Quorum Systems Constructed from Combinatorial Designs

Charles J. Colbourn
Department of Computer Science
University of Vermont
Burlington VT 05405

Jeffrey H. Dinitz
Department of Mathematics
University of Vermont
Burlington VT 05405

Douglas R. Stinson
Department of Computer Science and Engineering
University of Nebraska
Lincoln NE 68588

May 12, 1998

Abstract

A quorum system is a set system in which any two subsets have nonempty intersection. Quorum systems have been extensively studied as a method of maintaining consistency in distributed systems. Important attributes of a quorum system include the load, balancing ratio, rank (i.e., quorum size) and availability. Many constructions have been presented in the literature for quorum systems in which these attributes take on optimal or otherwise favorable values.

In this paper, we point out an elementary connection between quorum systems and the classical covering systems studied in combinatorial design theory. We look more closely at the quorum systems that are obtained from balanced incomplete block designs (BIBDs). We study the properties of these quorum systems, and observe that they have load, balancing ratio and rank that are all within a constant factor of being optimal.

We also provide several observations about computing the failure polynomials of a quorum system (failure polynomials are used to measure availability). Asymptotic properties of failure polynomials have previously been analyzed for certain infinite families of quorum systems. We give an explicit formula for the failure polynomials for an easily constructed infinite class of quorum systems. We also develop two algorithms that are useful for computing failure polynomials for quorum systems, and prove that computing failure polynomials is #P-hard. Computational results are presented for several "small" quorum systems obtained from BIBDs.

1 Introduction

Quorum systems have been extensively studied as a method of maintaining consistency in distributed systems, e.g. a distributed database. The idea is to identify certain special subsets, called

quorums, where it is required that any two quorums have a non-empty intersection. Any access to the system requires accessing all the elements in one of the quorums, and an updating operation requires updating all the elements in one of the quorums. The intersection property ensures that any quorum contains at least one element that is up-to-date, and therefore consistency of the system is maintained over time.

A *set system* is a pair $(X, \mathcal{A})$, where X is a finite set of *points* and $\mathcal{A}$ is a set of subsets of X, called *blocks*. (A set system is also called a *hypergraph*, in which case the points are referred to as *vertices* and the blocks are referred to as *edges*.) The *degree* of a point $x \in X$ is the number of blocks containing the point x. $(X, \mathcal{A})$ is *regular* (of degree d) if all points have the same degree, d. The *rank* of $(X, \mathcal{A})$ is the size of the largest block. If all blocks have the same size, say r, then $(X, \mathcal{A})$ is said to be *uniform* (of rank r). A subset of blocks $\mathcal{B} \subseteq \mathcal{A}$ is *spanning* if $\cup_{A \in \mathcal{B}} A = X$. We say that $\mathcal{B}$ is a *spanning set of i blocks* if $|\mathcal{B}| = i$.

The following lemma follows immediately by counting point-block incidences.

Lemma 1.1 *If $(X, \mathcal{A})$ is a set system that is regular of degree d and uniform of rank r, then $|X|d = |\mathcal{A}|r$.*

The *incidence matrix* of a set system $(X, \mathcal{A})$ is the $|X| \times |\mathcal{A}|$ matrix $M = (m_{x,A})$, in which the rows are indexed by points, the columns are indexed by blocks, and the entries are defined as follows:

$$m_{x,A} = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A. \end{cases}$$

If $(X, \mathcal{A})$ is a set system having incidence matrix M, then the *dual set system* is the set system having incidence matrix M^T, where the superscript "T" denotes transpose. The dual set system can be described as $(Y, \mathcal{B})$, where $Y = \mathcal{A}$ and

$$\mathcal{B} = \{\{A \in \mathcal{A} : x \in A\} : x \in X\}.$$

A set system is regular of degree d if and only if the dual set system is uniform of rank d.

A *quorum system* is a set system $(X, \mathcal{A})$ in which $A \cap B \neq \emptyset$ for all $A, B \in \mathcal{A}$. An *(n, m)-quorum system* is a quorum system $(X, \mathcal{A})$ in which $|X| = n$ and $|\mathcal{A}| = m$. The points in a quorum system are called *elements* and the blocks are called *quorums*.

A *covering system* is a set system $(X, \mathcal{A})$ such that, for every $x, y \in X$, there exists a block $A \in \mathcal{A}$ such that $\{x, y\} \subseteq A$. An *(n, m)-covering system* is a covering system $(X, \mathcal{A})$ in which $|X| = n$ and $|\mathcal{A}| = m$.

Theorem 1.2 *There exists an (n, m)-quorum system if and only if there exists an (m, n)-covering system.*

Proof. A set system is a quorum system if and only the dual set system is a covering system. □

1.1 Quorum systems from BIBDs

There is a considerable literature on covering systems; see [3] and [11], for example. In view of Theorem 1.2, any covering system yields a quorum system, and vice versa. It is perhaps surprising that covering systems have not been extensively investigated for their suitability as quorum systems. The most likely explanation for this is that quorum systems have been studied primarily within hypergraph theory as opposed to design theory, the branch of combinatorics that includes covering systems. A comprehensive reference for results in design theory is [3].

In this paper, we concentrate on a special type of covering system, which we define now. Suppose v and k are integers such that $v > k > 1$. A (v, k, λ)-*BIBD* (balanced incomplete block design) is a set system that is uniform of rank k, such that every pair of points occurs in exactly λ blocks. It is not difficult to see that a (v, k, λ)-BIBD is regular of degree $(v-1)/(k-1)$, and the number of blocks is $\lambda(v^2-v)/(k^2-k)$. A (v, k, λ)-BIBD is a covering system, so the dual system is a quorum system by Theorem 1.2.

We are mainly interested in the quorum systems that result from BIBDs with $\lambda = 1$, as recorded in the following theorem.

