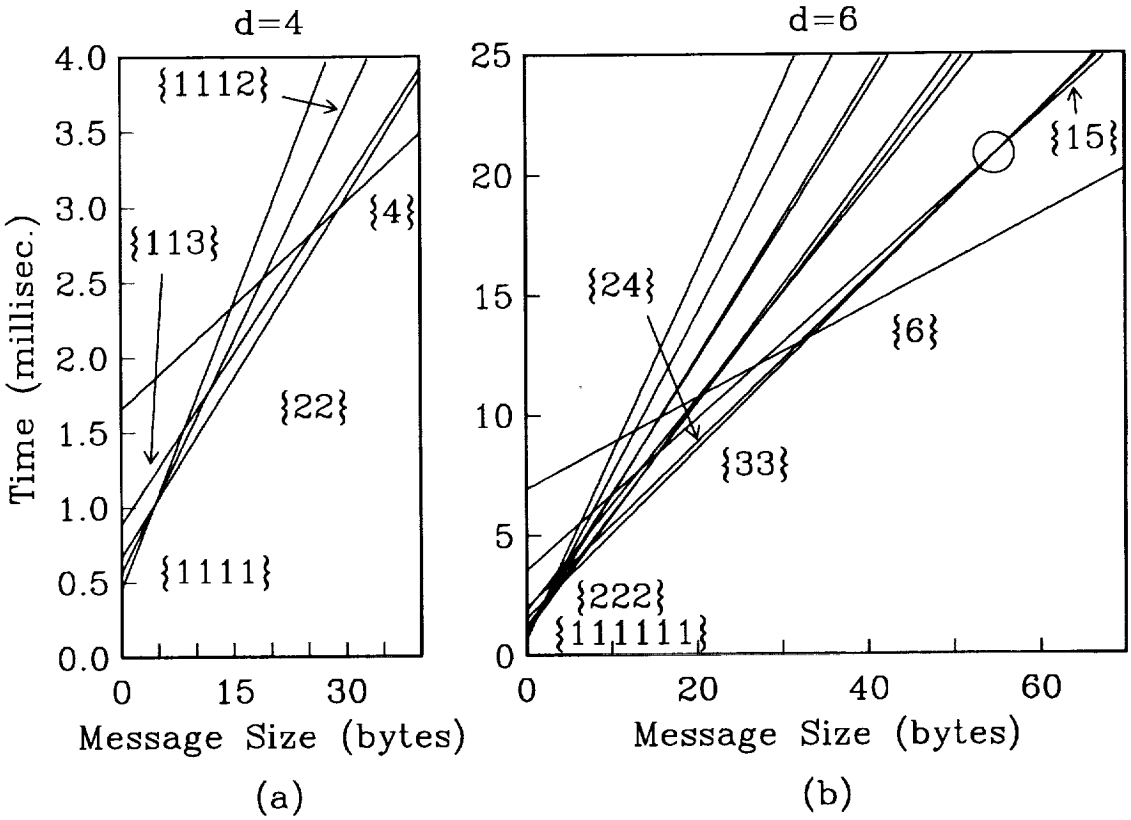

Figure 1: Run times for $d = 4, 6$. In this particular example, $\lambda = 100, \delta = 10$ (μsec.) and $\tau = 2, \rho = 1$ (μsec./byte). Circle indicates the point of intersection of all partitions of cardinality 2:{33},{24} and {15}

{111111}, {222}, {33} and {6}. Only a few of the remaining partitions are labeled to avoid a congested plot, but we can see that out of the 11 partitions of the integer 6, only the abovementioned 4 equipartitions contribute a face to the hull.

We now prove Properties 6 and 7 which are also illustrated in Figure 1.

Property 6 *For any d,*

(a) $\mathcal{E}_{d,1}$ *always lies on the hull, and*

(b) $\mathcal{E}_{d,d}$ *always lies on the hull.*

Proof. $\mathcal{A}_{d,n}$ represents an arbitrary partition of cardinality n. From (6) and (7) we have

$$\begin{aligned} t_{d,\mathcal{A}_{d,n}} &= \left[(n - 2^{\mathcal{A}_{d,n}})\tau + n\rho\right] 2^d m + (2^{\mathcal{A}_{d,n}} - n)(\lambda + \delta) \\ t_{d,\mathcal{E}_{d,1}} &= \left[(1 - 2^{\mathcal{E}_{d,1}})\tau - \rho\right] 2^d m + (2^{\mathcal{E}_{d,1}} - 1)(\lambda + \delta) \\ t_{d,\mathcal{E}_{d,d}} &= \left[(d - 2^{\mathcal{E}_{d,d}})\tau - d\rho\right] 2^d m + (2^{\mathcal{E}_{d,d}} - d)(\lambda + \delta) \end{aligned}$$

