Two New Quorum Based Algorithms for Distributed Mutual Exclusion

Wai-Shing Luk

Departement Computerwetenschappen
Katholieke Universiteit Leuven
Leuven, Belgium

Tien-Tsin Wong

Dept. of Computer Science & Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong

Abstract

Two novel suboptimal algorithms for mutual exclusion in distributed systems are presented. One is based on the modification of Maekawa's grid based quorum scheme. The size of quorums is approximately $\sqrt{2}\sqrt{N}$ where N is the number of sites in a network, as compared to $2\sqrt{N}$ of the original method. The method is simple and geometrically evident. The second one is based on the idea of difference sets in combinatorial theory. The resulting scheme is very close to optimal in terms of quorum size.

1 Introduction

Consider a communication network that contains a set of N sites $\{P_1, P_2, \ldots, P_N\}$. The problem of mutual exclusion arises in distributed systems whenever concurrent access to shared resources by several sites is involved [1]. For recent developments of distributed mutual exclusion algorithms, see the survey by Chang [2]. Among many algorithms proposed, Maekawa's algorithm is one of the well-known methods in the literature [3]. In this algorithm, one of the considerations is to design a group of subsets $\{S_1, S_2, \ldots, S_N\}$ that satisfies the following properties:

A1. P_i is contained in S_i, for all $i \in 1, 2, \ldots, N$;

A2. (*Non-empty intersection property*)
$S_i \cap S_j \neq \emptyset$, for all $i, j \in 1, 2, \ldots, N$;

A3. (*Equal work property*)
$|S_i| = k$, for all $i \in 1, 2, \ldots, N$; where k is an integer less than N.

A4. (*Equal responsibility property*)
P_i is contained in k S_j's, for all $i \in 1, 2, \ldots, N$.

The set of sites S_i is called *quorum*. P_i is referred to as the *requesting site* of S_i. When P_i wants to get a shared resource, it needs only to send requests to the sites in S_i. Property A2 ensures that any two quorums have a non-empty intersection, thus every pair of quorums has at least one common site that mediates conflicts between the pair. Properties A3 and A4 enforce that all sites perform an equal amount of work. A set of quorums is called *symmetric* if the two properties are both satisfied. Since the communication cost is proportional to the size of quorums k, the research problem is to minimize k while the four properties are retained. For example when $N = 6$, one of the possible solutions is:

$$\begin{aligned} S_1 &= \{P_1, P_2, P_4\}, \\ S_2 &= \{P_2, P_3, P_5\}, \\ S_3 &= \{P_3, P_4, P_6\}, \\ S_4 &= \{P_1, P_4, P_5\}, \\ S_5 &= \{P_2, P_5, P_6\}, \\ S_6 &= \{P_1, P_3, P_6\}. \end{aligned}$$

In this example, k is equal to 3, which is the minimal value. Maekawa showed that for a fixed k, the maximal possible value of N in which all the properties can be satisfied is equal to $k(k-1)+1$, by assuming that any two quorums have only one intersection site [3]. Hence the theoretical lower bound of the quorum size is approximately equal to $\sqrt{N}$. Furthermore, he also mentioned that finding the possible solution for $N = k(k-1)+1$ is equivalent to finding a *finite projective plane* of order n, where $n = k-1$. Nevertheless, not all finite projective planes exist and we only know how to construct those with prime power order $n = p^i$, where p is a prime number and i is an integer [4].

In theory, we can solve the optimal quorum problem by testing all the combinations and see which one satisfies all the properties. However, the time complexity of this exhaustive search method is exponential and it is intractable even with fastest computer in the world. One example is that in 1988, the non-existence of fi-

0-8186-7813-5/97 $10.00

nite projective plane of order 10 was proven with the assistance of more than 3,000 hours of computation on a supercomputer by the exhaustive search [5].