Theorem 1.3 *Suppose there exists a $(v, k, 1)$-BIBD. Then the dual system is an (n, m)-quorum system, where $n = (v^2-v)/(k^2-k)$ and $m = v$, that is regular of degree k and uniform of rank $(v-1)/(k-1)$.*

A $(v, k, 1)$-BIBD yields an (n, m)-quorum system that is uniform of rank r, where $r \approx c\sqrt{n}$ and $c = \sqrt{\frac{k}{k-1}}$. A *projective plane* of order n is an $(n^2+n+1, n+1, 1)$-BIBD. Such planes are known to exist only when n is a prime or a power of a prime. Maekawa [10] first suggested using projective planes as quorum systems. To our knowledge, the only other example in the literature where quorum systems are constructed from BIBDs is by Luk and Wong [9], who give a direct construction of a set system that can be seen to be isomorphic to the dual of a $(v, 2, 1)$-BIBD. (This BIBD is in fact a complete graph K_v, which trivially exists for all integers v.) Luk and Wong also suggested the use of "difference covers" to construct quorum systems. Difference covers are a generalization of the difference sets that are used to construct projective planes.

There are many known infinite classes of BIBDs, any of which could be employed as quorum systems (for a summary of results on BIBDs, see [3]). Examples include BIBDs with "small" block size, which have undergone a tremendous amount of study. In fact, existence of $(v, k, 1)$-BIBDs with $k \leq 5$ has been completely determined as follows (see [3], for example).

Theorem 1.4

1. *A $(v, 2, 1)$-BIBD exists for all integers $v \geq 3$.*
2. *A $(v, 3, 1)$-BIBD exists if and only if $v \equiv 1, 3 \bmod 6$, $v \geq 7$.*
3. *A $(v, 4, 1)$-BIBD exists if and only if $v \equiv 1, 4 \bmod 12$, $v \geq 13$.*
4. *A $(v, 5, 1)$-BIBD exists if and only if $v \equiv 1, 5 \bmod 20$, $v \geq 21$.*

The $(v, k, 1)$-BIBDs with $k = 2, 3, 4$ and 5 yield (n, m)-quorum systems that are uniform of rank (roughly) $c\sqrt{n}$, where $c \approx 1.41, 1.23, 1.15$ and 1.12, respectively. (Note that by Theorem 2.4 below, $\sqrt{n}$ is the best possible). Various other constructions for (n, m)-quorum systems that are uniform of rank $c\sqrt{n}$ can be found in the literature; see, for example, [1, 2, 7, 9, 10, 12]. Most of these constructions have $c = 2$ or 1.41 (except for the finite projective plane construction from [10] and the difference cover construction from [9], which have $c \approx 1$).

2 Attributes of quorum systems

We discuss several measures of merit for quorum systems in this section, including load, balancing ratio, rank and availability. We then apply these measures to quorum systems obtained from BIBD's.

2.1 Load and balancing ratio

Suppose $\mathcal{Q} = (X, \mathcal{A})$ is a quorum system and $p_{\mathcal{A}}$ is a probability distribution defined on $\mathcal{A}$. For any $x \in X$, define

$$\mathcal{L}(p_{\mathcal{A}}, \mathcal{Q}, x) = \sum_{\{A \in \mathcal{A} : x \in A\}} p_{\mathcal{A}}(A).$$

Then $\mathcal{L}(p_{\mathcal{A}}, \mathcal{Q}, x)$ measures the fraction of time that element x is busy under the probability distribution $p_{\mathcal{A}}$.

Define

$$\mathcal{L}(p_{\mathcal{A}}, \mathcal{Q}) = \max \{\mathcal{L}(p_{\mathcal{A}}, \mathcal{Q}, x) : x \in X\}.$$

Then $\mathcal{L}(p_{\mathcal{A}}, \mathcal{Q})$, the *load* of the system (under the probability distribution $p_{\mathcal{A}}$), measures the fraction of time that the busiest element in X is used under the probability distribution $p_{\mathcal{A}}$. In general, we choose the probability distribution $p_{\mathcal{A}}$ so as to minimize the load. Therefore, Naor and Wool ([12]) defined the quantity

$$\mathcal{L}(\mathcal{Q}) = \min_{p_{\mathcal{A}}} \mathcal{L}(p_{\mathcal{A}}, \mathcal{Q}),$$

where the minimum is computed over all probability distributions $p_{\mathcal{A}}$.

Another desirable property, especially when the elements in the system are all "similar", is to balance the loads. The *balancing ratio* of the system (under the probability distribution $p_{\mathcal{A}}$) is defined by Holzman, Marcus and Peleg ([5]) to be the quantity

$$\rho(p_{\mathcal{A}}, \mathcal{Q}) = \frac{\min \{\mathcal{L}(p_{\mathcal{A}}, \mathcal{Q}, x) : x \in X\}}{\max \{\mathcal{L}(p_{\mathcal{A}}, \mathcal{Q}, x) : x \in X\}}.$$

The quantity $\rho(p_{\mathcal{A}}, \mathcal{Q})$ is to be maximized. Thus, we define

$$\rho(\mathcal{Q}) = \max_{p_{\mathcal{A}}} \rho(p_{\mathcal{A}}, \mathcal{Q}),$$

where the maximum is computed over all probability distributions $p_{\mathcal{A}}$.

Theorem 2.1 *([12, Prop. 5.4]) $\mathcal{L}(\mathcal{Q}) \geq \frac{1}{\sqrt{n}}$ for any (n, m)-quorum system $\mathcal{Q}$.*

Theorem 2.2 *([12, Prop. 5.9]) If $\mathcal{Q}$ is an (n, m)-quorum system that is regular of degree d and uniform of rank r, then $\mathcal{L}(\mathcal{Q}) = r/n = d/m$.*

Theorem 2.3 *([5, Prop. 3.1]) If $\mathcal{Q}$ is a regular quorum system, then $\rho(\mathcal{Q}) = 1$.*

2.2 Rank

The following result was proved by Lovász [8]; a generalization is proved in [5, Theorem 5.10].