(a) The expression

$$\begin{aligned} & t_{d,\mathcal{A}_{d,n}} - t_{d,\mathcal{E}_{d,1}} \\ &= \left[(n - 1 - 2^{-\mathcal{A}_{d,n}} + 2^{-\mathcal{E}_{d,1}})\tau + (n-1)\rho)\right] + (2^{\mathcal{A}_{d,n}} - 2^{\mathcal{E}_{d,1}} - n + 1)(\lambda + \delta) \\ &\geq \left[(\frac{n}{2} + 2^{-d} - 1)\tau + (n-1)\rho\right] 2^d m + (2^{\mathcal{A}_{d,n}} - 2^{\mathcal{E}_{d,1}} - n + 1)(\lambda + \delta) \\ &\quad \text{(by Property 1(b))} \end{aligned}$$

which is always positive for sufficiently large m and $n > 1$ (for $n = 1, \mathcal{A}_{d,n} = \mathcal{E}_{d,1}$ and the property hold vacuously). Thus $\mathcal{E}_{d,1}$ lies below any $\mathcal{A}_{d,n}$ for sufficiently large m.

(b) At $m = 0$, the expression

$$t_{d,\mathcal{A}_{d,n}} - t_{d,\mathcal{E}_{d,d}} = (2^{\mathcal{A}_{d,n}} - 2^{\mathcal{E}_{d,d}} - n + d)(\lambda + \delta)$$

is greater than zero, since $2^{\mathcal{A}_{d,n}} > 2^{\mathcal{E}_{d,d}}$ (by Property 1(a)), and $d \geq n$). Thus, $\mathcal{E}_{d,d}$ lies below $\mathcal{A}_{d,n}$ for $m = 0$. ∎

The partition $\mathcal{E}_{d,1}$ corresponds to the Direct algorithm, while $\mathcal{E}_{d,d}$ is equivalent to the Standard exchange. These two algorithms are extreme cases of the Multiphase algorithm. Property 6 tells us that the Direct algorithm is always optimal for large values of m, while Standard exchange is always best for very small values.

Property 7 *Of all partitions of cardinality 2, only $\mathcal{E}_{d,2}$ can lie on the hull.*

Proof. Consider the two partitions $\mathcal{E}_{d,2} = \{e, d-e\}$ and $\mathcal{A}_{d,2} = \{a, d-a\}$. We have

$$\begin{aligned} t_{d,\mathcal{E}_{d,2}} &= \left[(2 - 2^{\mathcal{E}_{d,2}})\tau + 2\rho\right] 2^d m + (2^{\mathcal{E}_{d,2}})(\lambda + \delta) \\ t_{d,\mathcal{A}_{d,2}} &= \left[(2 - 2^{\mathcal{A}_{d,2}})\tau + 2\rho\right] 2^d m + (2^{\mathcal{A}_{d,2}})(\lambda + \delta) \end{aligned}$$

Solving for $t_{d,\mathcal{E}_{d,2}} = t_{d,\mathcal{A}_{d,2}}$ we obtain

$$\begin{aligned} m &= \frac{(2^{\mathcal{A}_{d,2}} - 2^{\mathcal{E}_{d,2}})(\lambda+\delta)}{(2^{-\mathcal{A}_{d,2}} - 2^{-\mathcal{E}_{d,2}})\tau 2^d} \\ &= \frac{(2^{\mathcal{A}_{d,2}} - 2^{\mathcal{E}_{d,2}})(\lambda+\delta)}{(2^{-a} + 2^{-d+a} - 2^{-e} - 2^{-d+e})\tau 2^d} \\ &= \frac{(2^{\mathcal{A}_{d,2}} - 2^{\mathcal{E}_{d,2}})(\lambda+\delta)}{(2^{d-a} + 2^a - 2^{d-e} - 2^e)\tau} \\ &= \frac{(2^{\mathcal{A}_{d,2}} - 2^{\mathcal{E}_{d,2}})(\lambda+\delta)}{(2^{\mathcal{A}_{d,2}} - 2^{\mathcal{E}_{d,2}})\tau} \\ &= \frac{\lambda+\delta}{\tau} \end{aligned}$$

which is independent of $\mathcal{E}_{d,2}$ and $\mathcal{A}_{d,2}$. Thus all partitions of cardinality 2 intersect at a point.