For this reason, several schemes have been proposed to generate the near-optimal solution. We will review these algorithms in the next section. The rest of this paper is organized as follows. In Section 3, Maekawa's grid based scheme is reviewd and a modification of this scheme is proposed. Section 4 contains another novel sub-optimal scheme which is based on the idea of difference sets in combinatorial theory. In Section 5, we discuss the possibility of hybrid use of those proposed schemes.

Note that the problem described above is an idealized mathematical problem. In practice, more factors, such as deadlock and starvation, should be taken into consideration for a distributed system. Note also that the original Maekawa's algorithm is prone to deadlocks [1] and the description of handling this problem can be found in [1]. Another application of Maekawa's algorithm is in task migration [6].

2 Previous works

In Maekawa's original paper, he suggested two suboptimal schemes. The first method inserts certain amount of "virtual sites" such that the corresponding finite projective plane exists. However, the resulting set of quorums is non-symmetric. The second method avoids the construction of finite projective planes and is classified as *grid based scheme.* The sites are logically organized in a grid in a shape of *square* (Figure 1). A quorum for a requesting site includes the union of the row and the column that the requesting site corresponds to. Therefore, the quorum size is roughly twice of the theoretical lower bound, i.e. $k = 2\sqrt{N}$. The advantage of this algorithm is that it is simple and geometrically evident, which means the four properties can be obviously verified graphically. However, this algorithm is not well-optimized in the sense that S_i intersects with S_j in two sites for all $i \neq j$.

Agrawal *et. al.* proposed an alternative grid based method which brings down the quorum size $2\sqrt{N}$ to $\sqrt{2}\sqrt{N}$ [7]. The idea is to resemble billiard ball paths on a modified grid. However, as mentioned by the authors, the method does not satisfy the equal responsibility property. Also, the size of each quorum has to be an odd integer.

In case that the symmetry is a criticism, Lien and Yuan proposed a scheme whose quorum size is $2\lceil\sqrt{N}\rceil - 1$ [8]. Ng and Ravishankar proposed another suboptimal scheme that also satisfies all the properties strictly with the quorum size in $O(N^{0.63})$ [9].

In this paper, we present two suboptimal algorithms. The first one is obtained by logically reorganizing the grid in a shape of ***triangle***. The size of quorums is approximately $\sqrt{2}\sqrt{N}$. In the second method, we construct a scheme which is based on the ideas of *cyclic difference set* and *cyclic block design* in combinatorial theory [10]. The method works for arbitrary N and the quorum set is strictly symmetric. Moreover, the resulting quorum size is close to the theoretical lower bound.

3 Grid based algorithms

Before we describe our approach, let us first review Maekawa's grid based algorithm. Assume that N is a perfect square integer. In Maekawa's approach, the sites are organized in a grid in a shape of ***square*** as shown in Figure 1. The node in black color in this figure represents the requesting site P_i. The quorum S_i for the requesting site P_i includes the union of the row and the column that P_i corresponds to. The sites inside the quorum S_i are denoted in gray color. It is easy to see that the configuration satisfies all the properties described in Section 1. This method is simple and geometrically evident. However, the quorum size is $2\sqrt{N}-1$, roughly twice of the theoretical lower bound. Note that if N is not a square integer, Properties 3 and 4 have to be slightly relaxed.

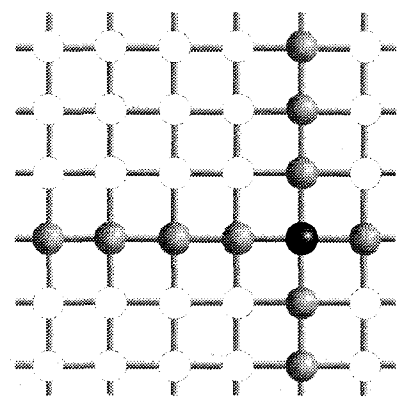

Figure 1: The "square" configuration.