Theorem 2.4 *If $(X, \mathcal{A})$ is an (n, m)-quorum system that is regular of degree d and uniform of rank r, then then $n \leq r^2 - r + 1$.*

In a uniform quorum system, the rank specifies the number of elements in any quorum. In general, we want the rank to be as small as possible. Theorem 2.4 provides a lower bound on the rank for uniform, regular quorum systems, i.e., $r > \sqrt{n}$.

2.3 Failure polynomials

The "availability" of a quorum system is studied in several papers by means of the so-called failure polynomial. We begin by reviewing the probabilistic model and the necessary definitions.

Let $\mathcal{Q} = (X, \mathcal{A})$ be an (n, m)-quorum system. For $Y \subseteq X$, define $\mathsf{fail}(Y) = 1$ if $Y \cap A \neq \emptyset$ for all $A \in \mathcal{A}$, and define $\mathsf{fail}(Y) = 0$ otherwise. Assuming that each element fails independently with probability p, the *failure probability* of $\mathcal{Q}$ is computed as

$$F_{\mathcal{Q}}(p) = \sum_{Y \subseteq X} \mathsf{fail}(Y) p^{|Y|} (1-p)^{|X|-|Y|}.$$

Then $F_{\mathcal{Q}}(p)$ is a polynomial in p of degree n, the *failure polynomial* of the quorum system $\mathcal{Q}$.

For $0 \leq i \leq n$, suppose we define F_i to be the number of spanning sets of i blocks in the dual set system (i.e., the covering system). Then $F_{\mathcal{Q}}(p)$ can be written as follows:

Theorem 2.5 *[13, Lemma 2.21] Let $\mathcal{Q}$ be an (n, m)-quorum system, and suppose the values F_i are defined as above. Then*

$$F_{\mathcal{Q}}(p) = \sum_{i=0}^{n} F_i \, p^i (1-p)^{n-i}.$$

2.4 An example

We present a small example. Define $X = \{1, 2, 3, 4, 5, 6, 7\}$ and

$$\mathcal{A} = \{\{1,2,4\}, \{2,3,5\}, \{3,4,6\}, \{4,5,7\}, \{5,6,1\}, \{6,7,2\}, \{7,1,3\}\}.$$

Then $\mathcal{Q} = (X, \mathcal{A})$ is a $(7, 3, 1)$-BIBD (in fact, a projective plane of order 2, which is unique up to isomorphism). The dual of this BIBD is a $(7, 7)$-quorum system that is uniform of rank 3 and regular of degree 3. By Theorem 2.2, its load is $3/7$, and from Theorem 2.3, its balancing ratio is 1.

To compute the failure polynomial $F_{\mathcal{Q}}(p)$ for this quorum system, we begin by computing the coefficients F_i, $0 \leq i \leq 7$:

- No set of two or fewer blocks is spanning, so $F_0 = F_1 = F_2 = 0$.
- A set of three blocks is spanning if and only if it consists of the three blocks through a given point. Hence, $F_3 = 7$.

- A set of four blocks is spanning if and only if it is not the complement of the three blocks through a given point. Hence, $F_4 = \binom{7}{4} - 7 = 28$.
- Any set of at least five blocks is spanning. Hence $F_5 = \binom{7}{5} = 21$, $F_6 = \binom{7}{6} = 7$ and $F_7 = \binom{7}{7} = 1$.

Hence, the failure polynomial is

$$F_{\mathcal{Q}}(p) = 7p^3(1-p)^4 + 28p^4(1-p)^3 + 21p^5(1-p)^2 + 7p^6(1-p) + p^7.$$

2.5 Properties of quorum systems constructed from BIBDs

Applying Theorem 1.3 and 2.2, the (n, m)-quorum system $\mathcal{Q}$ constructed from a $(v, k, 1)$-BIBD has load $\mathcal{L}(\mathcal{Q}) = k/v \approx c/\sqrt{n}$, where $c = \sqrt{\frac{k}{k-1}}$. This is a constant factor c above the lower bound of Theorem 2.1. The quorum system also has $\rho(\mathcal{Q}) = 1$, so it is optimally balanced. The rank of the quorum system is approximately $c\sqrt{n}$, so it is also a constant factor c above the lower bound of Theorem 2.4.

3 Computing the failure polynomial using inclusion-exclusion

In this section, we describe a method of computing failure polynomials for quorum systems that uses the principle of inclusion-exclusion. Suppose that $(Y, \mathcal{B})$ is the dual of an (n, m)-quorum system $(X, \mathcal{A})$. Hence, $|Y| = |\mathcal{A}| = m$ and $|\mathcal{B}| = |X| = n$. For a subset $Z \subseteq Y$, define

$$\mathsf{span}(Z) = \{B \in \mathcal{B} : B \cap Z \neq \emptyset\}.$$

Thus, $\mathsf{span}(Z)$ consists of all the blocks in $\mathcal{B}$ that hit the set Z. Next, for $0 \leq i \leq m$, $0 \leq j \leq n$, define

$$a_{i,j} = |\{Z \subseteq Y : |Z| = i \text{ and } |\mathsf{span}(Z)| = j\}|.$$

Thus $a_{i,j}$ denotes the number of ways to select i points from Y that are hit by exactly j blocks from $\mathcal{A}$. Using inclusion-exclusion, we obtain the following alternative formula for $F_{\mathcal{Q}}(p)$:

Theorem 3.1 *Let $\mathcal{Q}$ be an (n, m)-quorum system, and suppose the values $a_{i,j}$ are defined as above. Then*

$$F_{\mathcal{Q}}(p) = \sum_{i=0}^{m} \sum_{j=0}^{n} (-1)^i a_{i,j} (1-p)^j.$$

As an example, we return to the $(7, 3, 1)$-BIBD considered earlier. The non-zero values $a_{i,j}$ are easily computed by hand to be the following:

$$\begin{array}{lll} a_{0,0} = 1, & a_{1,3} = 7, & a_{2,5} = 21, \\ a_{3,6} = 28, & a_{3,7} = 7, & a_{4,6} = 7, \\ a_{4,7} = 28, & a_{5,7} = 21, & a_{6,7} = 7, \\ a_{7,7} = 1. & & \end{array}$$

Hence, $F_{\mathcal{Q}}(p) = 1 - 7(1-p)^3 + 21(1-p)^5 - 21(1-p)^6 + 6(1-p)^7$.