Since $2^{-\mathcal{E}_{d,2}} < 2^{-\mathcal{A}_{d,2}}$ and $2^{\mathcal{E}_{d,2}} < 2^{\mathcal{A}_{d,2}}$ (by Property 2), $t_{\mathcal{E}_{d,2}}$ has greater slope and lesser intercept than $t_{\mathcal{A}_{d,2}}$. Therefore only $t_{\mathcal{E}_{d,2}}$ can lie on the hull for $m < (\lambda+\delta)/\tau$.

At $m = (\lambda+\delta)/\tau$ we have

$$\begin{aligned} & t_{d,\mathcal{E}_{d,2}} - t_{d,\mathcal{E}_{d,1}} \\ &= \left[(1 - 2^{-\mathcal{E}_{d,2}} + 2^{-\mathcal{E}_{d,1}})\tau + \rho\right] 2^d \frac{(\lambda+\delta)}{\tau} + (2^{\mathcal{E}_{d,2}} - 2^{\mathcal{E}_{d,1}} - 1)(\lambda+\delta) \\ &= \left[(1 - 2^{-e} - 2^{-d-e} + 2^{-d})2^d + (2^e + 2^{d-e} - 2^d - 1) + \rho 2^d/\tau\right](\lambda+\delta) \\ &= \frac{\rho 2^d(\lambda+\delta)}{\tau} \end{aligned}$$

which is always positive. This means that the line $t_{\mathcal{E}_{d,1}}$ always passes *below* the common point of intersection of all cardinality 2 partitions. We have already shown that of all cardinality 2 partitions, only $\mathcal{E}_{d,2}$ can lie on the hull below this point. Hence of all cardinality 2 partitions only $\mathcal{E}_{d,2}$ can lie on the hull. ■

In the following, we shall prove that a non-equipartition cannot contribute a face to the hull of optimality and further that a large number of equipartitions can at most touch the hull at a vertex. Therefore, although there is an exponential number of partitions of an integer d, we shall prove that the number of faces on the hull of optimality is $\Theta(\sqrt{d})$.

3.4 Main Theorems

The properties proved above permit us to determine the maximum number of faces on the hull of optimality. Table 4 lists all partitions of the integers $1 \cdots 7$.

Table 4: Partitions of the integers $1 \cdots 7$.

$\mathcal{A}_{d,n}$																													
7							6							5					4				3			2		1	
						7						6					5				4				3			2	1
					[1]	6					1	5				1	4			1	3			1	2		1	1	
					2	5					2	4				2	3			2	2		1	1	1				
					3	4					3	3			1	1	3		1	1	2								
				[1]	1	5				1	1	4			1	2	2	1	1	1	1								
				[1]	2	4				1	2	3		1	1	1	2												
				[1]	3	3				2	2	2	1	1	1	1	1												
				2	2	3			1	1	1	3																	
			[1]	1	1	4			1	1	2	2																	
			[1]	1	2	3		1	1	1	1	2																	
			[1]	2	2	2	1	1	1	1	1	1																	
		[1]	1	1	1	3																							
		[1]	1	1	2	2																							
	[1]	1	1	1	1	2																							
[1]	1	1	1	1	1	1																							

Turning our attention to the partitions of 7, we see that if we select all those partitions that have a '1' in them (these are boxed in the table) and then delete a '1' from each of these, we obtain the partitions of 6, which are given in the next column. Similarly, selecting all partitions of 7 that have a '2' in them and then deleting a '2' from each of these will result in the partitions of the integer 5 and so on. It is thus clear that the set of all partitions of the integer d is composed of the union of the sets of all partitions of the integers $d - a$, $1 \leq a < d$, each augmented by a and the integer d. For

a specific partition we have

$$\mathcal{A}_{k,m} = \mathcal{A}_{k-a,m} + \{a\}$$

where we take the '+' operation to mean the addition of an element to a partition. The following property is evident.

Property 8 $t_{d,\mathcal{A}_{k,m}} = t_{d,\mathcal{A}_{k-a,m}} + t_{d,\{a\}}$.

It follows that the straight lines describing the run times of all partitions of an integer d can be obtained by adding $t_{d,\{a\}}$ to the run times of all partitions of the integer $d-a$, and then adding the line $t_{\mathcal{E}_{d,1}}$. This permits us to prove the following Theorem.

Theorem 1 *A non-equipartition cannot touch the hull of optimality.*

Proof. By induction on the partitions of integers $\leq d$.
Basis step: The smallest integer that has a non-equipartition is 4, which has only one: $\{13\}$. As the basis step of our induction, we shall prove that $t_{d,\{13\}}$ can never touch the hull.