As mentioned in Section 1, the method is not well-optimized in the sense that any two quorums have two intersections. Therefore, it is better if we can make each quorum contains only either a row or a column. In fact, it can be done by re-organizing the sites in a "triangle" shape as shown in Figure 2.

Figure 2 illustrates the configuration for $N = 21$. A quorum is constructed in the following manner. We start to draw a line from the leftmost node on the first row and go horizontally to the right. Any node joined by this line is included in the quorum. When there is

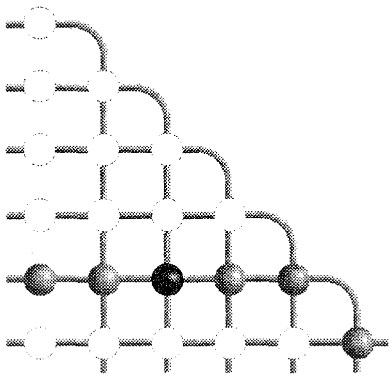

Figure 2: The "triangle" configuration with a "row" illustrated.

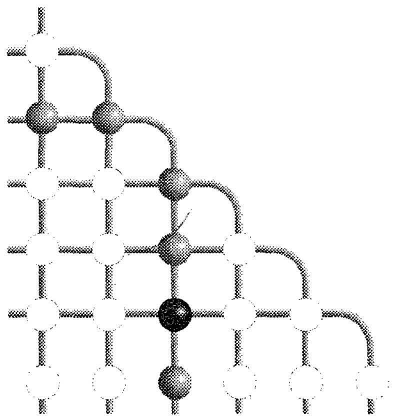

Figure 3: The "triangle" configuration with a "column" illustrated.

no more node to join on the right, the line is turned 90 degree to the bottom. It stops whenever it reaches the bottommost node. All nodes joined by this line are included in one quorum denoted by R. Similarly, we can construct other quorum by starting the horizontal line drawing on different row. The requesting site P_i should send request to the quorum R_i, which refers to the line that horizontally passes through P_i. We called this construction scheme the *row based* construction because we start the line horizontally along the row first. Another construction scheme (Figure 3) which we called *column based* construction is quite similar. This time we start the line from the bottommost node on one column and go vertically upward. It turns 90 degree to the left whenever no more node on its path and stops when it reaches the leftmost node. All nodes joined by this line are included in a quorum denoted by C. This time P_i should send request to the quorum C_i, which refers to the line that vertically passes through P_i.

Any two different lines meet at exactly one node. Thus, any two quorums have exactly one intersection site. Moreover, all quorums have the same size. However if the row based scheme is chosen, the sites near the top of the grid will be contained in fewer quorums and thus the equal responsibility property cannot be satisfied. Similarly, the unbalance occurs in a column based scheme. Hence, we propose alternating use the row based scheme and the column based scheme in order to retain the fairness. For example, P_i requests for shared resources T_1, T_2 and T_3. It should send request to the sites in the row based quorum for T_1. When it requests for T_2, it should send request to the column based quorum. And it send request to row based quorum for T_3. This means that each requesting site can have two quorums. This slightly modifies the originally problem setting that allows only one quorum in each requesting site. Take an example of $N = 10$ and the sites are organized as follows:

$$\begin{array}{llll} P_1 & & & \\ P_2 & P_3 & & \\ P_4 & P_5 & P_6 & \\ P_7 & P_8 & P_9 & P_{10} \end{array}$$