As another illustration of the use of Theorem 3.1, we compute an explicit formula for the failure polynomials for an infinite class of quorum systems. For an integer $v \geq 2$, define $\mathcal{Q}_v$ to be the quorum system that is the dual of a $(v, 2, 1)$-BIBD, say $(Y, \mathcal{B})$. (As mentioned earlier, $(Y, \mathcal{B})$ is a complete graph K_v.) $\mathcal{Q}_v$ is an (n, m)-quorum system in which $n = \binom{v}{2}$ and $m = v$. By Theorem 2.2, its load is $2/v$.

The value of $F_{\mathcal{Q}_v}(p)$ is the probability that a random graph is spanning, in the usual random graph model where every edge occurs with probability p. We obtain an explicit formula for $F_{\mathcal{Q}_v}(p)$ by computing the values $a_{i,j}$ and then applying Theorem 3.1. For $0 \leq i \leq v$, we have that

$$a_{i,j} = \begin{cases} \binom{v}{i} & \text{if } j = \binom{v}{2} - \binom{v-i}{2} \\ 0 & \text{otherwise.} \end{cases}$$

Hence, we have the following result.

Theorem 3.2 *For any integer $v \geq 2$, suppose $\mathcal{Q}_v$ is as defined above. Then*

$$F_{\mathcal{Q}_v}(p) = \sum_{i=0}^{v} \binom{v}{i} (-1)^i (1-p)^{\binom{v}{2} - \binom{v-i}{2}}.$$

As is typical with many properties of random graphs, when p becomes less than $\ln(v)/v$, $F_{\mathcal{Q}_v}(p)$ approaches 0 very rapidly. The values of $F_{\mathcal{Q}_v}(\ln(v)/v)$ turn out to be close to .4.

3.1 A conversion formula

We now show how to convert $F_{\mathcal{Q}_v}(p)$ from the form given in Theorem 3.1 to that of Theorem 2.5. Using the fact that

$$1 = ((1-p) + p)^{n-j},$$

we have

$$\begin{aligned}
&\sum_{i=0}^{m} \sum_{j=0}^{n} (-1)^i a_{i,j} (1-p)^j \\
&= \sum_{i=0}^{m} \sum_{j=0}^{n} \left(\sum_{k=0}^{n-j} \binom{n-j}{k} p^k (1-p)^{n-j-k} \right) (-1)^i a_{i,j} (1-p)^j \\
&= \sum_{i=0}^{m} \sum_{j=0}^{n} \sum_{k=0}^{n-j} (-1)^i a_{i,j} \binom{n-j}{k} p^k (1-p)^{n-k} \\
&= \sum_{k=0}^{n} \sum_{i=0}^{m} \sum_{j=0}^{n-k} (-1)^i a_{i,j} \binom{n-j}{k} p^k (1-p)^{n-k}.
\end{aligned}$$

Equating the coefficients of this last formula with those of Theorem 2.5, we obtain the following relation of the values F_k to the $a_{i,j}$s.

Theorem 3.3

$$F_k = \sum_{i=0}^{m} \sum_{j=0}^{n-k} (-1)^i \binom{n-j}{k} a_{i,j}.$$

3.2 The failure polynomial for the dual of a $(v,3,1)$-BIBD

Certain information can be obtained about $F_{\mathcal{Q}}(p)$ in the case where the quorum $\mathcal{Q}$ is the dual of a $(v,3,1)$-BIBD. For $i \leq 5$, the values $a_{i,j}$ are constant (i.e., independent of the particular $(v,3,1)$-BIBD chosen), and hence can be computed as a function of v. Suppose we write $v = 2r+1$ and $n = v(v-1)/6$; then the following are computed by elementary counting:

$$\begin{aligned}
a_{0,0} &= 1 \\
a_{1,r} &= v \\
a_{2,2r-1} &= \binom{v}{2} \\
a_{3,3r-2} &= n \\
a_{3,3r-3} &= \binom{v}{3} - n \\
a_{4,4r-5} &= n(v-3) \\
a_{4,4r-6} &= \binom{v}{4} - n(v-3) \\
a_{5,5r-8} &= v\binom{r}{2} \\
a_{5,5r-9} &= n\left(\binom{v-3}{2} - 3(r-1)\right) \\
a_{5,5r-10} &= \binom{v}{5} - v\binom{r}{2} - n\left(\binom{v-3}{2} - 3(r-1)\right).
\end{aligned}$$

Most of these values are obtained by counting occurrences of small configurations in $(v,3,1)$-BIBDs. For example, $a_{5,5r-8}$ is the number of pairs of intersecting blocks; $a_{5,5r-9}$ is the number of ways to select five points which contain exactly one block; and $a_{5,5r-10}$ is the number of ways to select five points that contain no block.

If $i \leq 5$ and $a_{i,j}$ is not given above, then $a_{i,j} = 0$. When we proceed to $i = 6$, it is possible to show that $a_{6,j} = 0$ unless $6r - j \in \{11, \ldots, 15\}$. In general, these values are not "constant". For example, $a_{6,6r-11}$ is equal to the number of Pasch configurations in the BIBD. (A *Pasch configuration* is a set of four blocks in a $(v,3,1)$-BIBD, whose union contains exactly six points. A Pasch configuration is isomorphic to the following set of blocks: $\{1,2,3\},\{1,4,5\},\{2,4,6\},\{3,5,6\}$. For more information about small configurations in $(v,3,1)$-BIBDS, see [4], for example.) From the fact that all values $a_{i,j}$ are constant for $j \leq 6r-16$, Theorem 3.3 implies that the value F_i is constant for $i \leq b-6r+16$. (Of course, it is also the case that $F_i = 0$ if $i < v/3$.)