The equations for the 5 partitions of 4 are, from (1),

$$\begin{aligned}
t_{d,\{4\}} &= 15\lambda + 15\sigma + 2^d m \left(\rho + \frac{15\tau}{16}\right) \\
t_{d,\{13\}} &= 8\lambda + 8\sigma + 2^d m \left(2\rho + \frac{11\tau}{8}\right) \\
t_{d,\{22\}} &= 6\lambda + 6\sigma + 2^d m \left(2\rho + \frac{3\tau}{2}\right) \\
t_{d,\{122\}} &= 5\lambda + 5\sigma + 2^d m \left(3\rho + \frac{7\tau}{4}\right) \\
t_{d,\{1111\}} &= 4\lambda + 4\sigma + 2^d m \left(4\rho + 2\tau\right)
\end{aligned}$$

The point of intersection of $t_{d,\{4\}}$ and $t_{d,\{22\}}$ is

$$m = \frac{144\ (\lambda + \sigma)}{2^d\ (16\rho + 9\tau)}.$$

At this value of m we have

$$t_{d,\{4\}} - t_{d,\{13\}} = \frac{-32\rho\ (\lambda + \sigma)}{16\rho + 9\tau}$$

which is always negative.

The lines $t_{d,\{22\}}$, $t_{d,\{122\}}$ and $t_{d,\{1111\}}$ intersect at a single point which occurs at

$$m = \frac{4\,(\lambda+\sigma)}{2^d\,(4\,\rho+\tau)}.$$

At this value of m we have

$$t_{d,\{1111\}} - t_{d,\{13\}} = \frac{-\left((\lambda+\sigma)\,(16\,\rho+3\,\tau)\right)}{2\,(4\,\rho+\tau)}$$

which is also always negative. Thus the partition $\{13\}$ can never touch the hull of optimality.

Induction: Suppose the theorem is true for all partitions of the integer $k < d$. Partitions of the integer $k+1$ can be obtained by adding $1, 2, \cdots, k$ to the partitions of the integers $k, k-1, \cdots, 1$, and then adding $\mathcal{E}_{k,1} = k$ as discussed above. The corresponding run times are obtained by adding $t_{d,1}, t_{d,2}, \cdots, t_{d,k}$ to run times of all the constituent partitions, as stated in Property 8. Each time we add $t_{d,a}$ to all the partitions of a certain integer we raise the hull of optimality and all other lines by a linear amount. The resultant hull of optimality of cardinality $k+1$ will be the intersection of the hulls of cardinality $1, 2, \cdots, k$. A line that did not touch one of the constituent hulls cannot touch the intersected hull.

When a partition is augmented, a new non-equipartition of cardinality k can be created by augmenting (1) a non-equipartition of cardinality j or (2) an equipartition of cardinality j. In the first case our hypothesis continues to hold since a non-equipartition not touching the hull is transformed into an non-equipartition that still does not touch the hull.

The second case requires careful analysis. When an existing equipartition of cardinality $j, \mathcal{E}_{d,j}$, that by hypothesis must touch the hull, is transformed into a non-equipartition $\mathcal{E}_{d,j} + \{k-j\}$ we have two possibilities

$k \geq 3$ Consider the partition obtained by deleting one of the original elements, $m \in \mathcal{E}_{d,j}$ from $\mathcal{E}_{d,j} + \{k-j\}$. This new partition *must* be a non-equipartition of cardinality $d-m$. In the hull for $d-m$ it could not have touched the hull, being 'masked' by equipartitions of cardinality $d-m$, and therefore it can now also not touch the hull after augmentation.

$k = 2$ In this case Property 7 states that only the equipartition of cardinality 2 can lie on the hull.

Therefore no non-equipartition can touch the new augmented hull of optimality for $k+1$. We have proved that if the theorem is true for k is is also true for $k+1$. We have already shown that it is true for $k = 2, 3, 4$. Thus it is true for all k. ■

An important consequence of Theorem 1 is the fact that even though there is an exponential number of partitions of d, the total number of faces on the hull of optimality cannot exceed d, the number of equipartitions. We shall continue with further investigations into the properties of equipartitions. These will permit us to improve the bound on the number of faces to $O(\sqrt{d})$. At this point we prove a theorem that shall permit us to place a lower bound on the number of faces on the hull.

Theorem 2 *Every equipartition must touch the hull of optimality.*

Proof. By induction on d.
Basis step: The theorem is true for $d = 2$, since by Property 6, both $\{11\}$ and $\{2\}$ must lie on the hull.
Induction: Assume the theorem is true for $d = n$. Then the hull of optimality is touched by all equipartitions $\mathcal{E}_{n,i}, 1 \leq i \leq n$. The set of equipartitions of the integer $n+1$, that is $\mathcal{E}_{n+1,i}$, can be formed from the set of equipartitions of the integer n, by adding 1 to the smallest element of each $\mathcal{E}_{n,i}$, and then adding the new equipartition $\mathcal{E}_{n+1,n+1}$.