Let C_i and R_i denote the column quorum and the row quorum of P_i respectively. The configuration of our scheme will be:

$$\begin{array}{ll}
C_1 = \{P_1, P_2, P_4, P_7\}, & R_1 = \{P_1, P_3, P_5, P_8\}, \\
C_2 = \{P_1, P_2, P_4, P_7\}, & R_2 = \{P_2, P_3, P_6, P_9\}, \\
C_3 = \{P_1, P_3, P_5, P_8\}, & R_3 = \{P_2, P_3, P_6, P_9\}, \\
C_4 = \{P_1, P_2, P_4, P_7\}, & R_4 = \{P_4, P_5, P_6, P_{10}\}, \\
C_5 = \{P_1, P_3, P_5, P_8\}, & R_5 = \{P_4, P_5, P_6, P_{10}\}, \\
C_6 = \{P_2, P_3, P_6, P_9\}, & R_6 = \{P_4, P_5, P_6, P_{10}\}, \\
C_7 = \{P_1, P_2, P_4, P_7\}, & R_7 = \{P_7, P_8, P_9, P_{10}\}, \\
C_8 = \{P_1, P_3, P_5, P_8\}, & R_8 = \{P_7, P_8, P_9, P_{10}\}, \\
C_9 = \{P_2, P_3, P_6, P_9\}, & R_9 = \{P_7, P_8, P_9, P_{10}\}, \\
C_{10} = \{P_4, P_5, P_6, P_{10}\}, & R_{10} = \{P_7, P_8, P_9, P_{10}\}.
\end{array}$$

Each requesting site is contained in exactly eight quorums (i.e., $2k$) and therefore the equal responsibility property is retained. Note that the requesting site does not need to notify other sites that which quorum (row or column) it is using and therefore no extra communication overhead is induced. For a fixed quorum size k, the maximum number of sites that this approach can handled is $k(k+1)/2$. Hence the quorum size is approximately $\sqrt{2}\sqrt{N}$.

This algorithm is simple and geometrically evident.

4 The "cyclic" quorum scheme

In this section, we propose another scheme which is based on the ideas of cyclic block design and cyclic difference sets in combinatorial theory [10]. The solution set can be strictly symmetric for arbitrary N. First of all, we define some concepts which help us to prove that the proposed algorithm satisfy all four properties A1-A4.

Definition 1 *A set $D : \{a_1, \ldots, a_k\}$ modulo N, $a_i \in 0, \ldots, N-1$, is called a cyclic (N, k, λ)-difference set if for every $d \not\equiv 0 \pmod{N}$ there are exactly λ ordered pairs (a_i, a_j), $a_i, a_j \in D$ such that $a_i - a_j \equiv d \pmod{N}$.*

For example, the set {0,1,2,4,5,8,10} modulo 15 is a cyclic (15,7,3)-difference set. The cyclic $(n^2+n+1, n+1, 1)$-difference sets of prime power order n are related to the finite projective planes of the same order, and can be constructed efficiently by constructing the Singer difference sets (see [11, p. 299]). Now we are ready to describe our method using the similar idea. For convenience of discussion, *we now simply use the index $i-1$ to represent the site P_i*. In our scheme, we restrict ourself to searching for the optimal solution within cyclic quorum sets defined as follows,

Definition 2 *A group of cyclic quorums is a group of sets $\{B_0, B_1, \ldots, B_{N-1}\}$ which satisfies the following properties:*

B1. *i is contained in B_i, for all $i \in 0, 1, \ldots, N-1$;*

B2. *$B_i \cap B_j \neq \emptyset$, for all $i, j \in 0, 1, \ldots, N-1$;*

B3. *$B_i = \{a_1 + i, a_2 + i, \ldots, a_k + i\}$ modulo N.*

Properties B1 and B2 are same as properties A1 and A2. Note that the last property automatically implies Properties A3 and A4 in Section 1. In other words, a group of cyclic quorums satisfies all four properties A1-A4 and can be used immediately as a solution for a specific mutual exclusion problem. For example if $B_0 = \{0, 1, 2, 4\}$ and $N = 8$, then

$$\begin{aligned} B_1 &= \{1, 2, 3, 5\}, \\ B_2 &= \{2, 3, 4, 6\}, \\ B_3 &= \{3, 4, 5, 7\}, \\ B_4 &= \{4, 5, 6, 0\}, \\ B_5 &= \{5, 6, 7, 1\}, \\ B_6 &= \{6, 7, 0, 2\}, \\ B_7 &= \{7, 0, 1, 3\}. \end{aligned}$$