4 Algorithms for computing the failure polynomial

In this section, we describe two simple but effective algorithms that can be used for computing the failure polynomial. These algorithms can be applied to any quorum system. The first algorithm is a backtracking algorithm and the second is a dynamic programming algorithm. Since both algorithms require exponential time in the worst case, it is natural to establish a complexity result justifying the development of exponential time algorithms. We pursue this next.

4.1 The complexity of computing availability

Calculating failure polynomials involves solving an enumeration problem for each coefficient. Thus natural complexity classes to consider involve machines with a capability for counting. Valiant [16] introduced the *counting Turing machine*, which is a nondeterministic Turing machine with an auxiliary write-only tape. When such a machine accepts its input, it writes the number of accepting computations on the auxiliary tape. Valiant then defines #P to be the class of enumeration problems that can be solved in polynomial time on a counting Turing machine. He establishes the existence of "most difficult" problems in this class. A problem is #P-*hard* if every problem in #P can be reduced in polynomial time to it, and a problem is #P-*complete* if it is a member of #P and also is #P-hard.

Theorem 4.1 *Computing the failure polynomial of a quorum system is #P-hard.*

Proof. Valiant [16] establishes that computing the number of perfect matchings in a graph is #P-complete. We give a polynomial-time reduction from the problem of computing the number of perfect matchings in a graph to the computation of one of the coefficients in the failure polynomial of a quorum system.

Let $G = (V, E)$ be a graph without isolated vertices, with $|V| = 2s$ and $|E| = t$. Let ρ be the number of perfect matchings in G; this is the quantity that we wish to calculate. Now let X be a set of s elements disjoint from V. Form a covering on $V \cup X$, taking as blocks:

1. the set X;
2. the set V; and
3. the set $\{x\} \cup e$ for each $x \in X$ and $e \in E$.

It is easy to verify, using the fact that G does not contain isolated vertices, that this set of blocks is a covering. The dual set system is an (n, m)-quorum system with $n = st + 2$ and $m = 3s$.

We claim that the number of ways to choose precisely s blocks of the covering which span all $3s$ points is precisely

$$\binom{st}{s-2} + \rho \cdot s!$$

To see this, observe that if the blocks X and V are both chosen, then we are free to choose any $s-2$ further blocks from the st blocks of cardinality 3, and the selection then spans all $3s$ points (indeed, V and X alone do that). If V is not chosen but X is, then s further blocks must be chosen from the blocks of cardinality 3 to cover all points of V, necessitating the selection of at least $s+1$ blocks. Symmetrically, if V is chosen but not X, then s further blocks must be chosen from the blocks of cardinality 3 to cover all points of X, again requiring more than s blocks. Finally, if neither V nor X is chosen, then we must choose blocks of cardinality 3 which, when restricted to the points of V, form s edges comprising a perfect matching in G. Every such perfect matching corresponds to precisely $s!$ different selections of spanning blocks of cardinality 3.

Let us examine the consequences of this. If we can calculate the coefficient F_s for the failure polynomial of the quorum system, then, knowing s and t, we can solve for ρ and hence determine the number of perfect matchings in G. But since the quroum system can be constructed in polynomial time given the graph G, this provides a polynomial time reduction from the problem of counting perfect matchings in G to the problem of computing the failure polynomial of a quorum system. Thus, computing the failure polynomial is #P-hard. ∎

The computation of the failure polynomial fails for technical reasons to fall into the class of #P-complete problems, as it involves a counting problem for each coefficient. Nevertheless, the computation of each coefficient separately can be done in polynomial time by a counting Turing machine. Hence the computation of the failure polynomial is polynomially equivalent to the problem of counting perfect matchings in a graph.

While we do not know from this argument whether the result carries over to regular coverings (and hence uniform quorum systems), we expect that simple variations of this technique can be used to establish similar complexity results. The main consequence, however, is that in the evaluation of failure polynomials, we can expect to need methods requiring exponential time in the worst case if we are to determine precise results. In a practical direction, it suggests that approximations to the failure polynomial are of interest.

4.2 A backtracking algorithm

Suppose that $\mathcal{Q}$ is an (n, m)-quorum system. We describe our algorithm in terms of the dual set system, which is a covering system. This covering system, $(Y, \mathcal{B})$, has m points and n blocks. By Theorem 2.5, $F_{\mathcal{Q}}(p) = \sum F_i\, p^i(1-p)^{n-i}$, where F_i is the number of spanning sets of i blocks in the covering system.

Suppose that $\mathcal{B} = \{B_j : 1 \leq j \leq n\}$ and $\alpha = (\alpha_1, \ldots, \alpha_n) \in \{0,1\}^n$. The n-tuple α can be thought of as a method of encoding a subset of blocks $\mathcal{C}_\alpha \subseteq \mathcal{B}$, where

$$\alpha_j = 1 \Leftrightarrow B_j \in \mathcal{C}_\alpha.$$

If $\mathcal{C}_\alpha$ is a spanning set of i blocks, then α contributes a term $p^i(1-p)^{n-i}$ to $F_{\mathcal{Q}}(p)$.

An elementary backtracking algorithm can be used to generate all 2^n n-tuples α, and compute $F_{\mathcal{Q}}(p)$ using the above observation. This algorithm has complexity $\Theta(2^n)$, and thus is not practical for large values of n. However, we can speed up the algorithm significantly by reducing the number of n-tuples that need to be considered.

The speedup is based on the observation that, if

$$B_j \subseteq \bigcup_{i=1}^{j-1} B_{\alpha_i},$$

then the value of α_j has no effect on whether $\mathcal{C}_\alpha$ is a spanning set of blocks. We can thus set $\alpha_j = *$ for this value of j, where $*$ means "don't care".