Turning to the corresponding run times, this operation is equivalent to adding

$$t_{d,\{1\}} = (\frac{1}{2}\tau + \rho)2^d m + (\lambda + \delta)$$

to each of $t_{\mathcal{E}_{n,i}}, 1 \leq i \leq n$ (see equation (1)). Since the same linear expression is added to each $t_{\mathcal{E}_{n,i}}$, the relationships between these lines is undisturbed and the augmented hull is touched by all of the augmented equipartitions. Now consider $t_{\mathcal{E}_{n+1,n+1}}$; this must touch the hull because of Property 6(b). Thus the hull of optimality of the integer $n+1$ is touched by all equipartitions of $n+1$.

We have proved the theorem to be true for $d = 3$. We have shown that if it is true for $d = n$ it is also true for $d = n+1$. It is therefore true for all d. ■

An equipartition $\mathcal{E}_{d,n}$ can only have two distinct elements: $\lfloor d/n \rfloor$ and $\lceil d/n \rceil$. In some cases sequences of several different equipartitions have the same two distinct elements. For example, in Table 4, $\mathcal{E}_{6,6} = [6,1]$, $\mathcal{E}_{6,5} = [5,1][1,2]$, $\mathcal{E}_{6,4} = [2,1][2,2]$ and $\mathcal{E}_{6,3} = [3,2]$. All these partitions are composed of 1's and 2's exclusively. Similarly, the following equipartitions of the integer 19,

$$\begin{aligned} \mathcal{E}_{19,7} &= \{2233333\} \\ \mathcal{E}_{19,8} &= \{22222333\} \\ \mathcal{E}_{19,9} &= \{222222223\} \end{aligned}$$

are all composed of 2's and 3's exclusively. We call such equipartitions *indistinct*. It is clear that indistinct partitions always have successive cardinality values.

Theorem 3 *The run time functions of indistinct equipartitions are linearly dependent.*

Proof. Consider three indistinct equipartitions of cardinality p, $p+1$ and $p-1$ that are composed of the elements Ω and $\Omega+1$. Then for some α, β, γ

$$\begin{aligned} \mathcal{E}_{d,p} &= [\alpha, \ \Omega] \ [p-\alpha, \ \Omega+1] \\ \mathcal{E}_{d,p+1} &= [\beta, \ \Omega] \ [p+1-\beta, \ \Omega+1] \\ \mathcal{E}_{d,p-1} &= [\gamma, \ \Omega] \ [p-1-\gamma, \ \Omega+1] \end{aligned}$$

The times for these equipartitions are

$$t_{\mathcal{E}_{d,p}} = \alpha t_{d,\Omega} + (p-\alpha) t_{d,\Omega+1} \tag{8}$$

$$t_{\mathcal{E}_{d,p+1}} = \beta t_{d,\Omega} + (p+1-\beta) t_{d,\Omega+1} \tag{9}$$

$$t_{\mathcal{E}_{d,p-1}} = \gamma t_{d,\Omega} + (p-1-\gamma) t_{d,\Omega+1} \tag{10}$$

Since we are dealing with equipartitions of the integer d,

$$\begin{aligned} d &= \alpha\Omega + (p-\alpha)(\Omega+1) \\ &= \beta\Omega + (p+1-\beta)(\Omega+1) \\ &= \gamma\Omega + (p-1-\gamma)(\Omega+1) \end{aligned}$$

These yield the following relations

$$\beta = 1 + \alpha + \Omega \tag{11}$$

$$\gamma = -1 + \alpha - \Omega \tag{12}$$

Substituting (11) and (12) in (9) and (10) we obtain the system

$$t_{\mathcal{E}_{d,p}} = \alpha t_{d,\Omega} + (p-\alpha)t_{d,\Omega+1} \quad (13)$$

$$t_{\mathcal{E}_{d,p+1}} = (1+\alpha+\Omega)t_{d,\Omega} + (p+\alpha-\Omega)t_{d,\Omega+1} \quad (14)$$

$$t_{\mathcal{E}_{d,p-1}} = (-1+\alpha-\Omega)t_{d,\Omega} + (p-\alpha+\Omega)t_{d,\Omega+1} \quad (15)$$

Adding (14) and (15) we obtain

$$\begin{aligned} t_{\mathcal{E}_{d,p+1}} + t_{\mathcal{E}_{d,p-1}} &= 2\alpha t_{d,\Omega} + 2(p-\alpha)t_{d,\Omega+1} \\ &= 2t_{\mathcal{E}_{d,p}} \end{aligned}$$

Hence the system is linearly dependent. ∎

Theorem 3 assures us that all members of a set of indistinct equipartitions intersect at a single point. Therefore only two of these can contribute faces to the hull of optimality, since they have successively decreasing slopes and increasing intercepts (Property 5). For example, in the hull for $d = 4$ (Figure 1), we can see that the equipartitions $\{1111\}$, $\{112\}$ and $\{22\}$ intersect at a point and only $\{1111\}$ and $\{22\}$ contribute faces to the hull.