The definition above is similar to the cyclic block design defined in combinatorial theory [10]. In fact, certain block designs can be solutions of certain quorum sets. The major difference between them, however, is that the non-empty intersection property in quorum set is not requested in block designs. Similar to the definition of difference set in combinatorial theory, we define the relaxed difference set as follows,

Definition 3 *A set $D : \{a_1, \ldots, a_k\}$ modulo N, $a_i \in 0, \ldots, N-1$, is called a relaxed (cyclic) (N, k)-difference set if for every $d \not\equiv 0 \pmod{N}$ there exists at least one ordered pair (a_i, a_j), $a_i, a_j \in D$ such that $a_i - a_j \equiv d \pmod{N}$.*

Note that a relaxed (N, k)-difference set is a (N, k, λ)-difference set if there are exactly λ such ordered pairs for each d. For example the set {0,1,2,4} modulo 8 is a relaxed (8,4)-difference set since

$$\left.\begin{aligned} 1 &\equiv 1-0 \\ 2 &\equiv 2-0 \\ 3 &\equiv 4-1 \\ 4 &\equiv 4-0 \\ 5 &\equiv 1-4 \\ 6 &\equiv 2-4 \\ 7 &\equiv 0-1 \end{aligned}\right. \pmod{8}.$$

Theorem 1 *A group of sets $B_i : \{a_1 + i, a_2 + i, \ldots, a_k + i\}$ modulo N, $i \in 0, \ldots, N-1$, is a group of cyclic quorum sets if and only if $D : \{a_1, a_2, \ldots, a_k\}$ is a relaxed (N, k)-difference set.*

Proof: First, we prove that if D is a relaxed difference set, then B_i's form a group of cyclic quorum sets. As mentioned above, since the cyclic property automatically implies the equal work property A3 and the equal responsibility property A4, we only need to prove the non-empty intersection property A2 or B2, i.e. $B_i \cap B_j \neq \emptyset$, is satisfied for all i and j. Without loss of generality, we assume that $j > i$. Consider the l-th element of B_i and the m-th element of B_j, denoted by $b_{i,l}$ and $b_{j,m}$ respectively. We will show that $b_{i,l} = b_{j,m}$ for some l and m. The difference $b_{i,l} - b_{j,m}$ is equal to $(a_l - a_m + i - j) \bmod N$ by Property B3 in Definition 2. Since l and m can be chosen from any value from 1 to k, there must be some l and m such that

$$a_l - a_m \equiv j - i \pmod{N}$$

where $j - i \in 1, \ldots, N-1$ if D is a relaxed difference set. Therefore, we can always choose a pair of l and m such that $b_{i,l} - b_{j,m} \equiv 0 \pmod{N}$. Since all the elements are between 0 and $N-1$, this implies $b_{i,l} = b_{j,m}$ and then $B_i \cap B_j \neq \emptyset$.

Next, we show that the converse is also true, i.e., if $B_i \cap B_j \neq \emptyset$, $\forall i, j$ then D is a relaxed difference set. We will prove it by contradiction. Assume that $B_i \cap B_j \neq \emptyset$ and D is not a relaxed difference set. Then there exists a number in $1, \ldots, N-1$, say t, in which $a_i - a_j \not\equiv t \pmod N$, $\forall i, j$ (see Definition 3). Take the m-th element of B_0 and the l-th element of B_t, we have

$$b_{t,l} - b_{0,m} \equiv a_l - a_m + t \pmod N.$$

However we know that for some l and m, $b_{t,l} - b_{0,m} = 0$ since $B_0 \cap B_t \neq \emptyset$. This implies that $a_m - a_l \equiv t \pmod N$, which contradicts the assumption. ∎