Suppose that $\alpha = (\alpha_1, \ldots, \alpha_n) \in \{0, 1, *\}^n$ is an n-tuple where

$$\alpha_j = * \Leftrightarrow B_j \subseteq \bigcup_{i=1}^{j-1} B_{\alpha_i}.$$

Such an n-tuple is called *reduced*. Now, suppose that $\mathcal{C}_\alpha$ is a spanning set of i blocks, where α is reduced, and α contains j entries equal to 0. Then α contributes a term $p^i(1-p)^j$ to $F_{\mathcal{Q}}(p)$.

It is a simple matter to construct a backtracking algorithm to generate all the reduced n-tuples α. Since each entry $*$ replaces two recursive calls, the size of the search tree is reduced considerably. In fact, we do not need to explicitly keep track of the α's that are generated. We just need to record the number of 1's and the number of 0's in α, as well as the points spanned by the blocks in $\mathcal{C}_\alpha$. These correspond to the parameters i, j and U in the algorithm presented in Figure 1.

In this algorithm, $F_{\mathcal{Q}}(p)$ is first computed in the recursive procedure `bktrk` the form

$$F_{\mathcal{Q}}(p) = \sum_{i=0}^{n} \sum_{j=0}^{n} f_{i,j} p^i (1-p)^j.$$

(Such a representation of $F_{\mathcal{Q}}(p)$ is not unique, of course.) After this is done, $F_{\mathcal{Q}}(p)$ is converted to its usual representation, as given in Theorem 2.5, by essentially the same technique used in Section 3.1.

Figure 1: Backtracking algorithm to compute $F_{\mathcal{Q}}(p)$

```
procedure bktrk(ℓ, U, i, j);
begin procedure
  if ℓ = n + 1 then
    if U = Y then
      f_{i,j} ← f_{i,j} + 1
    else
  else
    if B_ℓ ⊆ U then
      bktrk(ℓ + 1, U, i, j)
    else
      bktrk(ℓ + 1, U, i, j + 1)
      bktrk(ℓ + 1, U ∪ B_ℓ, i + 1, j)
end procedure

begin main
  input B_1, ..., B_n
  for i ← 0 to n do
  for j ← 0 to n do
    f_{i,j} ← 0
  bktrk(1, ∅, 0, 0)
  for i ← 0 to n do
    F_i ← 0
  for i ← 0 to n do
  for j ← 0 to n − i do
  for k ← 0 to n − i − j do
    F_{i+k} ← F_{i+k} + C(n − i − j, k) f_{i,j}
  output F_0, ..., F_n
end main
```

4.3 A dynamic programming algorithm

Dynamic programming involves generating a table of values, using a recurrence relation, in a bottom-up fashion. We illustrate how this strategy can be used to compute the coefficients F_i of

the failure polynomial. As in the previous section, we work with the covering system $(Y, \mathcal{B})$, which has m points and n blocks.

Suppose $Z \subseteq Y$ and $1 \leq i \leq j \leq n$. Define $f(Z, i, j)$ to be the number of ways of selecting a subset of blocks $\mathcal{C} \subseteq \{B_1, \ldots, B_j\}$ in such a way that

$$|\mathcal{C}| = i \text{ and } \bigcup_{B \in \mathcal{C}} B = Z.$$

We have

$$F_i = f(Y, i, n),$$

$i = 1, \ldots, n$.

We compute all the values $f(Z, i, j)$ by means of a recurrence relation. It is convenient to define $f(Z, i, i-1) = 0$ for all Z and i.

If $B_j \not\subseteq Z$, then we cannot include B_j in a set $\mathcal{C}$ as defined above. In this case, we have $f(Z, i, j) = f(Z, i, j-1)$ (noting that $f(Z, i, i-1) = 0$).

On the other hand, if $B_j \subseteq Z$, then we can include B_j in $\mathcal{C}$ if we wish. Here, we have

$$f(Z, i, j) = f(Z, i, j-1) + \sum_{\{Z_0 : Z \setminus B_j \subseteq Z_0 \subseteq Z\}} f(Z_0, i-1, j-1).$$

There are $2^{|B_j|}$ terms in the above sum.

We can compute all values $f(Z, i, j)$ by this method. If the covering system has rank ℓ, then the resulting algorithm has complexity $\Theta(n^2 2^{m+\ell})$.

4.4 Computational results for small BIBDs

In this section, we compute the failure polynomials for all quorum systems obtained from $(v, k, 1)$-BIBDs with $k \geq 3$ and $v \leq 15$. All nonisomorphic BIBDs with parameters in this range have been enumerated; see [3], for example, where these designs are explicitly presented. The following list provides a summary:

- There is a unique $(7, 3, 1)$-BIBD up to isomorphism.
- There is a unique $(9, 3, 1)$-BIBD up to isomorphism.
- There are two non-isomorphic $(13, 3, 1)$-BIBDs.
- There is a unique $(13, 4, 1)$-BIBD up to isomorphism.
- There are precisely 80 non-isomorphic $(15, 3, 1)$-BIBDs.

We have already presented $F_{\mathcal{Q}}(p)$ when $\mathcal{Q}$ is the dual of a $(7, 3, 1)$-BIBD.