3.5 Faces on the Hull

From the foregoing discussion we can see that all equipartitions touch the hull. Each distinct equipartition contributes a single face to the hull while each set of indistinct equipartitions contributes two faces. To find a bound on the number of faces on the hull, refer to Figure 2 which plots $\lfloor d/n \rfloor$, and $\lceil d/n \rceil$ versus n for $d = 11$ and 16. In each of these plots, the dashed curve represents the the continuous function d/n. The values of $\lfloor d/n \rfloor$, and $\lceil d/n \rceil$ are indicated by heavy dots. When $\lfloor d/n \rfloor < \lceil d/n \rceil$, there is a vertical line joining these dots. In the plot for $d = 11$ we have enumerated all the equipartitions in full, while in the plot for $d = 16$ we have used the compact notation (3). The lines marked with '+'s are tangents, with slope -1, at the point $\lfloor\sqrt{d}\rfloor, \lfloor\sqrt{d}\rfloor$.

Over the range $1 \leq n \leq \sqrt{d}$ the slope of these hyperbolas is less than -1 and therefore no two consecutive equipartitions can have an element in common. All equipartitions in this range are distinct and their number is equal to the number of integers in this range, which is $\lfloor\sqrt{d}\rfloor$. This equals the number of '+'s on the tangent between $n = 1$ and $n = \lfloor\sqrt{d}\rfloor$.

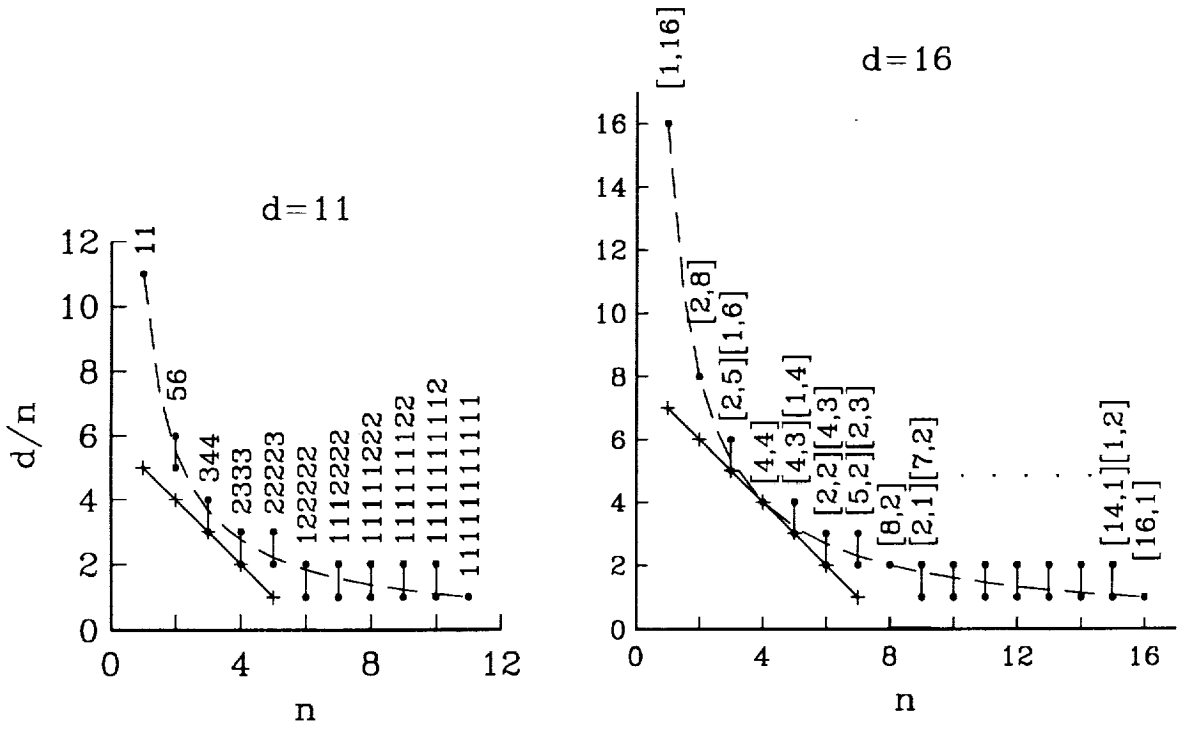

Figure 2: Plots of $\lfloor d/n \rfloor$, $\lceil d/n \rceil$ for $d = 11$ and 16.