Armed with this theorem, we can check if a configuration satisfies the non-empty intersection property easily by just examining whether $\{a_1, a_2, \ldots, a_k\}$ is a relaxed difference set. Define a $k \times k$ matrix M whose element $m_{i,j}$ is equal to $(a_i - a_j) \bmod N$. We can determine whether a given set is a relaxed difference set by checking if M contains all the numbers from 0 to $N-1$. For example, given a set {0,1,3,6} and $N = 8$. The corresponding matrix M is:

$$M = \begin{bmatrix} 0 & 7 & 5 & 2 \\ 1 & 0 & 6 & 3 \\ 3 & 2 & 0 & 5 \\ 6 & 5 & 3 & 0 \end{bmatrix}.$$

Since M does not contain 4, the set {0,1,3,6} cannot form a group of cyclic quorums.

The maximum possible N for a fixed k is that M contains all distinguish numbers from 1 to $N-1$ at off-diagonal. In this particular case, $N = k^2 - k + 1$. Hence, although the quorum sets are restricted to be cyclic, the theoretical lower bound is the same as the original one. In fact, the achievement of the theoretical lower bound can be guaranteed for the cases of $N = k^2 - k + 1$ plus $k - 1$ is a prime power, by the construction of the Singer difference sets.

We search for an optimal solution of this scheme for all $N \leq 111$. The method that we used is by exhaustive search with the quorum size k starting from the theoretical lower bound. If a solution does not exist, we increase the quorum size by one and repeat the process. This procedure guarantees the solution that we find is optimal within the cyclic quorum sets. Although the exhaustive search method is used, since search space is much smaller now than that of the original problem setting and we have an efficient checking method by Theorem 1, it is more tractable by computers. The results for $N \leq 100$ are shown in Tables 1 and 2. We found that the optimal quorum size of $N \neq k^2 - k + 1$ is very close to the theoretical lower bound. It is interesting to note that since the cyclic difference set exists whenever the corresponding finite projective plane exists, this scheme can also achieve the lower bound for such cases.

The advantage of this scheme is that the solution set can be specified by only one quorum. For moderate value of N, practitioners may simply choose to use a table lookup method once the table such as Tables 1 and 2 are formed. Also, the solution set is strictly symmetric for arbitrary N.

5 Discussion

The choice of methods discussed above depends on what is the value of N and on whether the symmetry is a criticism. In this section, we discuss some guidelines in using these methods. If N is equal to $k^2 + k + 1$ where $k - 1$ is a prime power, the best way is to construct a finite projective plane of order $k - 1$, or a cyclic $(n^2 + n + 1, n + 1, 1)$-difference set where $n = k - 1$, because the solution set is both symmetric and optimal. Tables 12.23-24 in [11] list the cyclic $(n^2 + n + 1, n + 1, 1)$-difference sets for $2 \leq n \leq 97$. If N is not one of such values, and the symmetry is not a criticism, Maekawa's first method (i.e., by inserting "virture sites") can be employed because this method usually generates a smaller size of quorum sets. However, if the symmetry is a criticism, the cyclic quorum method that we proposed should be a better choice. The quorum set is strictly symmetric and the size is very close to the optimal one. Nevertheless, one of the problems of this method is that exhaustive search methods may not be practical for large N. Non-exhaustive search methods, such as simulated annealing and hill-climbing techniques may be used for searching for good solutions. We plan to extend Tables 1 and 2 in order to make practical use. Finally, if all the above methods fail to attack the problem, then the grid based algorithms can be considered.

After the submission of an earlier version of this paper, we discovered that the definition of relaxed difference set has been appeared in [12], known as *difference cover*. The authors in that paper discussed the use of this notion in VLSI chip design. However, the non-empty intersection property, which is the key for distributed mutual exclusion problem, is exploited only the first time.

Acknowledgments

The authors would like to thank our colleague Yui-Wah Lee for bringing Maekawa's paper to our attention.