For the $(9, 3, 1)$-BIBD, we illustrate how the F_i's can be computed by hand, using the conversion formula proved in Theorem 3.3. First, it is easy to see that the following hold:

$$\begin{aligned} a_{0,0} &= 1, \\ a_{1,4} &= 9, \\ a_{2,7} &= \binom{9}{2} = 36, \\ a_{3,9} &= \binom{9}{3} - 12 = 72. \end{aligned}$$

Then we have the following:

$$\begin{aligned}
F_{12} &= \binom{12}{12}a_{0,0} &&= 1\\
F_{11} &= \binom{12}{11}a_{0,0} &&= 12\\
F_{10} &= \binom{12}{10}a_{0,0} &&= 66\\
F_{9} &= \binom{12}{9}a_{0,0} &&= 220\\
F_{8} &= \binom{12}{8}a_{0,0} - \binom{8}{8}a_{1,4} &&= 486\\
F_{7} &= \binom{12}{7}a_{0,0} - \binom{8}{7}a_{1,4} &&= 720\\
F_{6} &= \binom{12}{6}a_{0,0} - \binom{8}{6}a_{1,4} &&= 672\\
F_{5} &= \binom{12}{5}a_{0,0} - \binom{8}{5}a_{1,4} + \binom{5}{5}a_{2,7} &&= 324\\
F_{4} &= \binom{12}{4}a_{0,0} - \binom{8}{4}a_{1,4} + \binom{5}{4}a_{2,7} &&= 45\\
F_{3} &= \binom{12}{3}a_{0,0} - \binom{8}{3}a_{1,4} + \binom{5}{3}a_{2,7} - \binom{3}{3}a_{3,9} &&= 4.
\end{aligned}$$

For the remaining BIBDs, the failure polynomials are best determined using the algorithms we have described. In the case of a $(13,3,1)$-BIBD, all F_i's are constant, except for F_5. For both $(13,3,1)$-BIBDs, we have

$$\begin{array}{lll}
F_6 = 5408, & F_7 = 66950, & F_8 = 382980,\\
F_9 = 1316900, & F_{10} = 3141255, & F_{11} = 5648890,\\
F_{12} = 8055580, & F_{13} = 9401030, & F_{14} = 9154990,\\
F_{15} = 7524686, & F_{16} = 5248750, & F_{17} = 3109730,\\
F_{18} = 1559805, & F_{19} = 657540, & F_{20} = 230217,\\
F_{21} = 65780, & F_{22} = 14950, & F_{23} = 2600,\\
F_{24} = 325, & F_{25} = 26, & F_{26} = 1.
\end{array}$$

For design #1 in [3] we have $F_5 = 112$; for design #2 in [3], we have $F_5 = 117$.

For the $(13,4,1)$-BIBD, the coefficients of $F_{\mathcal{Q}}(p)$ are as follows:

$$\begin{array}{lll}
F_0 = 0, & F_1 = 0, & F_2 = 0,\\
F_3 = 0, & F_4 = 13, & F_5 = 117,\\
F_6 = 702, & F_7 = 1248, & F_8 = 1170,\\
F_9 = 702, & F_{10} = 286, & F_{11} = 78,\\
F_{12} = 13, & F_{13} = 1. &
\end{array}$$

In the case of a $(15,3,1)$-BIBD, all F_i's are constant, except for F_5, F_6, F_7 and F_8. The values of $F_9, \ldots, F_{35}$ are as follows:

$$
\begin{array}{lll}
F_9 = 9378775, & F_{10} = 46479118, & F_{11} = 163911510, \\
F_{12} = 443373350, & F_{13} = 965916245, & F_{14} = 1751496450, \\
F_{15} = 2704160200, & F_{16} = 3611429815, & F_{17} = 4218219600, \\
F_{18} = 4341489075, & F_{19} = 3956487150, & F_{20} = 3201345840, \\
F_{21} = 2302201110, & F_{22} = 1470686805, & F_{23} = 832977600, \\
F_{24} = 416918775, & F_{25} = 183530256, & F_{26} = 70601790, \\
F_{27} = 23535400, & F_{28} = 6724505, & F_{29} = 1623160, \\
F_{30} = 324632, & F_{31} = 52630, & F_{32} = 6545, \\
F_{33} = 595, & F_{34} = 35, \text{ and} & F_{35} = 1.
\end{array}
$$

The values of F_5, F_6, F_7 and F_8 for the 80 nonisomorphic $(15,3,1)$-BIBDs are presented in Tables 1 and 2 (the numbering used for the designs is as in [3]).

Given that we have described two algorithms for computing failure polynomials, it is natural to ask which is faster. This of course can depend on the implementation. For our programs, we found that computing the failure polynomial for (the dual of) a $(13,3,1)$-BIBD was done approximately seven times faster using the backtracking algorithm. On the other hand, computing the failure polynomial for (the dual of) a $(15,3,1)$-BIBD was done approximately two times faster using the dynamic programming algorithm. In general, we expect the dynamic programming algorithm to run faster for "larger" quorum systems. However, it should be recognized that the dynamic programming algorithm also has a much higher memory requirement than the backtracking algorithm.

The observations used in developing these methods hinge on an understanding of small configurations in BIBDs. In this context, we have seen that the number of Pasch configurations plays a role. Minimizing the number of Pasch configurations is explored in [6], while maximizing this number is examined in [15]. Other configurations also play a role. In the failure polynomial of a $(3s,3,1)$-BIBD, the coefficient F_s is the number of *parallel classes*, i.e., sets of blocks of the design which contain each element exactly once. It is unknown at present whether for all odd $s \geq 5$ there exists a $(3s,3,1)$-BIBD having no parallel classes [14], nor is the complexity of determining the existence of a parallel class known. That these difficult questions on $(3s,3,1)$-designs remain unsolved suggest the difficulty, and perhaps also another importance, of the study of failure polynomials.

5 Summary

We have pointed out that BIBDs provide many examples of quorum systems that have good values of load, balancing ratio and rank. We also developed several formulas and algorithms that can be used for exact computation of failure polynomials of arbitrary quorum systems. An explicit formula was obtained for the failure polynomial of an easily constructed infinite class of quorum systems, and computational results were presented for quorum systems constructed from "small" BIBDs.

Acknowledgments

Research of the authors is supported by ARO grant DAAG55-98-1-0272 (Colbourn) and NSF grant CCR-9610138 (Stinson).