Indistinct equipartitions can only occur over the range $\sqrt{d} < n \leq d$. The number sets of in distinct equipartitions is no more than the number of distinct values of $\lfloor d/n \rfloor$, which is the number of '+'s on the tangent between $\sqrt{d}$ and d, and is again $\lfloor \sqrt{d} \rfloor$.

In the range $1 \leq n \leq \sqrt{d}$, there are no indistinct equipartitions, so one face is contributed to the hull by each equipartition, giving us a total of $\lfloor \sqrt{d} \rfloor$ faces. In the range $\sqrt{d} \leq n \leq d$ there may be up to $\lfloor \sqrt{d} \rfloor$ sets of indistinct equipartitions, each contributing at most 2 faces and at least one face to the hull. An upper bound on the total number of faces on the hull is therefore $3\lfloor \sqrt{d} \rfloor$. To obtain a lower bound note that the hyperbola is symmetric about the line $n = d/n$ (the line through the origin with slope 1). If the point

$\lfloor\sqrt{d}\rfloor, \lfloor\sqrt{d}\rfloor$ lies on this line the number of distinct levels is $2\lfloor\sqrt{d}\rfloor - 1$ and is $2\lfloor\sqrt{d}\rfloor$ otherwise. Each level must contribute at least one face to the hull. Thus the lower bound on the number of faces is $2\lfloor\sqrt{d}\rfloor - 1$.

The equipartitions that contribute to the hull can be found by visiting all $\Theta(\sqrt{d})$ points on the tangent. For $1 \leq n \leq \lfloor\sqrt{d}\rfloor$, each point corresponds to the equipartition $\mathcal{E}_{d,n}$. For each n in the range $\lfloor\sqrt{d}\rfloor \cdots 2\lfloor\sqrt{d}\rfloor$ there is a sequence of indistinct equipartitions extending from $\lceil d/(n+1)\rceil$ to $\lfloor d/n\rfloor$. We need consider only the first and last members of these sequences. Thus all partitions contributing to the hull can be found in $\Theta(\sqrt{d})$ time. Once these partitions have been found, the vertices of the hull can be discovered by computing the intersection points of adjacent partitions, again in $\Theta(\sqrt{d})$ time. The intersection points will be computed in order and, once they have been stored, the optimal algorithm for any value of m can be found using a binary search in $\Theta(\log d)$ time.

4 Conclusions

We have analyzed the multiphase complete exchange algorithm and shown that the total number of optimal algorithms lies between $2\sqrt{d} - 1$ and $3\sqrt{d}$. This holds under the assumption that the time for transmitting a message is independent of the number of communication links traversed. High performance parallel machines satisfy this assumption.

In addition to its theoretical interest, this result is of considerable practical importance. It allows us to compute the optimal algorithm for any given values of hypercube performance parameters and message length very quickly. When dealing with an application where the performance parameters (Table 1) are fixed and the message lengths for complete exchange vary from time to time, the values of message length m at which vertices of the hull of optimality occur can be computed ahead of time and stored in a sorted list. During the course of program execution, a fast binary search will locate the optimal algorithm for the current message size.

When the performance parameters vary with time, as would happen if the communication network were shared among several subcubes, our results provide a fast method for computing the optimal algorithm from scratch. A related situation is where the same application is run on hypercubes of different sizes.

Among the future directions of this research, the foremost issue is an extension to 2 and 3 dimensional meshes. Preliminary results on 2–dimensional meshes appear in [5]. Since the time required for 'direct' complete exchanges on N-processor 2 and 3-d meshes is $O(N^{3/2})$ and $O(N^{4/3})$ respectively[15], compared to the hypercube's $O(N)$, any improvements will be especially welcome.

Acknowledgements

I am grateful to David Nicol for many useful discussions and for pointing out that Properties 1–3 are a direct consequence of Schur-convexity. I thank Piyush Mehrotra for valuable comments on a draft of the manuscript and Gordon Erlebacher for help with Mathematica.

References

[1] G. E. Andrews. *The theory of partitions.* Addison-Wesley, Reading, Massachusetts, 1976.

[2] S. H. Bokhari. Multiphase complete exchange on a circuit switched hypercube. In *Proc. 1991 International Conf. Parallel Processing*, pages 525–529, 1991.