References

[1] M. Singhal and N. G. Shivaratri, *Advanced concepts in operating systems: distributed, database, and multiprocessor operating systems*, chapter 6, McGraw-Hill, 1994.

[2] Y.-I. Chang, "A simulation study on distributed mutual exclusion," *Journal of Parallel and Distributed Computing*, vol. 33, pp. 107–121, 1996.

[3] M. Maekawa, "A $\sqrt{N}$ algorithm for mutual exclusion in decentralized systems," *ACM Trans. Comput. Systs.*, pp. 145–159, May 1985.

[4] D. R. Hughes and F. C. Piper, *Projective planes*, Springer Verlag, 1973.

[5] C. W. H. Lam, "The search for a finite projective plane of order 10," *American Mathematical Monthly*, vol. 98, pp. 305–318, 1991.

[6] T.-Y. Suen and S.-K. Wong, "Efficient task migration algorithm for distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 3, no. 4, pp. 488–499, July 1992.

[7] D. Agrawal, Ömer Eğecioğlu, and A. El Abbadi, "Billiard quorums on the grid," Tech. Rep. TRCS95-12, Department of Computer Science, University of California, Santa Barbara, April 1995.

[8] H.-M. Lien and S.-M. Yuan, "A new approach of constructing information structure for mutual exclusion in distributed systems," in *ICPADS: International Conference on Parallel and Distributed Systems, National Tsing Hua University, Hsinchu, Taiwan, Republic of China*, 1994, pp. 588–590.

[9] W. K. Ng and C. V. Ravishankar, "Coterie templates: A new quorum construction method," in *Proceedings of the 15th International Conference on Distributed Computing Systems*, Vancouver, Canada, May 1995, pp. 92–99.

[10] Marshall Hall, Jr., *Combinatorial Theory*, chapter 11, John Wiley & Sons, 1986.

[11] C. J. Colbourn and J. H. Dinitz, Eds., *The CRC Handbook of Combinatorial Designs*, CRC press, 1996.

[12] J. Kilian, S. Kipnis, and C. E. Leiserson, "The organization of permutation architectures with bused interconnections," *IEEE Transactions on Computers*, vol. 39, no. 11, pp. 1346–1358, Nov. 1990.

Table 1: Optimal cyclic quorum scheme ($N = 4$ to 57)

N	B_1								
4	1	2	3						
5	1	2	3						
6	1	2	4						
7	1	2	4						
8	1	2	3	5					
9	1	2	3	5					
10	1	2	3	6					
11	1	2	3	6					
12	1	2	4	8					
13	1	2	4	10					
14	1	2	3	4	8				
15	1	2	3	4	8				
16	1	2	3	6	9				
17	1	2	3	5	13				
18	1	2	3	6	12				
19	1	2	3	7	10				
20	1	2	3	4	7	11			
21	1	2	5	15	17				
22	1	2	3	4	8	12			
23	1	2	3	4	8	12			
24	1	2	3	4	8	16			
25	1	2	3	4	9	13			
26	1	2	3	6	10	16			
27	1	2	3	6	14	23			
28	1	2	5	16	21	23			
29	1	2	3	4	5	10	15		
30	1	2	3	4	5	10	20		
31	1	2	4	9	13	19			
32	1	2	3	4	8	12	20		
33	1	2	3	4	7	17	28		
34	1	2	3	4	8	13	21		
35	1	2	3	4	9	13	22		
36	1	2	3	6	13	15	21		
37	1	2	3	5	11	16	23		
38	1	2	3	4	5	9	15	24	
39	1	2	3	5	14	19	34		
40	1	2	3	4	5	10	15	25	
41	1	2	3	4	5	10	16	26	
42	1	2	3	4	5	10	16	26	
43	1	2	3	4	5	11	16	27	
44	1	2	3	4	7	17	28	39	
45	1	2	3	4	6	13	19	27	
46	1	2	3	4	7	19	26	39	
47	1	2	3	4	6	17	23	41	
48	1	2	3	6	10	21	27	37	
49	1	2	3	6	25	34	37	45	
50	1	2	4	9	18	29	33	39	
51	1	2	3	6	12	19	31	39	
52	1	2	3	4	5	7	15	22	31
53	1	2	3	4	5	8	22	30	45
54	1	2	3	4	5	10	16	22	32
55	1	2	3	4	5	7	20	27	48
56	1	2	3	4	5	12	17	34	40
57	1	2	4	14	33	37	44	53	