Table 1: Coefficients of failure polynomials for the nonisomorphic $(15, 3, 1)$-BIBDs (designs #1–60)

	F_5	F_6	F_7	F_8		F_5	F_6	F_7	F_8
1	56	1890	77715	1193185	2	24	1794	77619	1193153
3	8	1746	77571	1193137	4	8	1730	77547	1193129
5	16	1738	77551	1193129	6	12	1710	77515	1193117
7	32	1722	77511	1193113	8	4	1702	77513	1193117
9	2	1688	77493	1193111	10	6	1692	77495	1193111
11	6	1676	77471	1193103	12	1	1689	77496	1193112
13	4	1694	77501	1193113	14	0	1698	77511	1193117
15	8	1682	77479	1193105	16	0	1722	77547	1193129
17	12	1686	77481	1193105	18	4	1678	77477	1193105
19	16	1674	77457	1193097	20	1	1665	77460	1193100
21	1	1665	77460	1193100	22	4	1662	77451	1193097
23	1	1661	77454	1193098	24	0	1662	77457	1193099
25	1	1665	77461	1193100	26	0	1670	77470	1193103
27	3	1655	77443	1193094	28	2	1656	77446	1193095
29	0	1662	77457	1193099	30	3	1655	77443	1193094
31	5	1665	77456	1193098	32	2	1652	77439	1193093
33	1	1649	77436	1193092	34	1	1649	77436	1193092
35	0	1650	77439	1193093	36	1	1645	77430	1193090
37	5	1641	77418	1193086	38	4	1646	77428	1193089
39	1	1649	77436	1193092	40	0	1650	77439	1193093
41	1	1649	77436	1193092	42	5	1645	77425	1193088
43	3	1647	77431	1193090	44	1	1641	77423	1193088
45	2	1644	77427	1193089	46	2	1640	77420	1193087
47	1	1645	77429	1193090	48	1	1641	77423	1193088
49	2	1640	77420	1193087	50	7	1643	77419	1193086
51	2	1644	77427	1193089	52	0	1642	77426	1193089
53	1	1645	77429	1193090	54	2	1648	77433	1193091
55	2	1644	77427	1193089	56	1	1641	77423	1193088
57	4	1638	77414	1193085	58	3	1643	77423	1193088
59	0	1650	77439	1193093	60	6	1644	77422	1193087

Table 2: Coefficients of failure polynomials for the nonisomorphic $(15, 3, 1)$-BIBDs (designs #61–80)

	F_5	F_6	F_7	F_8		F_5	F_6	F_7	F_8
61	7	1659	77442	1193094	62	0	1638	77418	1193087
63	6	1644	77421	1193087	64	3	1647	77430	1193090
65	2	1640	77420	1193087	66	3	1639	77417	1193086
67	4	1638	77414	1193085	68	1	1637	77416	1193086
69	4	1638	77414	1193085	70	2	1644	77427	1193089
71	2	1636	77413	1193085	72	4	1638	77414	1193085
73	9	1645	77420	1193086	74	3	1643	77424	1193088
75	6	1644	77422	1193087	76	1	1645	77430	1193090
77	1	1629	77402	1193082	78	9	1645	77420	1193086
79	17	1653	77424	1193086	80	11	1635	77400	1193080

References

[1] D. Agrawal, O. Eğecioğlu and A. El Abbadi. Billiard quorums on the grid. *Information Processing Letters* **64** (1997), 9–16.

[2] S. Y. Cheung, M. H. Ammar and M. Ahamad. The grid protocol: a high performance scheme for maintaining replicated data. *6th IEEE Conference on Data Engineering*, 1990, pp. 438–445.

[3] C. J. Colbourn and J. H. Dinitz, eds. *The CRC Handbook of Combinatorial Designs*, CRC Press, Inc., 1996.

[4] M. J. Grannell, T. S. Griggs and E. Mendelsohn. A small basis for four-line configurations in Steiner triple systems. *Journal of Combinatorial Designs* **3** (1995), 51–59.

[5] R. Holzman, Y. Marcus and D. Peleg. Load balancing in quorum systems. *SIAM Journal on Discrete Mathematics* **10** (1997), 223–245.

[6] A. C. H. Ling, C. J. Colbourn, M. J. Grannell and T. S. Griggs. Construction techniques for anti-Pasch Steiner triple systems. Preprint, 1997.

[7] L. Lovász. Coverings and colorings of hypergraphs. *4th Southeastern Conference on Combinatorics, Graph Theory and Computing*, 1973, pp. 3–12.

[8] L. Lovász. On the minimax theorems of combinatorics (in Hungarian). *Mathematikai Lapok* **26** (1975), 209–264.

[9] W.-S. Luk and T.-T. Wong. Two new quorum based algorithms for distributed mutual exclusion. *17th International Conference on Distributed Computing Systems*, 1997, pp. 100–106.

[10] M. Maekawa. A $\sqrt{N}$ algorithm for mutual exclusion in decentralized systems. *ACM Transactions on Computing Systems* **3** (1985), 145–159.

[11] W. H. Mills and R. C. Mullin. Coverings and packings. In *Contemporary Design Theory: A Collection of Surveys*, J. H. Dinitz and D. R. Stinson, eds., John Wiley & Sons, 1992, pp. 371–399.

[12] M. Naor and A. Wool. The load, capacity and availability of quorum systems. *35th IEEE Symposium on the Foundations of Computer Science*, 1994, pp. 214–225.

[13] D. Peleg and A. Wool. The availability of quorum systems. *Information and Computation* **123** (1995), 210–223.

[14] A. Rosa and C. J. Colbourn, Colorings of block designs. In *Contemporary Design Theory: A Collection of Surveys*, J. H. Dinitz and D. R. Stinson, eds., John Wiley & Sons, 1992, pp. 401–430.

[15] D. R. Stinson and Y. J. Wei. Some results on quadrilaterals in Steiner triple systems. *Discrete Math.* **105** (1992), 220–245.

[16] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Computing* 8 (1979), 410–421.