[3] S. H. Bokhari. Complete exchange on the Intel iPSC-860 hypercube. ICASE Report No. 91-4, NASA CR-187498.

[4] S. H. Bokhari. Multiphase complete exchange on a circuit switched hypercube. ICASE Report No. 91-5, NASA CR-187499.

[5] S. H. Bokhari and S. Berryman. Complete exchange on a circuit switched mesh. In *Proc. Scalable High Performance Computing Conf.*, pages 300–306, 1992.

[6] J. Douglas and J. E. Gunn. A general formulation of alternating direction methods. *Numer. Math.*, 6(5), 1964.

[7] T. Dunigan. Hypercube clock synchronization. *Concurrency: Practice and Experience*, 4(3):257–268, May 1992.

[8] E. Grosswald. *Topics from the Theory of numbers.* Birkhäuser, Boston, 1984.

[9] C-T. Ho and M. T. Raghunath. Efficient communication primitives on hypercubes. In *Proc. 6th. DMCC*, pages 390–397, 1991.

[10] C-T. Ho and M. T. Raghunath. Efficient communication primitives on hypercubes. Technical Report RJ 7932 (72915), IBM, T. J. Watson Center, January 1991.

[11] S. L. Johnsson and C-T. Ho. Matrix transposition on boolean n-cube configured ensemble architectures. *SIAM J. Matrix Anal. Appl.*, 9(3):419–454, July 1988.

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

[13] D. W. Peaceman and H. H. Rachford. The numerical solution of parabolic and elliptic differential equations. *SIAM J.*, 3(1), 1955.

[14] T. Schmiermund and S. R. Seidel. A communication model for the Intel iPSC/2. Technical Report CS-TR 9002, Dept. of Computer Science, Michigan Tech. Univ., April 1990.

[15] D. S. Scott. Efficient all-to-all communication patterns in hypercube and mesh topologies. In *Proc. 6th. Conf. Distributed Memory Concurrent Computers*, pages 398–403, 1991.

[16] S. R. Seidel. Circuit switched vs. store-and-forward solutions to symmetric communication problems. In *Proc. 4th. Conf. Hypercube Concurrent Computers and Applications*, pages 253–255, 1989.

[17] R. Take. A routing method for the all-to-all burst on hypercube network. In *Proc. 35th. National Conf. Info. Proc. Soc. Japan*, pages 151–152, 1987. In Japanese.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1993	Contractor Report

4. TITLE AND SUBTITLE
MULTIPHASE COMPLETE EXCHANGE: A THEORETICAL ANALYSIS

5. FUNDING NUMBERS
C NAS1-19480
C NAS1-18605
WU 505-90-52-01

6. AUTHOR(S)
Shahid H. Bokhari

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Computer Applications in Science and Engineering
Mail Stop 132C, NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER
ICASE Report No. 93-64

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-191531
ICASE Report No. 93-64

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Michael F. Card
Final Report
Submitted to IEEE Transactions on Computers

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited

Subject Category 60, 61

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Complete Exchange requires each of N processors to send a unique message to each of the remaining $N-1$ processors. For a circuit switched hypercube with $N = 2^d$ processors, the Direct and Standard algorithms for Complete Exchange are optimal for very large and very small message sizes, respectively. For intermediate sizes, a hybrid Multiphase algorithm is better. This carries out Direct exchanges on a set of subcubes whose dimensions are a partition of the integer d. The best such algorithm for a given message size m could hitherto only be found by enumerating all partitions of d.

The Multiphase algorithm is analyzed assuming a high performance communication network. It is proved that only algorithms corresponding to *equipartitions* of d (partitions in which the maximum and minimum elements differ by at most 1) can possibly be optimal. The run times of these algorithms plotted against m form a hull of optimality. It is proved that, although there is an exponential number of partitions, (1) the number of faces on this hull is $\Theta(\sqrt{d})$, (2) the hull can be found in $\Theta(\sqrt{d})$ time, and (3) once it has been found, the optimal algorithm for any given m can be found in $\Theta(\log d)$ time.

These results provide a very fast technique for minimizing communication overhead in many important applications, such as matrix transpose, Fast Fourier transform and ADI.

14. SUBJECT TERMS
all-to-all personalized, circuit switching, complete exchange, communication overhead, hypercube, multiphase algorithm, parallel computing

15. NUMBER OF PAGES
27

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

☆U.S. GOVERNMENT PRINTING OFFICE: 1993 - 728-064/86063

National Aeronautics and
Space Administration
Code JTT
Washington, D.C.
20546-0001

Official Business

Penalty for Private Use, $300

BULK RATE
POSTAGE & FEES PAID
NASA
Permit No. G-27

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return