Table 2: Optimal cyclic quorum scheme ($N = 58$ to 111)

N	B_1											
58	1	2	3	4	8	22	34	38	51			
59	1	2	3	4	7	14	22	36	45			
60	1	2	3	5	10	16	26	31	43			
61	1	2	3	4	8	16	26	37	46			
62	1	2	3	5	11	33	40	47	52			
63	1	2	3	7	9	21	39	42	55			
64	1	2	3	6	15	17	35	43	60			
65	1	2	3	7	11	29	36	52	55			
66	1	2	3	4	5	6	14	20	40	47		
67	1	2	3	4	5	6	13	21	27	40		
68	1	2	3	4	5	11	17	22	39	46		
69	1	2	3	4	5	11	18	23	34	46		
70	1	2	3	4	5	10	21	36	50	63		
71	1	2	3	4	5	11	19	24	35	47		
72	1	2	3	4	7	12	19	32	38	52		
73	1	2	4	8	16	32	37	55	64			
74	1	2	3	4	8	29	31	44	58	66		
75	1	2	3	6	9	19	31	33	42	57		
76	1	2	3	7	10	26	36	47	59	64		
77	1	2	3	5	11	16	38	50	57	62		
78	1	2	3	8	14	17	34	52	56	71		
79	1	2	3	7	14	29	32	48	49	72		
80	1	2	3	4	5	6	11	24	41	57	72	
81	1	2	3	4	5	6	13	21	27	40	54	
82	1	2	3	4	5	6	13	21	27	41	54	
83	1	2	3	4	5	6	13	22	28	41	55	
84	1	2	3	4	5	8	19	27	47	55	76	
85	1	2	3	4	5	10	14	26	41	55	69	
86	1	2	3	4	5	12	18	25	30	49	55	
87	1	2	3	4	5	11	43	55	63	68	74	
88	1	2	3	4	6	12	25	30	37	44	74	
89	1	2	3	4	6	13	19	44	58	66	72	
90	1	2	3	4	7	34	47	55	68	75	82	
91	1	2	4	10	28	50	57	62	78	82		
92	1	2	3	5	41	51	52	60	65	72	78	
93	1	2	3	6	15	21	25	32	53	61	69	
94	1	2	3	4	5	6	7	15	24	31	47	62
95	1	2	3	6	9	18	29	40	54	64	83	
96	1	2	3	4	5	6	9	22	31	54	63	87
97	1	2	3	4	5	6	10	18	34	44	55	80
98	1	2	3	4	5	6	12	28	41	55	70	82
99	1	2	3	4	5	6	13	22	28	35	49	63
100	1	2	3	4	5	6	14	21	29	35	57	64
101	1	2	3	4	5	6	13	50	64	73	79	86
102	1	2	3	4	5	7	14	29	35	43	51	86
103	1	2	3	4	5	8	39	54	63	78	86	94
104	1	2	3	4	5	10	20	33	47	58	73	85
105	1	2	3	4	5	11	16	37	40	62	67	90
106	1	2	3	4	6	49	54	70	77	83	90	98
107	1	2	3	4	6	21	28	36	43	49	59	99
108	1	2	3	4	8	13	21	35	42	50	58	86
109	1	2	3	4	8	16	40	50	59	84	90	95
110	1	2	3	7	18	26	40	44	47	53	81	101
111	1	2	3	6	13	28	37	39	45	53	66	94